Data Mining with Graphs and Matrices

Fei Wang[1] Tao Li[1] Chris Ding[2]

[1]School of Computing and Information Sciences
Florida International University

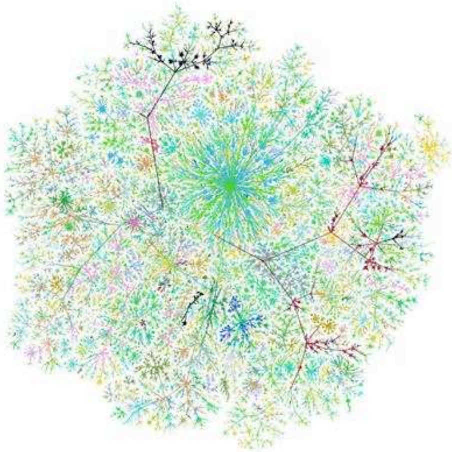
[2]Department of Computer Science and Engineering
University of Texas at Arlington

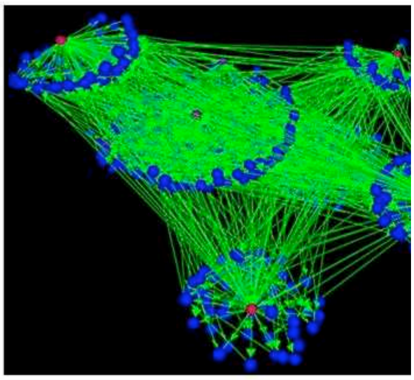
Tutorial at SDM 2009, Sparks, Nevada

http://feiwang03.googlepages.com/sdm-tutorial

Table of Contents

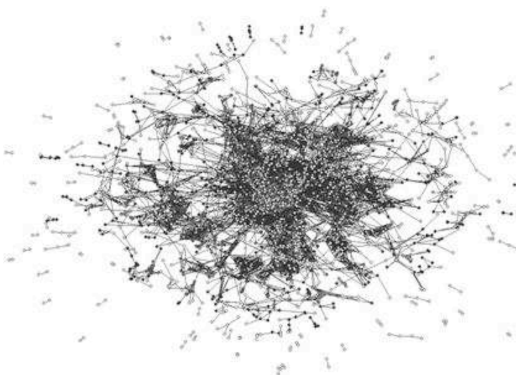
Internet Graph

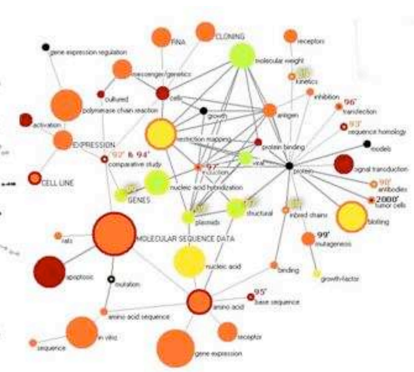




The images are downloaded from
http://www.maths.bris.ac.uk/~maarw/graphs/graph.html
and http://www.netdimes.org/new/?q=node/17

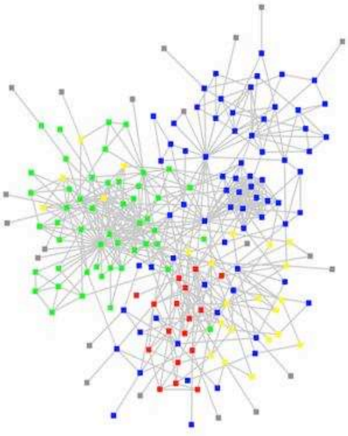
Citation Graph





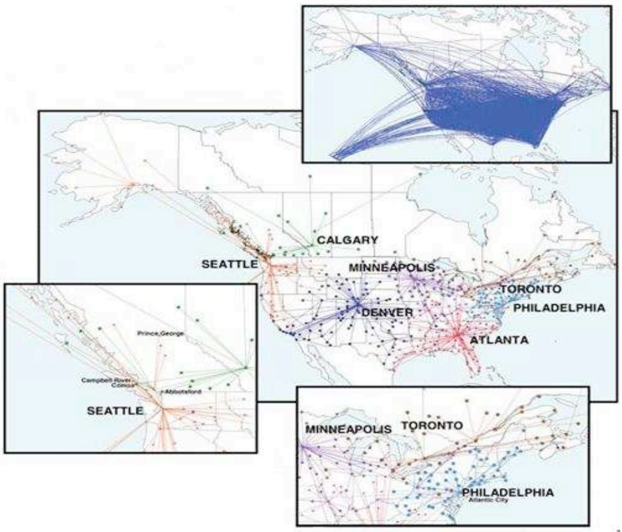
The images are downloaded from
http://www.emeraldinsight.com/fig/2780600403005.png
and www.bordalierinstitute.com/target1.html

Friendship Graph



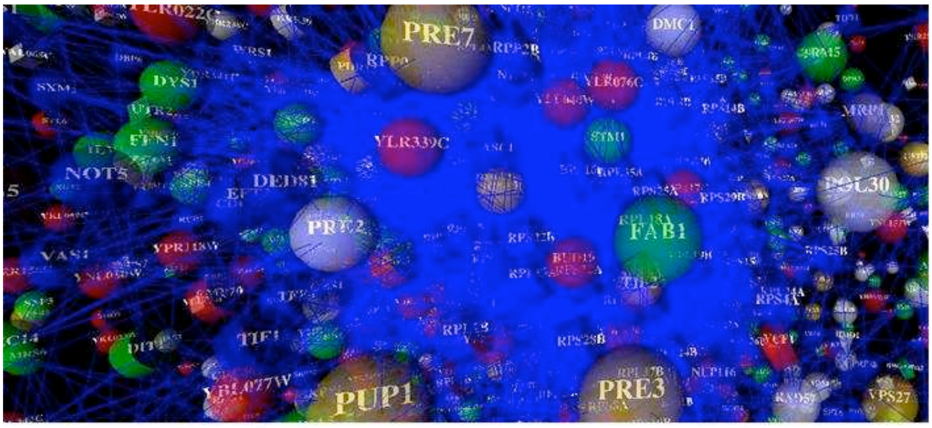
The images are downloaded from
http://www.thenetworkthinker.com/
and http://myweb20list.com/blog/2008/03/23/
new-amazing-facebook-photo-mapper/my-facebook-friend-graph/

Airline Graph



Copied from Brendan J. Frey and Delbert Dueck, University of Toronto
Clustering by Passing Messages Between Data Points. Science 315, 972-976

Protein Interaction Graph



The images are downloaded from
http://bioinformatics.icmb.utexas.edu/lgl/Images/rsomZoom.jpg

Social Network Analysis

- Email network
- Represents the email communications between users
  - Cluster users
  - Identify communities

				...	
	0	5	4	⋯	0
	6	0	3	⋯	2
⋮	⋮	⋮	⋮	⋱	⋮
	0	5	10	⋯	0

Document-Term Matrices

- A collection of documents is represented by an nDoc-by-nTerm matrix (bag-of-words model).
  - Cluster or classify documents
  - Find a subset of terms that (accurately) clusters or classifies documents

	image	Bayes	matrix	⋯	rock
	0	5	4	⋯	0
	6	0	3	⋯	2
⋮	⋮	⋮	⋮	⋱	⋮
	0	5	10	⋯	0

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions

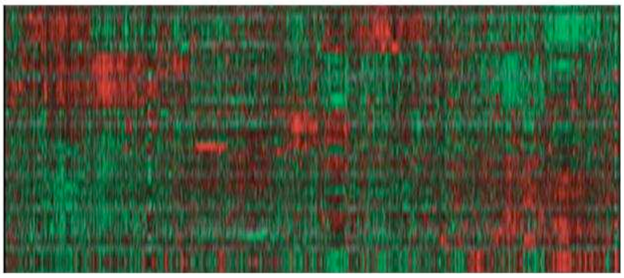
Recommendation Systems

- Collaborative filtering
  - Given the users' historical data, predict the ratings of a specific user to a new movie

	5	1	4	⋯	0
	1	5	3	⋯	2
⋮	⋮	⋮	⋮	⋱	⋮
	0	5	5	⋯	0

Bioinformatics

- Gene expression data
  - Pick a subset of genes (if it exists) that suffices in order to identify the "cancer type" of a patient

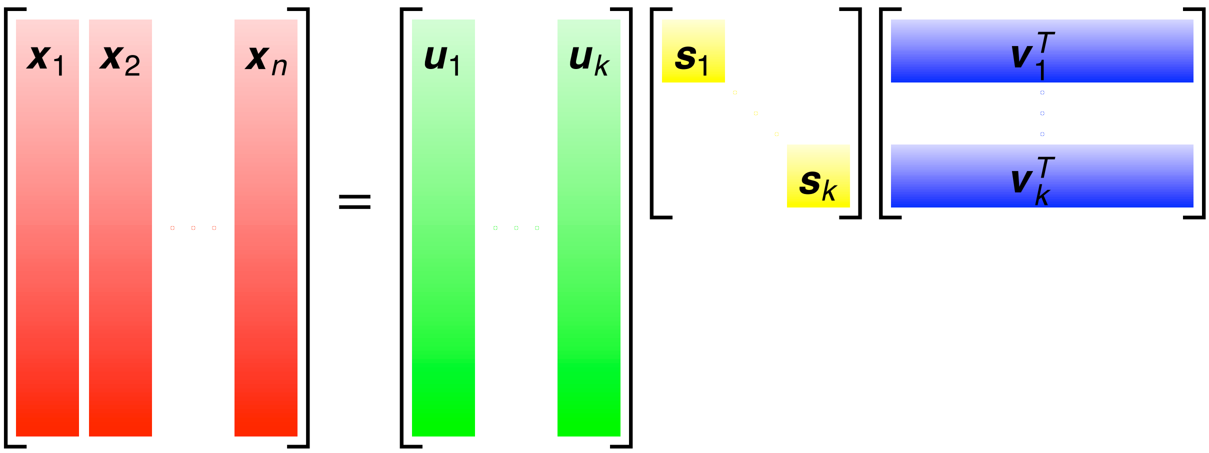


Some Notations & Preliminaries

- The data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$
- Generally, a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ can be described as a matrix
  - The columns and rows are indexed by $\mathcal{V}$
  - The elements are the strengths on the corresponding edges in $\mathcal{E}$
- Analyzing graphs is usually equivalent to perform analysis on matrices

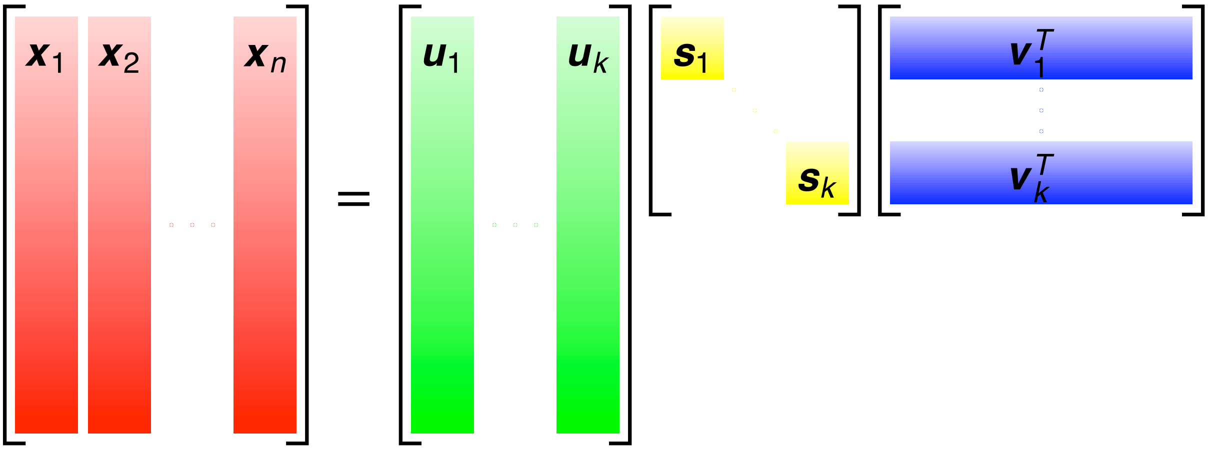
Table of Contents

Singular Value Decomposition



- Best rank-k approximation in Frobenius norm
- Exact computation of SVD takes $O(\min(dn^2, d^2n))$ time.
- The top k left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.

