Clustering

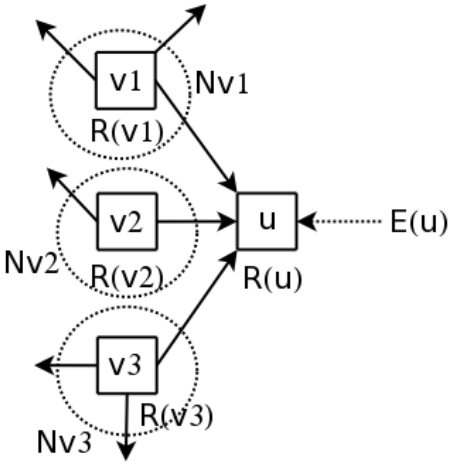
Chen-Hsiang Yeang
Institute of Statistical Science
Academia Sinica

Clustering: definition, uses

Clustering: group data points according to some similarity concept.

What can clustering do?

PageRank



$R(u)$: rank of webpage u, N_v: # pages eminating from v.

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} + cE(u).$$

A random surfer model:

- Start from an arbitrary webpage.
- Randomly jump to one connected page with a uniform probability.
- $\mathbf{A}$: connection matrix.
- Occasionally explore an unlinked page.
- $R(u) \propto$ probability of visiting u.

$$\mathbf{R} = c(\mathbf{AR} + \mathbf{E}) = c(\mathbf{A} + \mathbf{E} \times \mathbf{I})\mathbf{R}.$$

The PageRank citation ranking: bringing order to the web. 1998. Page et al.

Outline

- Clustering by geometry.
  - K-means algorithm.
  - EM algorithm.
  - Hierarchical clustering.
  - Self Organizing Map.
  - Principal Component Analysis, Singular Value Decomposition.
  - Independent Component Analysis.
- Clustering on graphs.
  - Basic concepts.
  - Max flow - Min cut.
  - Normal cuts. Spectral clustering.
  - Community detection.
- Advanced topics.
  - Chinese restaurant process.
  - Affinity propagation.

K-means clustering

- Number of clusters is fixed to c.
- Assign each data point a fixed membership.
- Update the data point memberships and cluster means alternatively.

$$p(y = k|x; \theta^t) = \begin{cases} 1 & \text{if } d(x, \mu_k) = \min_{k'} d(x, \mu_{k'}). \\ 0 & \text{otherwise.} \end{cases}$$

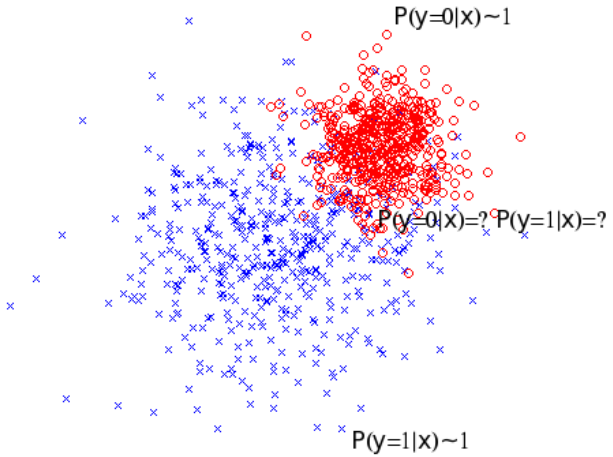
Algorithm:

- Initialize $\mu_1, \cdots, \mu_c$.
- do
  - Assign each data point to the nearest μ_i.
  - Recompute μ_i.

  until no change in μ_i.
- Return $\mu_1, \cdots, \mu_c$ and data point memberships.

Expectation-Maximization algorithm: mixture of Gaussians

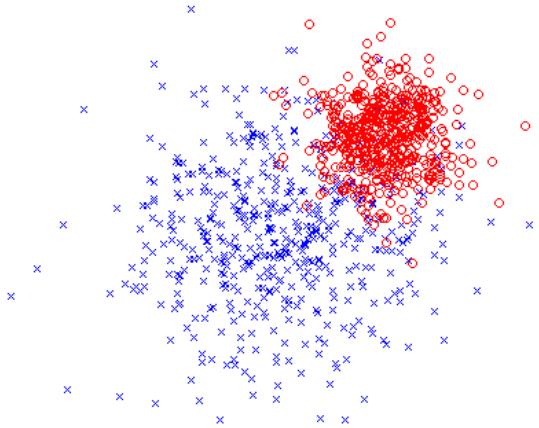
What about *soft assignments*?



$$p(y = k|x; \theta) > 0 \text{ for } k = 1, \cdots, c.$$

Expectation-Maximization algorithm: mixture of Gaussians

Mixture of two Gaussians:



Goals:

1. Estimate the mean and variance of each Gaussian distribution.

2. Assign the membership probabilities of each data point.

Expectation-Maximization algorithm: mixture of Gaussians

$$
\begin{aligned}
p(x, y = k|\theta) &= p(y = k)\frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}. \\
g(x; \mu_k, \sigma_k) &\equiv \frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}. \\
\theta &\equiv (\mu_k, \sigma_k). \\
L(X, Y|\theta) &= \prod_i p(y_i)\frac{1}{\sqrt{2\pi\sigma_{y_i}^2}}e^{-\frac{(x_i-\mu_{y_i})^2}{2\sigma_{y_i}^2}}.
\end{aligned}
$$

Problem: Simultaneous estimation of θ and $P(Y)$ is difficult.

Solution: Iteratively fix θ or $P(Y)$ and optimize the other.

Expectation-Maximization algorithm: mixture of Gaussians

- E-step:

$$p(y_i|x_i;\theta^t) \quad = \quad \frac{p(y_i|\theta^t)p(x_i|y_i;\theta^t)}{\sum_y p(y|\theta^t)p(x_i|y;\theta^t)}.$$

- M-step:

$$\theta^{t+1} \quad = \quad \arg\max_\theta \sum_Y P(Y|X,\theta^t) \log L(X,Y|\theta).$$

Iteratively execute E and M steps until both θ and $P(Y|X,\theta)$ converge.

Expectation-Maximization algorithm: general concepts

Goal: Find the optimal model to fit the data with missing variables.

Idea:

- D_o: observed data. D_h: hidden data.
- E-step: Estimate the joint probability given the observed data and a model.
- $Q(\theta;\theta^t) = E_h[\log P(D_o, D_h;\theta|D_o,\theta^t)]$.
- $E_h[.]$ denotes taking expectation over $P(D_h|D_o,\theta^t)$.
- M-step: Find the optimal parameters for the completed probability.
- $\theta^{t+1} = \arg\max_\theta Q(\theta;\theta^t)$.

Algorithm:

- Initialize $\theta^0, t = 0$.
- do $t \leftarrow t+1$.
  - E-step: compute $P(D_h|D_o,\theta^t)$, $Q(\theta;\theta^t)$.
  - M-step: $\theta^{t+1} \leftarrow \arg\max_\theta Q(\theta;\theta^t)$.
- until $Q(\theta^{t+1};\theta^t) - Q(\theta^t;\theta^{t-1}) \leq T$.

Expectation-Maximization algorithm: mixture models

Why does $Q(\theta; \theta^t)$ make sense?

x: observed variables, y: hidden variables.

- Ideally, we should maximize the expected likelihood over y.
- Equivalent to maximizing log(expected likelihood).
- Cumbersome to deal with log of a linear function.
- Thus convert it into a linear combination of log functions.

$$\begin{aligned} \mathbf{L} &= \sum_i \log E_{p(y|x_i;\theta^t)}[p(x_i, y|\theta)] \\ &\geq \sum_i E_{p(y|x_i;\theta^t)}[\log p(x_i, y|\theta)] \\ &= \sum_i \sum_y p(y|x_i; \theta^t) \log p(x_i, y|\theta) \\ &\equiv Q(\theta; \theta^t). \end{aligned}$$

Jensen's inequality: $\log E_x[x] \geq E_x[\log x]$.

