# Learning with Local Consistency

**Deng Cai (蔡登)**

College of Computer Science
Zhejiang University

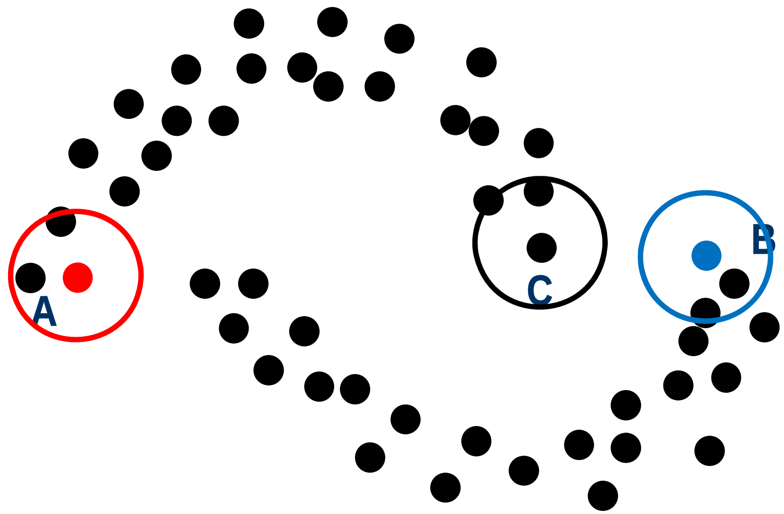
dengcai@gmail.com

Chinese Workshop on Machine Learning and Applications
2010

# What is Local Consistency?

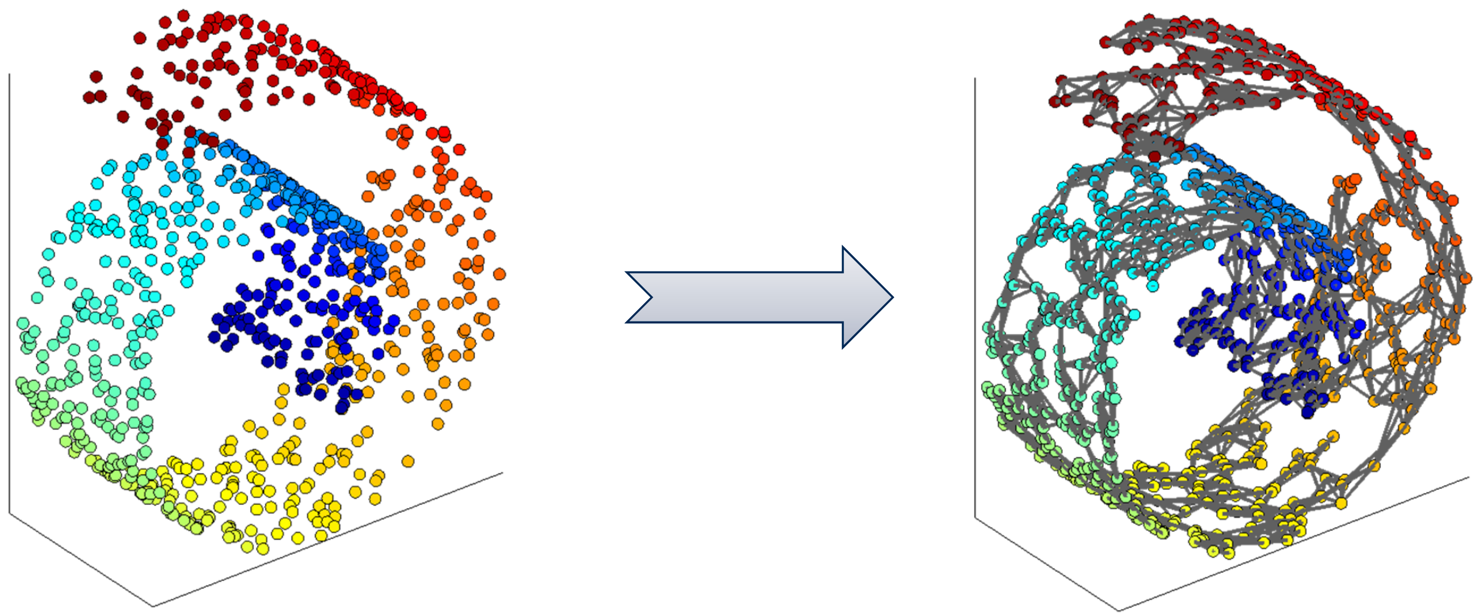
- **Nearby points (neighbors) share *similar properties*.**
- Traditional machine learning algorithms:
  - $k$-nearest neighbor classifier

# Local Consistency Assumption




- A lot of **unlabeled** data
- **Local** consistency
  - $k$-nearest neighbors
  - $\epsilon$-neighbors
  - …

# Local Consistency Assumption



- Put edges between neighbors (nearby data points)
- Two nodes in the graph connected by an edge share *similar properties*.

# Local Consistency Assumption

- Similar ***properties***
  - Labels
  - Representations
  - $\boldsymbol{x}$: $f(\boldsymbol{x})$

- $W \in \mathcal{R}^{n\times n}$: weight matrix of the graph

$$\min \frac{1}{2}\sum_{i,j} W_{ij}\left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2$$

$$y_i = f(\boldsymbol{x}_i)$$

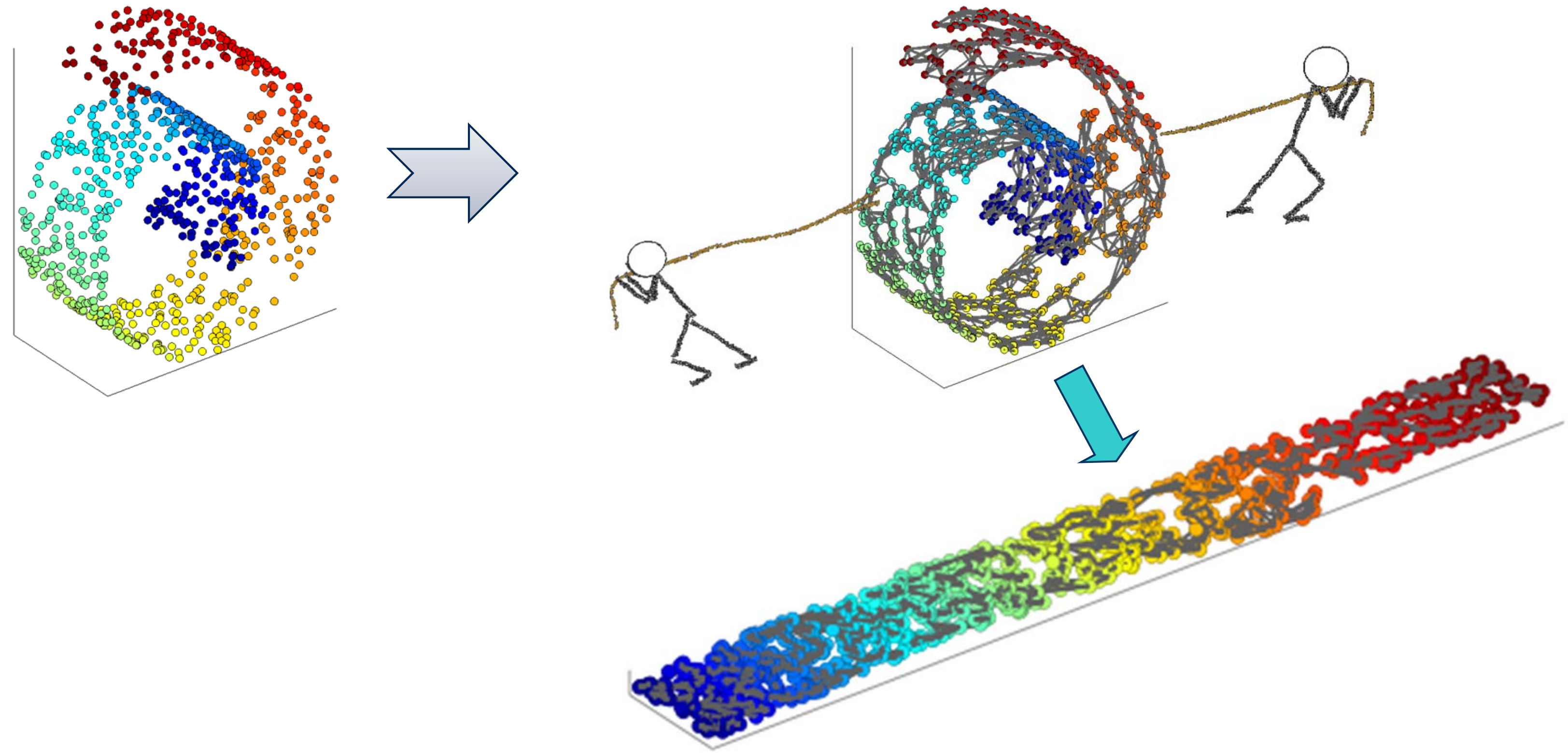
$$\boldsymbol{y} = [y_1, \cdots, y_n]^T$$

$$\min y^T(D - W)y \qquad L \equiv D - W$$

$$\min y^T L y$$
$$s.t. \quad y^T D y = 1$$

M. Belkin, and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, NIPS 2001.

# Local Consistency and Manifold Learning



- Manifold learning
- We only need **local consistency**

$$\min \sum_{i,j} W_{ij} \left( f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2$$

- How to use the local consistency idea?

