Machine Learning

Spectral & graph learning models

Introduction and Spectral clustering & graph clustering models

- Introduction & the problem
- Recall: classical clustering methods i.e. kmeans & hierarchical clustering
- Graph Laplacian
- Graph cut & Normalized cut
- Spectral clustering algorithms
- Validation indexes (intern & extern)
- Examples of applications & Conclusions

INTRODUCTION

Generalization

DATA

A data ?

Data is a basic description of a phenomenon

A phenomenon can be described by one or more criteria, these criteria are called variables:

A single criterion: univariate data
Two criteria: **bivariate data**
Several criteria: multivariate data

A phenomenon is thus described by a set of univariate or multivariate data.

More data is generated:

- Bank, telecom, other business transactions ...
- Scientific data: astronomy, biology, etc
- Web, text, and e-commerce

Samples and variables

- Population

  Group or set of individuals that are analyzed.

- Variables

  Set of characteristics of a population.

Classical data matrix

- For *n* objects and *p* variables, we difine the data table :

  X which is rectangular matrix containing *n* lines and *p* columns

$$X = (x^1, \ldots, x^p) = \begin{bmatrix} x_1^1 & x_1^2 & & \cdots & & x_1^p \\ x_2^1 & x_2^2 & & & & \\ & & \ddots & & & \\ \vdots & & & x_i^j & & \vdots \\ & & & & \ddots & \\ x_n^1 & & & \cdots & & x_n^p \end{bmatrix}$$

Example of a data table (matrix)

- Objects : x1,x2,…,x10.
- Variables :Y1,Y2,…,Y4.

Matrix of data

	Y1	Y2	Y3	Y4
x1	10	6	45	41
x2	13	8	35	78
x3	15	23	87	64
x4	19	56	96	43
x5	40	47	56	52
x6	45	34	43	42
x7	39	26	12	13
x8	40	12	14	16
x9	11	13	14	15
x10	39	26	12	13

Other definitions :
- Individuals: observations, objets, instances, transactions.
- Variables: attributs, dimensions, description, component.

Data types

- Quantitative variables
- Qualitative variables
- Others : text data

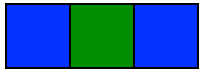
Quantitative variables

- Quantitative variables
  - Continuous (ex: the size, the weight of a person, the time of completion of a task, the volume of an object, the speed of a car)
  - Discrete (ex: counting: the number of persons in a room, the number of items from a list)

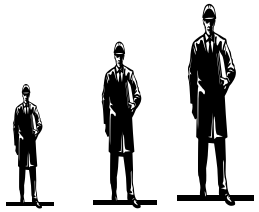
Qualitative variables

- Qualitative variables (categorical)
  Binary data: it can take two states (true or false, 0 or 1, yes or no;).
  Ex: Gender, having a credit or not ...

  Unordered categorical data (nominal):
  Ex: eye color

Colour : 

Ordered categorical data: data from a survey (1: very satisfied, 2: satisfied, ..)
Ex: low, medium, high, small, medium, large

Height: 

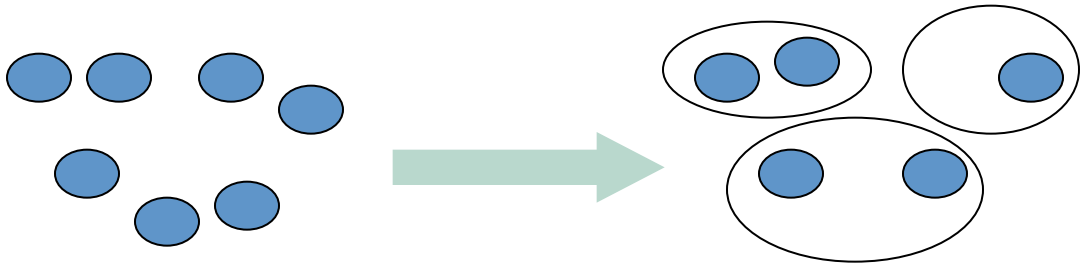
Goals …

- Answer a question or solve a problem
- Make data – intelligible
- Retrieve and select information
- Knowledge extraction
- Determine the validity of this information:
  Problem of sampling fluctuations
  Problem of generalization

Generalization
Clustering

Introduction : Clustering

- Grouping together of "similar" objects
  - Hard Clustering -- Each object belongs to a single cluster
  - Soft Clustering -- Each object is probabilistically assigned to clusters

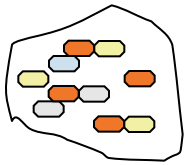


In general, the formalization of the Clustering problem is determined by the following components:

Data representation (categorical, binary, graph…)
The affinity measures (similarity, distance,..)
The objective function
The optimization procedure
Data distribution …

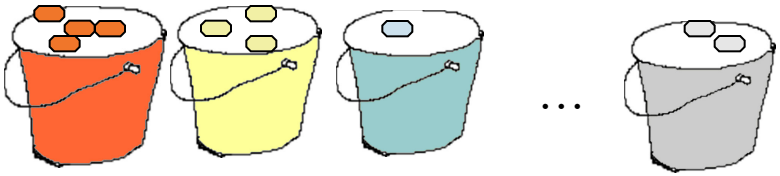
Introduction

Classification

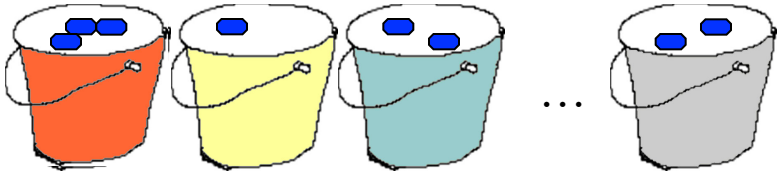


Clustering

The labels and the number of classes are known



The labels and the number of classes are unknown



Difficulties

- *The objects are not labeled, …*
- *We need to use a similarity measure (for which variables?)*
- *Do we need to know a priori the number of classes?*
- *How to characterize clusters***?**

-We need to use a similarity measure (for which variables?)

What is Similarity?
The quality or state of being similar; likeness; resemblance; as, a similarity of features.
Similarity is hard to define, but...

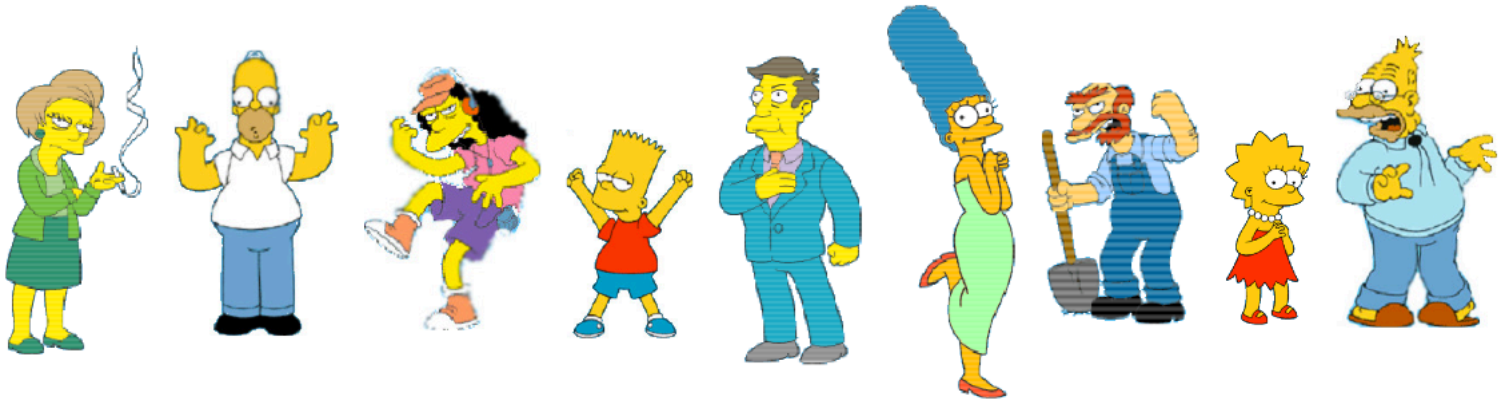
-We need to use a similarity measure (for which variables?)

What is Similarity?
The quality or state of being similar; likeness; resemblance; as, a similarity of features.
Similarity is hard to define, but...

"We know it when we see it"

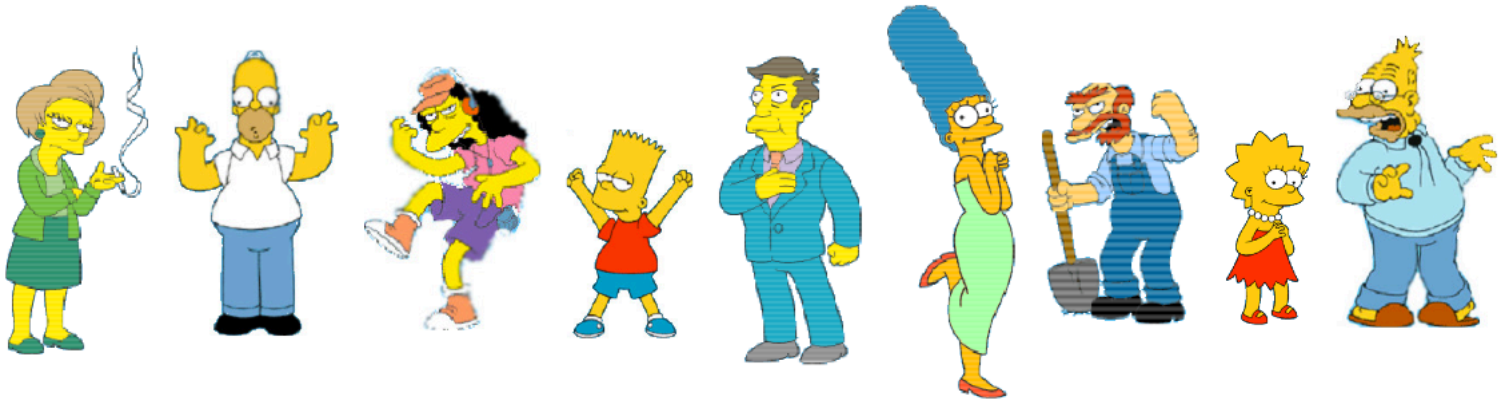
-Do we need to know a priori the number of classes?
*- How to characterize clusters**?***

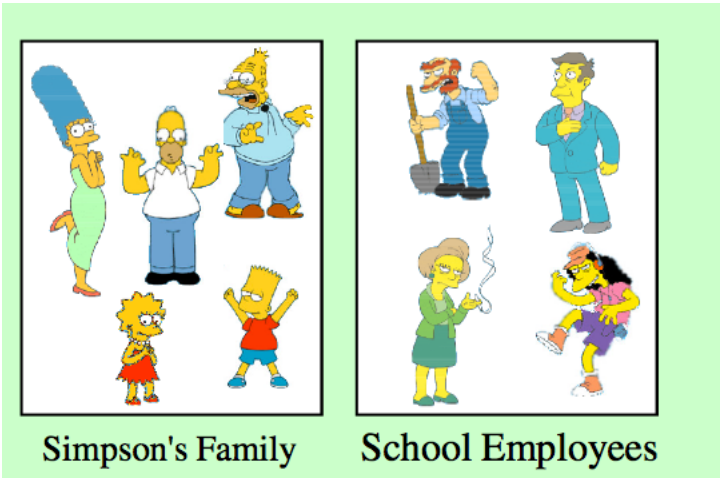
How to cluster the following objects?



-Do we need to know a priori the number of classes?
*- How to characterize clusters**?***

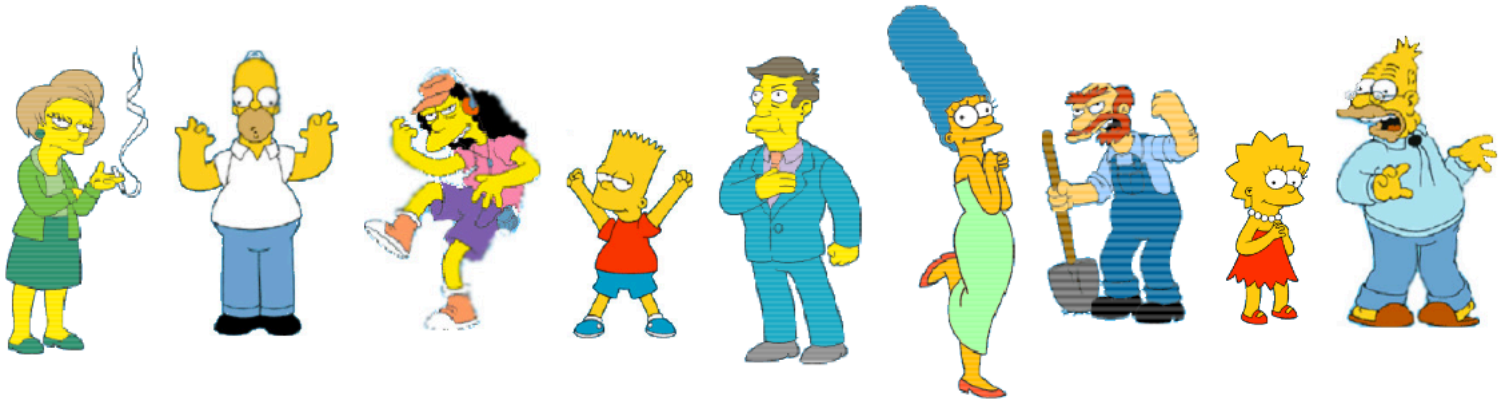
How to cluster the following objects?

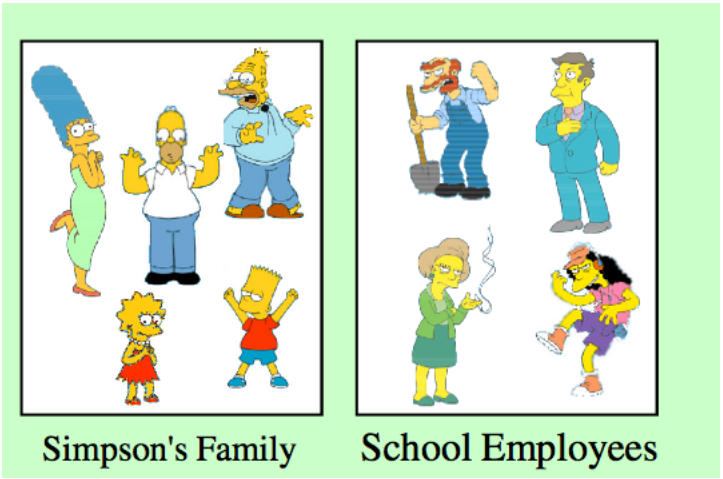


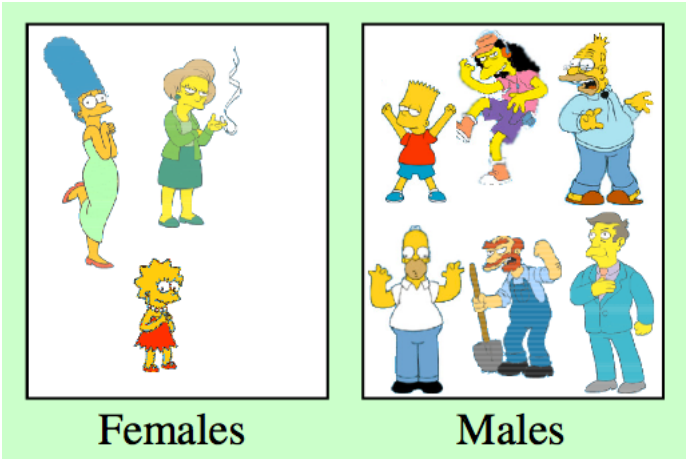


-Do we need to know a priori the number of classes?
*- How to characterize clusters**?***

How to cluster the following objects?