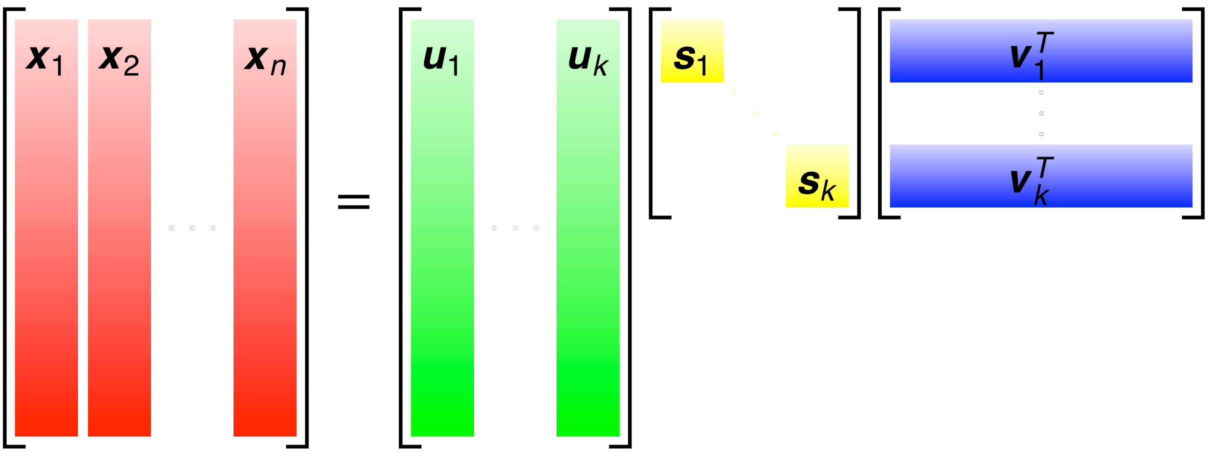
Singular Value Decomposition



- Best rank-k approximation in Frobenius norm
- Exact computation of SVD takes $O(\min(dn^2, d^2n))$ time.
- The top k left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

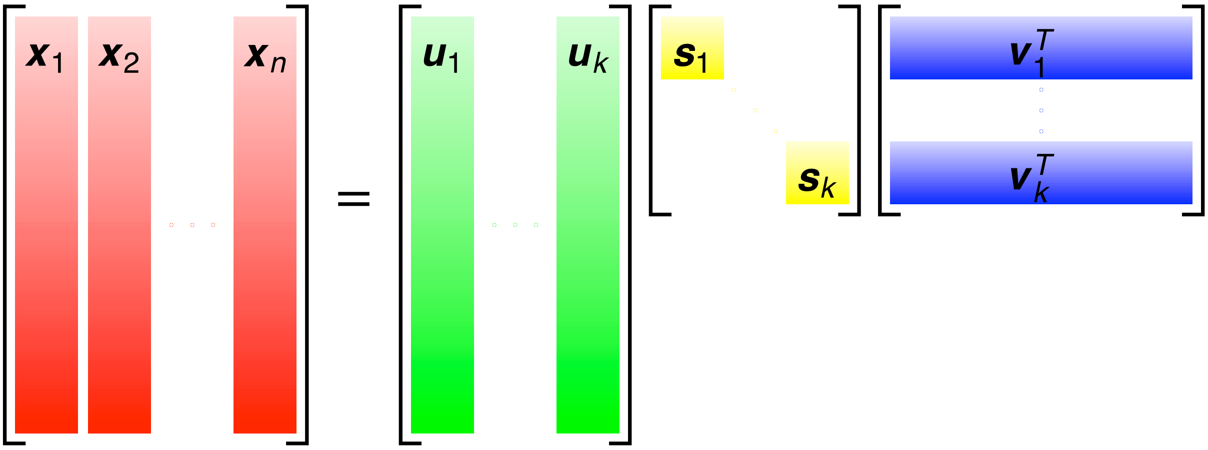
Singular Value Decomposition



- Best rank-k approximation in Frobenius norm
- Exact computation of SVD takes $O(\min(dn^2, d^2n))$ time.
- The top k left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

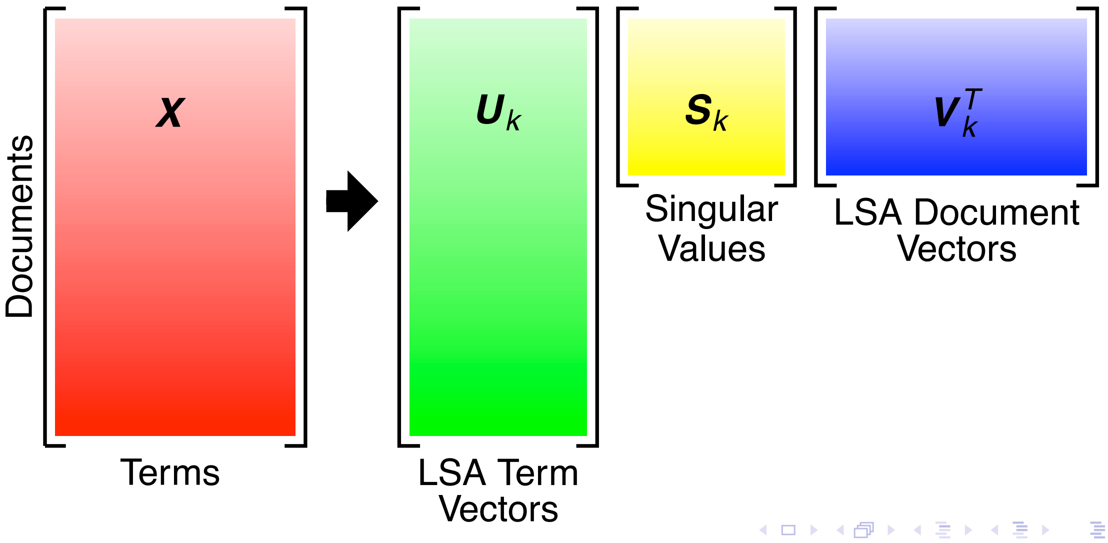
Singular Value Decomposition



- Best rank-k approximation in Frobenius norm
- Exact computation of SVD takes $O(\min(dn^2, d^2n))$ time.
- The top k left/right singular vectors/values can be computed faster using Lanczos/Arnoldi methods.

Latent Semantic Analysis

- k-dimensional semantic structure
- Similarity on reduced-space: D-D, D-T, T-T
- Folding-in queries: $\hat{\mathbf{q}} = \mathbf{S}_k^{-1}\mathbf{V}_k\mathbf{q}$



Principal Component Analysis

- Find a projection vector $\mathbf{u} \in \mathbb{R}^{d\times 1}$, such that the projected data points $\mathbf{Y} = \mathbf{u}^T\mathbf{X}$ own the largest variance, i.e., we should solve the following optimization problem

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{u}^T \frac{1}{n}\left(\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T\right)\mathbf{u} \\ s.t. \quad & \|\mathbf{u}\|^2 = 1 \end{aligned} \tag{1}$$

- From the standard theorem of Rayleigh-Ritz, we know that the optimal $\mathbf{u}$ is the eigenvector of the data covariance matrix $\mathbf{C}$ corresponding to its largest eigenvalue.

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

Principal Component Analysis

Find a projection vector $\mathbf{u} \in \mathbb{R}^{d\times 1}$, such that the projected data points $\mathbf{Y} = \mathbf{u}^T\mathbf{X}$ own the largest variance, i.e., we should solve the following optimization problem

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{u}^T \frac{1}{n}\left(\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T\right)\mathbf{u} \\ s.t. \quad & \|\mathbf{u}\|^2 = 1 \end{aligned} \tag{1}$$

From the standard theorem of Rayleigh-Ritz, we know that the optimal $\mathbf{u}$ is the eigenvector of the data covariance matrix $\mathbf{C}$ corresponding to its largest eigenvalue.

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

PCA & SVD

- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

PCA & SVD

- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

PCA & SVD

- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

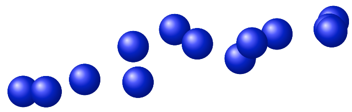
PCA & SVD

- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

PCA & SVD

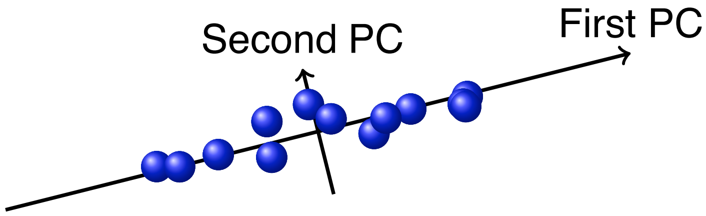
- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\mathbf{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD



Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

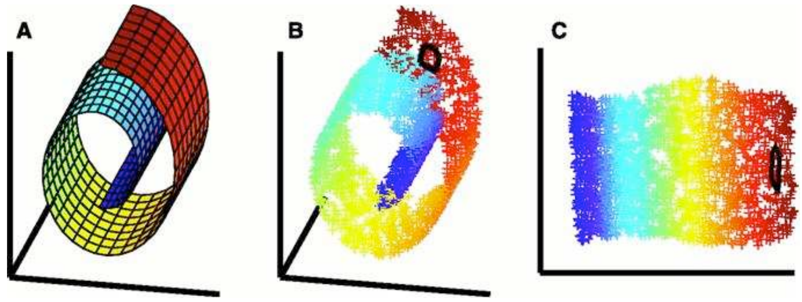
PCA & SVD

- If $\mathbf{X}$ is centralized, then the covariance matrix $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- Eigenvalue decomposition $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$
- SVD of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
  - $\frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T = \mathbf{U}\frac{1}{n}\mathbf{S}^2\mathbf{U}^T$
- Let $\mathbf{\Sigma} = \frac{1}{n}\mathbf{S}^2$, then PCA=SVD



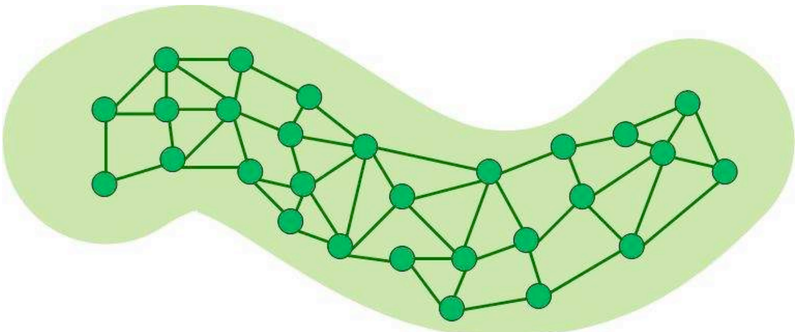
Nonlinear Embedding

- PCA is a linear method to project the data points
- What should we do if the data are nonlinearly distributed?