# Local Consistency in Semi-Supervised Learning

- Supervised learning

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} l(\mathbf{x}_i, y_i, f) + \lambda ||f||^2$$

  - Squared loss: ridge regression (regularized least squares)
  - Hinge loss: SVM

- Semi-Supervised learning (with local consistency)

$$f^* = \underset{f}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} l(\mathbf{x}_i, y_i, f) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j=1}^{n} W_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2$$

  - Laplacian least squares and Laplacian SVM.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: a Geometric Framework for Learning from Labeled and Unlabeled Examples, Journal of Machine Learning Research, 7(Nov):2399-2434, 2006.

# Manifold Regularization

- Semi-Supervised learning (with local consistency)

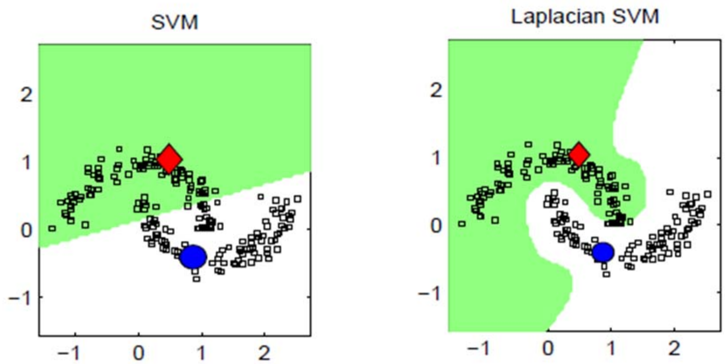
$$f^* = \operatorname*{argmin}_f \frac{1}{m}\sum_{i=1}^{m} l(\mathbf{x}_i, y_i, f) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j=1}^{n} W_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2$$

- Laplacian least squares

$$a^* = (XX^T + \lambda_1 I + \lambda_2 XLX^T)^{-1} X\boldsymbol{y}$$

- Ridge regression (regularized least squares)

$$a^* = (XX^T + \lambda I)^{-1} X\boldsymbol{y}$$




M. Belkin, P. Niyogi, and V. Sindhwani. Manifold Regularization: a Geometric Framework for Learning from Labeled and Unlabeled Examples, Journal of Machine Learning Research, 7(Nov):2399-2434, 2006.

# How to use the local consistency idea?

- Matrix factorization
  - Non-negative matrix factorization
- Topic modeling
  - Probabilistic latent semantic analysis
- Clustering
  - Gaussian mixture model

# Matrix Factorization (Decomposition)

- 
- $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathcal{R}^{p \times n} \rightarrow X \approx UV^T$

$$X \approx \tilde{X} = UV^T$$

approximation — left factor — right factor

# Matrix Factorization (Decomposition)

$$X \approx UV^T$$

$$\underset{m}{\overset{n}{\begin{bmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ x_{13} & x_{23} & \cdots & x_{n3} \\ \vdots & \vdots & & \vdots \\ x_{1m} & x_{2m} & \cdots & x_{nm} \end{bmatrix}}} \approx m \overset{k}{\begin{bmatrix} u_{11} & \cdots & u_{k1} \\ u_{12} & \cdots & u_{k2} \\ u_{13} & \cdots & u_{k3} \\ \vdots & & \vdots \\ u_{1m} & \cdots & u_{km} \end{bmatrix}} \times k \overset{n}{\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix}}$$

$$\begin{bmatrix} \\ \mathbf{x}_i \\ \\ \end{bmatrix} \approx v_{1i} \cdot \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} + v_{2i} \cdot \begin{bmatrix} \\ \mathbf{u}_2 \\ \\ \end{bmatrix} + \cdots + v_{ki} \cdot \begin{bmatrix} \\ \mathbf{u}_k \\ \\ \end{bmatrix}$$

# Singular Value Decomposition

- For an arbitrary matrix $X$ there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

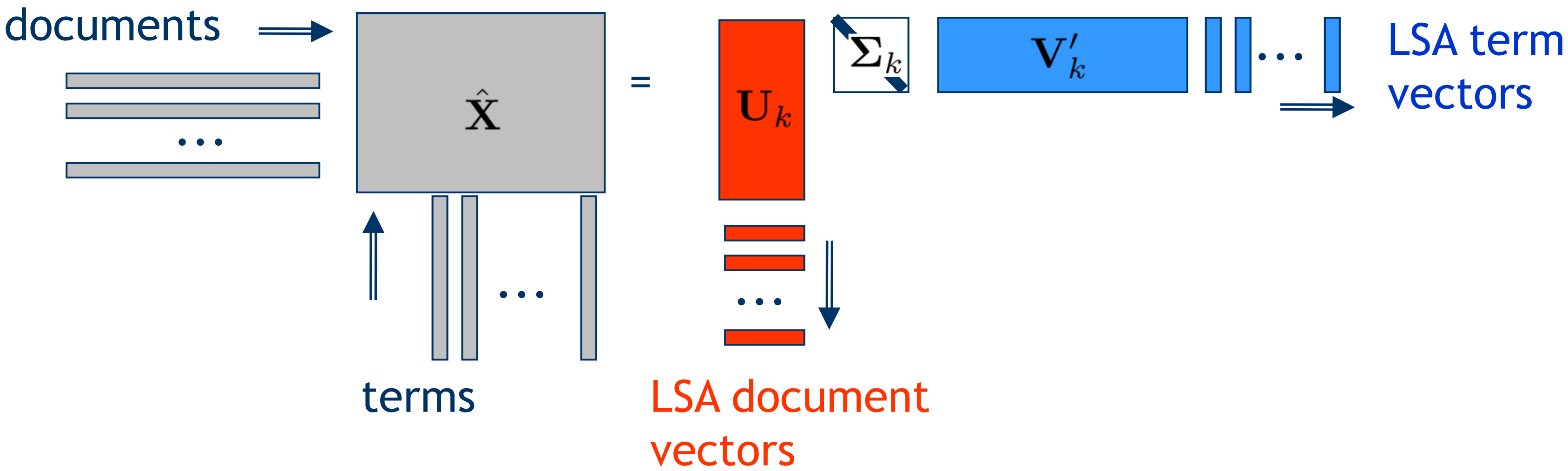
$$X = U \Sigma V^T \in \mathcal{R}^{n \times m}$$

- Where
  - (i) $U \in \mathcal{R}^{n \times k}$ $\quad \Sigma \in \mathcal{R}^{k \times k}$ $\quad V \in \mathcal{R}^{m \times k}$
  - (ii) $\mathbf{U}'\mathbf{U} = \mathbf{I}$ $\quad \mathbf{V}'\mathbf{V} = \mathbf{I}$ — Orthonormal columns
  - (iii) $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_k), \; \sigma_i \geq \sigma_{i+1}$ — Singular values (ordered)
  - (iv) $k = \mathrm{rank}(X)$

C. Eckart, G. Young, The approximation of a matrix by another of lower rank. Psychometrika, 1, 211-218, 1936.

# Latent Semantic Analysis (Indexing)

- The LSA via SVD can be summarized as follows:




- Document **similarity**

- Folding-in **queries**

$$\hat{\mathbf{q}} = \Sigma_k^{-1} \mathbf{V}_k \mathbf{q}$$


M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. SIAM Review, 37(4):573-595, 1995.