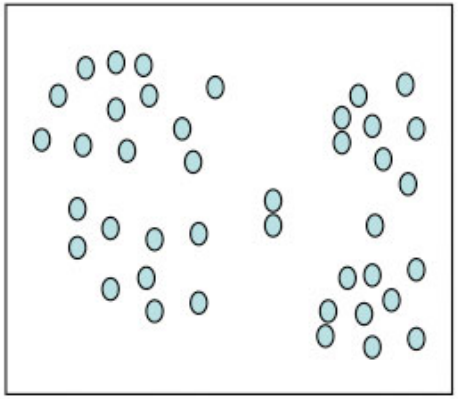




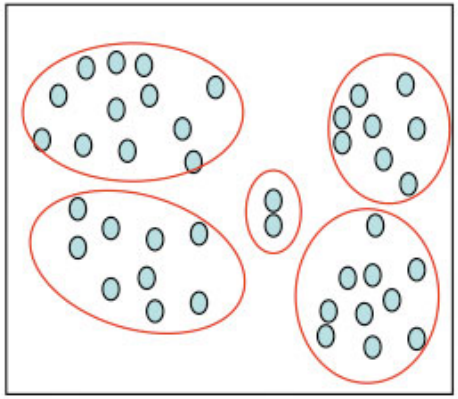


The problem

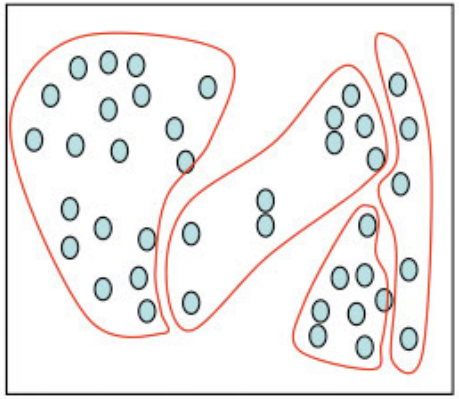
Different clustering results can be obtained



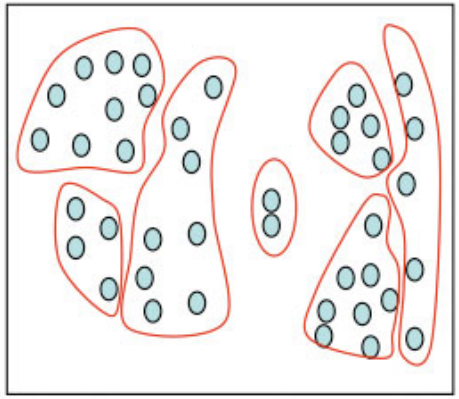
Group of houses



Clustered based on geographic distance



Clustered based on value



Clustered based on size and value

K-means & Hierarchical Clustering (some recalls)

Example of calculation of CAH

Example for CAH

- Update distance matrix (iteration 1)

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\to A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\to B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\to C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

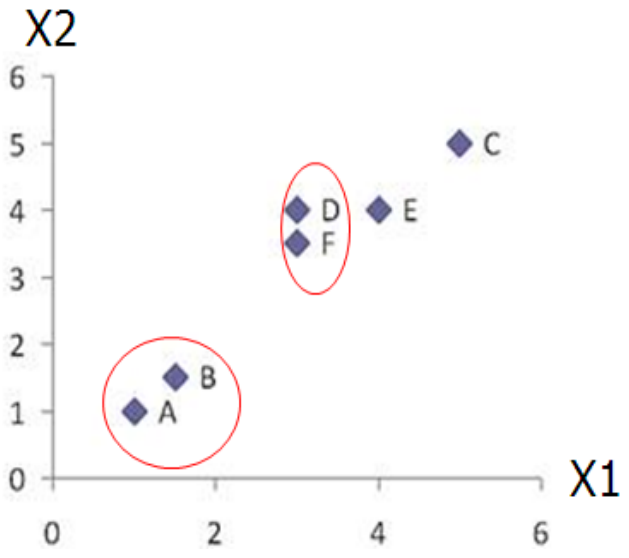
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Example for CAH

- Merge two closest clusters (iteration 2)



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Example for CAH

- Update distance matrix (iteration 2)

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

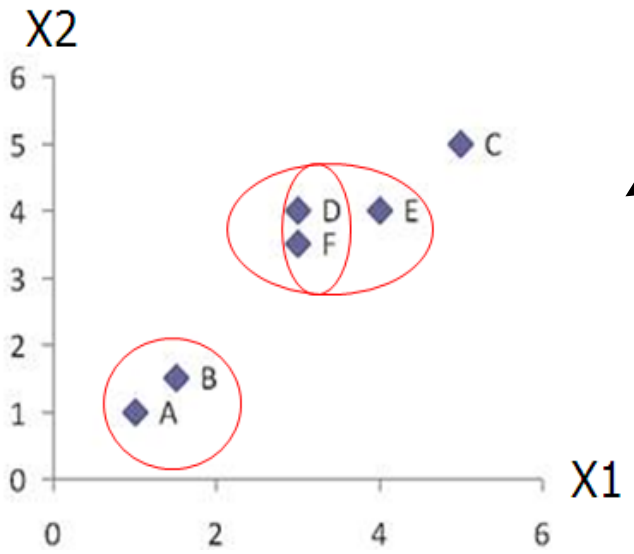
Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Example for CAH

- Merge two closest clusters/update distance matrix (iteration 3)



Min Distance (Single Linkage)

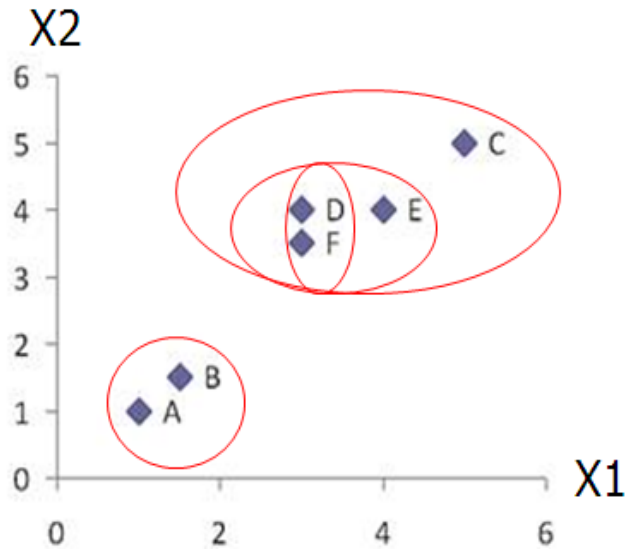
Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Example for CAH

- Merge two closest clusters/update distance matrix (iteration 4)



Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

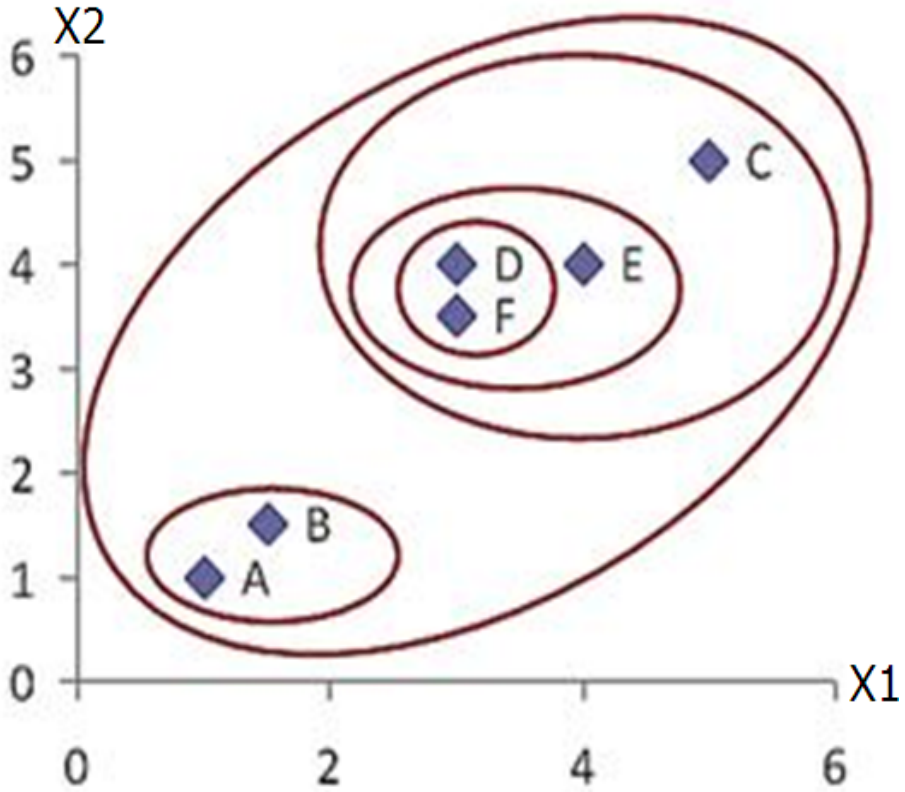
Min Distance (Single Linkage)

Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Example for CAH

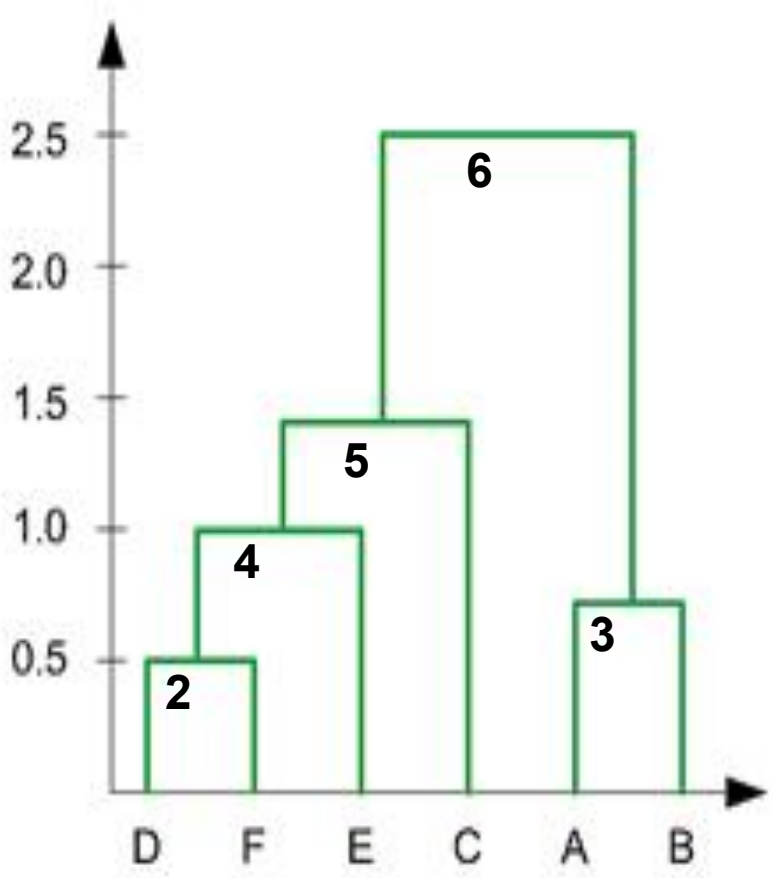
- Final result (meeting termination condition)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



Example for CAH

- Dendrogram tree representation



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge cluster D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge cluster E and (D, F) into ((D, F), E) at distance 1.00
5. We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

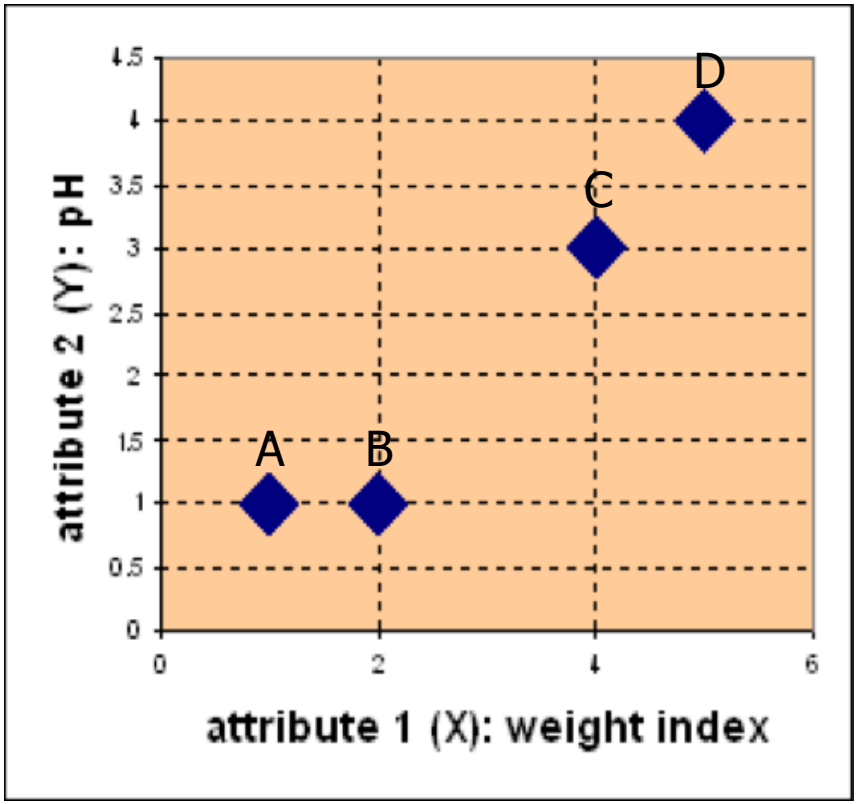
Example of calculation of K-means

Example for k-means

- Problem

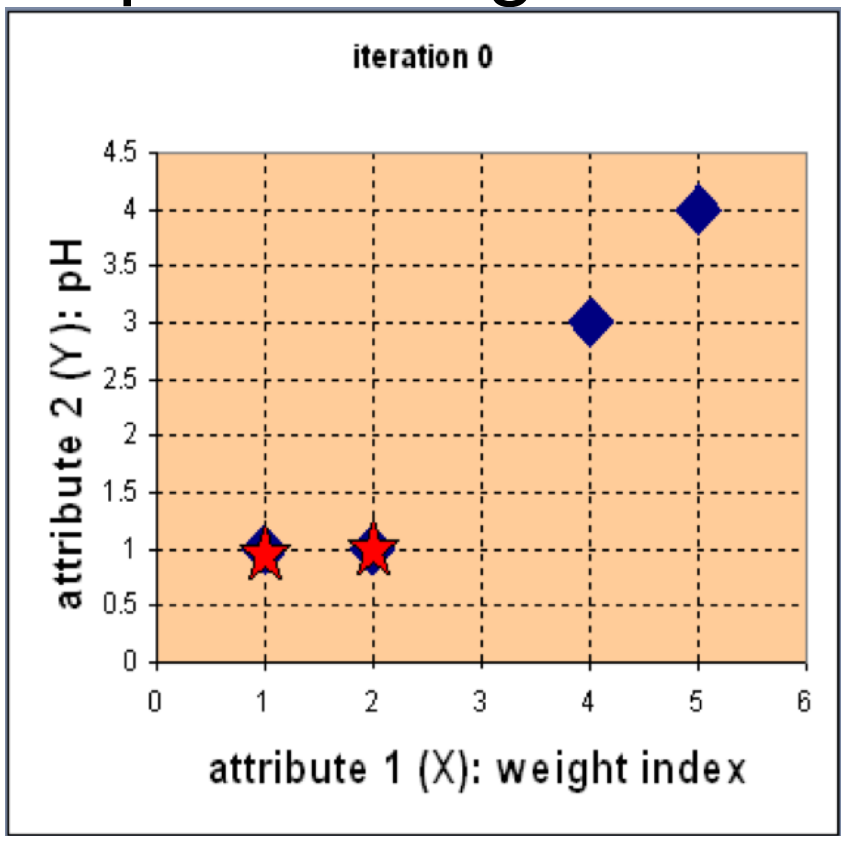
Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into *K=2* group of medicine.