Iteratively estimate $Q(\theta; \theta^t)$ and compute the optimal parameters.

Expectation-Maximization algorithm: mixture of Gaussians

- E step:

$$p(y|x_i;\theta^t) = \frac{p(y|\theta^t)p(x_i|y;\theta^t)}{\sum_y p(y|\theta^t)p(x_i|y;\theta^t)}.$$

- M step:

$$\begin{aligned}
\mu_k^{t+1} &= \frac{\sum_i p(y|\theta^t)x_i}{\sum_i p(y|\theta^t)}. \\
\sigma_k^{t+1} &= \sqrt{\frac{\sum_i p(y|\theta^t)(x_i-\mu_k^{t+1})^2}{\sum_i p(y|\theta^t)}}. \\
p(y|\theta^{t+1}) &= \frac{1}{N}\sum_i p(y|x_i;\theta^t).
\end{aligned}$$

A short tutorial on using expectation-maximization with mixture models, Jason Rennie
Estimating Gaussian mixture densities with EM – a tutorial, Carlo Tomasi.

Finding the number of clusters: model selection

- What about the number of clusters is unknown a priori?
- This problem pertains to a more general and important topic: model selection.
- How do we select among a collection of models to fit the data?

Finding the number of clusters: model selection

Suppose there are n clusters. Each cluster constitutes one data point.

$$L(X, Y|\theta) = \prod_{i=1}^{n} P(y_i = i) \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_i - x_i)^2}{2\sigma_i^2}} = 1.$$

This clustering yields the highest joint likelihood. But is this optimal?

Finding the number of clusters: model selection

- To select a model, we should consider both fitness of the data and model complexity.
- The more complex a model is, the less likely it can be generalized beyond the training data.
- Occam's razor: one should proceed to simpler theories until simplicity can be traded for greater explanatory power.
- Bayesian Information Criteria: $BIC = 2l_{\mathcal{M}}(x;\widehat{\theta}) - m_{\mathcal{M}}\log(n)$.
- $\mathcal{M}$: model.
- $\widehat{\theta}$: maximum likelihood parameter(s).
- $l_{\mathcal{M}}(x;\widehat{\theta})$: maximum log-likelihood.
- $m_{\mathcal{M}}$: number of free parameters in $\mathcal{M}$.
- n: number of data points.
- $2l_{\mathcal{M}}(x;\widehat{\theta})$: fitness to the data.
- $m_{\mathcal{M}}\log(n)$: penalty for model complexity.
- Vary the cluster number and calculate BIC accordingly. Choose the cluster number with the highest BIC.

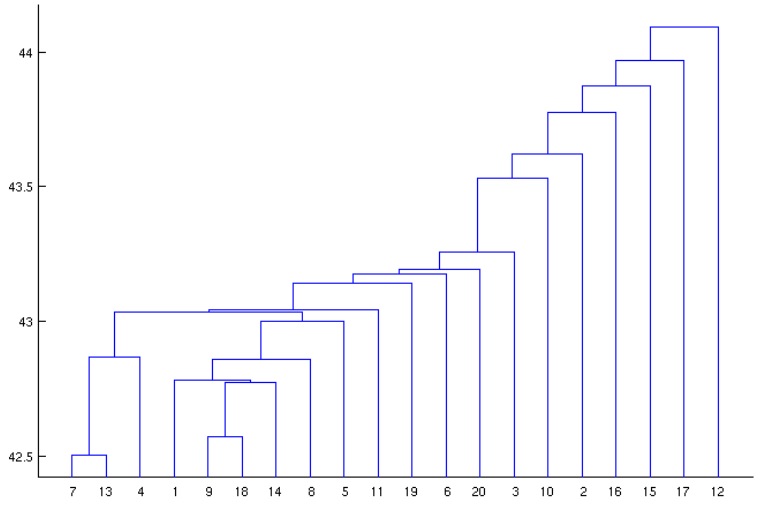
C. Fraley and A. Raftery, The Computer Journal 1998, 41(8):578-588.

Hierarchical clustering

Agglomeratively merge similar clusters.

- Initially assign each data point i to a cluster C_i. N data points.
- For $n = 1$ to N
  - Find the nearest pair of clusters C_i and C_j.
  - $C_i \leftarrow C_i \cup C_j$.
  - Remove C_j from the list.

The outcome of the algorithm is a dendrogram.



Self Organizing Map

Motivation: How to map a set of points (cells) to a manifold and preserve the topology of these points?

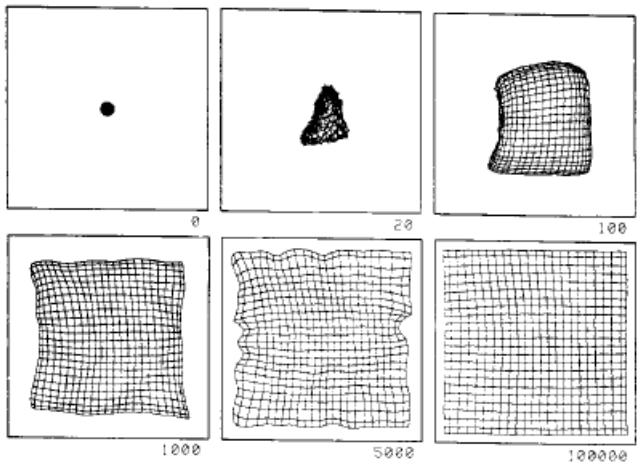


Fig. 3. Weight vectors during the ordering process, two-dimensional array.

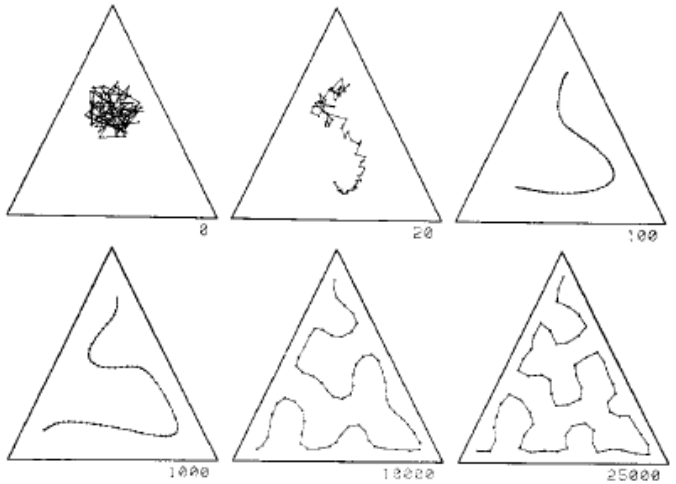


Fig. 4. Weight vectors during the ordering process, one-dimensional array.

Self Organizing Map

Idea: Start with a set of *weight vectors* with a fixed topology. Move the weight vectors to fit the data points.