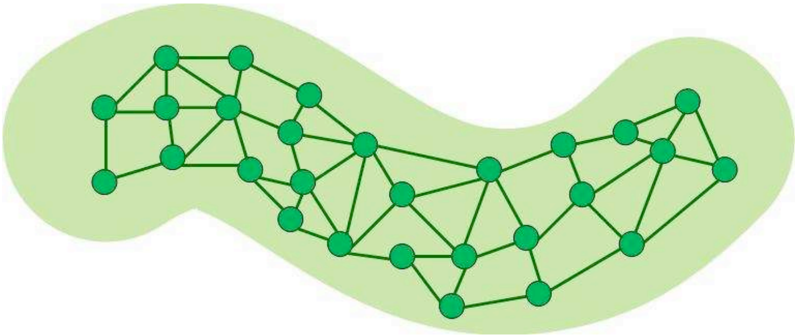
Manifold & Graph

- We usually assume that the high-dimensional data points reside (nearly) on a low-dimensional nonlinear manifold
- Find the low-dimensional embeddings of the data which preserve the graph structure



Manifold & Graph

- We usually assume that the high-dimensional data points reside (nearly) on a low-dimensional nonlinear manifold
- Find the low-dimensional embeddings of the data which preserve the graph structure



Local Linear Embedding (LLE)

- Assume each data point can be linearly reconstructed from its neighborhood, *i.e.*, for each $\mathbf{x}_i$, we minimize

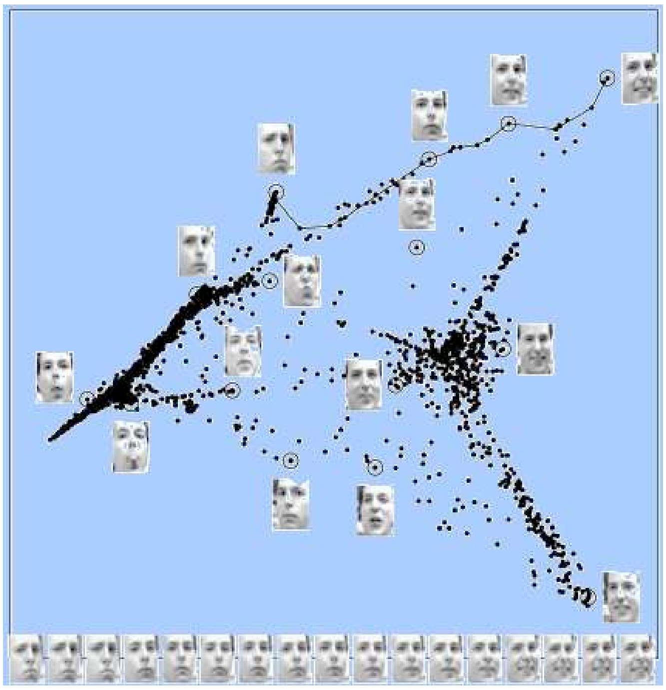
$$\begin{aligned} \varepsilon_i &= \sum\nolimits_{\mathbf{x}_j \in \mathcal{N}_i} \|\mathbf{x}_i - w_{ij}\mathbf{x}_j\|^2 \\ s.t. & \sum\nolimits_j w_{ij} = 1 \end{aligned} \quad (2)$$

- Then we use all $\{w_{ij}\}$ to recover the low-dimensional embedding of the data points $\mathbf{Y}$ by solving

$$\begin{aligned} \mathcal{J} &= \sum\nolimits_{i=1}^{n} \|\mathbf{y}_i - \sum\nolimits_{\mathbf{y}_j \in \mathcal{N}_i} w_{ij}\mathbf{y}_j\|^2 \\ s.t. & \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \end{aligned} \quad (3)$$

  - $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]$ is the low-dimensional embedded data matrix

An Example of LLE



Laplacian Eigenmaps (LE)

- The embedded data should be sufficiently smooth with respect to the intrinsic data manifold.
- We minimize

$$\min_{\mathbf{Y}} \quad \sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2$$
$$s.t. \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \qquad (4)$$

  - w_{ij} represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$
- Writing in matrix form $\sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 = tr(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T)$
  - $\mathbf{W}(i,j) = w_{ij}$ is the similarity matrix
  - $\mathbf{D} = diag(\sum_j w_{1j}, \cdots, \sum_j w_{2j})$
- We call $\mathbf{L} = \mathbf{D} - \mathbf{W}$ the *Laplacian matrix*

Laplacian Eigenmaps (LE)

- The embedded data should be sufficiently smooth with respect to the intrinsic data manifold.
- We minimize

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \end{aligned} \tag{4}$$

  - w_{ij} represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$
- Writing in matrix form $\sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 = tr(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T)$
  - $\mathbf{W}(i,j) = w_{ij}$ is the similarity matrix
  - $\mathbf{D} = diag(\sum_j w_{1j}, \cdots, \sum_j w_{2j})$
- We call $\mathbf{L} = \mathbf{D} - \mathbf{W}$ the *Laplacian matrix*

Laplacian Eigenmaps (LE)

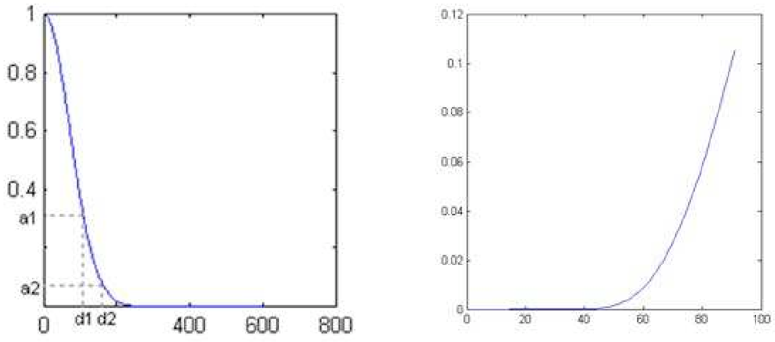
- The embedded data should be sufficiently smooth with respect to the intrinsic data manifold.
- We minimize

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ s.t. \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \end{aligned} \tag{4}$$

  - w_{ij} represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$
- Writing in matrix form $\sum_{i\sim j} w_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 = tr(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T)$
  - $\mathbf{W}(i,j) = w_{ij}$ is the similarity matrix
  - $\mathbf{D} = diag(\sum_j w_{1j}, \cdots, \sum_j w_{2j})$
- We call $\mathbf{L} = \mathbf{D} - \mathbf{W}$ the *Laplacian matrix*

Graph Similarities

- Node similarities: $s_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$



"Closer" nodes will get larger similarity

Weight as a function of δ

Locality Preserving Projections (LPP)

- Linear version of Laplacian embedding
- Let **P** be the projection matrix, then the goal of LPP is just to solve the following problem

$$\begin{aligned} \min_{\mathbf{P}} \quad & tr(\mathbf{P}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}) \\ s.t. \quad & \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{5}$$

- Locality Preserving Indexing
- Laplacianface
- ...

Graph Embedding: A General Framework

- A general graph embedding framework:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \sum_{i\sim j} p_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ s.t. \quad & \mathbf{y}^T\mathbf{A}\mathbf{y} = c \end{aligned} \tag{6}$$

  - $i \sim j$ denotes that there is an edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$
  - c is a constant
- Linearization:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \sum_{i\sim j} p_{ij}\|\mathbf{p}^T\mathbf{x}_i - \mathbf{p}^T\mathbf{x}_j\|^2 \\ s.t. \quad & \mathbf{p}^T\mathbf{X}\mathbf{A}\mathbf{X}^T\mathbf{p} = c \end{aligned} \tag{7}$$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

Graph Embedding: A General Framework

- A general graph embedding framework:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \sum_{i\sim j} p_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ s.t. \quad & \mathbf{y}^T\mathbf{A}\mathbf{y} = c \end{aligned} \tag{6}$$

  - $i \sim j$ denotes that there is an edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$
  - c is a constant
- Linearization:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \sum_{i\sim j} p_{ij}\|\mathbf{p}^T\mathbf{x}_i - \mathbf{p}^T\mathbf{x}_j\|^2 \\ s.t. \quad & \mathbf{p}^T\mathbf{X}\mathbf{A}\mathbf{X}^T\mathbf{p} = c \end{aligned} \tag{7}$$

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

Summarization of Different Methods from a GE Perspective (Shuicheng Yan et al. CVPR'05)

Algorithm	$\mathbf{P}$	$\mathbf{A}$
PCA	$p_{ij} = 1/n, \forall i \neq j$	$\mathbf{A} = \mathbf{I}$
LDA	$p_{ij} = \delta_{l_1, l_j}/n_{l_i}$	$\mathbf{A} = \mathbf{I} - \mathbf{e}\mathbf{e}^T$
LLE	$\mathbf{P} = \mathbf{W} + \mathbf{W}^T - \mathbf{W}^T\mathbf{W}$	$\mathbf{A} = \mathbf{I}$
LPP	$p_{ij} = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$	$\mathbf{A} = \mathbf{D}$

Table of Contents

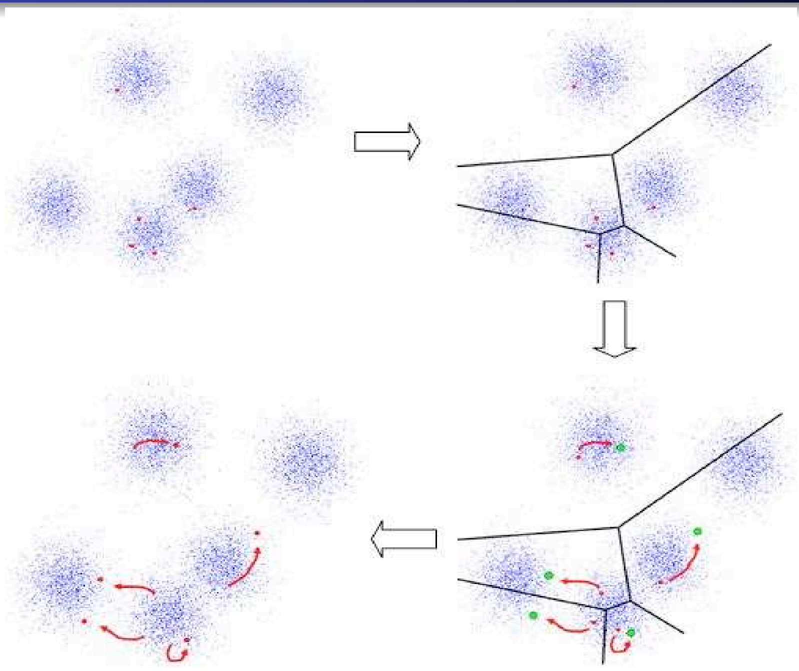
K-means

- The data points $\mathbf{X}$ comes from C clusters. We aim to find the cluster centers $\{\mathbf{f}_i\}_{i=1}^C$ together with the clusters such that the following criterion is minimized

$$\min \sum_{i=1}^{C} \sum_{\mathbf{x}_j \in \pi_i} \|\mathbf{x}_j - \mathbf{f}_i\|^2 \qquad (8)$$

  - π_i denotes the i-th cluster
- We can resort to iterative procedures to solve the problem.