Medicine	Weight	pH-Index
A	1	1
B	2	1
C	4	3
D	5	4



Example for k-means

- Step 1: Use initial seed points for partitioning



$c_1 = A, c_2 = B$

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad c_1 = (1,1) \quad group - 1 \quad c_2 = (2,1) \quad group - 2$$

Euclidean distance

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

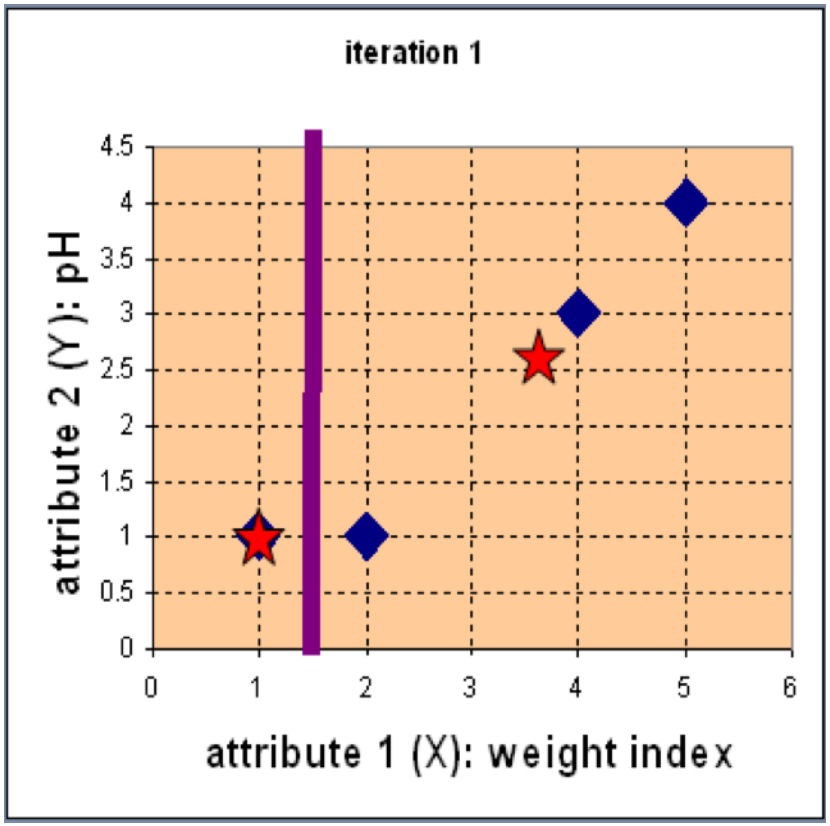
$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$d(D, c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Assign each object to the cluster with the nearest seed point

Example for k-means

- Step 2: Compute new centroids of the current partition



Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = (1,\ 1)$$

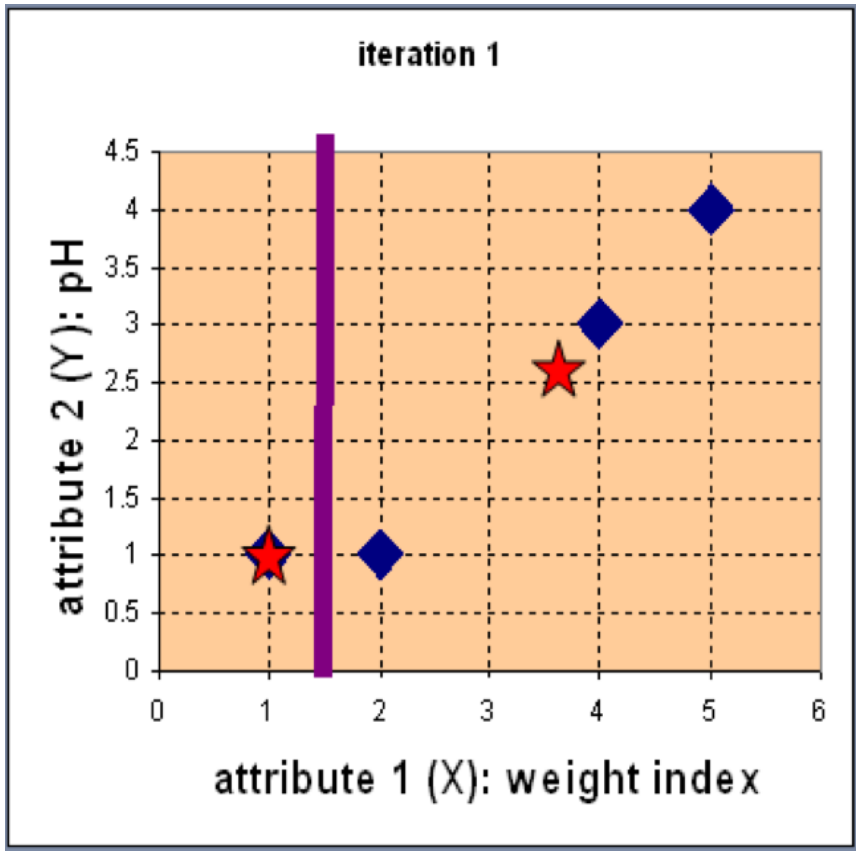
$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= (11/3,\ 8/3)$$

$$= (3.67,\ 2.67)$$

Example for k-means

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

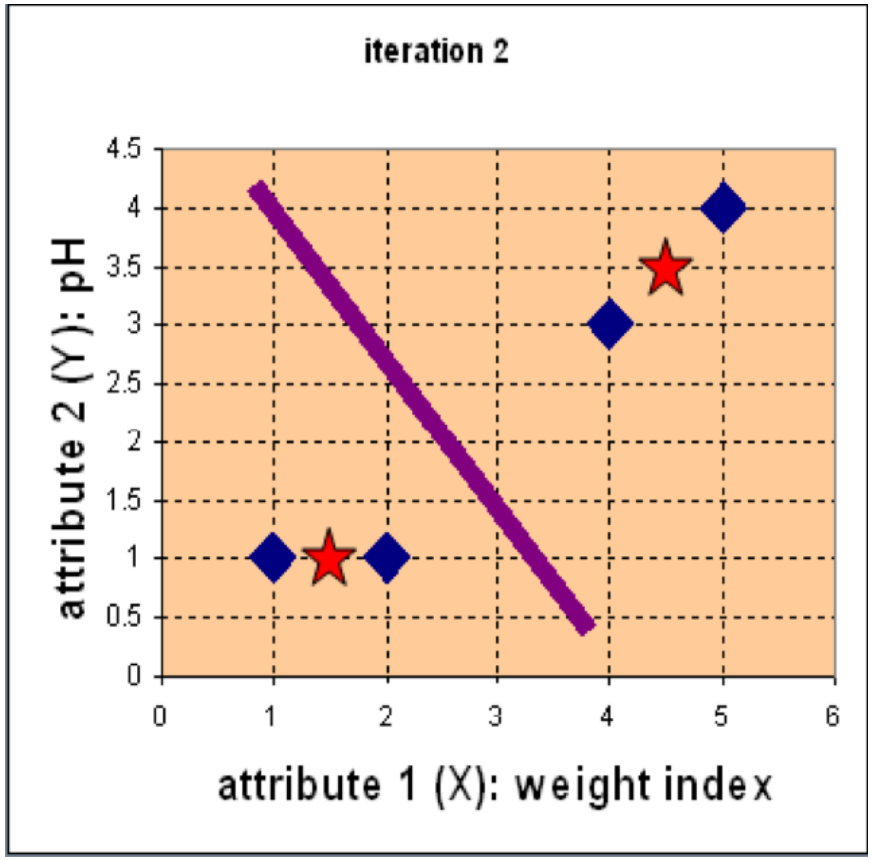
$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Assign the membership to objects

Example for k-means

- Step 3: Repeat the first two steps until its convergence



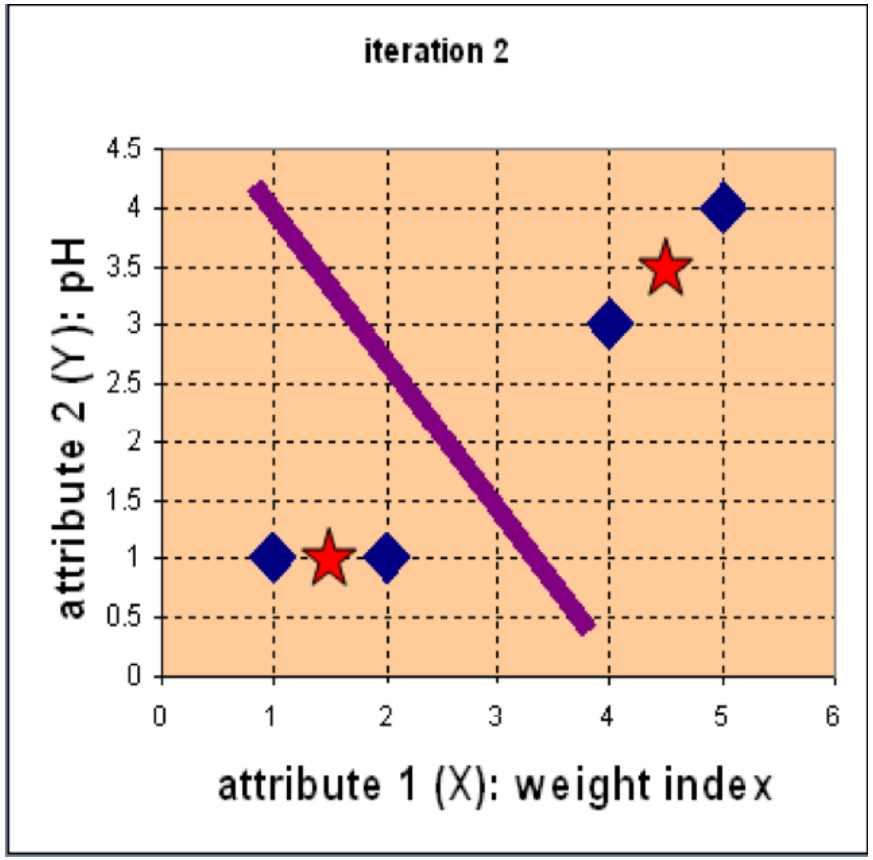
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

Example for k-means

- Step 3: Repeat the first two steps until its convergence



Compute the distance of all objects to the new centroids

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Stop due to no new assignment

Dimensionality Reduction

Singular Value Decomposition

Recall (1)

- Why dimensionality reduction?
- Some features may be irrelevant
- We want to visualize high dimensional data
- “Intrinsic” dimensionality may be smaller than the number of features

Recall (2)
Dimensionality Reduction Techniques

- **Visualize, categorize, or simplify large datasets.**
- **Principal Component Analysis: Finds the dimensions that capture the most variance**
- **MDS: Finds data points in lower dimensional space that best preserves the inter-point distance.**
- **Isomap: Estimates the distance between two points on a manifold by following a chain of points with shorter distances between them. (More accurate in representing global distances than LLE; slower than LLE)**
- **LLE (Local Linear Embedding): Only worries about representing the distances between local points. Faster than Isomap (no worry about global distances).**
- **Hesian Eigenmaps:**
- **Laplacian Eigenmaps**
- **Charting**
- **SOM (Kohonen's Self-organizing map)**

Dimensionality Reduction

- **The input space of many learning problems is of high dimensionality.**
- **This has computational implications and, it makes finding the intrinsic information content difficult.**
- **Dimensionality reduction methods usually try to address these two problems at the same time.**
- **Context: unsupervised learning**

given a collection of data points in an n-dimensional space find a good representation of the data in r-dimensional

Choosing sets of features

- Score each feature
- Forward/Backward elimination
  - Choose the feature with the highest/lowest score
  - Re-score other features
  - Repeat
- If you have lots of features (text mining for ex.)
  - Just select top K scored features

Unsupervised feature selection

- Differs from feature selection in two ways:
  - Instead of choosing subset of features,
  - Create new features (dimensions) defined as functions over all features
  - Don't consider class labels, just the data points

SVD: the problem

- For a large dimensional data, use PCA ?
  - for example: images ($d \geq 10^4$), gene expression data, text data,…
- The problem:
  - The covariance matrix is size (d^2), so the computational time is expensive.
- Solution:
  - Singular Value Decomposition (SVD)