# Non-negative Matrix Factorization

- $$X \approx \tilde{X} = UV^T, \min\|X - UV^T\|^2$$
  $$u_{ij} \geq 0, v_{ij} \geq 0$$

- *The Euclidean distance $\|X - UV^T\|^2$ is nonincreasing under the update rules*
  $$u_{ik} \leftarrow \frac{(XV)_{ik}}{(UV^TV)_{ik}} u_{ik} \qquad v_{jk} \leftarrow \frac{(X^TU)_{jk}}{(VU^TU)_{jk}} v_{jk}$$

- Can we incorporate the local consistency idea?

D. D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, NIPS 13, pp. 556-562, 2001.

# Locally Consistent NMF

$$X \approx UV^T$$

If $x_i$ and $x_j$ are neighbors

$$\mathbf{x}_i = v_{1i} \cdot \mathbf{u}_1 + v_{2i} \cdot \mathbf{u}_2 + \cdots + v_{ki} \cdot \mathbf{u}_k$$

$$\mathbf{x}_j = v_{1j} \cdot \mathbf{u}_1 + v_{2j} \cdot \mathbf{u}_2 + \cdots + v_{kj} \cdot \mathbf{u}_k$$

- Neighbor: prior knowledge, label information, *p*-nearest neighbors …

D. Cai, X. He, J. Han, and T. Huang, Graph regularized Non-negative Matrix Factorization for Data Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

# Locally Consistent NMF

$$\begin{bmatrix} \mathbf{x}_i \end{bmatrix} = v_{1i} \cdot \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} + v_{2i} \cdot \begin{bmatrix} \mathbf{u}_2 \end{bmatrix} + \cdots + v_{ki} \cdot \begin{bmatrix} \mathbf{u}_k \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x}_j \end{bmatrix} = v_{1j} \cdot \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} + v_{2j} \cdot \begin{bmatrix} \mathbf{u}_2 \end{bmatrix} + \cdots + v_{kj} \cdot \begin{bmatrix} \mathbf{u}_k \end{bmatrix}$$

$$\min \sum_{i,j} W_{ij} \left( f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2 \qquad \min \sum_k \sum_{i,j} W_{ij} (v_{ki} - v_{kj})^2$$

$$\min \mathrm{Tr}(V^T L V)$$

D. Cai, X. He, J. Han, and T. Huang, Graph regularized Non-negative Matrix Factorization for Data Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

# Objective Function

NMF: $\min \|X - UV^T\|^2$

$$u_{ik} \leftarrow \frac{(XV)_{ik}}{(UV^TV)_{ik}} u_{ik} \qquad v_{jk} \leftarrow \frac{(X^TU)_{jk}}{(VU^TU)_{jk}} v_{jk}$$

GNMF: $\min \|X - UV^T\|^2 + \lambda \mathrm{Tr}(V^TLV)$

Graph regularized NMF

$$u_{ik} \leftarrow \frac{(XV)_{ik}}{(UV^TV)_{ik}} u_{ik} \qquad v_{jk} \leftarrow \frac{(X^TU + \lambda WV)_{jk}}{(VU^TU + \lambda DV)_{jk}} v_{jk}$$

D. Cai, X. He, J. Han, and T. Huang, Graph regularized Non-negative Matrix Factorization for Data Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

# Clustering Results

| $K$ | NMF | GNMF |
|---|---|---|
| 4 | 81.0±14.2 | **93.5±10.1** |
| 6 | 74.3±10.1 | **92.4±6.1** |
| 8 | 69.3±8.6 | **84.0±9.6** |
| 10 | 69.4±7.6 | **84.4±4.9** |
| 12 | 69.0±6.3 | **81.0±8.3** |
| 14 | 67.6±5.6 | **79.2±5.2** |
| 16 | 66.0±6.0 | **76.8±4.1** |
| 18 | 62.8±3.7 | **76.0±3.0** |
| 20 | 60.5 | **75.3** |
| Avg. | 68.9 | **82.5** |

COIL20

| $K$ | NMF | GNMF |
|---|---|---|
| 5 | 95.5±10.2 | **98.5±2.8** |
| 10 | 83.6±12.2 | **91.4±7.6** |
| 15 | 79.9±11.7 | **93.4±2.7** |
| 20 | 76.3±5.6 | **91.2±2.6** |
| 25 | 75.0±4.5 | **88.6±2.1** |
| 30 | 71.9 | **88.6** |
| Avg. | 80.4 | **92.0** |

TDT2

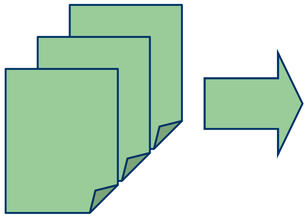
- Please check our papers for more details.
- http://www.zjucadcg.cn/dengcai/GNMF/index.html

# How to use the local consistency idea?

- Matrix factorization
  - Non-negative matrix factorization
- Topic modeling
  - Probabilistic latent semantic analysis
- Clustering
  - Gaussian mixture model

# What is Topic Modeling

Text Collection



Probabilistic Topic Modeling

Topic models (Multinomial distributions)

| | |
|---|---|
| term | 0.16 |
| relevance | 0.08 |
| weight | 0.07 |
| feedback | 0.04 |
| … | |

| | |
|---|---|
| web | 0.21 |
| search | 0.10 |
| link | 0.08 |
| graph | 0.05 |
| … | |

…

- Powerful tool for text mining
  - Topic discovery,
  - Summarization,
  - Opinion mining,
  - Many more …

# Language Model Paradigm in IR

- Probabilistic relevance model
  - Random variables

$$R_d \in \{0, 1\} \quad : \quad \text{relevance of document } d$$
$$q \subseteq \Sigma \quad : \quad \text{query, set of words}$$

  - Bayes' rule

probability of generating a query q to ask for relevant d

prior probability of relevance for document d (e.g. quality, popularity)

$$P(R_d = 1|q) = \frac{P(q|R_d = 1) \cdot P(R_d = 1)}{P(q)}$$

probability that document d is relevant for query q





J. Ponte and W.B. Croft, A Language Model Approach to Information Retrieval, ACM SIGIR, 1998.

# Language Model Paradigm

$$P(R_d = 1|q) \propto \underbrace{P(q|R_d = 1)}_{②} \; \underbrace{P(R_d = 1)}_{①}$$

- First contribution: **prior probability of relevance**
  - ① simplest case: uniform (drops out for ranking)
  - **popularity**: document usage statistics (e.g. library circulation records, download or access statistics, hyperlink structure)

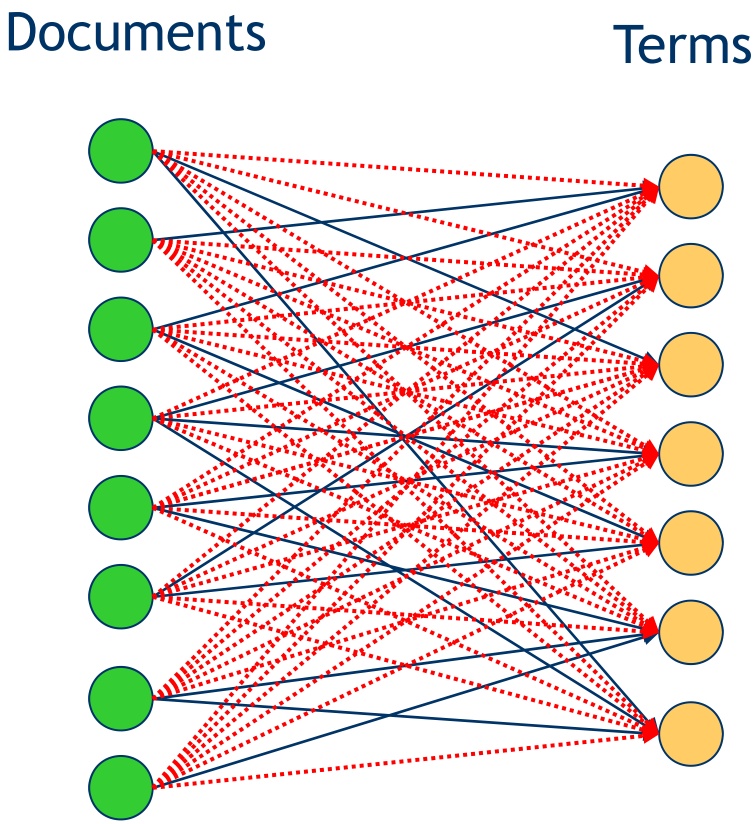
- Second contribution: **query likelihood**
  - ② query terms $q$ are treated as a **sample** drawn from an (unknown) relevant document