Algorithm:

- Initialize weight vectors $\mathbf{m}_1(0), \cdots, \mathbf{m}_K(0)$.
- do $t \leftarrow t + 1$.
  - Pick up a data point $\mathbf{x}(t)$.
  - Find the closest weight vector $\mathbf{m}_c(t) = \arg\min_{\mathbf{m}_i(t)} d(\mathbf{x}(t), \mathbf{m}_i(t))$.
  - Find the neighboring weight vectors $N_c(t)$ of $\mathbf{m}_c(t)$.
  - Update the weight vectors in $N_c(t)$:

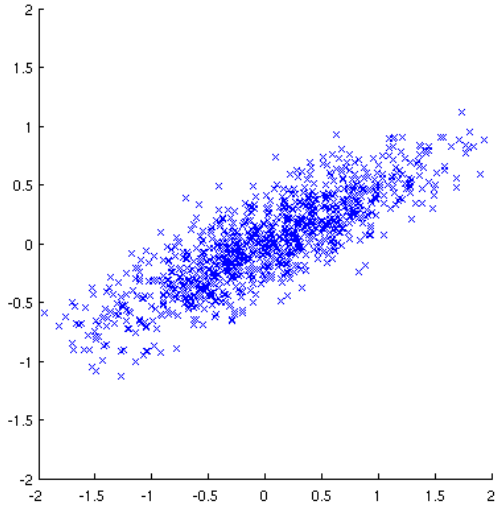
$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] & \text{if } i \in N_c(t). \\ \mathbf{m}_i(t) & \text{otherwise.} \end{cases}$$

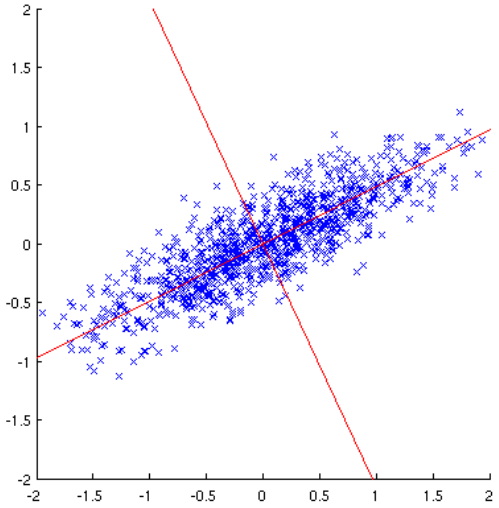
- until $\mathbf{m}_1(t), \cdots, \mathbf{m}_K(t)$ are stationary.

The self-organizing map. Kohonen T., 1990. Proceedings of the IEEE 78(9):1464-1480.

Principal Component Analysis

Motivation: What is the projected subspace that captures the variation of the data?





Problem: Find the coordinate transformation that diagonalize the covariance matrix.

Principal Component Analysis

Original data (shifted to zero-mean):

$$\mathbf{X} = \left(\begin{array}{c|c|c|c} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{array} \right).$$

X has n data points, each data point has m dimensions.

Coordinate transformation:

$$\mathbf{P} = \left(\begin{array}{c} \mathbf{p}_1 \\ \hline \mathbf{p}_2 \\ \hline \vdots \\ \hline \mathbf{p}_m \end{array} \right).$$

$\mathbf{p}_i \cdot \mathbf{p}_j = \delta_{ij}$.

P is an $m \times m$ matrix which transforms each $\mathbf{x}_i$ into $P\mathbf{x}_i$.

Transformed data:

$$\mathbf{Y} = \mathbf{PX}.$$

Principal Component Analysis

Design goal of P: transformed components are uncorrelated.

Covariance matrix of $\mathbf{X}$:

$$\mathbf{C_X} = \frac{1}{n}\mathbf{X}\mathbf{X}^\mathbf{T}.$$

Covariance matrix of $\mathbf{Y}$:

$$\begin{aligned}\mathbf{C_Y} &= \frac{1}{n}\mathbf{Y}\mathbf{Y}^\mathbf{T} \\ &= \frac{1}{n}(\mathbf{PX})(\mathbf{PX})^T \\ &= \mathbf{P}(\frac{1}{n}\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\ &= \mathbf{P}\mathbf{C_X}\mathbf{P}^\mathbf{T}.\end{aligned}$$

Find the $\mathbf{P}$ such that $\mathbf{P}\mathbf{C_X}\mathbf{P}^\mathbf{T}$ is diagonal.

Principal Component Analysis

Lemma: An invertible square matrix can be diagonalized as $\mathbf{A} = \mathbf{E}^T\mathbf{D}\mathbf{E}$ (1), where $\mathbf{D}$ is the diagonal matrix of eigenvalues of $\mathbf{A}$, $\mathbf{E}^T$ is the concatenation of (column) eigenvectors of $\mathbf{A}$, and the eigenvectors are orthonormal.

Proof: Concatenate $\mathbf{A}\mathbf{e_i} = \lambda_i\mathbf{e_i}$ in each column, $\mathbf{A}\mathbf{E}^T = \mathbf{E}^T\mathbf{D}$. (1) holds by using $\mathbf{E}\mathbf{E}^T = \mathbf{E}^T\mathbf{E} = \mathbf{I}$.

$\mathbf{C_X} = \mathbf{P^T}\mathbf{C_Y}\mathbf{P}$. $\mathbf{C_Y}$ is diagonal. Each column of $\mathbf{P^T}$ is an eigenvector of $\mathbf{C_X}$.

Singular Value Decomposition

An $n \times m$ matrix $\mathbf{A}$ (rank r) can be decomposed as:

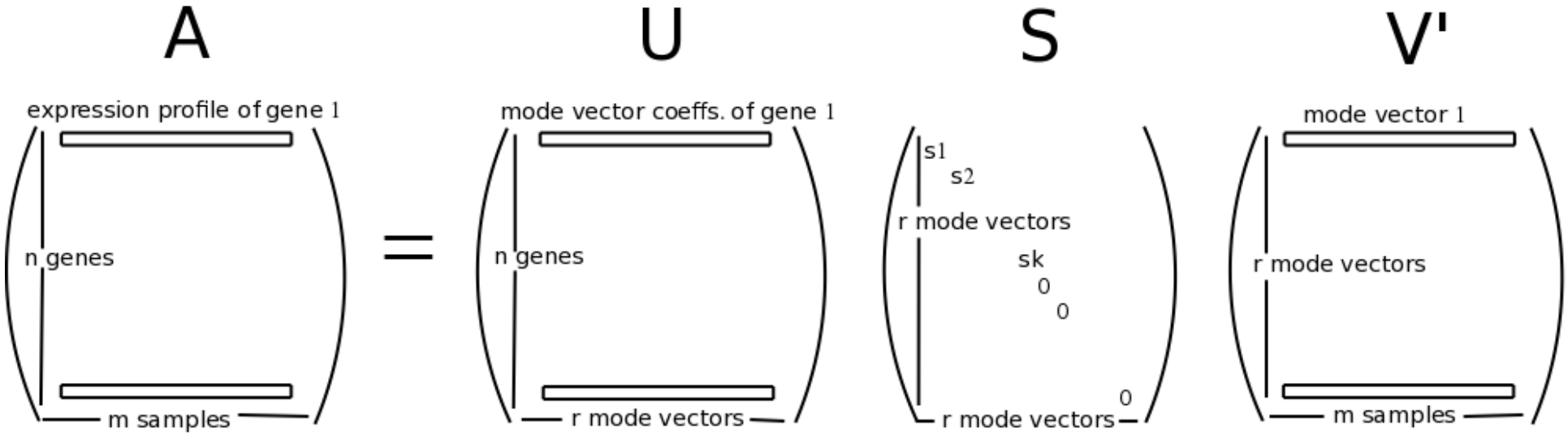
$$\mathbf{A} \;=\; \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

$\mathbf{U} : n \times r, \mathbf{\Sigma} : r \times r, \mathbf{V} : m \times r.$

- Each column $\mathbf{v}_i$ of $\mathbf{V}$ is an orthonormal eigenvector of $\mathbf{A}^T\mathbf{A}$.
- Nonzero entries $\sigma_1, \cdots, \sigma_r$ are on the diagonal.
- Each column $\mathbf{u}_i$ of $\mathbf{U}$ satisfies $\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i$.
- $\mathbf{U}$ is unitary ($\mathbf{U}\mathbf{U}^\mathbf{T} = \mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{I}$).