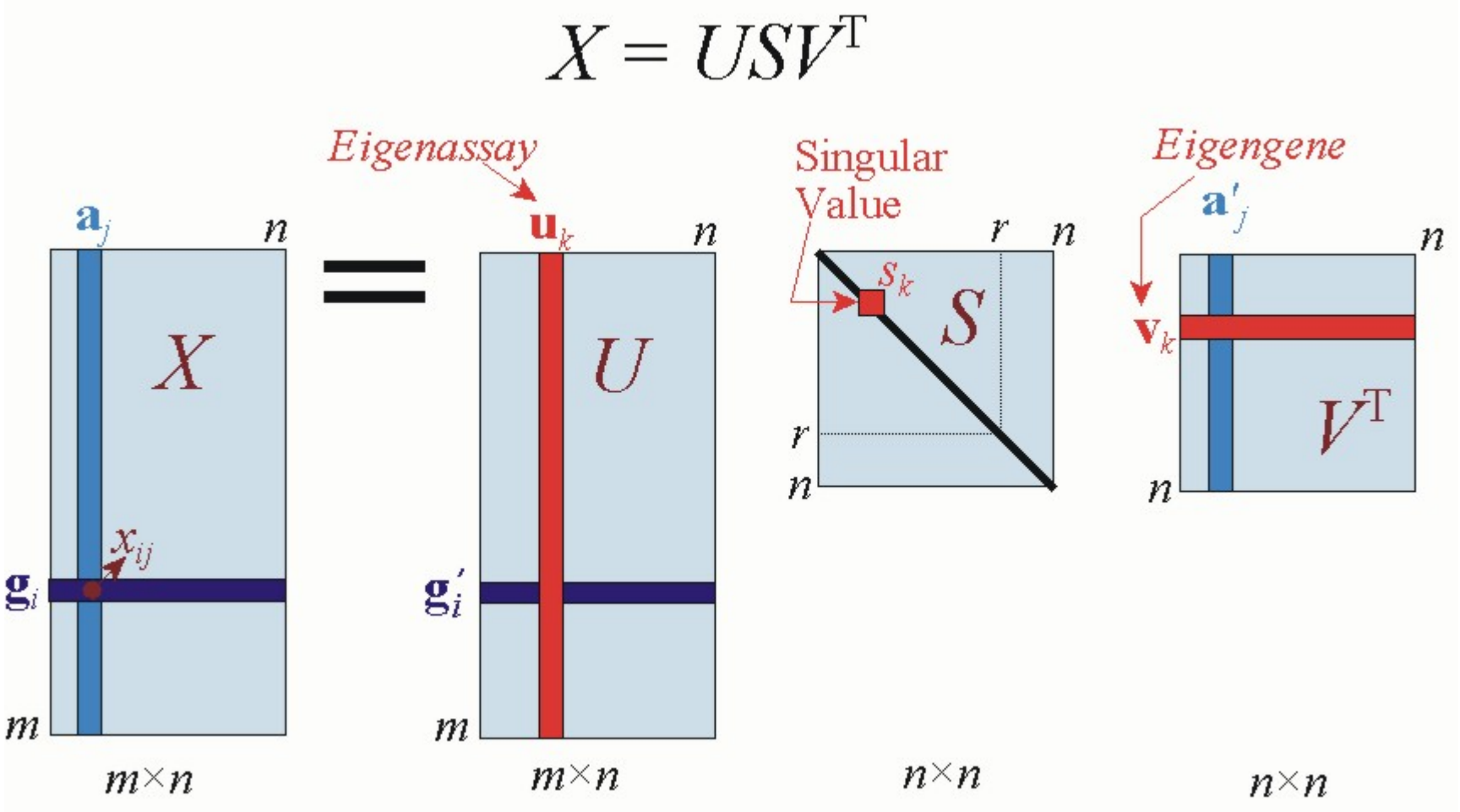
  - an efficient algorithm available in Matlab, R, SAS…

Definition

$$\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]}\, \mathbf{\Lambda}_{[r \times r]}\, (\mathbf{V}_{[m \times r]})^{T}$$

- **A**: *n x m* matrix (e.g., n documents, m terms)
- **U**: *n x r* matrix (n documents, r concepts)
- **Λ**: *r x r* diagonal matrix (strength of each 'concept') (r: rank of the matrix)
- **V**: m x r matrix (m terms, r concepts)

SVD : schema


$X = USV^{\mathrm{T}}$
Eigenassay
Singular Value
Eigengene
$\mathbf{a}_j$
n
$\mathbf{u}_k$
$\mathbf{a}'_j$
X
U
S
V^{T}
s_k
$\mathbf{v}_k$
x_{ij}
$\mathbf{g}_i$
$\mathbf{g}'_i$
m
r
$m \times n$
$n \times n$

SVD - Interpretation

'documents', 'terms' and 'concepts':

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept similarity matrix
- **Λ**: its diagonal elements: 'strength' of each concept

Projection:

- best axis to project on: ('best' = min sum of squares of projection errors)

In matlab:

- svd(A);
- svds(A) – for sparse data.

- The svd command computes the matrix singular value decomposition.

- s = svd(X) returns a vector of singular values.
- [U,S,V] = svd(X) produces a diagonal matrix S of the same dimension as X, with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that X = U*S*V'.

Spectral analysis

Spectral analysis

- The Laplacian matrix $L = D - A$ where
  - A = the adjacency matrix
  - $D = \text{diag}(d_1, d_2, \ldots, d_n)$
    - d_i = degree of node i

- Therefore
  - $L(i,i) = d_i$
  - $L(i,j) = -1$, if there is an edge (i,j)

Laplacian Matrix properties

- The matrix L is symmetric and positive semi-definite
  - all eigenvalues of L are positive

- The matrix L has 0 as an eigenvalue, and corresponding eigenvector $w_1 = (1,1,\ldots,1)$
  - $\lambda_1 = 0$ is the smallest eigenvalue

The second smallest eigenvalue

- The second smallest eigenvalue (also known as Fielder value) λ_2 satisfies

$$\lambda_2 = \min_{x \perp w_1, \|x\|=1} x^T L x$$

- The vector that minimizes λ_2 is called the Fielder vector. It minimizes

$$\lambda_2 = \min_{x \neq 0} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_i x_i^2} \quad \text{where} \quad \sum_i x_i = 0$$

Spectral ordering

- The values of x minimize

$$\min_{x \neq 0} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_i x_i^2} \qquad \sum_i x_i = 0$$

- For weighted matrices

$$\min_{x \neq 0} \frac{\sum_{(i,j)} A[i,j](x_i - x_j)^2}{\sum_i x_i^2} \qquad \sum_i x_i = 0$$

- The ordering according to the x_i values will group similar (connected) nodes together

- Physical interpretation: The stable state of springs placed on the edges of the graph

Spectral partition

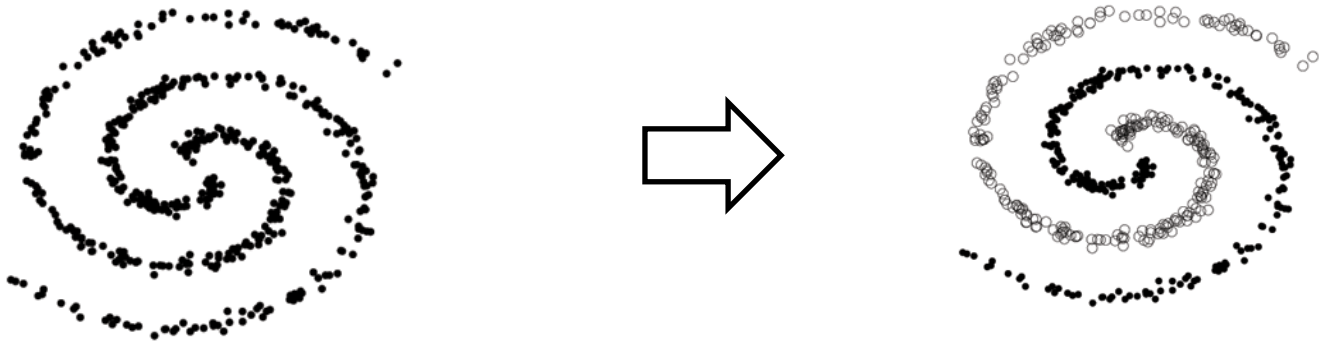
- Partition the nodes according to the ordering induced by the Fielder vector
- If $u = (u_1, u_2, \ldots, u_n)$ is the Fielder vector, then split nodes according to a value s
  - bisection: s is the median value in u
  - ratio cut: s is the value that minimizes α
  - sign: separate positive and negative values ($s=0$)
  - gap: separate according to the largest gap in the values of u
- This works well (provably for special cases)

Spectral Clustering

Referencie:
Andrew Rosenberg,
Speech Lab @ Queens College

Spectral Clustering

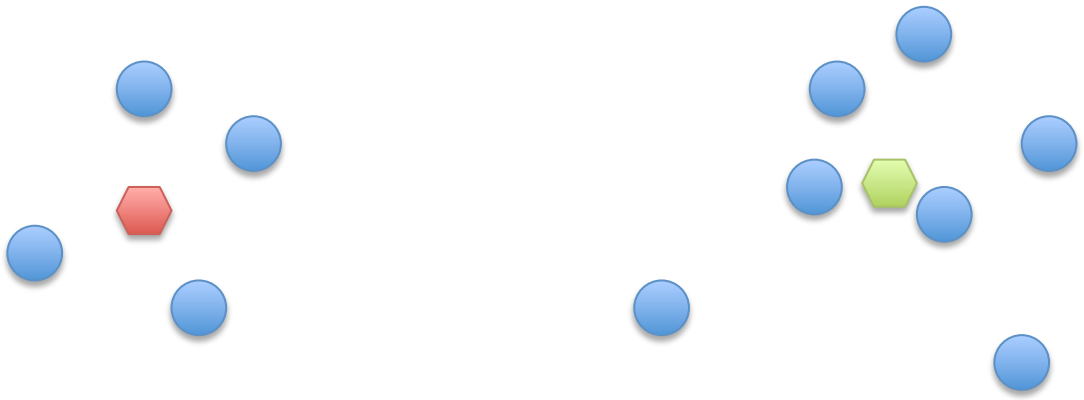
- Spectral Clustering
  - Simple, but powerful method of clustering
  - Requires less assumptions on the form of clusters
  - Outperforms the traditional approaches, such as k-means clustering.



(http://www.squobble.com/academic/publications/FFF_MIMO/node4.html)

Partitional Clustering

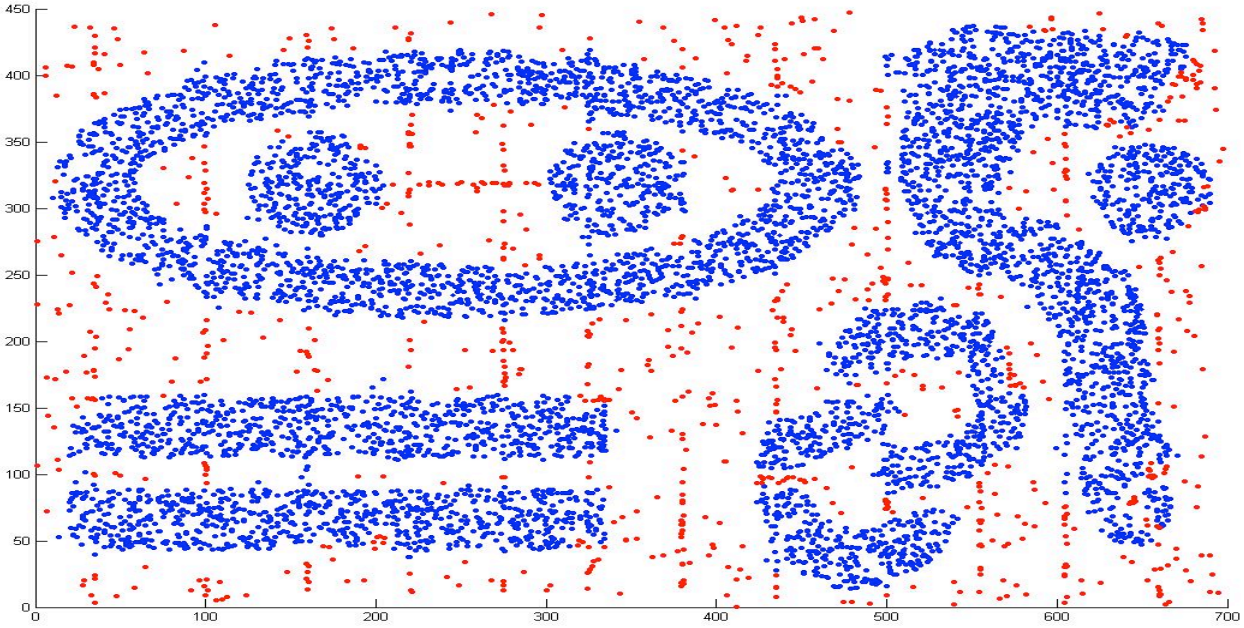
- How do we partition a space to make the best clusters?
- Proximity to a cluster centroid.



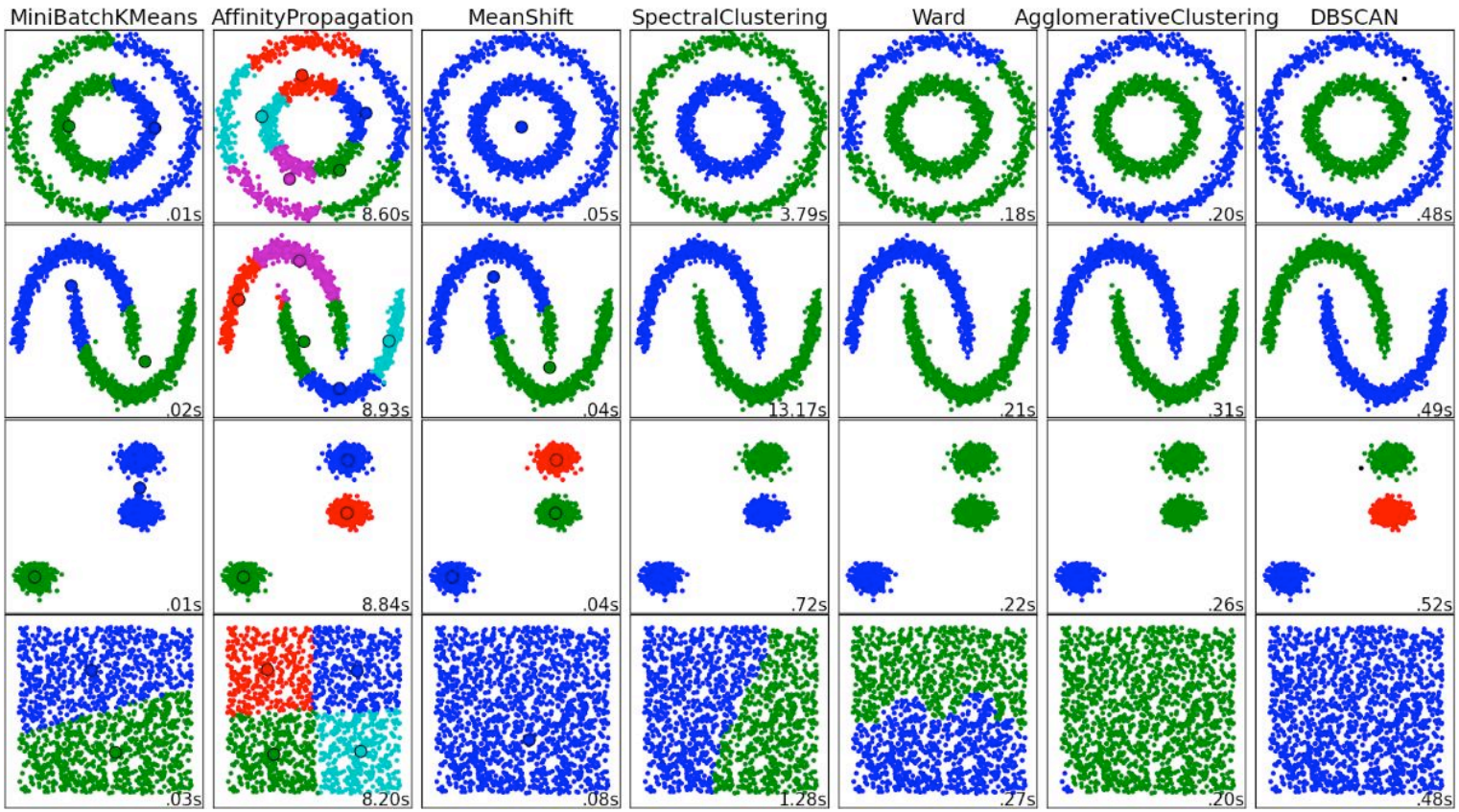
Difficult Clusterings

- But some clusterings don't lend themselves to a "centroid" based definition of a cluster.