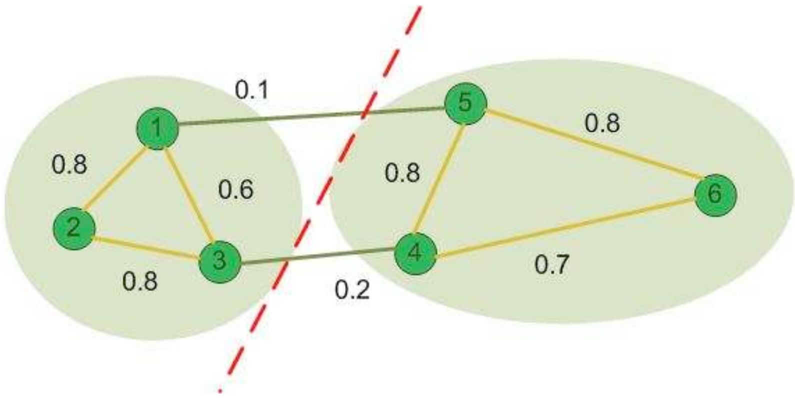
K-means Procedure



The figures come from

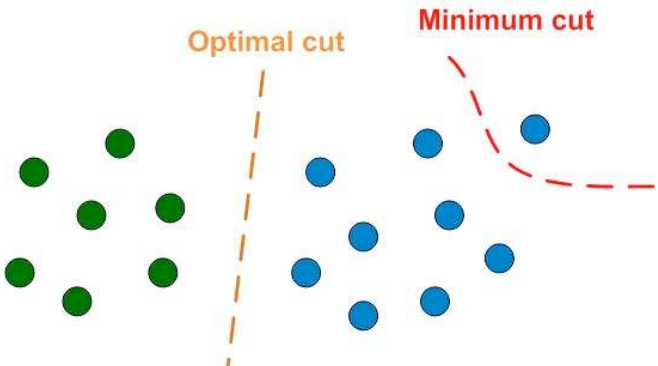
Graph Clustering

- Partition the nodes $\mathcal{V}$ in graph $\mathcal{G}$ into disjoint clusters
- Cut: Set of edges with points belonging to different clusters
- Association: Set of edges with points belonging to the same cluster



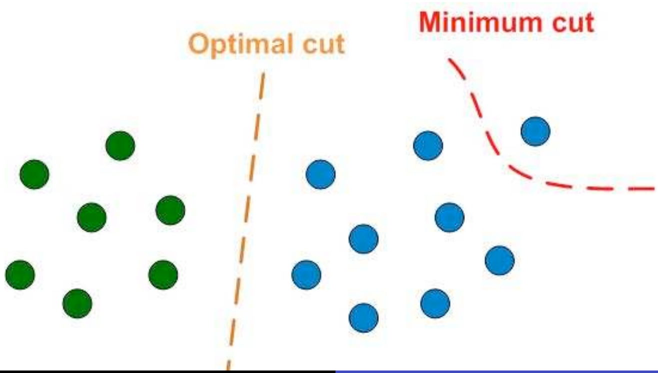
Graph Cut Criteria

- *MinCut*: Minimize the association between groups min $cut(\mathcal{A}, \mathcal{B})$
- *Normalized graph cut criterions*:
  - RatioAssociation: max $\frac{asso(\mathcal{A},\mathcal{A})}{|\mathcal{A}|} + \frac{asso(\mathcal{B},\mathcal{B})}{|\mathcal{B}|}$
  - RatioCut: min $\frac{cut(\mathcal{A},\mathcal{B})}{|\mathcal{A}|} + \frac{cut(\mathcal{B},\mathcal{A})}{|\mathcal{B}|}$
  - NormalizedCut: min $\frac{cut(\mathcal{A},\mathcal{B})}{vol(\mathcal{A})} + \frac{cut(\mathcal{B},\mathcal{A})}{vol(\mathcal{B})}$



Graph Cut Criteria

- *MinCut*: Minimize the association between groups
  min $cut(\mathcal{A}, \mathcal{B})$
- *Normalized graph cut criterions*:
  - RatioAssociation: max $\frac{asso(\mathcal{A},\mathcal{A})}{|\mathcal{A}|} + \frac{asso(\mathcal{B},\mathcal{B})}{|\mathcal{B}|}$
  - RatioCut: min $\frac{cut(\mathcal{A},\mathcal{B})}{|\mathcal{A}|} + \frac{cut(\mathcal{B},\mathcal{A})}{|\mathcal{B}|}$
  - NormalizedCut: min $\frac{cut(\mathcal{A},\mathcal{B})}{vol(\mathcal{A})} + \frac{cut(\mathcal{B},\mathcal{A})}{vol(\mathcal{B})}$



Some Definitions on Graphs

- Weight Matrix $\mathbf{W}$: $\mathbf{W}_{ij}$ is the weight on the edge e_{ij}
- Degree Matrix $\mathbf{D}$: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$
- Partition Matrix $\mathbf{P}$: $\mathbf{P}_{ij} = 1$ if $\mathbf{x}_i$ belongs to partition j; Otherwise $\mathbf{P}_{ij} = 0$
- Scaled Partition Matrix $\widetilde{\mathbf{P}}$: $\widetilde{\mathbf{P}}_{ij} = 1/\sqrt{n_j}$ if $\mathbf{x}_i$ belongs to partition j, n_j is the size of the j-th cluster; Otherwise $\tilde{\mathbf{P}}_{ij} = 0$
- The goal of graph clustering is to solve $\mathbf{P}$ or $\widetilde{\mathbf{P}}$

Some Definitions on Graphs

- Weight Matrix $\mathbf{W}$: $\mathbf{W}_{ij}$ is the weight on the edge e_{ij}
- Degree Matrix $\mathbf{D}$: $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$
- Partition Matrix $\mathbf{P}$: $\mathbf{P}_{ij} = 1$ if $\mathbf{x}_i$ belongs to partition j; Otherwise $\mathbf{P}_{ij} = 0$
- Scaled Partition Matrix $\widetilde{\mathbf{P}}$: $\widetilde{\mathbf{P}}_{ij} = 1/\sqrt{n_j}$ if $\mathbf{x}_i$ belongs to partition j, n_j is the size of the j-th cluster; Otherwise $\tilde{\mathbf{P}}_{ij} = 0$
- The goal of graph clustering is to solve $\mathbf{P}$ or $\widetilde{\mathbf{P}}$

Spectral Clustering

- The solutions of the above optimization problems can be finally obtained by spectral analysis of some matrices
- Ratio association: Do eigenvalue decomposition to $\mathbf{W}$
- Ratio cut: Do eigenvalue decomposition to $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- Normalized cut: Do eigenvalue decomposition to $\hat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$

Spectral Clustering

- The solutions of the above optimization problems can be finally obtained by spectral analysis of some matrices
- Ratio association: Do eigenvalue decomposition to $\mathbf{W}$
- Ratio cut: Do eigenvalue decomposition to $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- Normalized cut: Do eigenvalue decomposition to $\hat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$

Spectral Clustering II

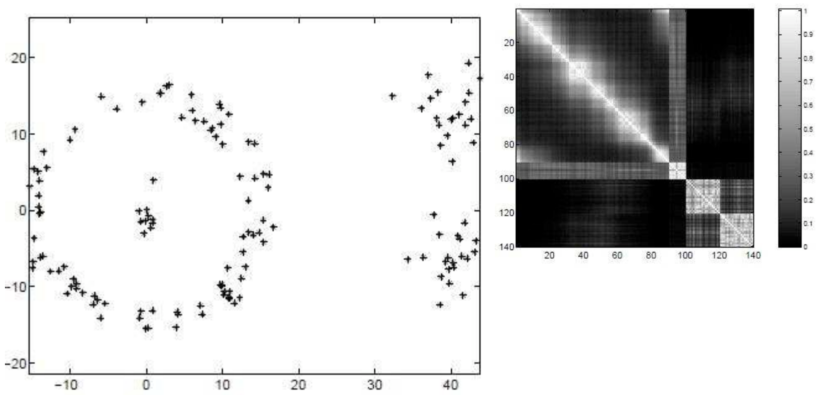
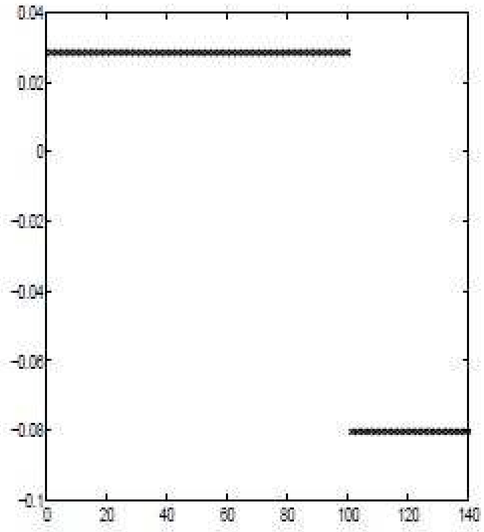


Figure from Shi & Malik. PAMI 2000.

The Eigenvectors of The Normalized Laplacian Matrix



Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

Nonnegative Matrix Factorization

- Analyzing nonnegative matrices (document-word matrix, image matrix...)
- For a nonnegative matrix $\mathbf{X}$, we decompose it into two nonnegative matrices

$$\min_{\mathbf{F}\geqslant 0,\mathbf{G}\geqslant 0} \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|^2 \quad (9)$$

- Multiplicative update rule to solve the problem

$$\mathbf{F}_{ij} \longleftarrow \mathbf{F}_{ij}\frac{(\mathbf{XG})_{ij}}{(\mathbf{FG}^T\mathbf{G})_{ij}}, \quad \mathbf{G}_{ij} \longleftarrow \mathbf{G}_{ij}\frac{(\mathbf{F}^T\mathbf{X})_{ij}}{(\mathbf{F}^T\mathbf{F}\mathbf{G}^T)_{ij}}$$

- Parts-based representation

Nonnegative Matrix Factorization

- Analyzing nonnegative matrices (document-word matrix, image matrix...)
- For a nonnegative matrix $\mathbf{X}$, we decompose it into two nonnegative matrices

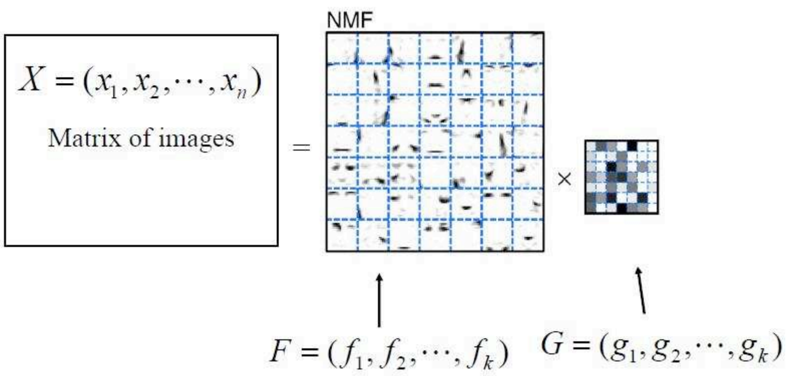
$$\min_{\mathbf{F}\geqslant 0, \mathbf{G}\geqslant 0} \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|^2 \tag{9}$$