Singular Value Decomposition

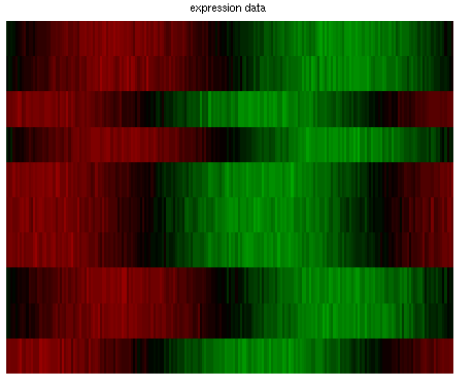
An interpretation of SVD:

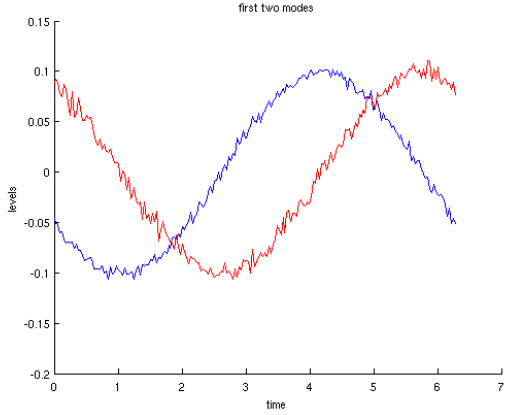


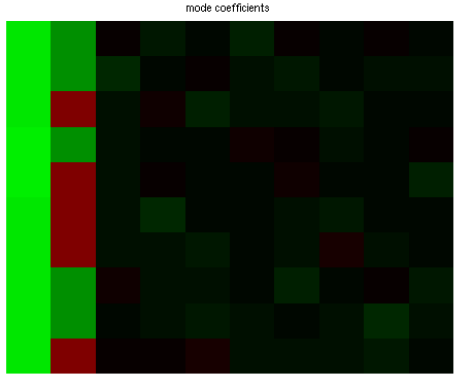
- $\mathbf{A}$ is the expression data.
- $\mathbf{V}$ constitutes the "modes" of gene expressions (e.g., responses at different cell cycle phases).
- $\mathbf{US}$ constitutes the coefficients of mode vectors for each gene.

Singular Value Decomposition

An example of SVD.







- Left: A gene expression data (A).
- Middle: First two modes (columns of V).
- Right: Mode coefficients (submatrix of $U\Sigma$).

PCA and SVD

$$
\begin{aligned}
\mathbf{Z} &\equiv \frac{1}{\sqrt{n}}\mathbf{X}^T. \\
\mathbf{Z}^T\mathbf{Z} &= \mathbf{C}_\mathbf{X}. \\
\mathbf{Z} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \\
\mathbf{Z}^T\mathbf{Z} &= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\
&= \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T.
\end{aligned}
$$

Relations of PCA and SVD:

- Columns of $\mathbf{V}$ are the principal components of $\mathbf{X}$.
- Eigenvalues $\lambda_i = \sigma_i^2$.

Tutorial on principal component analysis. Jonathon Shlens.

Independent Component Analysis

Cocktail-Party Problem: Separate the voice of each attendant of a cocktail party.

$$\mathbf{x} \;=\; \mathbf{A}\mathbf{s}.$$

$\mathbf{s}$: independent component signals. $\mathbf{A}$: mixture coefficients. $\mathbf{x}$: recorded signals.

Independent Component Analysis

Goal: Find the inverse transformation $\mathbf{W}$ to reconstruct $\mathbf{s}$.

$$\mathbf{s} \quad = \quad \mathbf{W}\mathbf{x}.$$

Assumptions:

- s_i and s_j are independent for each $i \neq j$.
- $E\{s_i\} = 0$.
- $E\{s_i^2\} = 1$.

Independent Component Analysis

Which objective function $F(\mathbf{W})$ is adequate?

Ideally, components of the inverse-transformed variables

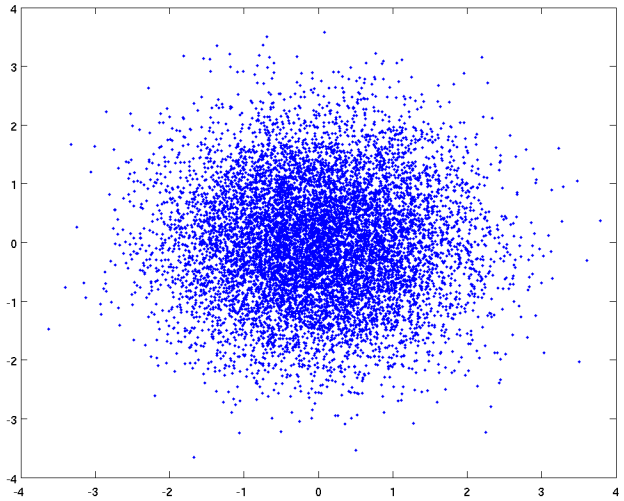
$$\mathbf{y} = \mathbf{W}\mathbf{x}.$$

should be independent.

$$\begin{aligned} p_{\mathbf{y}}(\mathbf{y}) &= \frac{p_{\mathbf{x}}(\mathbf{x})}{|\mathbf{J}|}. \\ |\mathbf{J}| &= ||\mathbf{W}| \prod_i \frac{\partial y_i}{\partial x_i}| \\ &= |\mathbf{W}| \prod_i W_{ii}. \end{aligned}$$

Independent Component Analysis

Gaussian variables are forbidden.



- All variables have unit variance.
- Any linear orthogonal transformation yields independent distributions.

Independent Component Analysis

Criterion1 : Kurtosis.

$$\begin{aligned} \text{kurt}(y) &= E\{y^4\} - 3(E\{y^2\})^2 \\ &= E\{y^4\}(\text{unit variance}). \end{aligned}$$

Non-Gaussian random variables have nonzero kurtosis.

Find the $\mathbf{W}$ that maximizes the fourth moment of $\mathbf{y}$.

Independent Component Analysis

Criterion 2: Mutual information.

$$\begin{aligned} I(\mathbf{y}) &= \textstyle\sum_i H(y_i) - H(\mathbf{y}). \\ H(y_i) &= -\int p_{y_i}(y_i) \log p_{y_i}(y_i) dy_i. \\ H(\mathbf{y}) &= -\int p_{\mathbf{y}}(\mathbf{y}) \log p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y}. \end{aligned}$$

Independent random variables have zero mutual information.

Find the $\mathbf{W}$ that minimizes the mutual information.

Independent Component Analysis

$$\begin{aligned} I(\mathbf{y}) &= \sum_i H(y_i) - H(\mathbf{x}) - \log|\mathbf{W}| \\ &= \sum_i H(y_i) - \log|\mathbf{W}| + C. \end{aligned}$$

Can approximate $H(y_i)$ with higher-order moments of y_i.

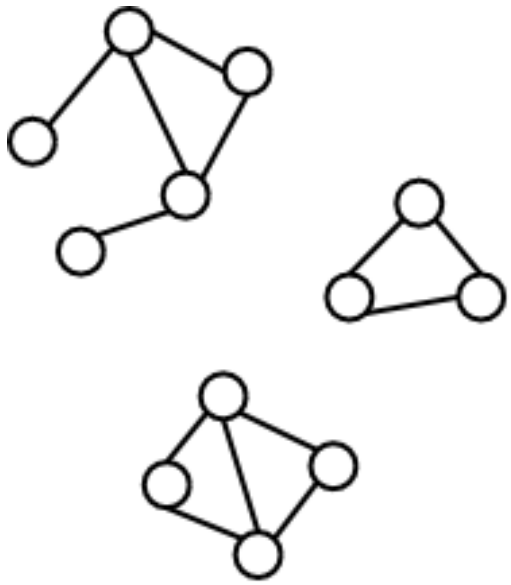
Independent Component Analysis: Algorithms and Applications. 2000. Hyvarinen A. and Oja E. Neural Networks 13(4-5):411-430.

Clustering on graphs

Sometimes the relations of objects are represented by a graph.