# Query Likelihood

- $$P(q|R_d = 1) \equiv P(q|d)$$
- $q = (w_1, \cdots, w_q)$
- Independent Assumption
$$P(q|d) = \Pi_{w \in q} P(w|d)$$

$$P(w|d)?$$

# Naive Approach



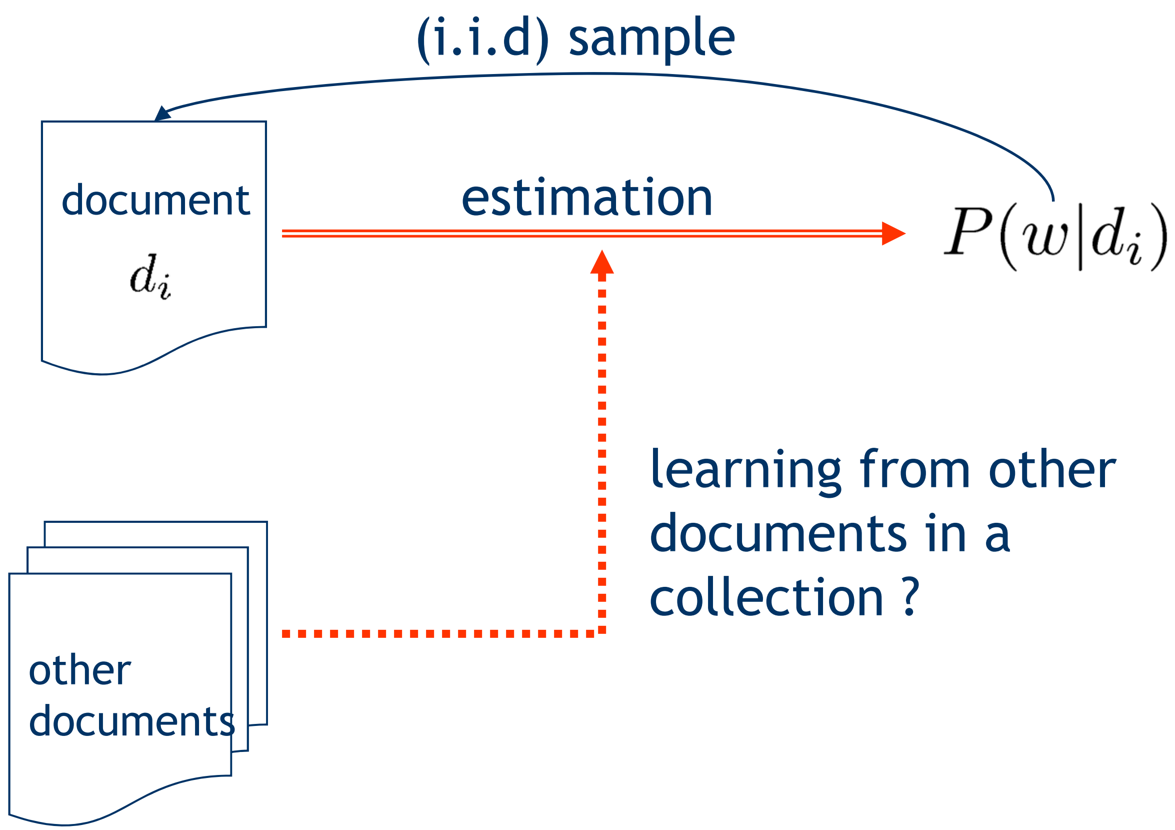

Maximum Likelihood Estimation

number of occurrences of term w in document d

$$\hat{P}_{\mathrm{ML}}(w|d) = \frac{n(d,w)}{\sum_{w'} n(d,w')}$$

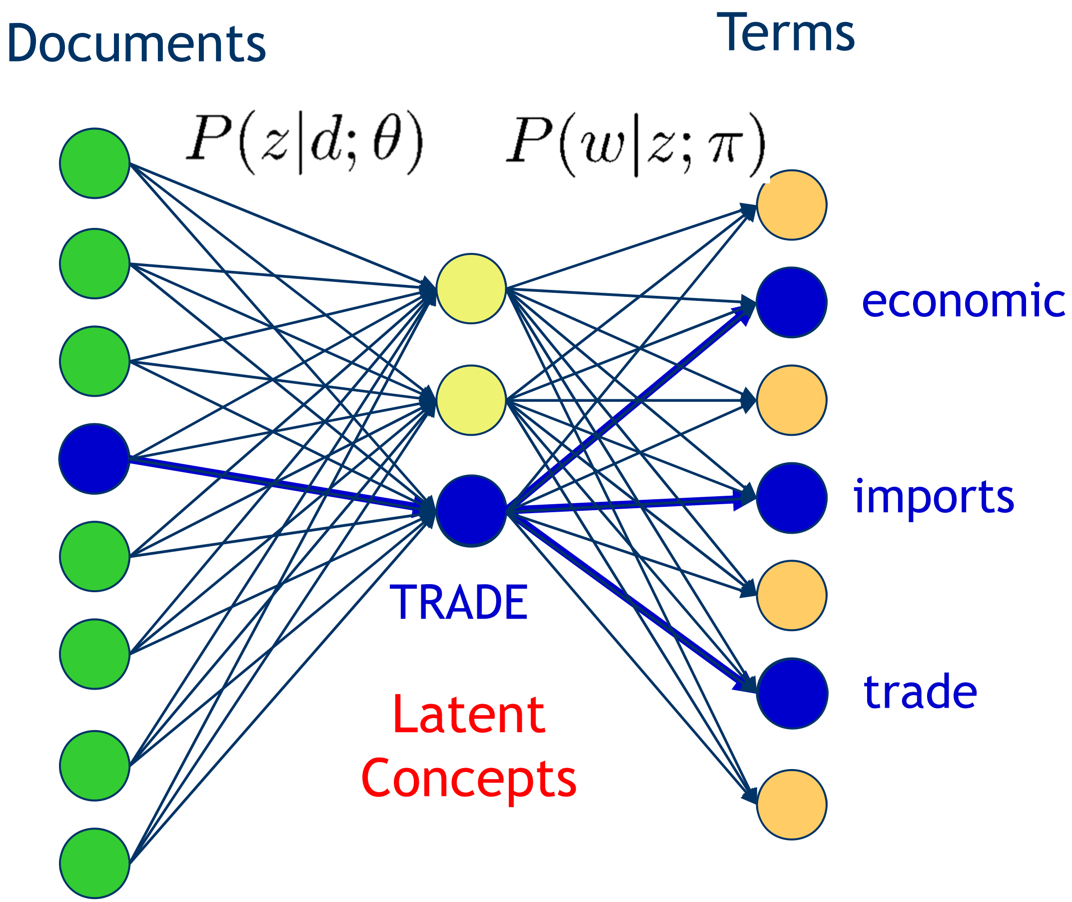
Zero frequency problem: terms not occurring in a document get zero probability

# Estimation Problem




- **Crucial question**: In which way can the document collection be utilized to improve probability estimates?

# Probabilistic Latent Semantic Analysis




Model fitting

$$\hat{P}_{\text{LSA}}(w|d) = \sum_z P(w|z;\theta)P(z|d;\pi)$$



T. Hofmann, Probabilistic Latent Semantic Analysis, UAI 1999.

# pLSA via Likelihood Maximization

- **Log-Likelihood**

$$l(\theta, \pi; \mathbf{N}) = \sum_{d,w} n(d,w) \log(\sum_{z} P(w|z;\theta)P(z|d;\pi))$$

- **Goal**: Find model parameters that maximize the log-likelihood, i.e. maximize the average predictive probability for observed word occurrences (**non-convex optimization problem**)

# Expectation Maximization Algorithm

- E step: posterior probability of latent variables ("concepts")

$$P(z|d,w) = \frac{P(z|d;\pi)P(w|z;\theta)}{\sum_{z'} P(z'|d;\pi)P(w|z';\theta)}$$

Probability that the occurence of term w in document d can be "explained" by concept z