- Multiplicative update rule to solve the problem

$$\mathbf{F}_{ij} \longleftarrow \mathbf{F}_{ij}\frac{(\mathbf{X}\mathbf{G})_{ij}}{(\mathbf{F}\mathbf{G}^T\mathbf{G})_{ij}}, \quad \mathbf{G}_{ij} \longleftarrow \mathbf{G}_{ij}\frac{(\mathbf{F}^T\mathbf{X})_{ij}}{(\mathbf{F}^T\mathbf{F}\mathbf{G}^T)_{ij}}$$

- Parts-based representation

NMF: An Illustrative Example



Clustering Results on TDT Data

Performance comparisons using TDT2 corpus

k	Mutual Information				Accuracy			
	AA	NC	NMF	NMF-NCW	AA	NC	NMF	NMF-NCW
2	0.834	0.954	0.854	0.972	0.934	0.990	0.946	0.993
3	0.754	0.890	0.790	0.931	0.863	0.951	0.899	0.981
4	0.743	0.846	0.786	0.909	0.830	0.918	0.866	0.953
5	0.696	0.802	0.740	0.874	0.758	0.857	0.812	0.925
6	0.663	0.761	0.701	0.823	0.712	0.802	0.773	0.880
7	0.679	0.756	0.704	0.816	0.707	0.783	0.750	0.857
8	0.624	0.695	0.651	0.782	0.641	0.717	0.697	0.824
9	0.663	0.741	0.683	0.804	0.664	0.754	0.708	0.837
10	0.656	0.736	0.681	0.812	0.638	0.729	0.685	0.835
average	0.701	0.798	0.732	0.858	0.750	0.833	0.793	0.898

From Xu, Liu & Gong. SIGIR'03.

Clustering Results on Reuters Data

Performance comparisons using Reuters corpus

k	Mutual Information				Accuracy			
	AA	NC	NMF	NMF-NCW	AA	NC	NMF	NMF-NCW
2	0.399	0.484	0.437	0.494	0.784	0.821	0.824	0.837
3	0.482	0.536	0.489	0.574	0.709	0.765	0.731	0.803
4	0.480	0.581	0.487	0.604	0.629	0.734	0.655	0.758
5	0.565	0.590	0.587	0.600	0.655	0.695	0.686	0.722
6	0.537	0.627	0.559	0.650	0.611	0.678	0.650	0.728
7	0.560	0.599	0.575	0.624	0.584	0.654	0.624	0.696
8	0.559	0.592	0.578	0.606	0.581	0.613	0.618	0.651
9	0.603	0.633	0.614	0.659	0.599	0.640	0.634	0.692
10	0.607	0.647	0.626	0.661	0.600	0.634	0.634	0.677
average	0.532	0.588	0.550	0.608	0.639	0.693	0.673	0.729

See Xu, Liu & Gong. SIGIR'03.

NMF Variants

- If the data matrix $\mathbf{X}$ has mixed signs, then
- Singular Value Decomposition: $\mathbf{X}_{\pm} \approx \mathbf{F}_{\pm}\mathbf{G}_{\pm}^T$
- Semi-NMF: $\mathbf{X}_{\pm} \approx \mathbf{F}_{\pm}\mathbf{G}_{+}^T$
- Convex-NMF: $\mathbf{X}_{\pm} \approx \mathbf{X}_{\pm}\mathbf{W}_{+}\mathbf{G}_{+}^T$
- Kernel-NMF: $\phi(\mathbf{X}_{\pm}) \approx \phi(\mathbf{X}_{\pm})\mathbf{W}_{+}\mathbf{G}_{+}^T$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

The Relationships Between NMF and K-means

- *K-means* objective:

$$J_{km} = \sum_{c}\sum_{\mathbf{x}_i\in\pi_c}\|\mathbf{x}_i - \mathbf{f}_c\|^2 = \sum_{i=1}^{n}\sum_{c=1}^{c} g_{ic}\|\mathbf{x}_i - \mathbf{f}_c\|^2$$
$$= \left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|_F^2$$

  - Cluster center matrix: $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_C] \in \mathbb{R}^{n\times C}$
  - $\mathbf{G} \in \mathbb{R}^{n\times C}$ with $\mathbf{G}_{ij} = g_{ij}$, such that $g_{ij} = 1$, if $\mathbf{x}_i \in \pi_j$; $\mathbf{G}_{ij} = 0$,otherwise.
- K-means and NMF: the same objective, only different constraint
  - NMF: $\mathbf{F} \geq 0, \quad \mathbf{G} \geq 0$
  - K-means: $\mathbf{G}_{ij} \in \{0, 1\}, \quad \mathbf{G1} = \mathbf{1}$

The Relationships Between K-means and PCA

- $\varepsilon_k = \sum_{i=1}^{n_k} \|\mathbf{x}_i^{(k)} - \mathbf{m}_k\|^2 = \|\mathbf{X}_k - \mathbf{m}_k\mathbf{e}^T\|^2$
- $\varepsilon_k = trace\left(\mathbf{X}_k(\mathbf{I}_{n_k} - \mathbf{e}\mathbf{e}^T/n_k)\mathbf{X}_k^T\right)$
- Finally, $\varepsilon = \sum_{k=1}^{C} \varepsilon_k = \sum_{k=1}^{C}\left(trace(\mathbf{X}_i^T\mathbf{X}_i) - \left(\frac{e^T}{\sqrt{n_k}}\right)\mathbf{X}_k^T\mathbf{X}_k\left(\frac{e^T}{\sqrt{n_k}}\right)\right)$
- Let $\tilde{\mathbf{P}} = diag(\frac{\mathbf{e}_{n_1}}{\sqrt{n_1}}, \cdots, \frac{\mathbf{e}_{n_C}}{\sqrt{n_C}})$
- Then $\varepsilon = trace(\mathbf{X}^T\mathbf{X}) - trace(\tilde{\mathbf{P}}^T\mathbf{X}^T\mathbf{X}\tilde{\mathbf{P}})$ subject to $\tilde{\mathbf{P}}^T\tilde{\mathbf{P}} = \mathbf{I}$
- Therefore we need to maximize $trace(\tilde{\mathbf{P}}^T\mathbf{X}^T\mathbf{X}\tilde{\mathbf{P}})$ and get $\tilde{\mathbf{P}}$.
- According to the Ky Fan therorem, $\tilde{\mathbf{P}}$ is composed of the eigenvectors of $\mathbf{X}^T\mathbf{X}$ corresponding to its largest C eigevalues
- If $\mathbf{X}$ is centralized, then it is equivalent to PCA

The Relationships Between K-means and Spectral Clustering

- From last slide we can see that the relaxed solution of kmeans is equivalent to analyze the eigenstructure of $\mathbf{A} = \mathbf{X}^T\mathbf{X}$
- If we define the similarity matrix $\mathbf{W} = \mathbf{A}$, then kmeans is equivalent to ratio association
- Define the weighted kmeans criterion $\tilde{\varepsilon} = \sum_{k=1}^{C}\sum_{\mathbf{x}_i \in \pi_k} w_i\|\mathbf{x}_i - \mathbf{m}_k\|^2$
- Using similar derivation procedure, we can derive that optimizing the above criterion is equivalent to solve

$$max_{\tilde{\mathbf{P}}}\, trace(\tilde{\mathbf{P}}^T\mathbf{D}^{1/2}\mathbf{W}\mathbf{D}^{1/2}\tilde{\mathbf{P}}) \qquad (10)$$

The Relationships Between NMF and Spectral Clustering

- Let the normalized similarity matrix be $\tilde{\mathbf{W}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
- Then we have the following theorem

Theorem

Normalized Cut using similarity $\tilde{\mathbf{W}}$ is equivalent to the following symmetric nonnegative matrix factorization

$$\min_{\tilde{\mathbf{P}} \geqslant 0} \mathcal{J} = \|\tilde{\mathbf{W}} - \tilde{\mathbf{P}}\tilde{\mathbf{P}}^T\|^2 \tag{11}$$

Table of Contents

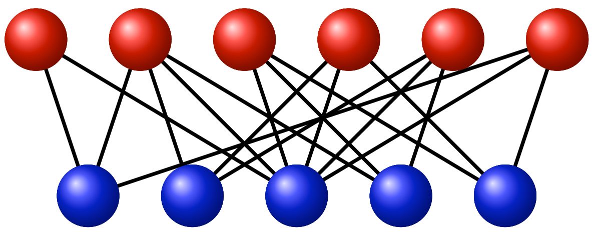
The Problem

- Usually the data we face with are relational, *i.e.*, there are multiple type of data interrelated with each other
- How to cluster those relational data simultaneously?

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

The Problem

- Usually the data we face with are relational, *i.e.*, there are multiple type of data interrelated with each other
- How to cluster those relational data simultaneously?



A Spectral Approach

- Define the similarity matrix on the bi-partite graph
$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}$
- Also the concatenated cluster membership vector
$\mathbf{x} = [\mathbf{x}_I^T, \mathbf{x}_{II}^T]^T$
- Then the co-clustering problem becomes a graph-cut problem on the bi-partite graph, *i.e.*, we should solve the following generalized eigenvalue decomposition problem

$$\mathbf{L}\mathbf{x} = \lambda \mathbf{D}\mathbf{x} \quad (12)$$

- where $\mathbf{D} = diag(\sum_j A_{1j}, \cdots, \sum_j A_{n_j})$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

A Spectral Approach

- Define the similarity matrix on the bi-partite graph
$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}$
- Also the concatenated cluster membership vector
$\mathbf{x} = [\mathbf{x}_I^T, \mathbf{x}_{II}^T]^T$
- Then the co-clustering problem becomes a graph-cut problem on the bi-partite graph, *i.e.*, we should solve the following generalized eigenvalue decomposition problem

$$\mathbf{L}\mathbf{x} = \lambda \mathbf{D}\mathbf{x} \quad (12)$$

- where $\mathbf{D} = diag(\sum_j A_{1j}, \cdots, \sum_j A_{n_j})$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

A Spectral Approach

- Define the similarity matrix on the bi-partite graph
$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}$$
- Also the concatenated cluster membership vector $\mathbf{x} = [\mathbf{x}_I^T, \mathbf{x}_{II}^T]^T$
- Then the co-clustering problem becomes a graph-cut problem on the bi-partite graph, *i.e.*, we should solve the following generalized eigenvalue decomposition problem

$$\mathbf{L}\mathbf{x} = \lambda \mathbf{D}\mathbf{x} \tag{12}$$

- where $\mathbf{D} = diag(\sum_j A_{1j}, \cdots, \sum_j A_{n_j})$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Dimensionality Reduction
Clustering
Co-Clustering

Nonnegative Matrix Tri-Factorization

- Factorize the user-movie rating matrix $\mathbf{X}$ into three matrices $\mathbf{F}, \mathbf{S}, \mathbf{G}$, such that
  - $\mathbf{F}$ represents the cluster memberships on the user side
  - $\mathbf{G}$ represents the cluster memberships on the movie side
- By relaxing the integer constrains on $\mathbf{F}, \mathbf{G}$, we need to solve the following optimization problem

$$\min_{\mathbf{F}\geq 0, \mathbf{S}\geq 0, \mathbf{G}\geq 0} \|\mathbf{X} - \mathbf{F}\mathbf{S}\mathbf{G}^T\|^2, \quad s.t.\ \mathbf{F}^T\mathbf{F} = \mathbf{I},\ \mathbf{G}\mathbf{G}^T = \mathbf{I} \tag{13}$$

- We can derive some multiplicative update rules to solve for the optimal $\mathbf{F}, \mathbf{S}, \mathbf{G}$

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

An Example of NMTF

Datasets	BiOR-NM3F			K-means		
	Purity	Entropy	ARI	Purity	Entropy	ARI
CSTR	0.754	0.402	0.436	0.712	0.412	0.189
WebKB4	0.583	0.372	0.428	0.534	0.442	0.418
Reuters	0.558	0.976	0.510	0.545	0.726	0.506
WebAce	0.541	0.889	0.449	0.546	0.868	0.452
Newsgroups	0.507	1.233	0.179	0.330	1.488	0.149

Performance Comparisons of clustering algorithms. Each entry is the corresponding performance value of the algorithm on the row dataset.