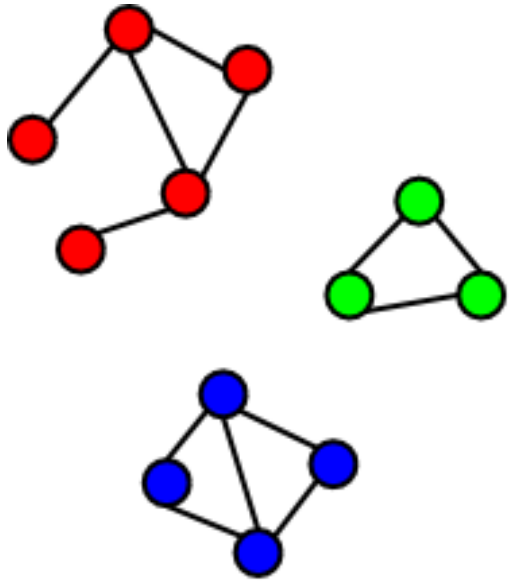
- Protein-protein interaction networks.
- Web documents.
- Social networks.
- Mobile phone infrastructure.
- .....

$G = (V, E)$. V: nodes. E: edges.



How to cluster the nodes on a graph?

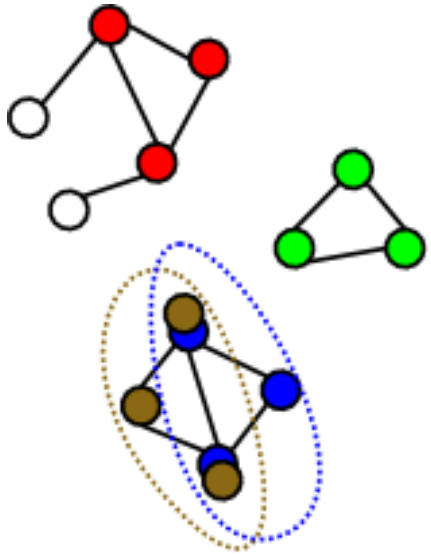
Connected Components



- Nodes u and v belong to the same component if u and v are connected by paths in G.

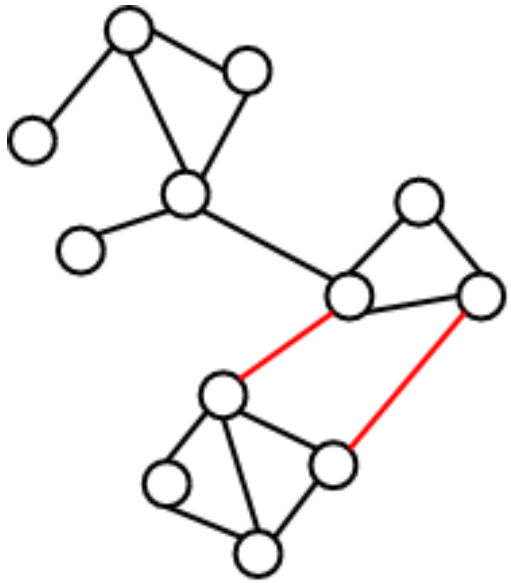
- Recursively label nodes to find all components.

Maximal cliques



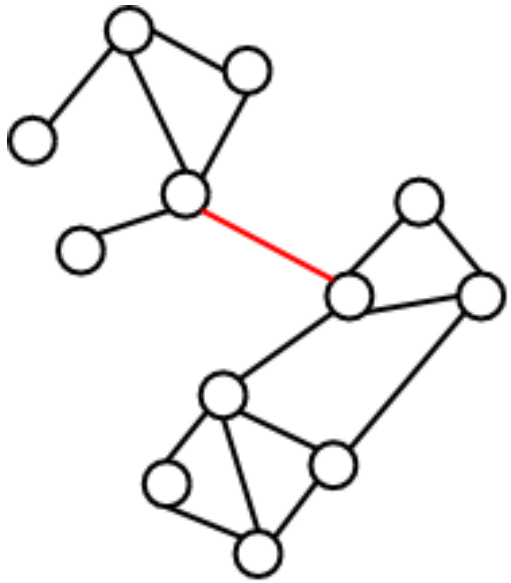
- A clique is a complete subgraph.
- Each pair of nodes in a clique are adjacent (connected by an edge).
- A maximal clique is not contained in another clique.
- One node can belong to multiple maximal cliques.
- Finding all maximal cliques is NP-hard.

Cuts



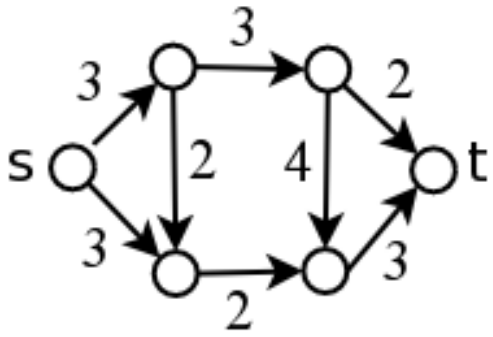
- A collection of edges $C \subseteq E$ in a connected graph G.
- Removal of C from E partitions G into two (or multiple) connected components.

Minimum Cuts



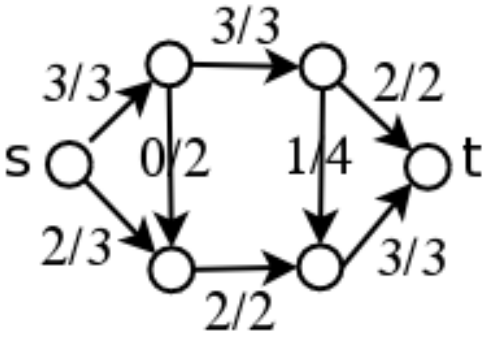
- Separate G into two disjoint node sets (S, T).
- Give each edge $(u, v) \in E$ a capacity (weight) $w(u, v)$.
- Capacity of a cut (S, T) is $c(S, T) = \sum_{u \in S, v \in T, (u,v) \in E} w(u, v)$.
- $c(S, T)$ is minimum among all cuts.
- Brute force takes $O(2^{|V|})$ time.
- How to find a minimum cut?

Flows



- A directed graph G.
- A source s.
- A destination t.
- Capacity for each edge in G.
- Want to transport some "commodities" from s to t.
- Can distribute the commodities to multiple edges.
- Conservation of commodities (except at s and t).
- The amount of commodities at each edge $\leq$ edge capacity.
- A flow is the amount of commodities transported from s to t.

Maximum Flows



- A valid flow.
- The flow is maximized.

Maximum Flows

How to find the maximum flow from s to t?

Ford-Fulkerson algorithm: Use the concept of *residual network*.

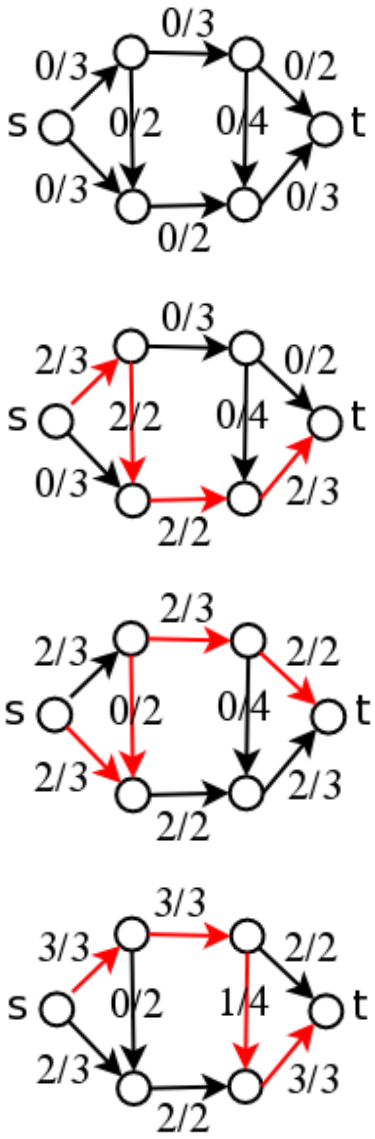
$w(u,v)$:capacity on (u,v). $f(u,v)$:flow on (u,v). $w_f(u,v) = w(u,v) - f(u,v)$:residual capacity on (u,v).