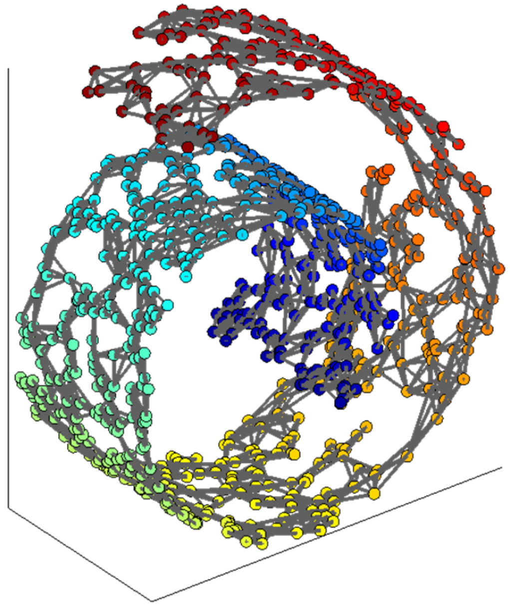
- M step: parameter estimation based on "completed" statistics

$$P(w|z;\theta) \propto \sum_{d} n(d,w)P(z|d,w),$$

$$P(z|d;\pi) \propto \sum_{w} n(d,w)P(z|d,w)$$

A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of Royal Statistical Society B, vol. 39, no. 1, pp. 1-38, 1977

# Local Consistency ?

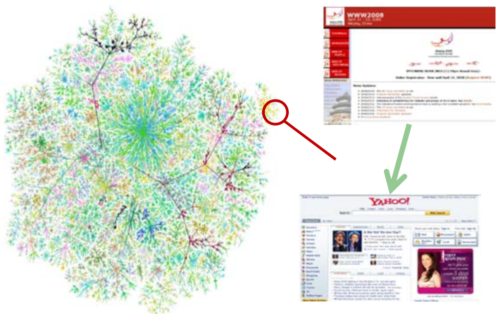


- Put edges between neighbors (nearby data points);
- Two nodes in the graph connected by an edge share similar properties.
- **Network data**
  - Co-author network, facebook, webpage

# Text Collections with Network Structure

Blog articles + friend network

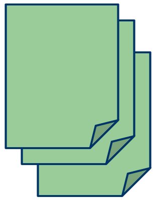
News + geographic network


Web page +
hyperlink structure

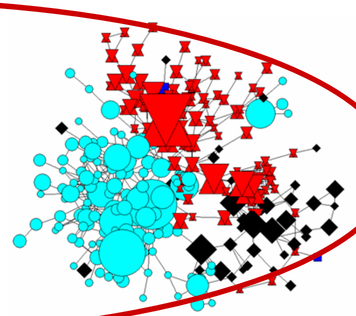
- Literature + coauthor/citation network
- Email + sender/receiver network
- …

# Importance of Topic Modeling on Network



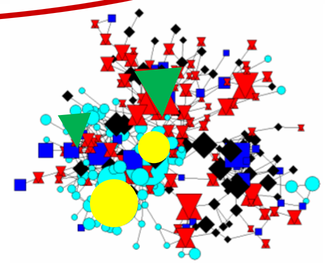
Computer Science Literature =

Information Retrieval +
Data Mining +
Machine Learning, …



or

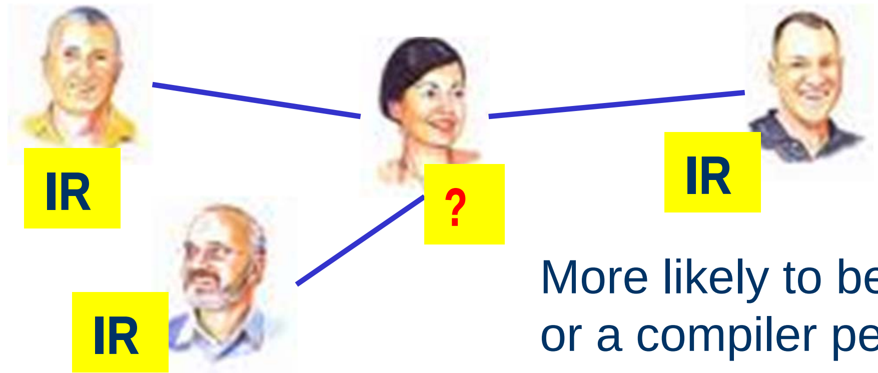
Domain Review +
Algorithm +
Evaluation, …



?

# Intuitions

- People working on the same topic belong to the same "topical community"
- Good community: coherent topic + well connected
- A topic is semantically coherent if people working on this topic also collaborate a lot

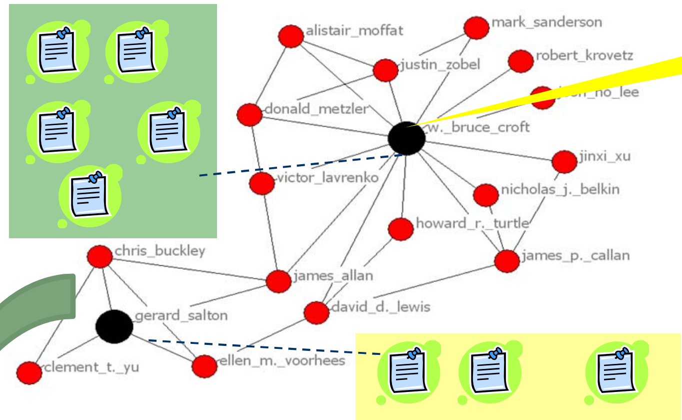



Intuition: my topics are similar to my neighbors

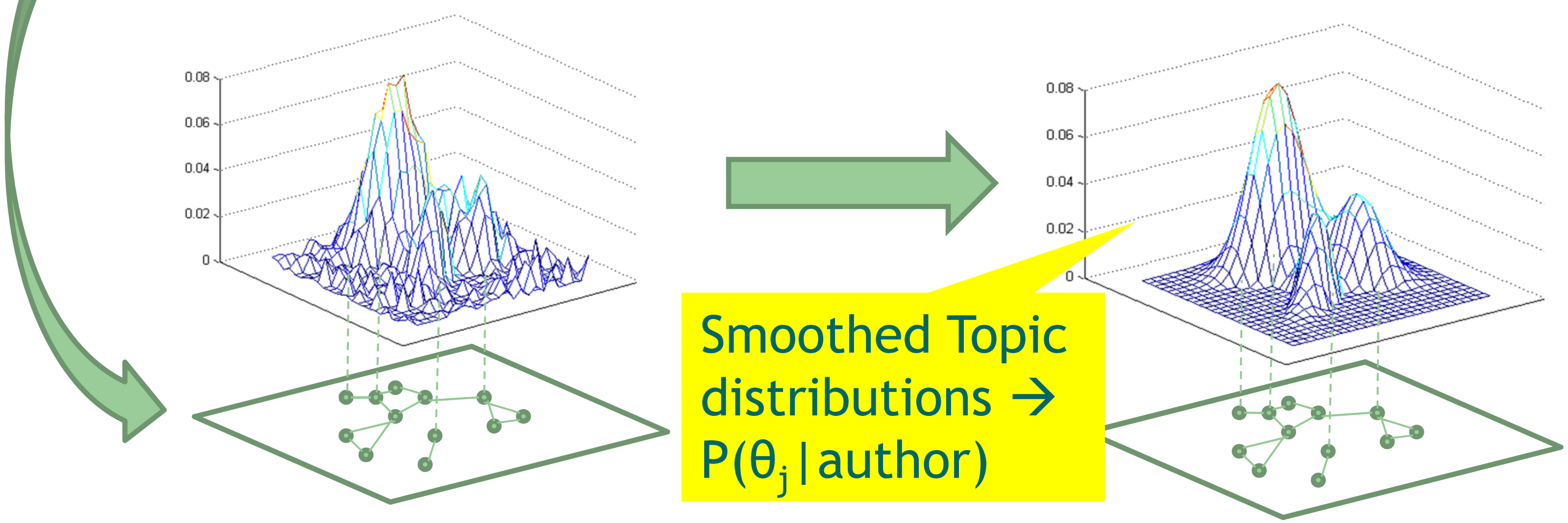
More likely to be an IR person or a compiler person?

# Social Network Context for Topic Modeling

e.g. coauthor network




- Context = author
- Coauthor = similar contexts
- Intuition: I work on similar topics to my neighbors




D. Cai, X. Wang, and X. He, Probabilistic Dyadic Data Analysis with Local and Global Consistency, ICML'09.
Q. Mei, D. Cai, D. Zhang, and C. Zhai, Topic Modeling with Network Regularization, WWW'08.