Spectral clustering allows us to address these sorts of clusters.

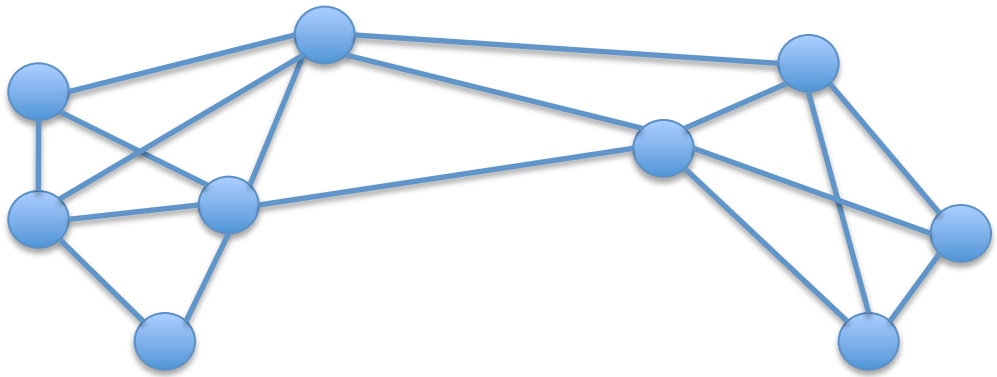


Difficult Clusterings



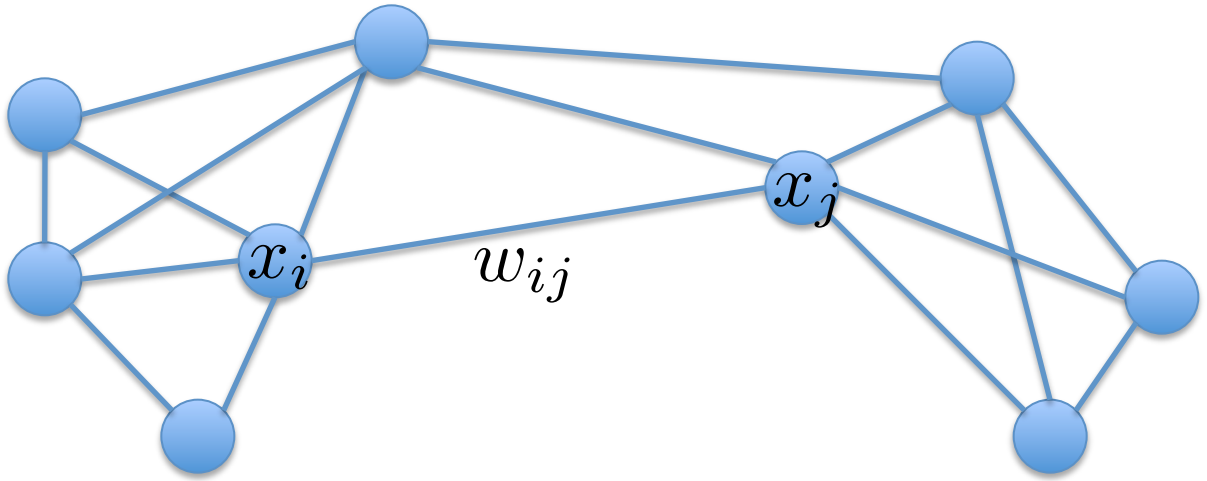
Spectral Clustering (1)

Given a set of data points $x_1, x_2, ..., x_n \ \ in\mathbb{R}^m$, spectral clustering first constructs an undirected graph $G = (V, E)$ represented by its adjacency matrix $W = (w_{ij})_{i,j=1}^n$, where $w_{ij} \geq 0$ denotes the similarity (affinity) between x_i and x_j. The degree matrix D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Let $L = D - W$, which is called Laplacian graph. Spectral clustering then use the top k eigenvectors of L (or, the normalized Laplacian $D^{-1/2}LD^{-1/2}$) corresponding to the k smallest eigenvalues as the low dimensional representations of the original data. Finally, k-means method is applied to obtain the clusters.



Spectral Clustering (2)

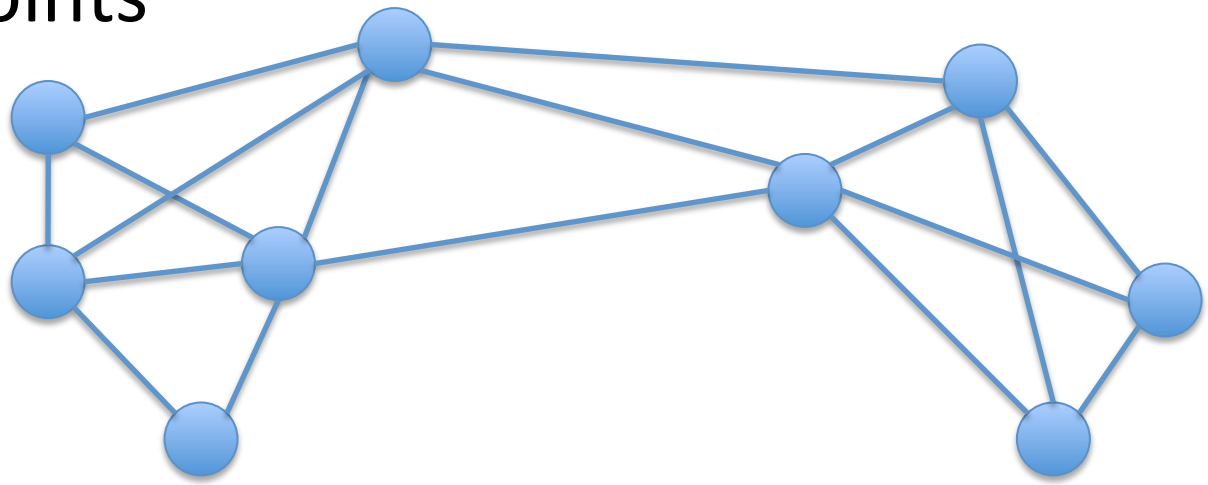
- Spectral clustering method needs to construct an adjacency matrix and calculate the eigen-decomposition of the corresponding Laplacian matrix. Both of these two steps are computational expensive. Then, it is not easy to apply spectral clustering on large-scale data sets.



pairwise affinity : $$w_{ij} = d(x_i, x_j) = \exp\left\{\frac{\|x_i - x_j\|}{\sigma^2}\right\}$$

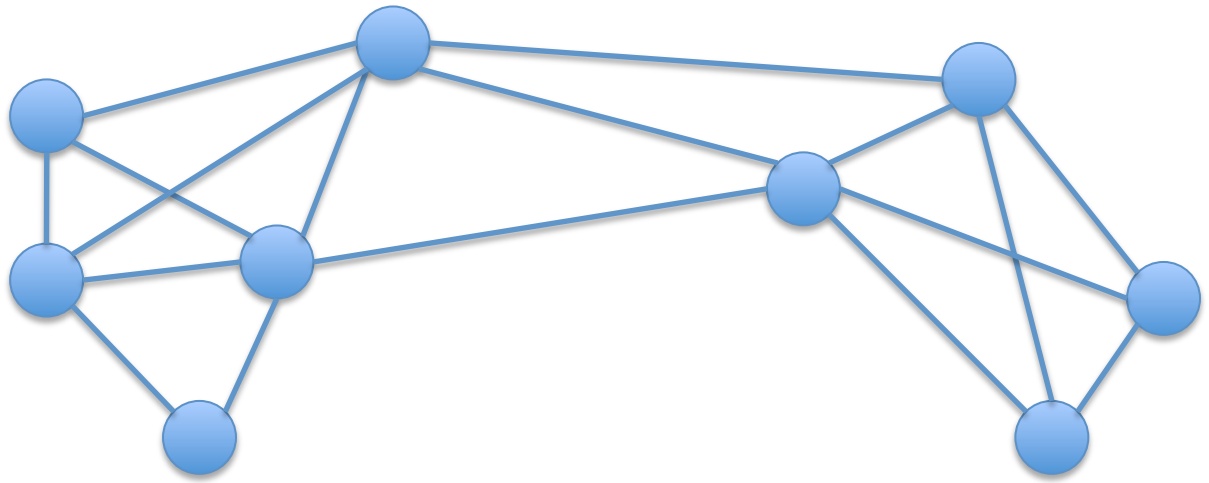
Graph Representation

- We can represent the relationships between data points in a graph.
- Weight the edges by the similarity between points



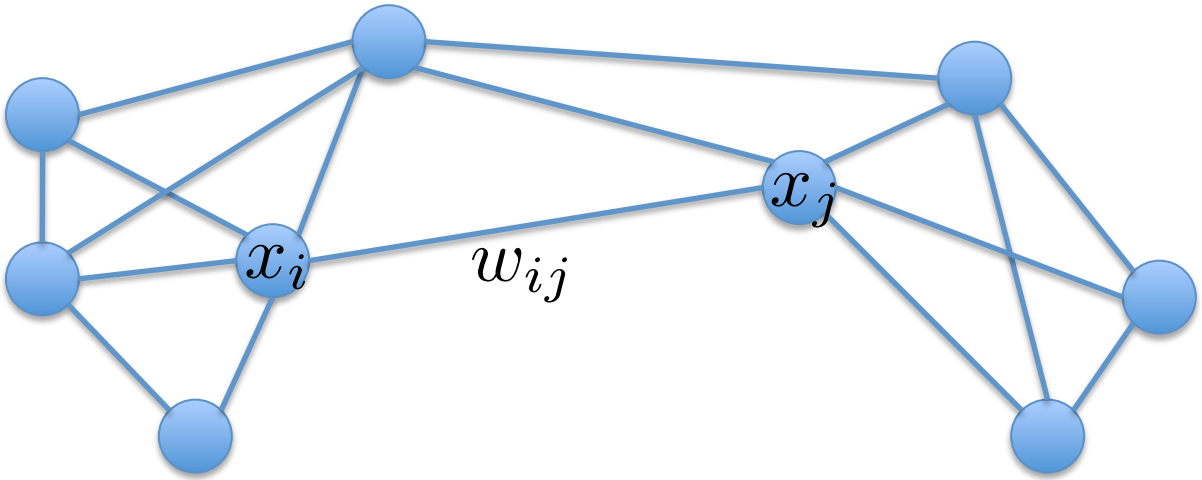
Representing data in a graph

- What is the best way to calculate similarity between two data points?
- Distance based: $d(x_i, x_j) = \exp\left\{ \frac{\|x_i - x_j\|}{\sigma^2} \right\}$



Graphs

- Nodes and Edges
- Edges can be directed or undirected
- Edges can have weights associated with them



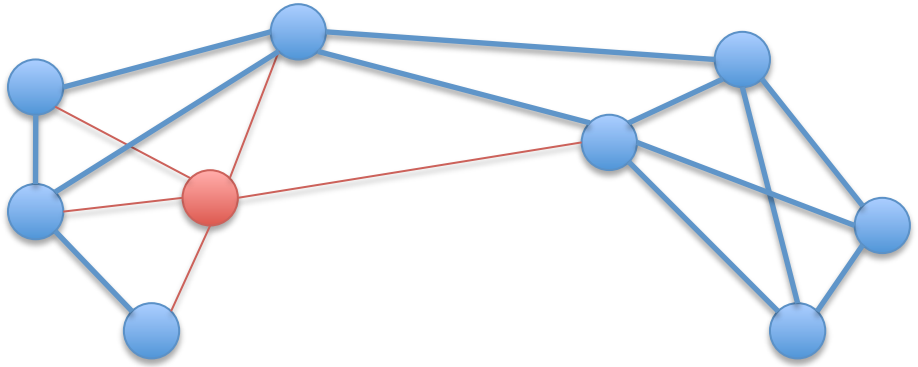
- Here the weights correspond to **pairwise affinity**