1. $f(u,v) \leftarrow 0$ for all edges (u,v).

2. Find a path π from s to t in G, such that $w_f(u,v) > 0, \forall (u,v) \in \pi$.
   (a) Compute $w_f(\pi) = \min\{w_f(u,v) | (u,v) \in \pi\}$.
   (b) For each $(u,v) \in \pi$,
      i. $f(u,v) \leftarrow f(u,v) + w_f(\pi)$.
      ii. $f(v,u) \leftarrow f(v,u) - w_f(\pi)$.

3. Stop when no valid path π can be found.

Maximum Flows



Max-Flow-Min-Cut Theorem

Theorem: The maximum flow from s to t equals to the capacity of a minimum cut separating s and t.

Proof Sketch:

- Max flow $\leq$ Min cut: Any total flow = Flow into any cut = Flow out of any cut $\leq$ Total capacity of any cut.

- Max flow $\geq$ Min cut: At max flow, the residual graph capacity of a cut separating s and t is 0. $\Rightarrow$ Max flow = Total capacity of some cut $\Rightarrow$ Max flow $\geq$ Min cut.

Find the mininum cut of a undirected graph G:

- Treat G as bidirectional.

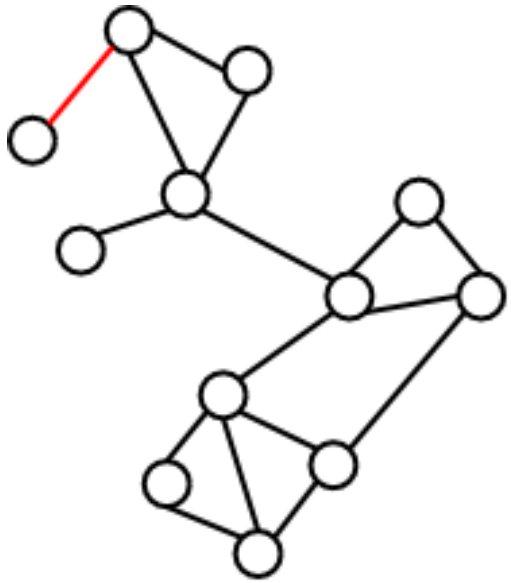
- For each pair of nodes (s, t), incur Ford-Fulkerson to find the max flow/min cut $C(s, t)$.

- Find the minimum $C(s, t)$ among all (s, t) pairs.

Maximum flow through a network. Ford LR and Fulkerson DR. 1956. Canadian Journal of Math 8:399-403.

Problem of Minimum Cuts

Problem of minimum cuts: unbalanced component sizes.

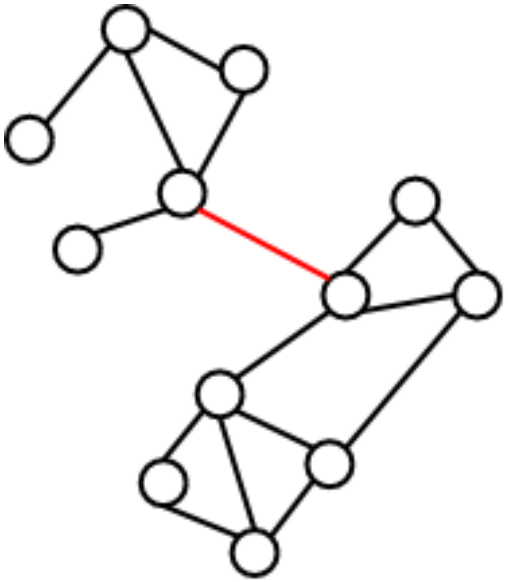


Need a proper score to gauge the *balanceness* of the partition.

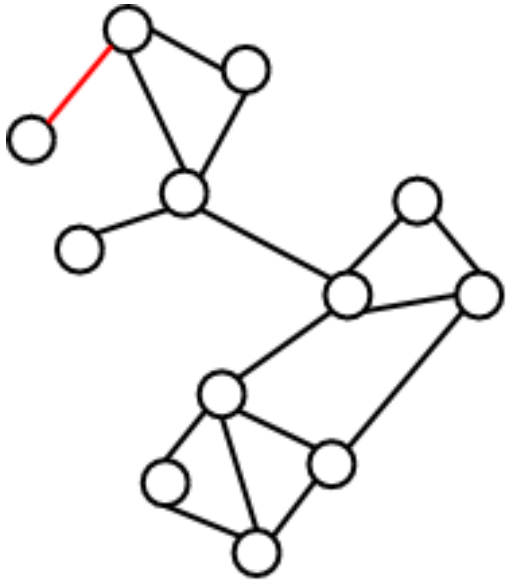
Normalized Cuts

$$nc(S,T) = \frac{c(S,T)}{assoc(S,V)} + \frac{c(S,T)}{assoc(T,V)}.$$

$$assoc(S,V) = \sum_{u \in S, v \in V} w(u,v).$$



$$nc = \frac{1}{6} + \frac{1}{11} = 0.258.$$



$$nc = \frac{1}{1} + \frac{1}{16} = 1.063.$$

Normalized Cuts $\equiv$ Spectral Clustering

$$
\begin{array}{lcl}
\mathbf{d} : d_i & = & \sum_j w(i,j). \\
\mathbf{x} : x_i & = & \text{indicator of node } i \in S (x_i \in \{-1, +1\}). \\
\mathbf{D} & = & \text{diagonal matrix of } d_i\text{'s.} \\
\mathbf{W} & = & \text{symmetric matrix of } w_{ij}\text{'s.} \\
k & = & \frac{\sum_{x_i>0} d_i}{\sum_i d_i}. \\
b & = & \frac{k}{1-k}. \\
\mathbf{y} & = & (1+\mathbf{x}) - b(1-\mathbf{x}). \\
nc(S,T) & = & \frac{c(S,T)}{assoc(S,V)} + \frac{c(S,T)}{assoc(T,V)} \\
 & = & \frac{\sum_{x_i>0, x_j<0} -w_{ij}\mathbf{x}_i\mathbf{x}_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0, x_j>0} -w_{ij}\mathbf{x}_i\mathbf{x}_j}{\sum_{x_i<0} d_i} \\
 & = & \frac{(1+\mathbf{x})^T(\mathbf{D}-\mathbf{W})(1+\mathbf{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{(1-\mathbf{x})^T(\mathbf{D}-\mathbf{W})(1-\mathbf{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} \\
 & = & \ldots \\
 & = & \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{b\mathbf{y}^T\mathbf{D}\mathbf{y}}. \\
\min_{\mathbf{x}} nc(\mathbf{x}) & = & \min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}. \\
y_i & \in & \{1, -b\} \\
\mathbf{y}^T\mathbf{D}\mathbf{1} & = & 0.
\end{array}
$$

Normalized Cuts $\equiv$ Spectral Clustering

The nc is minimized by solving

$$\begin{aligned} (\mathbf{D}-\mathbf{W})\mathbf{y} &= \lambda \mathbf{D}\mathbf{y}. \\ \mathbf{z} &= \mathbf{D}^{\frac{1}{2}}\mathbf{y}. \\ \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{z} &= \lambda \mathbf{z}. \end{aligned}$$

$\mathbf{z}_0 = \mathbf{D}^{\frac{1}{2}}\mathbf{1}$ is an eigenvector with eigenvalue 0.

$$\begin{aligned} \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{1} &= \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{1} \\ &= \mathbf{0}.(1) \end{aligned}$$

$\mathbf{z}_1$ is the eigenvector of the second smallest eigenvalue.

$$\begin{aligned} \mathbf{z}_1^T\mathbf{z}_0 &= \mathbf{y}_1^T\mathbf{D}\mathbf{1} \\ &= 0.(2) \end{aligned}$$

Given (1) and (2), the $\mathbf{z}$ that minimizes $\frac{\mathbf{z}^T\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{z}}{\mathbf{z}^T\mathbf{z}}$ is the eigenvector of the second smallest eigenvalue of $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}$.