# Objective Function

$$l(\theta, \pi; \mathbf{N}) = \sum_{d,w} n(d,w) \log(\sum_z P(w|z;\theta)P(z|d;\pi)) + \lambda R$$

$$\min \sum_{i,j} W_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2 \qquad f(\boldsymbol{x}_i) = f(d_i) \equiv P(z|d_i)$$

$$D\left(P(z|d_i)||P(z|d_j)\right) = \sum_z P(z|d_i) \log \frac{P(z|d_i)}{P(z|d_j)}$$

$$R = -\frac{1}{2}\sum_{i,j} W_{ij} \left(D\left(P(z|d_i)||P(z|d_j)\right) + D\left(P(z|d_j)||P(z|d_i)\right)\right)^2$$





D. Cai, X. Wang, and X. He, Probabilistic Dyadic Data Analysis with Local and Global Consistency, ICML'09.

# Parameter Estimation via EM

- E step: posterior probability of latent variables ("concepts")

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)}$$

Same as PLSA

- M step: parameter estimation based on "completed" statistics

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i, w_m)P(z_k|d_i, w_m)}$$

Same as PLSA

$$P(z_k \mid d_i) = ?$$





D. Cai, X. Wang, and X. He, Probabilistic Dyadic Data Analysis with Local and Global Consistency, ICML'09.

# Parameter Estimation via EM

- M step: parameter estimation based on "completed" statistics

$$\begin{bmatrix} P(z_k \mid d_1) \\ P(z_k \mid d_2) \\ \vdots \\ P(z_k \mid d_N) \end{bmatrix} = (\Omega + \lambda L)^{-1} \begin{bmatrix} \sum_{j=1}^{M} n(d_1, w_j) P(z_k \mid d_1, w_j) \\ \sum_{j=1}^{M} n(d_2, w_j) P(z_k \mid d_2, w_j) \\ \vdots \\ \sum_{j=1}^{M} n(d_N, w_j) P(z_k \mid d_N, w_j) \end{bmatrix}$$

$$\Omega = \begin{bmatrix} n(d_1) & & \\ & \ddots & \\ & & n(d_N) \end{bmatrix}$$

$L = D - W,$

Graph Laplacian

If $\lambda = 0$

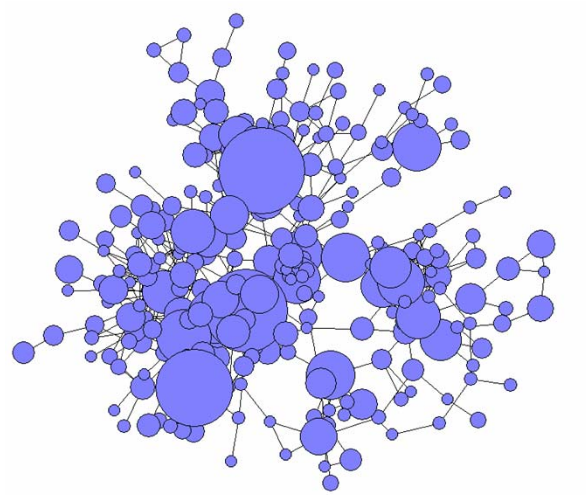
$$P(z_k \mid d_i) = \sum_{j=1}^{M} n(d_i, w_j) P(z_k \mid d_i, w_j) / n(d_i)$$

Same as PLSA

D. Cai, X. Wang, and X. He, Probabilistic Dyadic Data Analysis with Local and Global Consistency, ICML'09.

# Experiments

- Bibliography data and coauthor networks
  - DBLP: text = titles; network = coauthors
  - Four conferences (expect 4 topics): SIGIR, KDD, NIPS, WWW

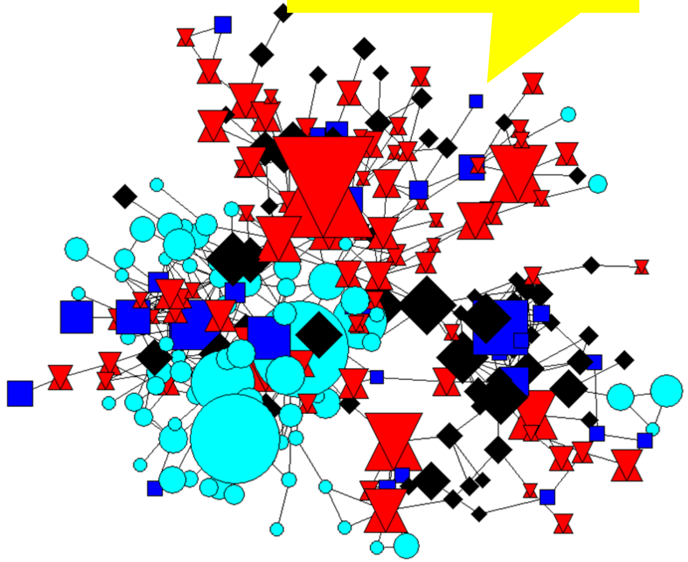


# Topical Communities with PLSA

| Topic 1 | | Topic 2 | | Topic 3 | | Topic 4 | |
|---|---|---|---|---|---|---|---|
| term | 0.02 | peer | 0.02 | visual | 0.02 | interface | 0.02 |
| question | 0.02 | patterns | 0.01 | analog | 0.02 | towards | 0.02 |
| protein | 0.01 | mining | 0.01 | neurons | 0.02 | browsing | 0.02 |
| training | 0.01 | clusters | 0.01 | vlsi | 0.01 | xml | 0.01 |
| weighting | 0.01 | stream | 0.01 | motion | 0.01 | generation | 0.01 |
| multiple | 0.01 | frequent | 0.01 | chip | 0.01 | design | 0.01 |
| recognition | 0.01 | e | 0.01 | natural | 0.01 | engine | 0.01 |
| relations | 0.01 | page | 0.01 | cortex | 0.01 | service | 0.01 |
| library | 0.01 | gene | 0.01 | spike | 0.01 | social | 0.01 |

? ? ? ?

*Noisy community assignment*



# Topical Communities with NetPLSA

| Topic 1 | | Topic 2 | | Topic 3 | | Topic 4 | |
|---|---|---|---|---|---|---|---|
| retrieval | 0.13 | mining | 0.11 | neural | 0.06 | web | 0.05 |
| information | 0.05 | data | 0.06 | learning | 0.02 | services | 0.03 |
| document | 0.03 | discovery | 0.03 | networks | 0.02 | semantic | 0.03 |
| query | 0.03 | databases | 0.02 | recognition | 0.02 | services | 0.03 |
| text | 0.03 | rules | 0.02 | analog | 0.01 | peer | 0.02 |
| search | 0.03 | association | 0.02 | vlsi | 0.01 | ontologies | 0.02 |
| evaluation | 0.02 | patterns | 0.02 | neurons | 0.01 | rdf | 0.02 |
| user | 0.02 | frequent | 0.01 | gaussian | 0.01 | management | 0.01 |
| relevance | 0.02 | streams | 0.01 | network | 0.01 | ontology | 0.01 |