Other Types of Co-Clustering Methods

- Information-Theoretic Co-clustering (Dhillon et al. KDD'03)
- Bayesian Co-Clustering (Shan & Banerjee. ICDM'08)
- Tensor Method (Banerjee et al. SDM'07)
- Collective Factorization on Related Matrices (Long et al. ICML'06)
- Multiple Latent Semantic Analysis (Wang et al. SIGIR'06)

Why Semi-supervised Learning

- Traditional learning problems
  - Supervised learning: learning with labeled data set
  - Unsupervised learning: learning with unlabeled data set
- Problems
  - Supervised learning: requires much human effort, expensive and time consuming
  - Unsupervised learning: unreliable
- Semi-supervised learning
  - Learning with partially labeled data
  - Learning with side-information

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Semi-supervised Learning with Partially Labeled Data
Semi-supervised Learning Using Side-Information

Why Semi-supervised Learning

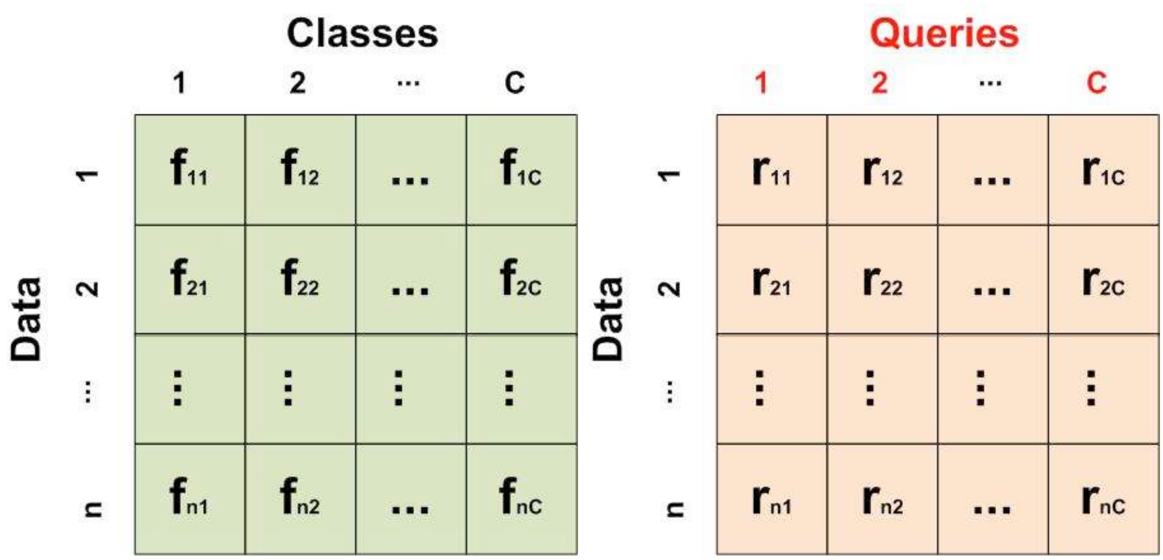
- Traditional learning problems
  - Supervised learning: learning with labeled data set
  - Unsupervised learning: learning with unlabeled data set
- Problems
  - Supervised learning: requires much human effort, expensive and time consuming
  - Unsupervised learning: unreliable
- Semi-supervised learning
  - Learning with partially labeled data
  - Learning with side-information

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

Why Semi-supervised Learning

- Traditional learning problems
  - Supervised learning: learning with labeled data set
  - Unsupervised learning: learning with unlabeled data set
- Problems
  - Supervised learning: requires much human effort, expensive and time consuming
  - Unsupervised learning: unreliable
- Semi-supervised learning
  - Learning with partially labeled data
  - Learning with side-information

The Similarity Between SSL and Ranking



The Similarity Between SSL and Collaborative Filtering

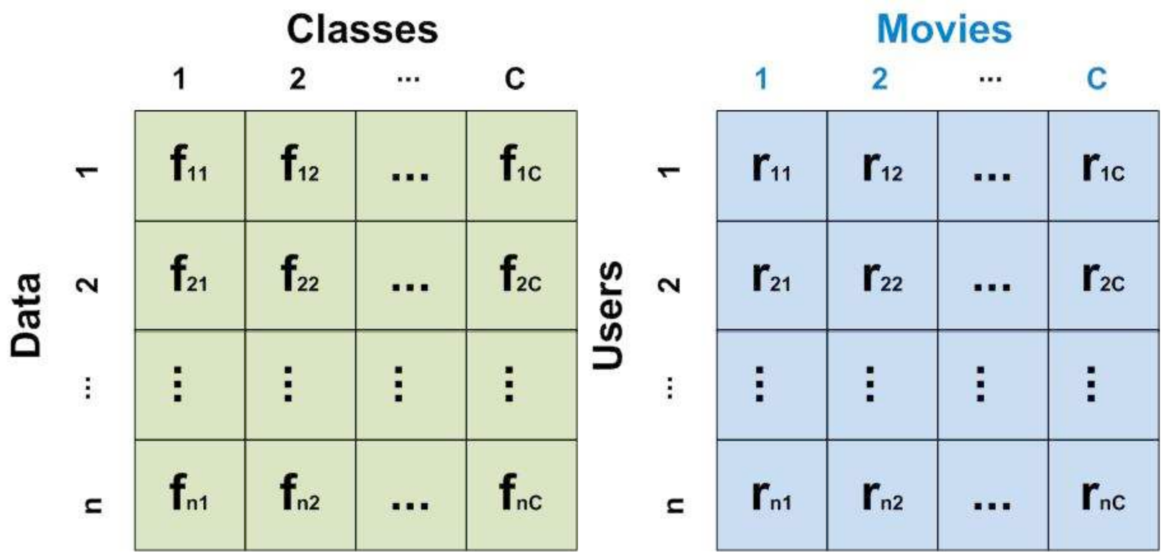


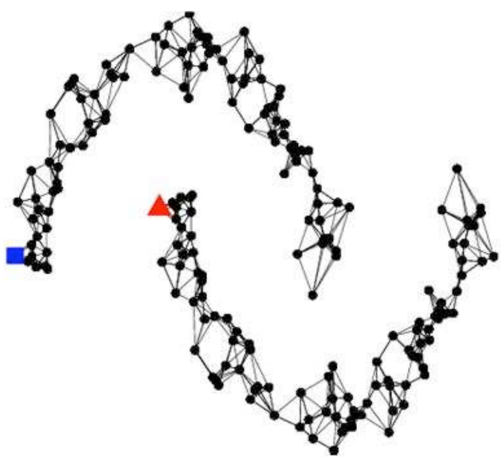
Table of Contents

Semi-supervised Assumption

- *Smoothness Assumption*: If two points $\mathbf{x}_1, \mathbf{x}_2$ in a high-density region are close, then so should be the corresponding outputs y_1, y_2
- *Cluster Assumption*: If points are in the same cluster, they are likely to be of the same class
- *Manifold Assumption*: The (high-dimensional) data lie (roughly) on a low-dimensional manifold

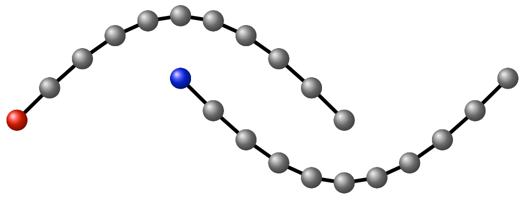
Label Propagation

- Connect the data points that are close to each other (Nearest Neighbor Graph)
- Propagate the class labels over the connected graph



Propagation Rules

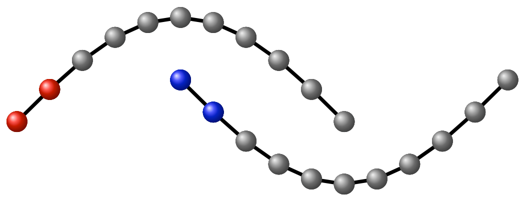
- Initial label vector: $\mathbf{y} \in \mathbb{R}^{n \times 1}$
  - $y_i = t_i$ if $\mathbf{x}_i$ is labeled as t_i; $y_i = 0$ if $\mathbf{x}_i$ is unlabeled
- $f_i^{(1)} = y_i$ if $\mathbf{x}_i$ is labeled; $f_i^{(1)} = \alpha \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{P}_{ij} y_j$ otherwise
  - $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the propagation matrix
  - Matrix form: $\mathbf{f}^{(1)} = \mathbf{y} + \alpha \mathbf{P} \mathbf{y}$
- $\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + \alpha \mathbf{P} \mathbf{f}^{(1)} = (\mathbf{I} + \alpha \mathbf{P} + \alpha^2 \mathbf{P}^2)\mathbf{y}$
- Finally $\mathbf{f}^{(\infty)} = \sum_{i=0}^{\infty} \alpha^i \mathbf{P}^i \mathbf{y} = (\mathbf{I} - \alpha \mathbf{P})^{-1} \mathbf{y}$



Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Semi-supervised Learning with Partially Labeled Data
Semi-supervised Learning Using Side-Information

Propagation Rules

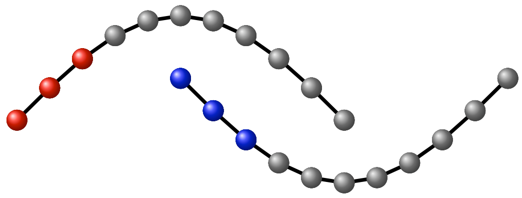
- Initial label vector: $\mathbf{y} \in \mathbb{R}^{n \times 1}$
  - $y_i = t_i$ if $\mathbf{x}_i$ is labeled as t_i; $y_i = 0$ if $\mathbf{x}_i$ is unlabeled
- $f_i^{(1)} = y_i$ if $\mathbf{x}_i$ is labeled; $f_i^{(1)} = \alpha \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{P}_{ij} y_j$ otherwise
  - $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the propagation matrix
  - Matrix form: $\mathbf{f}^{(1)} = \mathbf{y} + \alpha \mathbf{P} \mathbf{y}$
- $\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + \alpha \mathbf{P} \mathbf{f}^{(1)} = (\mathbf{I} + \alpha \mathbf{P} + \alpha^2 \mathbf{P}^2)\mathbf{y}$
- Finally $\mathbf{f}^{(\infty)} = \sum_{i=0}^{\infty} \alpha^i \mathbf{P}^i \mathbf{y} = (\mathbf{I} - \alpha \mathbf{P})^{-1} \mathbf{y}$



Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

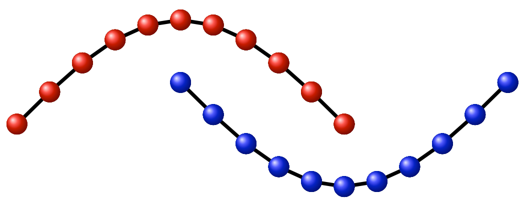
Propagation Rules

- Initial label vector: $\mathbf{y} \in \mathbb{R}^{n \times 1}$
  - $y_i = t_i$ if $\mathbf{x}_i$ is labeled as t_i; $y_i = 0$ if $\mathbf{x}_i$ is unlabeled
- $f_i^{(1)} = y_i$ if $\mathbf{x}_i$ is labeled; $f_i^{(1)} = \alpha \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{P}_{ij} y_j$ otherwise
  - $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the propagation matrix
  - Matrix form: $\mathbf{f}^{(1)} = \mathbf{y} + \alpha \mathbf{P}\mathbf{y}$
- $\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + \alpha \mathbf{P}\mathbf{f}^{(1)} = (\mathbf{I} + \alpha \mathbf{P} + \alpha^2 \mathbf{P}^2)\mathbf{y}$
- Finally $\mathbf{f}^{(\infty)} = \sum_{i=0}^{\infty} \alpha^i \mathbf{P}^i \mathbf{y} = (\mathbf{I} - \alpha \mathbf{P})^{-1} \mathbf{y}$

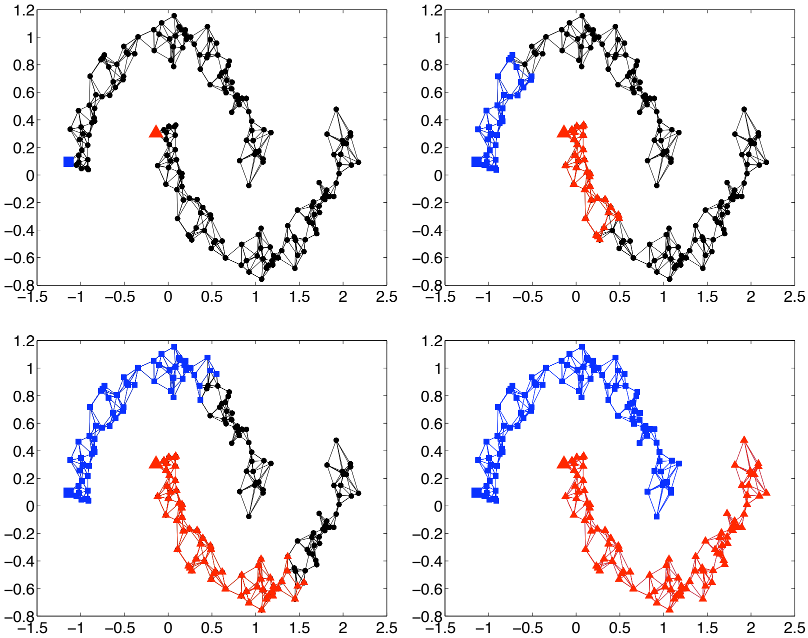


Propagation Rules

- Initial label vector: $\mathbf{y} \in \mathbb{R}^{n \times 1}$
  - $y_i = t_i$ if $\mathbf{x}_i$ is labeled as t_i; $y_i = 0$ if $\mathbf{x}_i$ is unlabeled
- $f_i^{(1)} = y_i$ if $\mathbf{x}_i$ is labeled; $f_i^{(1)} = \alpha \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{P}_{ij} y_j$ otherwise
  - $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the propagation matrix
  - Matrix form: $\mathbf{f}^{(1)} = \mathbf{y} + \alpha \mathbf{P}\mathbf{y}$
- $\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + \alpha \mathbf{P}\mathbf{f}^{(1)} = (\mathbf{I} + \alpha \mathbf{P} + \alpha^2 \mathbf{P}^2)\mathbf{y}$
- Finally $\mathbf{f}^{(\infty)} = \sum_{i=0}^{\infty} \alpha^i \mathbf{P}^i \mathbf{y} = (\mathbf{I} - \alpha \mathbf{P})^{-1} \mathbf{y}$



An Example



The Calculation of P

- Asymmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

- Symmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

- How to determine the optimal σ when computing $\mathbf{W}_{ij}$?
- Linear Neighborhood Similarity

$$\min_{\mathbf{W}_{ij}} \quad \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j\|^2$$

$$s.t. \quad \sum_j \mathbf{W}_{ij} = 1, \quad \mathbf{W}_{ij} \geq 0$$

The Calculation of P

- Asymmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

- Symmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

- How to determine the optimal σ when computing $\mathbf{W}_{ij}$?
- Linear Neighborhood Similarity

$$\min_{\mathbf{W}_{ij}} \quad \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j\|^2$$

$$s.t. \quad \sum_j \mathbf{W}_{ij} = 1, \quad \mathbf{W}_{ij} \geq 0$$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Semi-supervised Learning with Partially Labeled Data
Semi-supervised Learning Using Side-Information

The Calculation of P

- Asymmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

- Symmetrically Normalized Similarity Matrix:

$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

- How to determine the optimal σ when computing $\mathbf{W}_{ij}$?
- Linear Neighborhood Similarity

$$\min_{\mathbf{W}_{ij}} \quad \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j\|^2$$

$$s.t. \quad \sum_j \mathbf{W}_{ij} = 1, \quad \mathbf{W}_{ij} \geq 0$$

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

A Regularization Framework

- Label consistency: the predicted labels should be sufficiently close to the initial labels on the labeled data points
- Label smoothness: the predicted labels should be sufficiently smooth with respect to the data manifold (graph)

$$\min_{\mathbf{f}} \sum_{i=1}^{l}(f_i - t_i)^2 + \sum_{i=l+1}^{n} f_i^2 + \mu \sum_{i \sim j} w_{ij}(f_i - f_j)^2$$

  - The first term reflects label consistency
  - The second term guarantees the predicted label values should fall in a reasonable range for numerical stability
  - The third term reflects label smoothness
- $\mathbf{f} = (\mathbf{I} + \mu \mathbf{L})^{-1}\mathbf{y}$

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions
Semi-supervised Learning with Partially Labeled Data
Semi-supervised Learning Using Side-Information

A Regularization Framework

- Label consistency: the predicted labels should be sufficiently close to the initial labels on the labeled data points
- Label smoothness: the predicted labels should be sufficiently smooth with respect to the data manifold (graph)

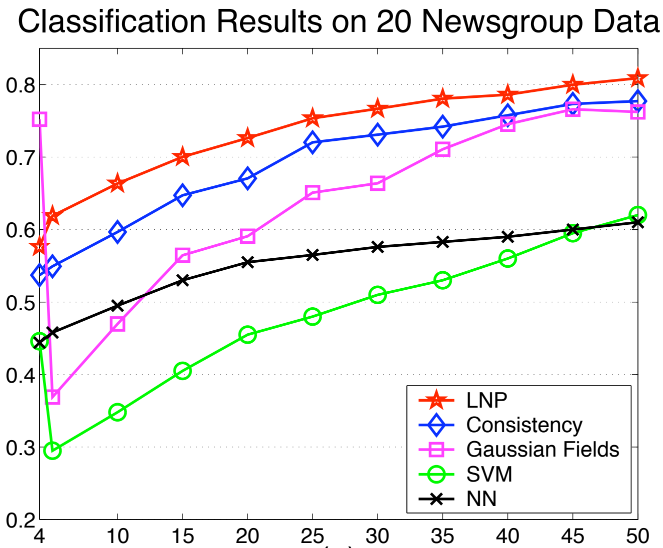
$$\min_{\mathbf{f}} \sum_{i=1}^{l}(f_i - t_i)^2 + \sum_{i=l+1}^{n} f_i^2 + \mu \sum_{i \sim j} w_{ij}(f_i - f_j)^2$$

  - The first term reflects label consistency
  - The second term guarantees the predicted label values should fall in a reasonable range for numerical stability
  - The third term reflects label smoothness
- $\mathbf{f} = (\mathbf{I} + \mu \mathbf{L})^{-1}\mathbf{y}$

Fei Wang, Tao Li, Chris Ding
Data Mining with Graphs & Matrices

Experimental Results on 20Newsgroup Data

autos, motorcycles, baseball, and hockey under rec



(a)

Table of Contents

What is Side-Information

- Types of side-information
  - Must-link: a pair of points should belong to the same class
  - Cannot-link: a pair of points should not appear in the same class
- Side-information is a type of prior knowledge weaker than partial labeling
  - Knowing the partial labeling, we can transform it into side-information
  - But not vice versa

Pairwise Constrained K-means Clustering

- *K-means* objective: $J_{km} = \sum_c \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2$
- Matrix form: $J_{km} = \left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|_F^2$
  - Cluster center matrix: $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_C] \in \mathbb{R}^{n \times C}$
  - $\mathbf{G} \in \mathbb{R}^{n \times C}$ with $\mathbf{G}_{ij} = 1$, if $\mathbf{x}_i \in \pi_j$; $\mathbf{G}_{ij} = 0$,otherwise.
- The objective of PCKM

$$J(\pi) = \sum_{c} \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i \neq l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij},$$

  - $\{\theta_{ij} \geqslant 0\}$: penalties for violating the must-link constraints
  - $\{\tilde{\theta}_{ij} \geqslant 0\}$: penalties for violating the cannot-link constraints

Pairwise Constrained K-means Clustering

- *K-means* objective: $J_{km} = \sum_c \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2$
- Matrix form: $J_{km} = \left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|_F^2$
  - Cluster center matrix: $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_C] \in \mathbb{R}^{n \times C}$
  - $\mathbf{G} \in \mathbb{R}^{n \times C}$ with $\mathbf{G}_{ij} = 1$, if $\mathbf{x}_i \in \pi_j$; $\mathbf{G}_{ij} = 0$,otherwise.
- The objective of PCKM

$$J(\pi) = \sum_c \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i \neq l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij},$$

  - $\{\theta_{ij} \geqslant 0\}$: penalties for violating the must-link constraints
  - $\{\tilde{\theta}_{ij} \geqslant 0\}$: penalties for violating the cannot-link constraints

Penalized Matrix Factorization

- Changing the penalties of violations in the constraints in $\mathcal{M}$ into the *awards* as

$$\begin{aligned} J(\pi) &= \sum_c \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 - \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i = l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij} \\ &= \sum_c \sum_{\mathbf{x}_i} \mathbf{G}_{ic} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_c \sum_{i,j} \mathbf{G}_{ic} \mathbf{G}_{jc} \Theta_{ij} \end{aligned}$$

- $\Theta_{ij} = \begin{cases} \tilde{\theta}_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ -\theta_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ 0, & \textit{otherwise} \end{cases}$
- Penalized matrix factorization objective

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \quad & J(\pi) = \left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|_F^2 + tr(\mathbf{G}^T \Theta \mathbf{G}) \\ s.t. \quad & \mathbf{G} \geqslant 0 \end{aligned} \quad (14)$$

Penalized Matrix Factorization

- Changing the penalties of violations in the constraints in $\mathcal{M}$ into the *awards* as

$$\begin{aligned} J(\pi) &= \sum_c \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 - \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i = l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij} \\ &= \sum_c \sum_{\mathbf{x}_i} \mathbf{G}_{ic} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_c \sum_{i,j} \mathbf{G}_{ic} \mathbf{G}_{jc} \Theta_{ij} \end{aligned}$$