Spectral Clustering

1. Build a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ from the data.

2. Find the eigenvector $\mathbf{z}_1$ of $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}$ with the second smallest eigenvalue.

3. Partition $\mathbf{V}$ according to the signs of entries in $\mathbf{z}_1$.

4. Recursively partition each component.

Normalized cuts and image segmentation. Shi J. and Malik J. 2000. IEEE Trans. pattern analysis & machine intelligence. 22(8):888-905.

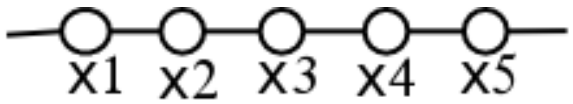
Spectral Clustering

Why is it "spectral"?

$\mathbf{L} \equiv \mathbf{D} - \mathbf{W}$ is the *Laplacian* of the graph.

Consider a chain graph $(x_1, x_2), (x_2, x_3), \cdots$.

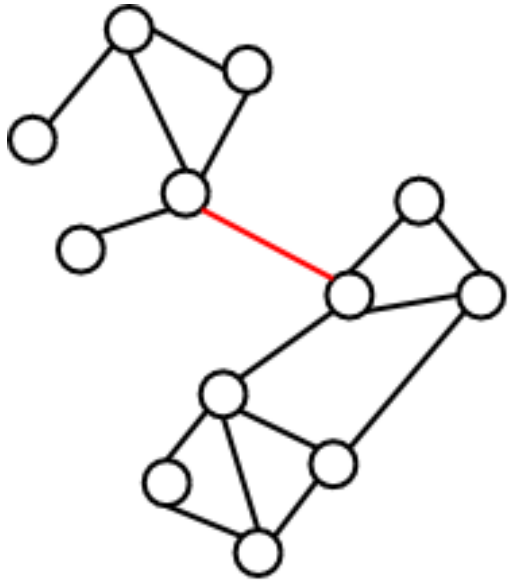
$$\mathbf{L} = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 0 & -1 & 2 & -1 & \cdots & \cdots \\ \cdots & 0 & 0 & -1 & 2 & -1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}.$$



Spectral Clustering

- Shrink the distance Δx between nodes $\rightarrow 0$.
- The chain becomes a real line $\mathbf{R}$.
- A function $\mathbf{f}$ on $\mathbf{R}$ is a vector on $\mathbf{R}$.
- Easy to show $\mathbf{Lf}(x) = -\mathbf{f}(x + \Delta x) - \mathbf{f}(x - \Delta x) + 2\mathbf{f}(x)$.
- $\frac{\mathbf{Lf}(x)}{(\Delta x)^2} = -\Delta \mathbf{f}(x)$ as $\Delta x \rightarrow 0$.
- $\Delta \mathbf{f}(x)$ is the Laplacian $\frac{\partial^2}{\partial x^2}$ of function $\mathbf{f}(x)$.
- Can be extended into a high-dimensional space.
- In a Euclidean space $\Delta \mathbf{f}(x) = \lambda \mathbf{f}(x)$ yields sinusoidal functions. λ's are the frequencies.
- The eigenvalues of an operator (matrix) constitute its *spectra*.

Community detection



- Intuition: remove the edges connected components.
- Define betweenness of an edge as the total number of short paths containing the edge.
- In a connected graph, iterate the following steps until no edges remain.
  - Calculate betweenness of each edge in the graph.
  - Remove the edge with the highest betweenness from the graph.
- Removal of edges creates isolated components (communities).

Community structure in social and biological networks. Girvan M. and Newman M.E.J. 2002. PNAS 99(12):7821-7826.

Exercise: recommend movies for your classmates

1. Pick up 7 of the 18 movies listed.
2. Rank those 7 movies according to your preference.
3. +1: like, -1: dislike, 0: no preference.
4. Divide the class into 2 groups.
5. Each group proposes a recommendation algorithm.
6. Each group fills the missing entries of members in the other group.
7. Validate the predictions.

Exercise: recommend movies for your classmates

Source: excerpt from IMDB.

Index	Title
1	Amelie
2	An inconvenient truth
3	Annie Hall
4	Avatar
5	Black swan
6	Cape number 7
7	Cries and whispers
8	Human centiped
9	Inception
10	Inglourious Basterds
11	Saw
12	The dark knight
13	The devil wears Prada
14	The Godfather
15	The incredibles
16	The king's speech
17	Titanic
18	Toy story 3

How to choose the number of clusters

- Set a threshold on some cluster coherence measure.
- Cross-validation errors.
- Model selection criteria (AIC, BIC, etc.).

What about a Bayesian approach?

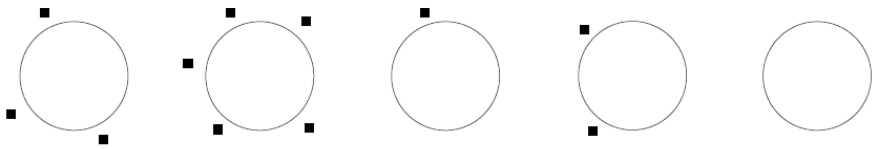
How to choose the number of clusters

- A clustering is a partition of data points.
- Impose a probability measure over the partitions of data points.
- Sample from the distribution of the partitions to determine the cluster number.
- Extend the probability measure to partitions of infinite data points.
- What is the adequate probability measure?

Chinese Restaurant Process (CRP)

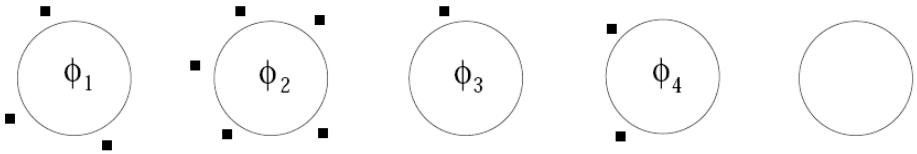
Goal: Sit customers of a Chinese restaurant with an infinite number of tables.

- First customer sits at the first table.
- The mth customer prefers to sit at a table with more customers.
- But he/she may also choose to sit at a new table.
- P(occupied table i|sitting configuration of $m-1$ customers) $\propto n_i$.
- P(unoccupied table|sitting configuration of $m-1$ customers) $\propto \alpha_0$.



CRP as sampling from mixture models

- Data points are customers.
- Tables are clusters.
- CRP defines a prior on partitioning and number of clusters.
- Prior: continuous over all new clusters (α).
- Posterior: discrete over occupied clusters (n_i).
- Sample a cluster according to CRP.
- Conditioned on the selected cluster sample parameters of the model.



CRP as a Bayesian estimation of multinomial distributions

Let $\pi = (\pi_1, \cdots, \pi_m)$ be the probabilities of a multinomial distribution.

$$\begin{aligned} \text{(Prior)} p(\pi|\alpha) &= \frac{\Gamma(\sum_{i=1}^m \alpha_i)}{\prod_{i=1}^m \Gamma(\alpha_i)} \prod_{i=1}^m \pi_i^{\alpha_i - 1}. \\ \text{(Posterior)} p(\pi|\alpha, D) &= \frac{\Gamma(\sum_{i=1}^m \alpha_i + n_i)}{\prod_{i=1}^m \Gamma(\beta_i)} \prod_{i=1}^m \pi_i^{\alpha_i + n_i - 1}. \end{aligned}$$

- Possible values in the multinomial distribution: tables.
- Counts of data points in each class: customers.
- Prior: Dirichlet prior over the parameters.
- Posterior: Weighted by the counts of data points in each class.