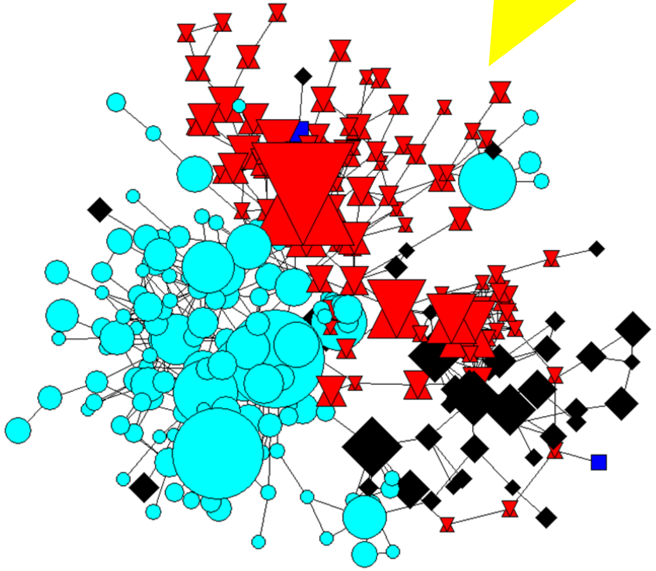
Information Retrieval

Data mining

Machine learning

Web

Coherent community assignment



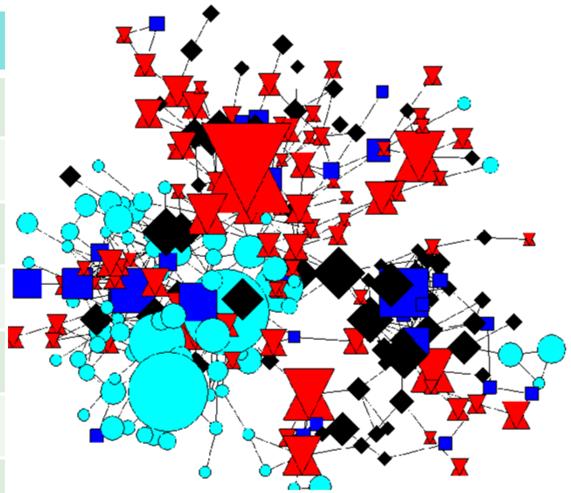


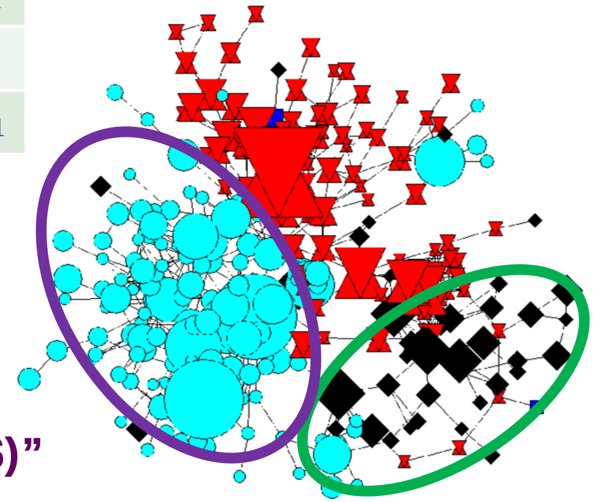
Q. Mei, D. Cai, D. Zhang, and C. Zhai, Topic Modeling with Network Regularization, WWW'08.

# Coherent Topical Communities

| PLSA | |
|---|---|
| visual | 0.02 |
| analog | 0.02 |
| neurons | 0.02 |
| vlsi | 0.01 |
| motion | 0.01 |
| chip | 0.01 |
| natural | 0.01 |
| cortex | 0.01 |
| spike | 0.01 |

| NetPLSA | |
|---|---|
| neural | 0.06 |
| learning | 0.02 |
| networks | 0.02 |
| recognition | 0.02 |
| analog | 0.01 |
| vlsi | 0.01 |
| neurons | 0.01 |
| gaussian | 0.01 |
| network | 0.01 |





**Semantics of community:**

**"machine learning (NIPS)"**

| PLSA | |
|---|---|
| peer | 0.02 |
| patterns | 0.01 |
| mining | 0.01 |
| clusters | 0.01 |
| stream | 0.01 |
| frequent | 0.01 |
| e | 0.01 |
| page | 0.01 |
| gene | 0.01 |

| NetPLSA | |
|---|---|
| mining | 0.11 |
| data | 0.06 |
| discovery | 0.03 |
| databases | 0.02 |
| rules | 0.02 |
| association | 0.02 |
| patterns | 0.02 |
| frequent | 0.01 |
| streams | 0.01 |

**Semantics of community:**

**"Data Mining (KDD)"**

Q. Mei, D. Cai, D. Zhang, and C. Zhai, Topic Modeling with Network Regularization, WWW'08.

# For More Detials

- Please check our papers
- http://www.zjucadcg.cn/dengcai/LapPLSA/index.html



D. Cai, X. Wang, and X. He, Probabilistic Dyadic Data Analysis with Local and Global Consistency, ICML'09.
Q. Mei, D. Cai, D. Zhang, and C. Zhai, Topic Modeling with Network Regularization, WWW'08.

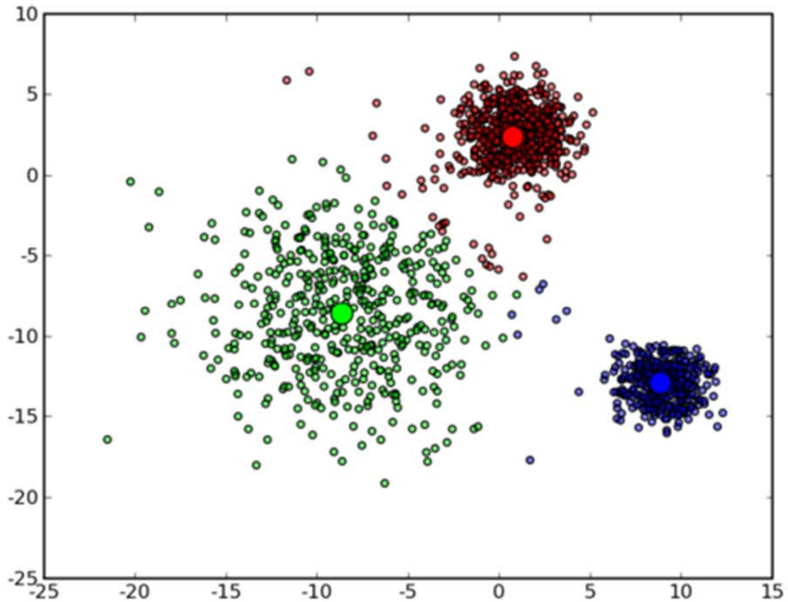
# How to use the local consistency idea?

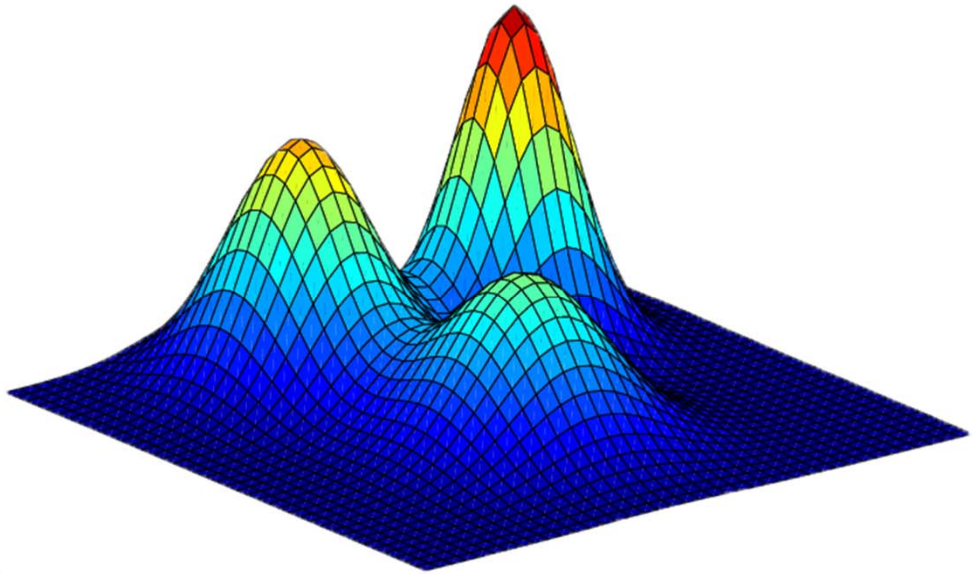
- Matrix factorization
  - Non-negative matrix factorization
- Topic modeling
  - Probabilistic latent semantic analysis
- Clustering
  - Gaussian mixture model

# Gaussian Mixture Model



- Gaussian Mixture Model (GMM) is one of the most popular clustering methods which can be viewed as a linear combination of different Gaussian components.