  - $\Theta_{ij} = \begin{cases} \tilde{\theta}_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ -\theta_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ 0, & otherwise \end{cases}$

- Penalized matrix factorization objective

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \quad & J(\pi) = \left\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \right\|_F^2 + tr(\mathbf{G}^T \Theta \mathbf{G}) \\ s.t. \quad & \mathbf{G} \geqslant 0 \end{aligned} \qquad (14)$$

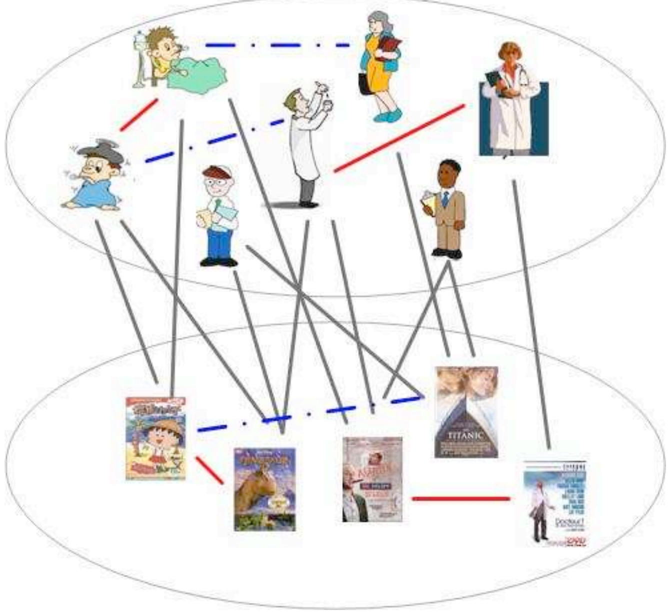
Updating Rules for PMF

Table: Penalized Matrix Factorization for Constrained Clustering

Inputs: Data matrix $\mathbf{X}$, Constraints matrix $\mathbf{\Theta}$.
Outputs: $\mathbf{F}$, $\mathbf{G}$.
1. Initialize $\mathbf{G}$;
2. Repeat the following steps until convergence:
   (a). Fixing $\mathbf{G}$, updating $\mathbf{F}$ by $\mathbf{F} = \mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$;
   (b). Fixing $\mathbf{F}$, updating $\mathbf{G}$ by

$$\mathbf{G}_{ij} \longleftarrow \mathbf{G}_{ij} \sqrt{\frac{(\mathbf{X}^T\mathbf{F})^+_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^-]_{ij} + \left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{X}^T\mathbf{F})^-_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^+]_{ij} + \left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}}.$$

Side-Information on Bi-partite Graph



PMF on Bi-partite Graph

$$\min_{\mathbf{G}_1 \geqslant 0, \mathbf{G}_2 \geqslant 0} J = \|\mathbf{R}_{12} - \mathbf{G}_1\mathbf{S}\mathbf{G}_2^T\|^2 + tr(\mathbf{G}_1^T\mathbf{\Theta}_1\mathbf{G}_1) + tr(\mathbf{G}_2^T\mathbf{\Theta}_2\mathbf{G}_2)$$

Table: PMF on Bi-partite Graph

Inputs: Relation matrix $\mathbf{R}_{12}$, Constraints matrices $\mathbf{\Theta}_1, \mathbf{\Theta}_2$.
Outputs: $\mathbf{G}_1$, $\mathbf{S}$, $\mathbf{G}_2$.
1. Initialize $\mathbf{G}_1$, $\mathbf{G}_2$;
2. Repeat the following steps until convergence:
   (a). Fixing $\mathbf{G}_1, \mathbf{G}_2$, updating $\mathbf{S}$ using
   $$\mathbf{S} \longleftarrow (\mathbf{G}_1^T\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2(\mathbf{G}_2^T\mathbf{G}_2)^{-1};$$
   (b). Fixing $\mathbf{S}, \mathbf{G}_2$, updating $\mathbf{G}_1$ using
   $$\mathbf{G}_{1ij} \leftarrow \mathbf{G}_{1ij}\sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^{+} + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^{-}]_{ij} + (\mathbf{\Theta}_1^{-}\mathbf{G}_1)_{ij}}{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^{-} + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^{+}]_{ij} + (\mathbf{\Theta}_1^{+}\mathbf{G}_1)_{ij}}};$$
   (c). Fixing $\mathbf{G}_1, \mathbf{S}$, updating $\mathbf{G}_2$ using
   $$\mathbf{G}_{2ij} \leftarrow \mathbf{G}_{2ij}\sqrt{\frac{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^{+} + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^{-}]_{ij} + (\mathbf{\Theta}_2^{-}\mathbf{G}_2)_{ij}}{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^{-} + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^{+}]_{ij} + (\mathbf{\Theta}_2^{+}\mathbf{G}_2)_{ij}}}.$$

Table: The F measure of three algorithms on the BBS data set

Data Sets	Algorithm	$d = 3$	$d = 4$	$d = 5$	$d = 6$
1	MLSA	0.7019	0.7079	0.7549	0.7541
1	SRC	0.7281	0.6878	0.6183	0.6183
1	Tri-SPMF	**0.7948**	**0.8011**	**0.8021**	**0.7993**
2	MLSA	0.7651	0.7429	0.7581	0.7309
2	SRC	0.7627	0.7226	0.7280	0.6965
2	Tri-SPMF	**0.8007**	**0.7984**	**0.7938**	**0.7896**
3	MLSA	0.6689	0.6511	0.6987	0.7301
3	SRC	0.7556	0.7666	0.7472	0.7125
3	Tri-SPMF	**0.8095**	**0.8034**	**0.7993**	**0.7874**

Tensor & Hypergraph Based Methods

- In knowledge & information management, we usually face with multi-relational data
  - Graph based methods can capture the pairwise relationships
  - Matrix is also only composed of two dimensions
- Hypergraph is more efficient in describing the multiple-wise relationships
- Tensor is also a structure that can capture multiple-wise relationships

Efficient & Large Scale Methods

- Matrix & Graph based methods usually involve high computational cost
  - eigenvalue decomposition
  - solving large scale linear equation systems
  - constrained optimization
- How to make the algorithm more efficient?
  - Exploring the sparsity
- How to improve scalability?
  - Smart sampling

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions

Efficient & Large Scale Methods

- Matrix & Graph based methods usually involve high computational cost
  - eigenvalue decomposition
  - solving large scale linear equation systems
  - constrained optimization
- How to make the algorithm more efficient?
  - Exploring the sparsity
- How to improve scalability?
  - Smart sampling

Fei Wang, Tao Li, Chris Ding Data Mining with Graphs & Matrices

Efficient & Large Scale Methods

- Matrix & Graph based methods usually involve high computational cost
  - eigenvalue decomposition
  - solving large scale linear equation systems
  - constrained optimization
- How to make the algorithm more efficient?
  - Exploring the sparsity
- How to improve scalability?
  - Smart sampling

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions

Efficient & Large Scale Methods

- Matrix & Graph based methods usually involve high computational cost
  - eigenvalue decomposition
  - solving large scale linear equation systems
  - constrained optimization
- How to make the algorithm more efficient?
  - Exploring the sparsity
- How to improve scalability?
  - Smart sampling

Fei Wang, Tao Li, Chris Ding Data Mining with Graphs & Matrices

Probabilistic Interpretations

- Potential problems of describing the data with matrices
  - Too large
  - Too complicated
  - Missing entries
  - Noisy entries
  - ......
- Probabilistic interpretations & graphical models
  - Discover latent structures
  - Relationships with matrix based methods?

Knowledge Transfer Across Different Domains

- The multi-relational data contain data points from different domains
  - We may easily get some prior knowledge on some domains
- How to transfer the knowledge from one domain to another?
  - What knowledge to transfer?
  - How?
  - Is it really helps?

Knowledge Transfer Across Different Domains

- The multi-relational data contain data points from different domains
  - We may easily get some prior knowledge on some domains
- How to transfer the knowledge from one domain to another?
  - What knowledge to transfer?
  - How?
  - Is it really helps?

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions

- Mikhail Belkin, Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. NIPS 2001.
- A. Banerjee, S. Basu, S. Merugu. Multi-way Clustering on Relation Graphs. SDM 2007.
- I. S. Dhillon, Y. Guan, and B. Kulis. Weighted Graph Cuts without Eigenvectors: A Multilevel Approach. PAMI 2007.
- I. S. Dhillon, S. Mallela, and D. S. Modha. Information-Theoretic Co-clustering. KDD 2003.
- I. S. Dhillon. Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning. KDD 2001.
- Chris Ding, Xiaofeng He, and Horst D. Simon. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. SDM 2005.
- Chris Ding, Tao Li, Wei Peng, Haesun Park. Orthogonal Nonnegative Matrix Tri-factorizations for Clustering. KDD 2006.
- Chris Ding, Tao Li, Michael I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. Technical Report. 2006.
- Chris Ding, Rong Jin, Tao Li, and Horst D. Simon. A Learning Framework Using Green's Function and Kernel Regularization with Application for Recommender System KDD 2007.

Fei Wang, Tao Li, Chris Ding | Data Mining with Graphs & Matrices

- Xiaofei He and Partha Niyogi. Locality Preserving Projections. NIPS 2003.
- Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma. Locality Preserving Indexing for Document Representation. SIGIR 2004.
- D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. Nature 1999.
- D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. NIPS 2000.
- Tao Li, Chris Ding, Yi Zhang, and Bo Shao. Knowledge Transformation from Word Space to Document Space. SIGIR 2008.
- Tao Li, Chris Ding, and Michael Jordan. Solving Consensus and Semi-supervised Clustering Problems Using Nonnegative Matrix Factorization ICDM 2007.
- Tao Li and Chris Ding. The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering. ICDM 2006.
- Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. Science 2000.
- Hanhuai Shan and Arindam Banerjee. Bayesian Co-clustering. ICDM 2008.
- J. Shi and J. Malik. Normalized Cuts and Image Segmentation. PAMI 2000.

Graphs and Matrices are Everywhere
Unsupervised Learning with Graphs & Matrices
Semi-supervised Learning with Graphs & Matrices
Future Research Directions

- Fei Wang, Shouchun Chen, Tao Li, Changshui Zhang. Semi-Supervised Metric Learning by Maximizing Constraint Margin. CIKM 2008.
- Fei Wang, Tao Li and Changshui Zhang. Semi-Supervised Clustering via Matrix Factorization. SDM 2008.
- Fei Wang, Sheng Ma, Liuzhong Yang, Tao Li. Recommendation on Item Graphs. ICDM 2006.
- Fei Wang, Changshui Zhang. Label Propagation Through Linear Neighborhoods. ICML 2006.
- X. Wang, J. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. SIGIR 2006.
- Wei Xu, Xin Liu, Yihong Gong. Document Clustering Based on Non-negative Matrix Factorization. SIGIR 2003.
- S. Yan, D. Xu, B. Zhang and H. Zhang. Graph Embedding: A General Framework for Dimensionality Reduction. CVPR 2005.
- D. Zhou , O. Bousquet, T.N. Lal, J. Weston and B. Schölkopf. Learning with Local and Global Consistency. NIPS 2003.
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. ICML 2003.

Fei Wang, Tao Li, Chris Ding Data Mining with Graphs & Matrices

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

Thank You!

`http://feiwang03.googlepages.com/sdm-tutorial`