$$w_{ij} = d(x_i, x_j) = \exp\left\{\frac{\|x_i - x_j\|}{\sigma^2}\right\}$$

Graphs

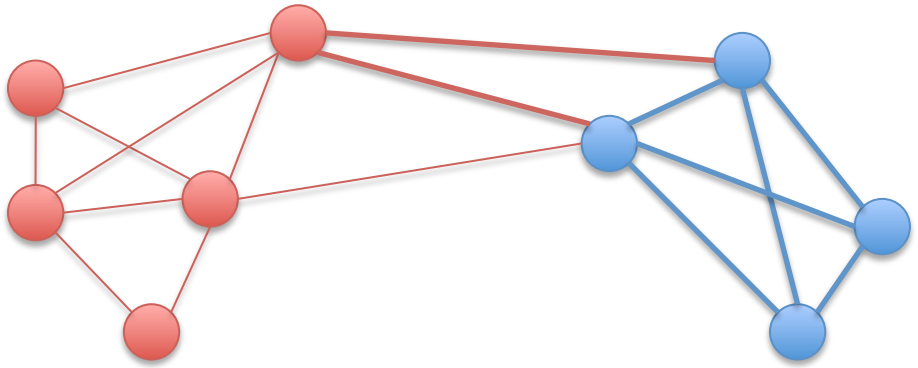
- Degree

$$D(x_i) = \sum_{j \in V} w_{ij}$$



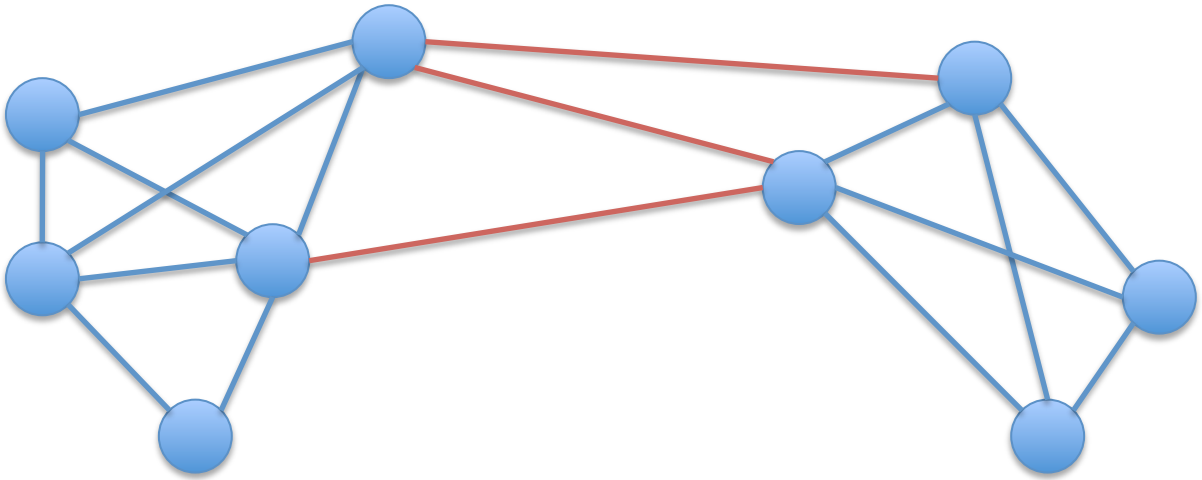
- Volume of a set

$$Vol(C) = \sum_{i \in C} D(x_i)$$



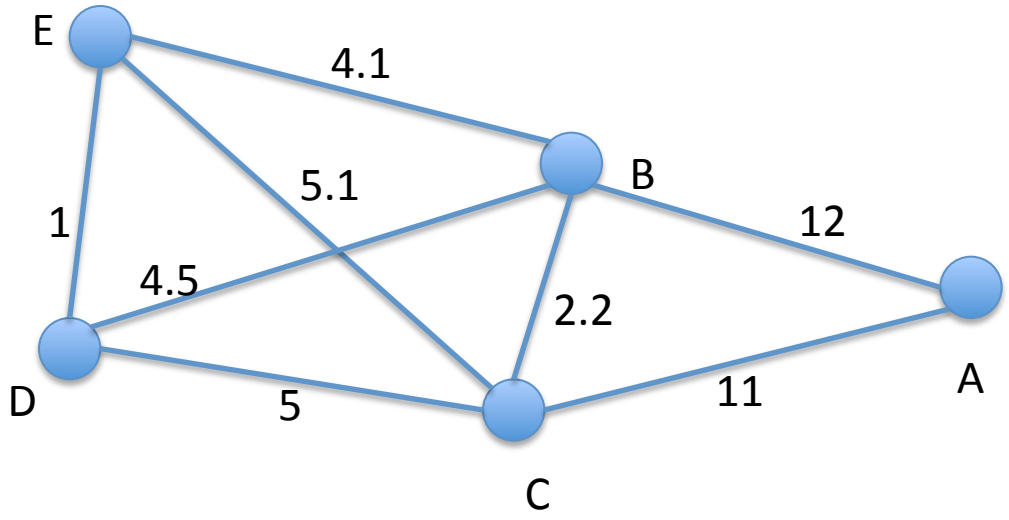
Graph Cuts

- The **cut** between two subgraphs is calculated as follows



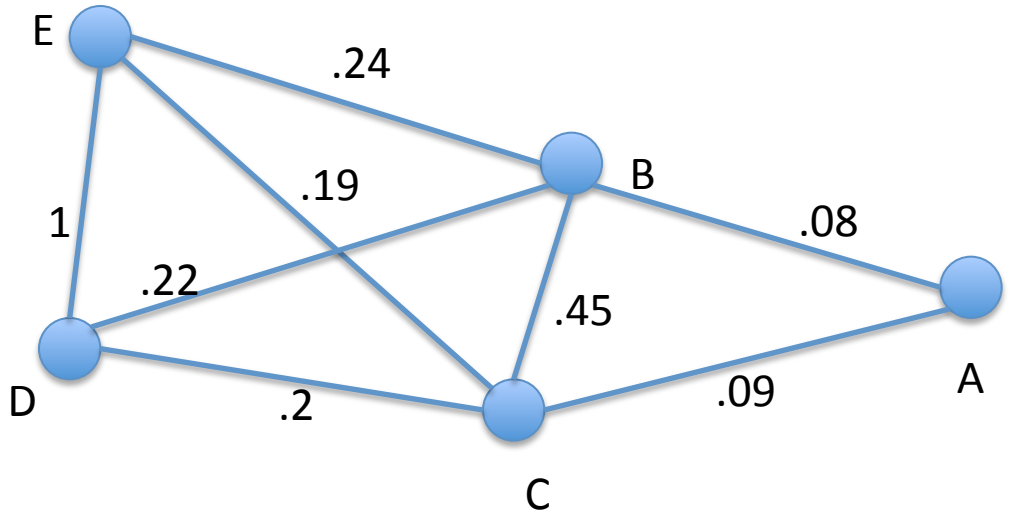
$$Cut(C_1, C_2) = \sum_{i \in C_1} \sum_{j \in C_2} w_{ij}$$

Graph Examples - Distance



Height	Weight
20	5
8	6
9	4
4	4
4	5

Graph Examples - Similarity



Height	Weight
20	5
8	6
9	4
4	4
4	5

Intuition

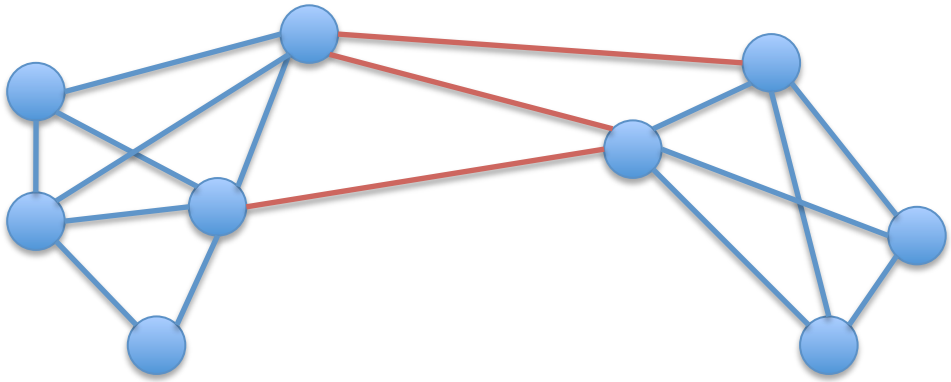
- The minimum cut of a graph identifies an optimal partitioning of the data.

- Spectral Clustering
  - Recursively partition the data set
    - Identify the minimum cut
    - Remove edges
    - Repeat until k clusters are identified

Graph Cuts

- Minimum (bipartitional) cut

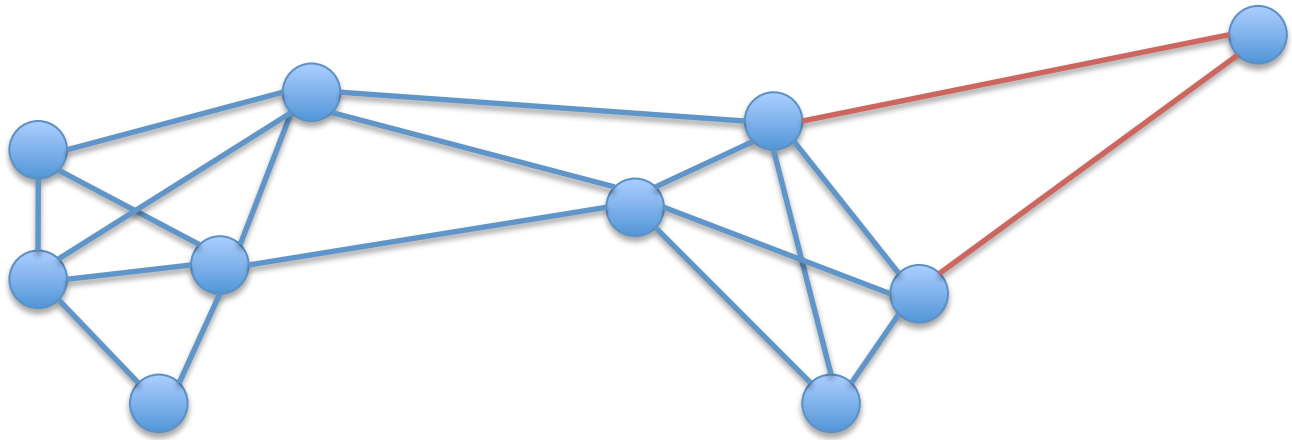
$$\min Cut(C_1, C_2) = \sum_{i \in C_1} \sum_{j \in C_2} w_{ij}$$



Graph Cuts

- Minimum (bipartitional) cut

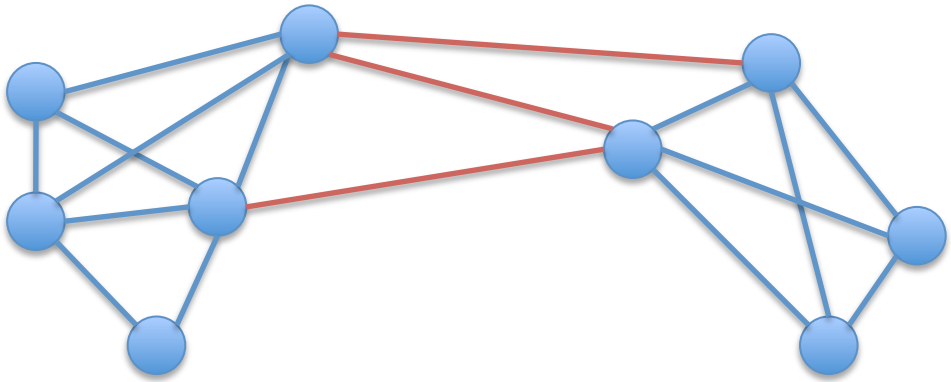
$$\min Cut(C_1, C_2) = \sum_{i \in C_1} \sum_{j \in C_2} w_{ij}$$



Graph Cuts

- Minimal (bipartitional) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left(\frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$



- Unnormalized cuts are attracted to outliers.

Graph definitions

- ε-neighborhood graph
  - Identify a threshold value, ε, and include edges if the affinity between two points is greater than ε.
- k-nearest neighbors
  - Insert edges between a node and its k-nearest neighbors.
  - Each node will be connected to (at least) k nodes.
- Fully connected
  - Insert an edge between every pair of nodes.

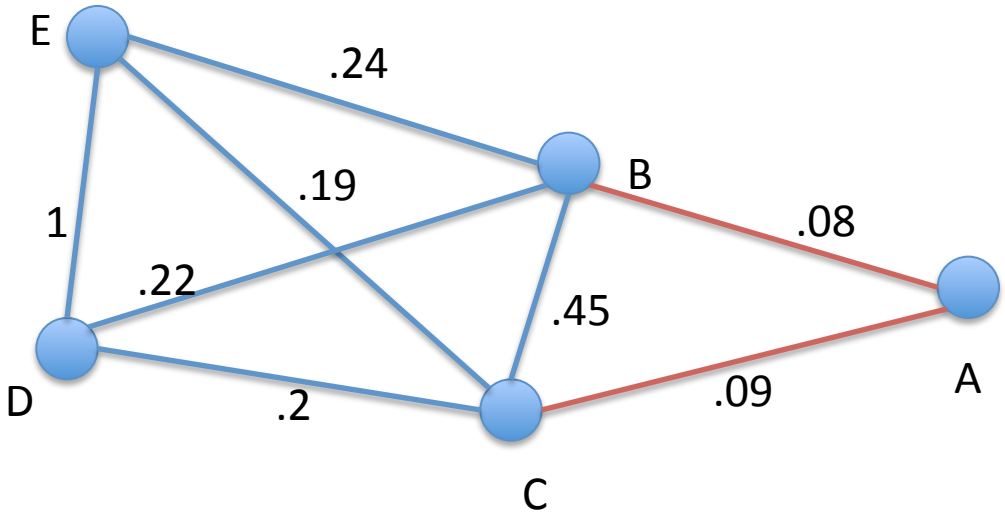
Intuition

- The minimum cut of a graph identifies an optimal partitioning of the data.

- Spectral Clustering
  - Recursively partition the data set
    - Identify the minimum cut
    - Remove edges
    - Repeat until k clusters are identified