# Gaussian Mixture Model

- Multivariate Gaussian
  - $\boldsymbol{\mu}$: mean of the distribution
  - $\Sigma$: covariance of the distribution

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$$

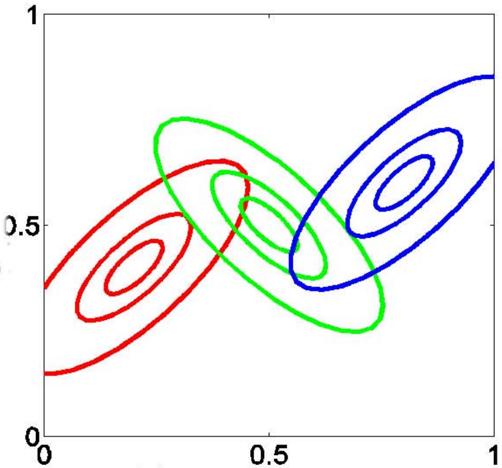
  - Maximum likelihood estimation

$$\begin{cases} \hat{\boldsymbol{\mu}} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \boldsymbol{x_i} \\ \hat{\Sigma} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} (\boldsymbol{x_i} - \hat{\boldsymbol{\mu}})(\boldsymbol{x_i} - \hat{\boldsymbol{\mu}})^{\boldsymbol{T}} \end{cases}$$

# Gaussian Mixture Model

- Linear combination of Gaussians
  - Assumption: $K$ Gaussians, each has a contribution of $\pi_k$ to the data points

$$
\begin{cases}
p(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x};\boldsymbol{\theta}_k) \\
\boldsymbol{\Theta} = \{\pi_1,\cdots,\pi_K,\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_K\}, \sum_{k=1}^{k} \pi_k = 1, \pi_k \in [0, \\
p_k(\boldsymbol{x};\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)
\end{cases}
$$

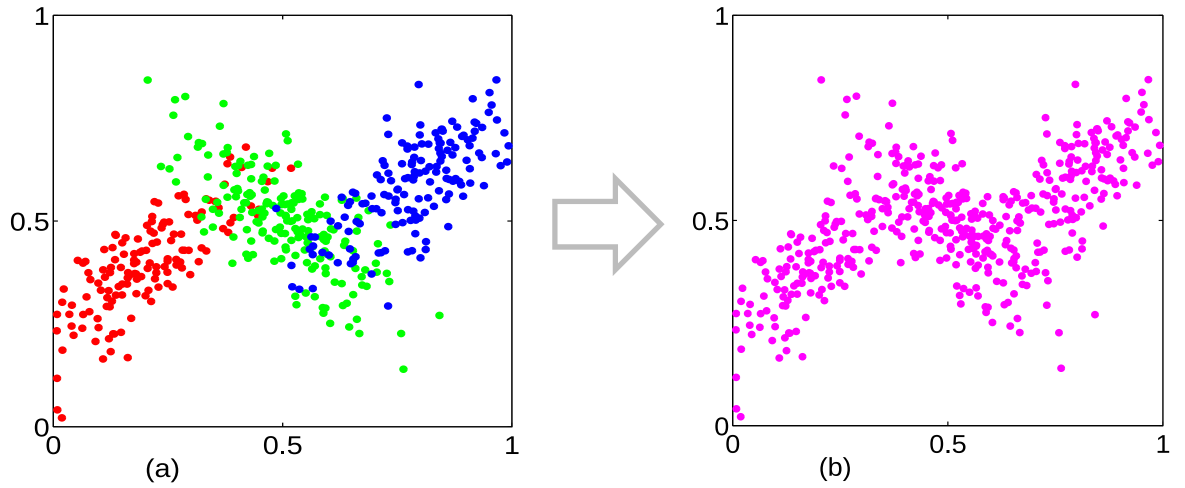


(a)

  - Parameters to be estimated: $\pi_k, \boldsymbol{\mu}_k, \Sigma_k$

# Gaussian Mixture Model

- The process of generating a data point
  - first pick one of the components with probability $\pi_k$
  - then draw a sample $\boldsymbol{x}_i$ from that component distribution
- Each data point is generated by one of $k$ components



# Gaussian Mixture Model

- The log-likelihood function:

$$\log \prod_{i=1}^{N} p\left(\boldsymbol{x}^{(i)}; \boldsymbol{\Theta}\right) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right)$$

- Using EM algorithm:

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{M} \sum_{\boldsymbol{z}^{(i)}} Q^i\left(\boldsymbol{z}^{(i)}\right) \log \frac{p\left(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}\right)}{Q^i\left(\boldsymbol{z}^{(i)}\right)}$$

$$\equiv \sum_{i=1}^{M} \sum_{k=1}^{K} Q^i\left(\boldsymbol{z}_k^{(i)}\right) \log \pi_k \mathcal{N}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

- E-step:

$$Q^i\left(\boldsymbol{z}_k^{(i)}\right) = p\left(\boldsymbol{z}_k^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\Theta}\right)$$

$$= \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

- M-step:
  - Take the derivative of the complete log likelihood to obtain estimates for $\pi_k, \mu_k, \Sigma_k$ directly

$$\pi_k = \frac{\sum_{i=1}^{M} Q^i\left(\boldsymbol{z}_k^{(i)}\right)}{M}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{M} \boldsymbol{x}^{(i)} Q^i\left(\boldsymbol{z}_k^{(i)}\right)}{\sum_{i=1}^{M} Q^i\left(\boldsymbol{z}_k^{(i)}\right)}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{M}\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_k\right)^T Q^i\left(\boldsymbol{z}_k^{(i)}\right)}{\sum_{i=1}^{M} Q^i\left(\boldsymbol{z}_k^{(i)}\right)}$$

- Do the iterations until convergence, then $Q^i\left(\boldsymbol{z}_k^{(i)}\right)$ can be used for clustering

# Objective Function

$$\min \sum_{i,j} W_{ij} \left( f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2 \qquad f(\boldsymbol{x}_i) \equiv P(z|\boldsymbol{x}_i)$$

$$D\left(P(z|\boldsymbol{x}_i)||P(z|\boldsymbol{x}_j)\right) = \sum_z P(z|\boldsymbol{x}_i) \log \frac{P(z|\boldsymbol{x}_i)}{P(z|\boldsymbol{x}_j)}$$

$$R = -\frac{1}{2} \sum_{i,j} W_{ij} \left( D\left(P(z|\boldsymbol{x}_i)||P(z|\boldsymbol{x}_j)\right) + D\left(P(z|\boldsymbol{x}_j)||P(z|\boldsymbol{x}_i)\right) \right)^2$$

$$\sum_{i=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) + \lambda R$$





J. Liu, D. Cai, and X. He, Gaussian Mixture Model with Local Consistency, AAAI'10.

# EM Equations

- E-step: $$P(c_k \mid x_i) = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}$$
- M-step:

$$S_{i,k} = (x_i - \mu_k)(x_i - \mu_k)^T$$

$$N_k = \sum_{i=1}^{N} P(c_k \mid x_i)$$

$$\pi_k = \frac{\sum_{i=1}^{N} P(c_k \mid x_i)}{N}$$

$$\mu_k = \frac{\sum_{i=1}^{N} x_i P(c_k \mid x_i)}{N_k} - \frac{\lambda \sum_{i,j=1}^{N} \big(P(c_k \mid x_i) - P(c_k \mid x_j)\big)(x_i \text{-} x_j) W_{ij}}{2N_k}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} P(c_k \mid x_i) S_{i,k}}{N_k} - \frac{\lambda \sum_{i,j=1}^{N} \big(P(c_k \mid x_i) - P(c_k \mid x_j)\big)\big(S_{i,k} - S_{j,k}\big) W_{ij}}{2N_k}$$

original GMM part

# Experiment

7 Real Data sets :

- The Yale face image database.
- The Waveform model described in “The Elements of Statistical Learning” .
- The Vowels data set which has steady state vowels of British English.
- The Libras movement data set containing hand movement pictures.
- The Control Charts data set consisting control charts.
- The Cloud data set is a simple 2 classes problem.
- The Breast Cancer Wisconsin data set computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

# Clustering Results

| Data set | LCGMM | GMM | K-means | Ncut | size | # of features | # of classes |
|---|---|---|---|---|---|---|---|
| Yale | 54.3 | 29.1 | 51.5 | **54.6** | 165 | 4096 | 15 |
| Libras | **50.8** | 35.8 | 44.1 | 48.6 | 800 | 21 | 3 |
| Chart | **70.0** | 56.8 | 61.5 | 58.8 | 990 | 10 | 11 |
| Cloud | **100.0** | 96.2 | 74.4 | 61.5 | 360 | 90 | 15 |
| Breast | **95.5** | 94.7 | 85.4 | 88.9 | 600 | 60 | 6 |
| Vowel | **36.6** | 31.9 | 29.0 | 29.1 | 2048 | 10 | 2 |
| Waveform | 75.3 | **76.3** | 51.9 | 52.3 | 569 | 30 | 2 |

# The Take-home Messages

- Local consistency is a very useful idea.
- It is very simple.
  - Nearby points (neighbors) share similar properties.

$$\min \sum_{i,j} W_{ij} \left( f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2$$

- It can be put everywhere (with a lot of unlabeled data)
  - The key: how to optimize the regularized objective function.

# Thanks!