Dirichlet Process

Dirichlet Process: Extend the Dirichlet prior to any possible partitioning.

- $(\Omega, \mathcal{B})$ is a probability space (possible outcomes of the multinomial distribution).
- G_0 is a probability measure on Ω (weight on each outcome).
- $(A_1, \cdots, A_m)$ is a partition of Ω (m possible values of the multinomial distribution).
- α_0 is a hyper-parameter.
- $G_0(A_i)$'s are the measures of each partitioned subspace (relative weight of each mixture component).

Dirichlet Process – Continued

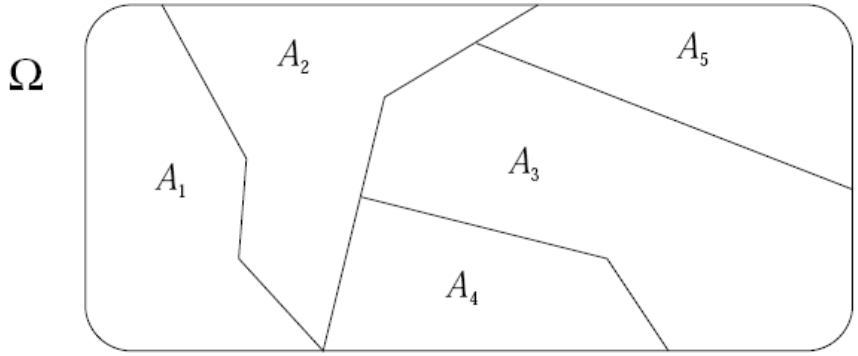
- $Dir(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_m))$ is a Dirichlet distribution of m components with parameters $\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_m)$.

- $Dir(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_m))$ gives a distribution of probabilities on each component $(\pi_1, \cdots, \pi_m)$.

- A random probability measure G over $(\Omega, \mathcal{B})$ can give probabilities on the components for any finite partition $(G(A_1), \cdots, G(A_m))$.

- The Dirichlet process is the distribution of G such that any finite partition yields a finite dimensional Dirichlet distribution:

$$(G(A_1), \cdots, G(A_m)) \quad \sim \quad Dir(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_m)).$$



Dirichlet processes, Chinese restaurant processes and all that. 2005. Michael Jordan.

How to choose the number of clusters

Is there any other way to choose the cluster numbers?

- K-means: treat the centroid of a cluster as its *exemplar*.
- In principle, any data point can be an exemplar.
- So, why not let them to compete?
- Clusters will emerge from the competition.

Affinity Propagation

- N data points.
- Similarity score $s(i, j)$ between each pair of points.
- Each point is assigned to an exemplar data point.
- $\mathbf{c} = (c_1, \cdots, c_N)$: exemplar label of each point.
- $c_i = i$ if i is an exemplar.

Affinity Propagation

Goal: Find the exemplar labels to maximize the intra-cluster similarity.

$$
\begin{aligned}
S(\mathbf{c}) &= \sum_{i=1}^{N} s(i, c_i) + \sum_{k=1}^{N} \delta_k(\mathbf{c}). \\
\delta_k(\mathbf{c}) &= \begin{cases} -\infty & \text{if } c_k \neq k, \exists i : c_i = k. \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

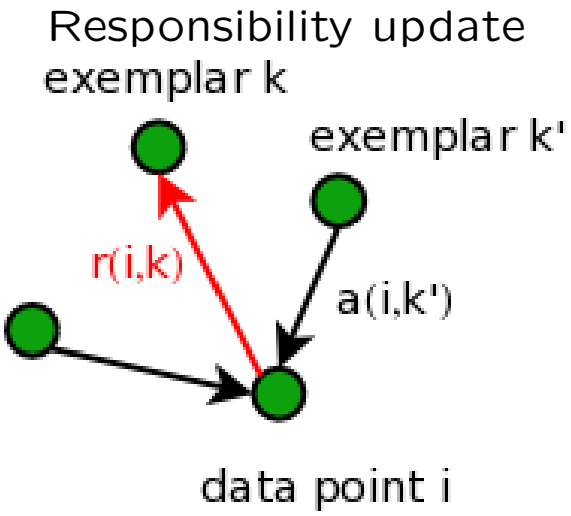
The problem can be approximated by a belief propagation algorithm on factor graphs.

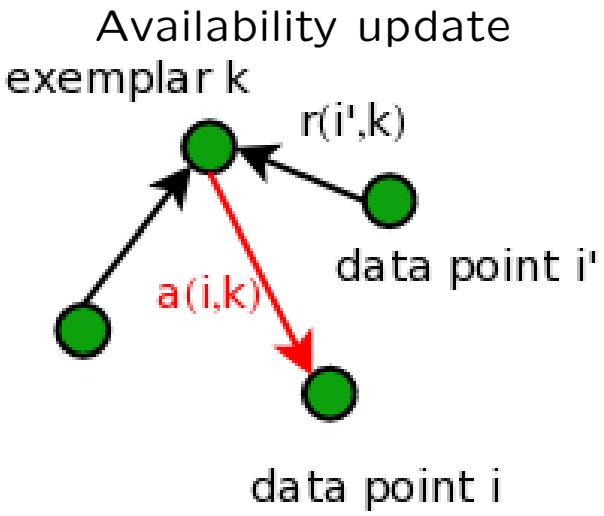
Affinity Propagation

- Each data point connects to all other data points.
- Two types of messages between data points and exemplars.
- Update messages according to the incoming messages from other neighbors.
- Stop when messages converge or after a fixed time.

Affinity Propagation

- Responsibility $r(i,k)$ from data point i to exemplar k.
- Availability $a(i,k)$ from exemplar k to data point i.
- $r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}$.
- $a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \notin \{i,k\}} max\{0, r(i',k)\}\}$.
- $r(i,k)$ high if i is close to k and i is not close to other likely exemplars.
- $a(i,k) < 0$ if other data points and k itself indicate k is not an exemplar.





Affinity Propagation

- Initially $a(i,k) = 0 \forall (i,j)$.
- Iteratively update $r(i,k)$ and $a(i,k)$.
- Stop when messages converge or after a fixed time.
- Exemplar of i is $\arg\max_k r(i,k)$.

Clustering by passing messages between data points. 2007. Frey BJ and Dueck D. Science 315:972-976.

Readings

- The PageRank citation ranking: bringing order to the web. 1998. Page et al.

- A short tutorial on using expectation-maximization with mixture models, Jason Rennie.

- Estimating Gaussian mixture densities with EM – a tutorial, Carlo Tomasi.

- C. Fraley and A. Raftery, The Computer Journal 1998, 41(8):578-588.

- The self-organizing map. Kohonen T., 1990. Proceedings of the IEEE 78(9):1464-1480.

- Tutorial on principal component analysis. Jonathon Shlens.

- Independent Component Analysis: Algorithms and Applications. 2000. Hyvarinen A. and Oja E. Neural Networks 13(4-5):411-430.

- Maximum flow through a network. Ford LR and Fulkerson DR. 1956. Canadian Journal of Math 8:399-403.

- http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm.

- Normalized cuts and image segmentation. Shi J. and Malik J. 20000. IEEE Trans. pattern analysis & machine intelligence. 22(8):888-905.

- Community structure in social and biological networks. Girvan M. and Newman M.E.J. 2002. PNAS 99(12):7821-7826.

- Dirichlet processes, Chinese restaurant processes and all that. 2005. Michael Jordan.

- Clustering by passing messages between data points. 2007. Frey BJ and Dueck D. Science 315:972-976.