Spectral Clustering Example

- Minimum Cut



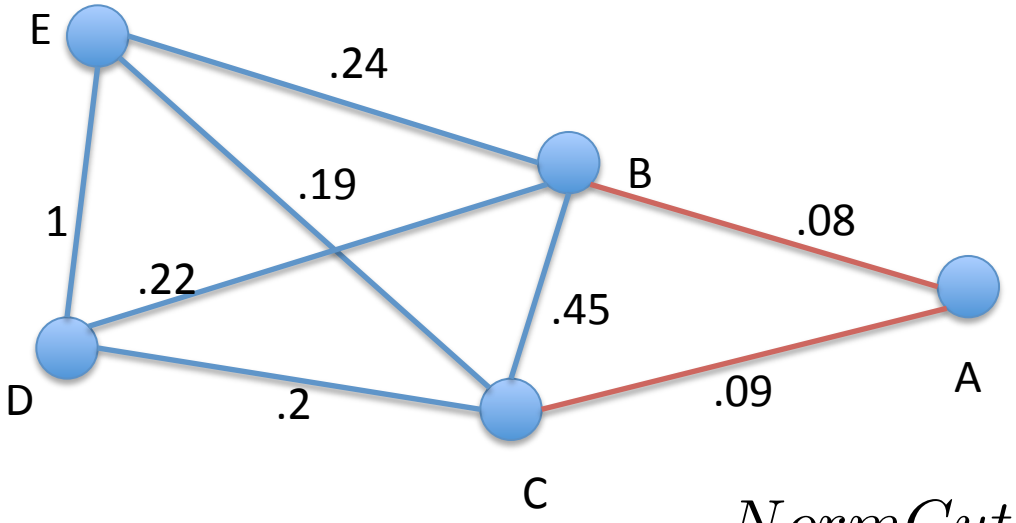
Height	Weight
20	5
8	6
9	4
4	4
4	5

$$Cut(BCDE, A) = 0.17$$

Spectral Clustering Example

- Normalized Minimum Cut

$$NormCut(C_1, C_2) = \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)}$$



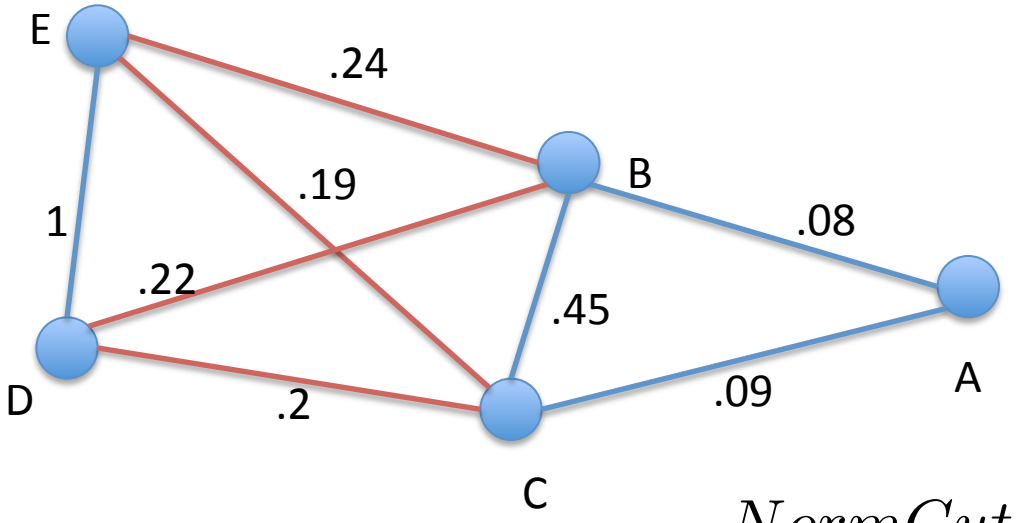
Height	Weight
20	5
8	6
9	4
4	4
4	5

$$NormCut(BCDE, A) = 1.067$$

Spectral Clustering Example

- Normalized Minimum Cut

$$NormCut(C_1, C_2) = \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)}$$



Height	Weight
20	5
8	6
9	4
4	4
4	5

$$NormCut(BCDE, A) = 1.067$$

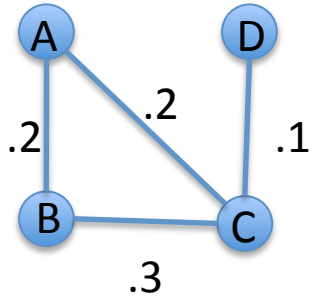
$$NormCut(ABC, DE) = 1.038$$

Problem

- Identifying a minimum cut is NP-hard.
- There are efficient approximations using linear algebra.
- Based on the Laplacian Matrix, or **graph Laplacian**

Spectral Clustering

- Construct an **affinity** matrix



$$W =$$

	A	**B**	**C**	**D**
A	.4	.2	.2	0
B	.2	.5	.3	0
C	.2	.3	.6	.1
D	0	0	.1	.1

Spectral Clustering

- Construct the graph Laplacian

$$D = diag(D_0, D_1, \ldots, D_n)$$

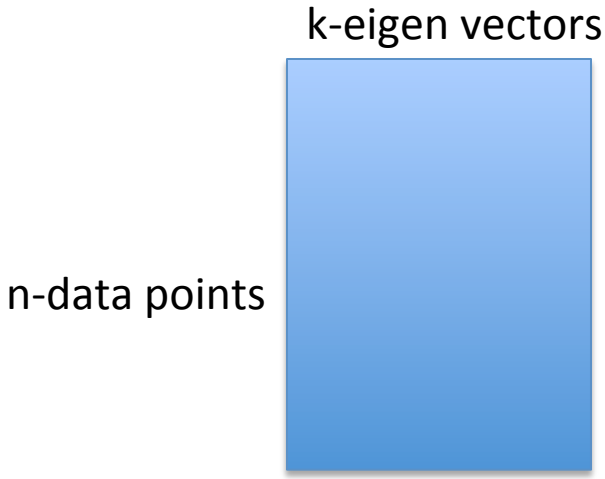
$$L = D - W$$

- Identify eigenvectors of the affinity matrix

$$Lv = \lambda v$$

Spectral Clustering

- K-Means on eigenvector transformation of the data.



Each Row represents a data point in the eigenvector space

- Project back to the initial data representation.

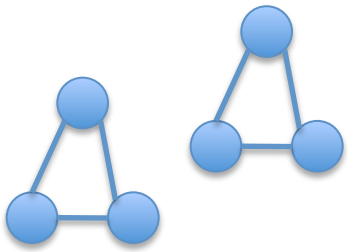
Overview: what are we doing?

- Define the affinity matrix
- Construct the Laplacian
- Identify eigenvalues and eigenvectors.
- K-means of transformed data
- Project back to original space

Why does this work?

- Ideal Case

$$Lv = \lambda v$$



1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

1	**0**
1	0
1	0
0	1
0	1
0	1

- What are we optimizing? Why do the eigenvectors of the laplacian include cluster identification information

Why does this work?

- How does this eigenvector decomposition address this? $f(x_j) = f_j$ cluster assignment

$$L = D - W$$

$$\begin{aligned} f^T L f &= f^T D f - f^T W f \\ &= \sum_i d_i f_i^2 - \sum_{ij} f_i f_j w_{ij} \\ &= \frac{1}{2}\left(\sum_i \left(\sum_j w_{ij}\right) f_i^2 - 2\sum_{ij} f_i f_j w_{ij} + \sum_j \left(\sum_i w_{ij}\right) f_j^2\right) \\ &= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2 \end{aligned}$$

Cluster objective function – normalized cut!

- if we let f be eigen vectors of L, then the eigenvalues are the cluster objective functions

Normalized Graph Cuts view

- Minimal (bipartitional) normalized cut.

$$\min \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_1, C_2)}{Vol(C_2)} = \min \left(\frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)} \right) Cut(C_1, C_2)$$

$$NCut(A, B) = \frac{y^T (D - W) y}{y^T D y}$$

$$\min_y y^T (D - W) y \text{ subject to } y^T D y = 1$$

$$(D - W) y = \lambda D y$$

- Eigenvalues of the laplacian are approximate solutions to mincut problem.

The Laplacian Matrix

- L = D-W
- Positive semi-definite $x^T Lx \geq 0$
- The lowest eigenvalue is 0, eigenvector is $\vec{1}$
- The second lowest contains the solution
  - The corresponding eigenvector contains the cluster indicator for each data point

$$\lambda_2 = \frac{Cut(A,B)}{|A|} + \frac{Cut(A,B)}{|B|}$$

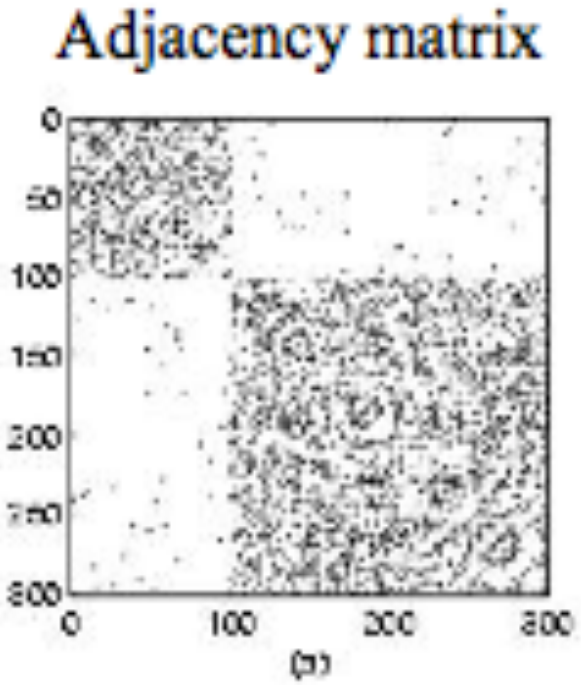
Using eigenvectors to partition

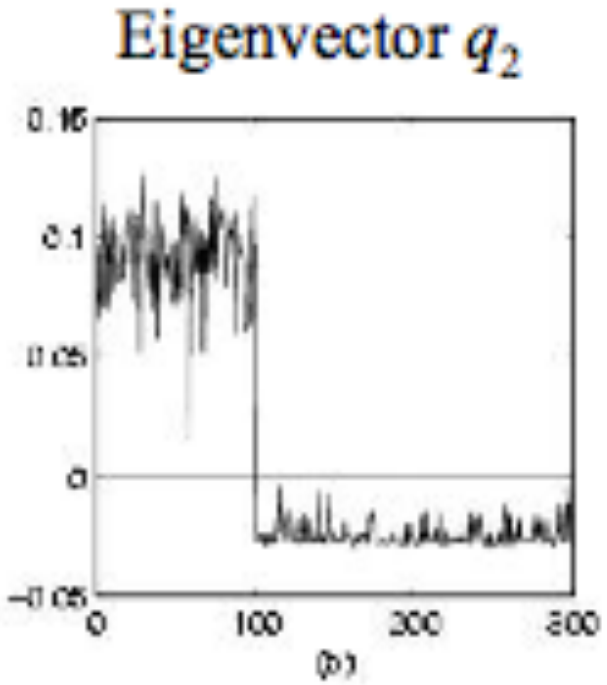
- Each eigenvector partitions the data set into two clusters.
- The entry in the second eigenvector determines the first cut.

- Subsequent eigenvectors can be used to further partition into more sets.

Example

- Dense clusters with some sparse connections





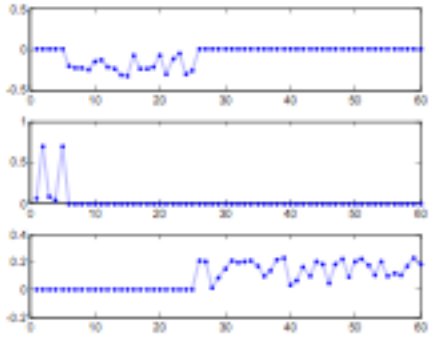
3 class Example

Affinity matrix

$W; A$

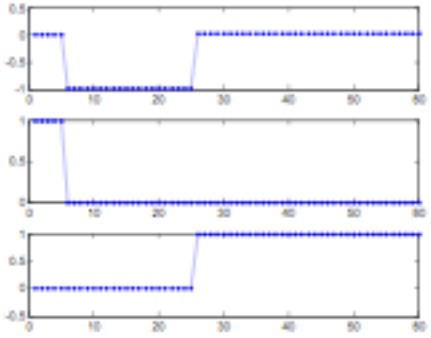
eigenvectors

$V = [v_1, v_2, v_3]$

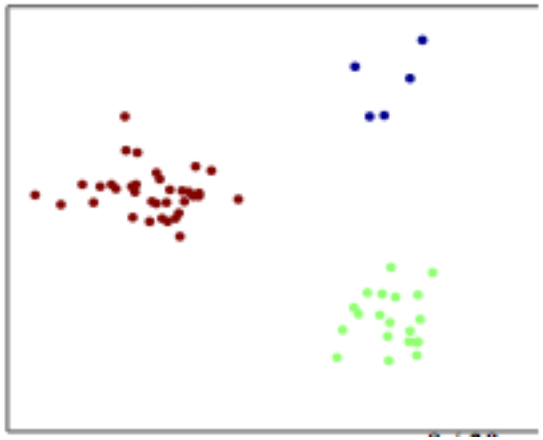


row normalization

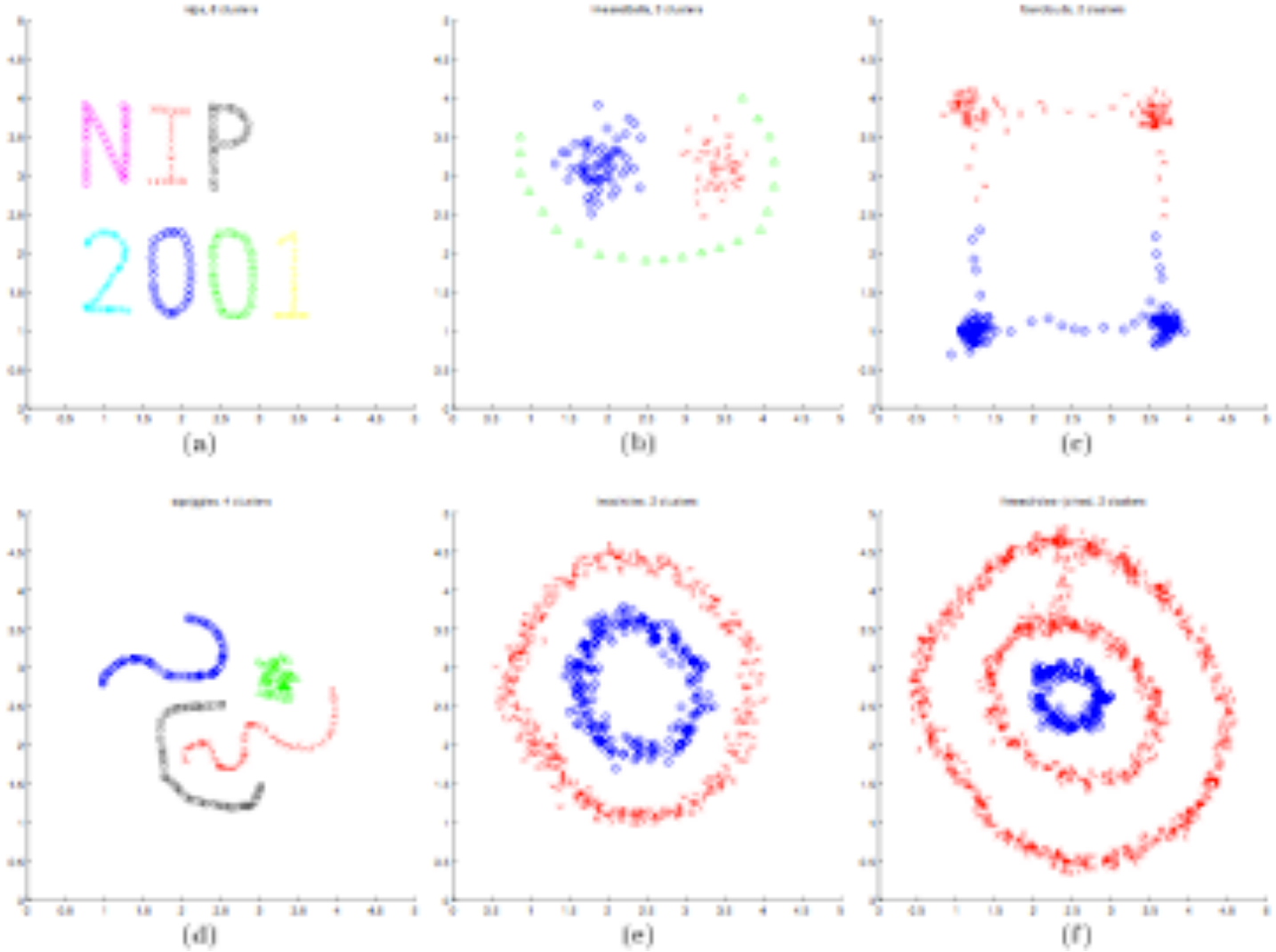
$U = [u_1, u_2, u_3]$



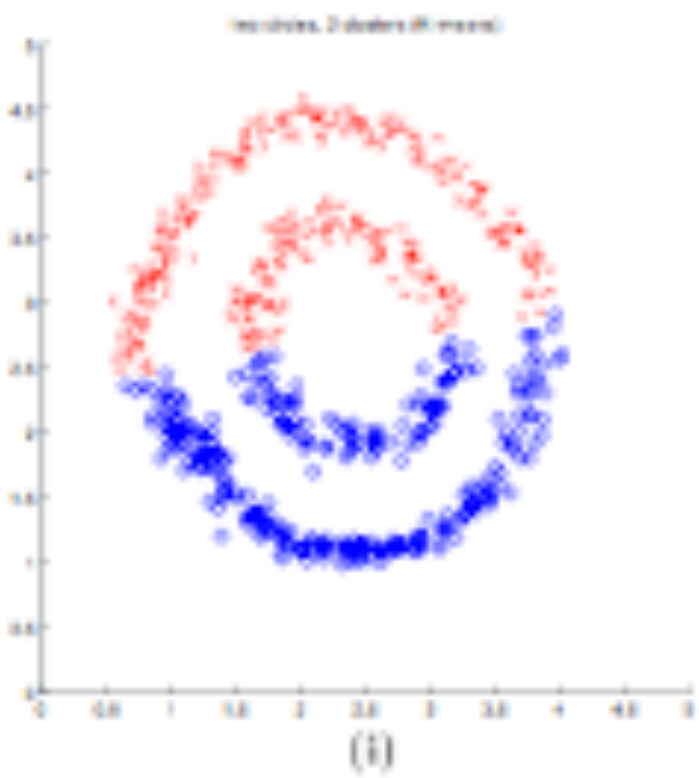
output



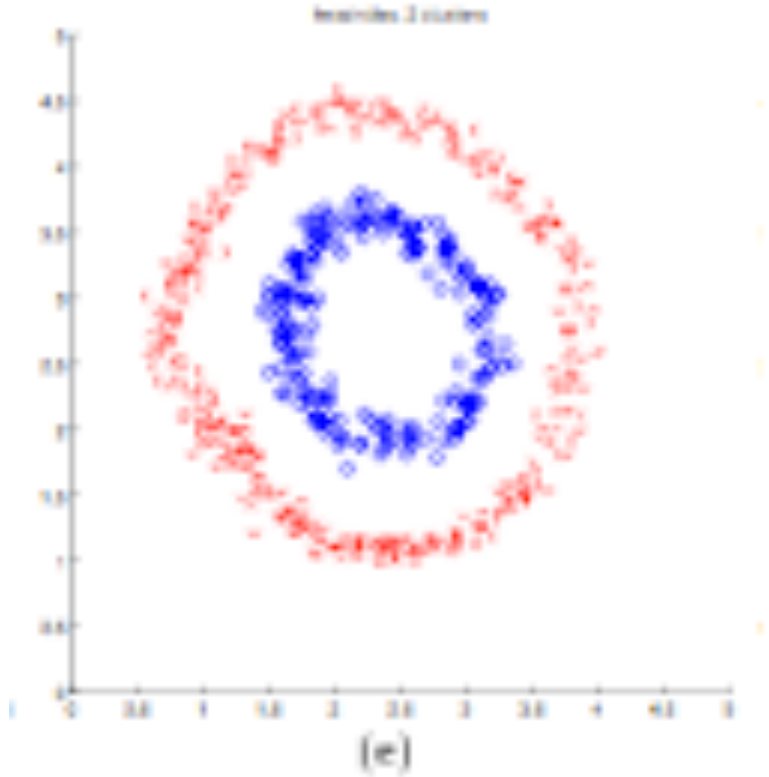
Example [Ng et al. 2001]



k-means vs. Spectral Clustering



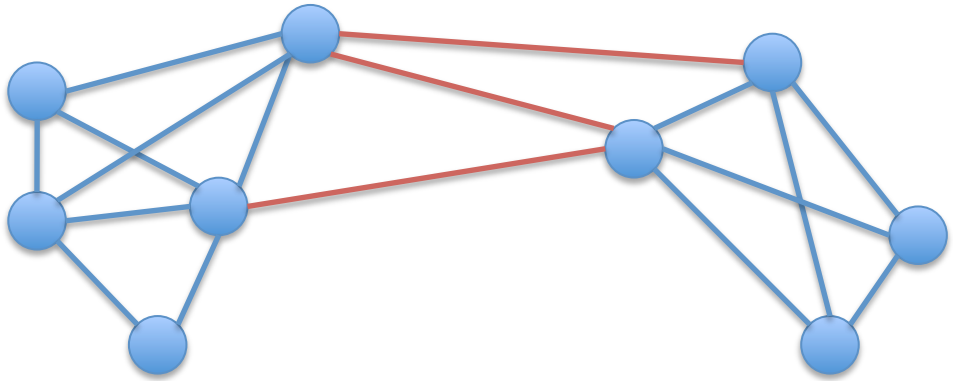
K-means



Spectral Clustering

Random walk view of clustering

- In a random walk, you start at a node, and move to another node with some probability.
- The intuition is that if two nodes are in the same cluster, you a randomly walk is likely to reach both points.

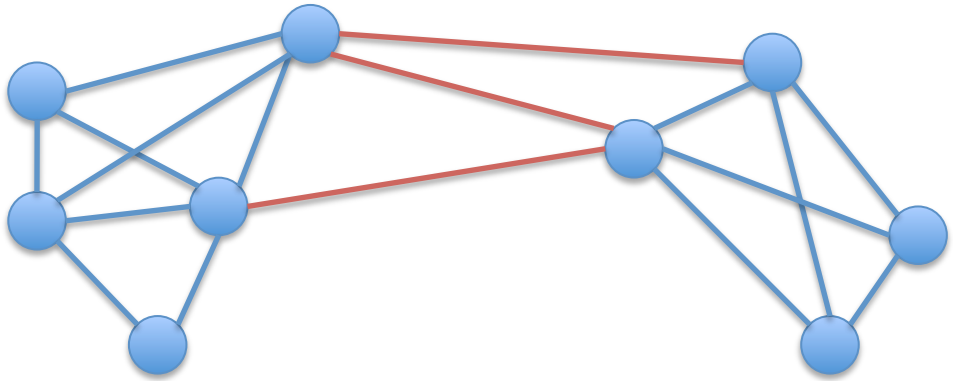


Random walk view of clustering

- Transition matrix:

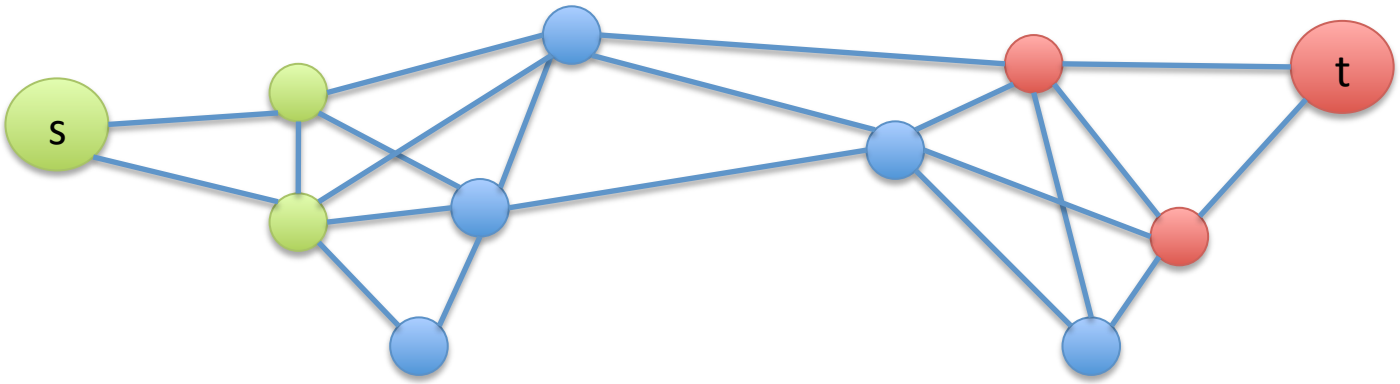
$$L_{rw} = \mathbb{I} - D^{-1}W = D^{-1}L$$

- The transition probability is related to the weight of given transition and the inverse degree of the current node.



Using minimum cut for semi supervised classification?

- Construct a graph representation of unseen data.
- Insert imaginary nodes s and t connected to labeled points with infinite similarity.
- Treat the min cut as a maximum flow problem from s to t



Kernel Method

- The weight between two nodes is defined as a function of two data points.

$$w_{ij} = d(x_i, x_j) = \exp\left\{\frac{\|x_i - x_j\|}{\sigma^2}\right\}$$

- Whenever we have this, we can use any valid Kernel.

$$w_{ij} = K(x_i, x_j)$$

Andrew Y. Ng spectral clustering method

Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-||s_i - s_j||^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i,i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Clustering evaluation

- Internal indexes
  - Based only on clustering results
- External indexes
  - Using the real label (usually used for supervised machine learning)

Internal index: Bic index

- The Bayesian information criterion (BIC) is devised to avoid overfitting, and is defined as:

$$BIC = -\ln(L) + vln(n)$$

where n is the number of objects, L is the likelihood of the parameters to generate the data in the model, and v is the number of free parameters in the Gaussian model. The BIC index takes into account both fit of the model to the data andthe complexity of the model. A model that has a smaller BIC is better.

Internal index: Calinski-Harabasz index

This index is computed by

$$CH = \frac{trace(S_B)}{trace(S_w)} \cdot \frac{n_p - 1}{n_p - k}$$

Where (S_B) is the between-cluster scatter matrix, (S_w) the internal scatter matrix, n_p the number of clustered samples, and k the number of clusters.

Internal index: Davies-Bouldin index (DB)

This index aim to identify sets of clusters that are compact and well separated. The Davies-Bouldin index is defined as:

$$BD = \frac{1}{c}\sum_{i=1}^{c} Max_{i \neq j} \left\{ \frac{d(X_i) + d(X_j)}{d(c_i, c_j)} \right\}$$

Were c denotes the number of clusters, i, j are cluster labels, then $d(X_i)$ and $d(X_j)$ are all samples in clusters i and j to their respective cluster centroids, $d(c_i, c_j)$ is the distance between these centroid. Smaller value of DB indicates a "better" clustering solution.

Internal index: Silhouette index

For a given cluster, $X_j (j = 1, .. c)$, the silhouette technique assigns to the ith sample of X_j a quality measure, $s(i) = (i = 1, ... m)$, known as the silhouette width. This value is a confidence indicator on the membership of the ith sample in the cluster X_j and it is defined a defined as:

$$s(i) = \frac{\big(b(i) - a(i)\big)}{Max\{a(i), b(i)\}}$$

Were $a(i)$ is the average distance between the ith sample and all of samples included in X_j ; $b(i)$ is the minimum average distance between the $it\ sh$ and all of the samples clustered in $X_k (k = 1, .. c; k \neq j)$

External measure: F-measure

Combines the precision and recall concepts from information retrival. We then calculate the recall and precision of that cluster for each class as:

$$Recall(i,j) = \frac{n_{ij}}{n_i}$$

And

$$Precision(i,j) = \frac{n_{ij}}{n_j}$$

Were n_{ij} is the number of objects of class i that are in cluster j, n_j is the number of objects in cluster j, and n_i, is the number of objects in class i. The $F - Measure$ of cluster j and class i is given by the following equation:

$$F(i,j) = \frac{2Recall(i,j)Precision(i,j)}{Precision(i,j) + Recall(i,j)}$$

The $F - Measure$ values are within the interval [0,1] and larger values indicate higher clustering quality.

External measure: NMI measure

Is called Normalized Mutual Information (NMI). The NMI of two labeled objects can be measured as:

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}$$

Where, $I(X,Y)$ denotes the mutual information between two random variables X and Y and $H(X)$ denotes the entropy of X,X will be consensus clustering while Y will be the true labels.

External measure: Purity (accuracy)

Purity is very similar to entropy. We calculate the purity of a set of clusters. First, we cancel the purity in each cluster. For each cluster, we have the purity $P_j = \frac{1}{n_j} Max_i\left(n_j^i\right)$ is the number of objects in j with class label i. In other words, P_j is a fraction of the overall cluster size that the largest class of objects assigned to that cluster represents. The overall purity of the clustering solution is obtained as a weighted sum of the individual cluster purities and given as:

$$Purity = \sum_{j=1}^{m} \frac{n_j}{n} P_j$$

Were n_j is the size of cluster j, m is the number of clusters,

External measure: Entropy

Entropy measures the purity of the clusters class labels. Thus, if all clusters consist of objects with only a single class label, the entropy is 0. However, as the class labels of objects in a cluster become more varied, the entropy increases. To compute the entropy of a dataset, we need to calculate the class distribution of the objects in each cluster as follows:

$$E_{j=} \sum_{i} p_{ij} \log (p_{ij})$$

Where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the weighted sum of the entropies of all clusters, as shown in the next equation

$$E = \sum_{j=1}^{m} \frac{n_j}{n} E_j$$

Were n_j is the size of cluster j, m is the number of clusters, and n is the total number of data points.

External measures

index	formula
Rand index	$Rand = \frac{a_{00} + a_{11}}{a_{00} + a_{01} + a_{10} + a_{11}}$
Adjusted Rand index	$AdjustedRand = \frac{a_{00} + a_{11} - n_c}{a_{00} + a_{01} + a_{10} + a_{11} - n_c}$
Jaccard index	$Jaccard = \frac{a_{11}}{a_{01} + a_{10} + a_{11}}$
Wallace's coefficient	$W_{P1 \to P2} = \frac{a_{11}}{a_{11} + a_{10}}$ and $W_{P2 \to P1} = \frac{a_{11}}{a_{11} + a_{01}}$
Adjusted Wallace index	$AW_{P1 \to P2} = \frac{W_{P1 \to P2} - Wi_{P1 \to P2}}{1 - Wi_{P1 \to P2}}$
Normalized Mutual Information	$NMI = \frac{-2 \sum_{ij} n_{ij} log \frac{n_{ij} N}{n_i n_j}}{\sum_i n_i log \frac{n_i}{N} + \sum_j n_j log \frac{n_j}{N}}$
Variation of Information	$VI = -2 \sum_{ij} \frac{n_{ij}}{N} log \frac{n_{ij} N}{n_i n_j} - \sum_i \frac{n_i}{N} log \frac{n_i}{N} - \sum_j \frac{n_j}{N} log \frac{n_j}{N}$

- Andrew Rosenberg, Speech Lab @ Queens College
  Computer Science Department
  New Science Building A332
  Queens College / CUNY
  CUNY Graduate Center Rm 4420