Cluster Analysis

[Based on the notes by D. Hebert, I. S. Dhillon, Kuei-Hsien, V. Kashyap]

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Measure the Quality of Clustering

- Dissimilarity/Similarity metric:

  Similarity is expressed in terms of a distance function, which is typically metric: *d*(*i, j*)
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for different types of data.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define "similar enough" or "good enough"
  - the answer is typically highly subjective.

Data Structures

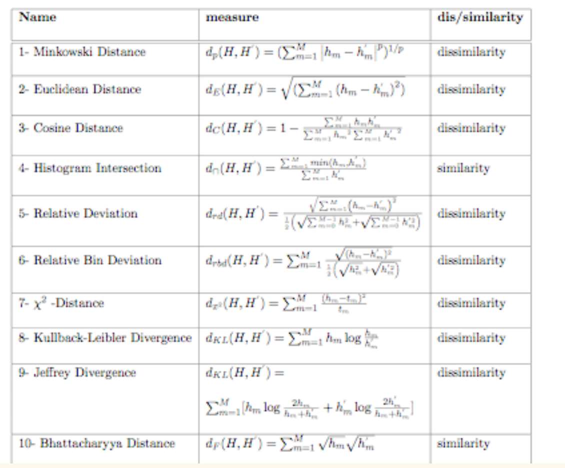
- Data matrix
  - (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Similarity Measures



Name	measure	dis/similarity
1- Minkowski Distance	$d_p(H,H') = (\sum_{m=1}^{M} \lvert h_m - h'_m \rvert^p)^{1/p}$	dissimilarity
2- Euclidean Distance	$d_E(H,H') = \sqrt{(\sum_{m=1}^{M} (h_m - h'_m)^2)}$	dissimilarity
3- Cosine Distance	$d_C(H,H') = 1 - \frac{\sum_{m=1}^{M} h_m h'_m}{\sum_{m=1}^{M} h_m^2 \sum_{m=1}^{M} h'^2_m}$	dissimilarity
4- Histogram Intersection	$d_\cap(H,H') = \frac{\sum_{m=1}^{M} min(h_m, h'_m)}{\sum_{m=1}^{M} h'_m}$	similarity
5- Relative Deviation	$d_{rd}(H,H') = \frac{\sqrt{\sum_{m=1}^{M} (h_m - h'_m)^2}}{\frac{1}{2}\left(\sqrt{\sum_{m=0}^{M-1} h_m^2} + \sqrt{\sum_{m=0}^{M-1} h'^2_m}\right)}$	dissimilarity
6- Relative Bin Deviation	$d_{rbd}(H,H') = \sum_{m=1}^{M} \frac{\sqrt{(h_m - h'_m)^2}}{\frac{1}{2}\left(\sqrt{h_m^2} + \sqrt{h'^2_m}\right)}$	dissimilarity
7- χ^2 -Distance	$d_{\chi^2}(H,H') = \sum_{m=1}^{M} \frac{(h_m - t_m)^2}{t_m}$	dissimilarity
8- Kullback-Leibler Divergence	$d_{KL}(H,H') = \sum_{m=1}^{M} h_m \log \frac{h_m}{h'_m}$	dissimilarity
9- Jeffrey Divergence	$d_{KL}(H,H') =$ $\sum_{m=1}^{M} [h_m \log \frac{2h_m}{h_m + h'_m} + h'_m \log \frac{2h'_m}{h_m + h'_m}]$	dissimilarity
10- Bhattacharyya Distance	$d_F(H,H') = \sum_{m=1}^{M} \sqrt{h_m}\sqrt{h'_m}$	similarity

Dissimilarity between Binary Variables

- Example – do patients have the same disease

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- gender is a symmetric attribute (use Jaccard coefficeint so ignore)
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

Most likely to have same disease

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Unlikely to have same disease

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p-m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- order is important, e.g., rank (gold, silver, bronze)
- Can be treated like interval-scaled-use following steps
  - *f* is variable from set of variables, value of *f* for *i*th object is x_{if}
  - replace each x_{if} by its' rank $r_{if} \in \{1,\ldots,M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables

Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.
- One may use a weighted formula to combine their effects.

$$d(i, j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

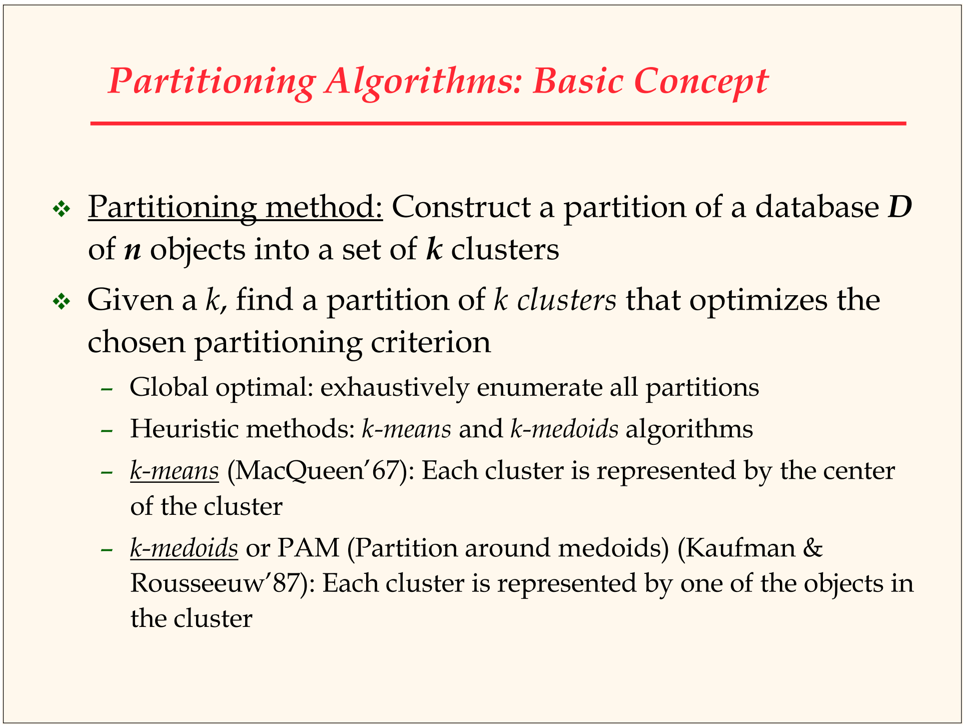
  - f is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ o.w.
  - f is interval-based: use the normalized distance
  - f is ordinal or ratio-scaled
    - compute ranks r_{if} and
    - and treat z_{if} as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

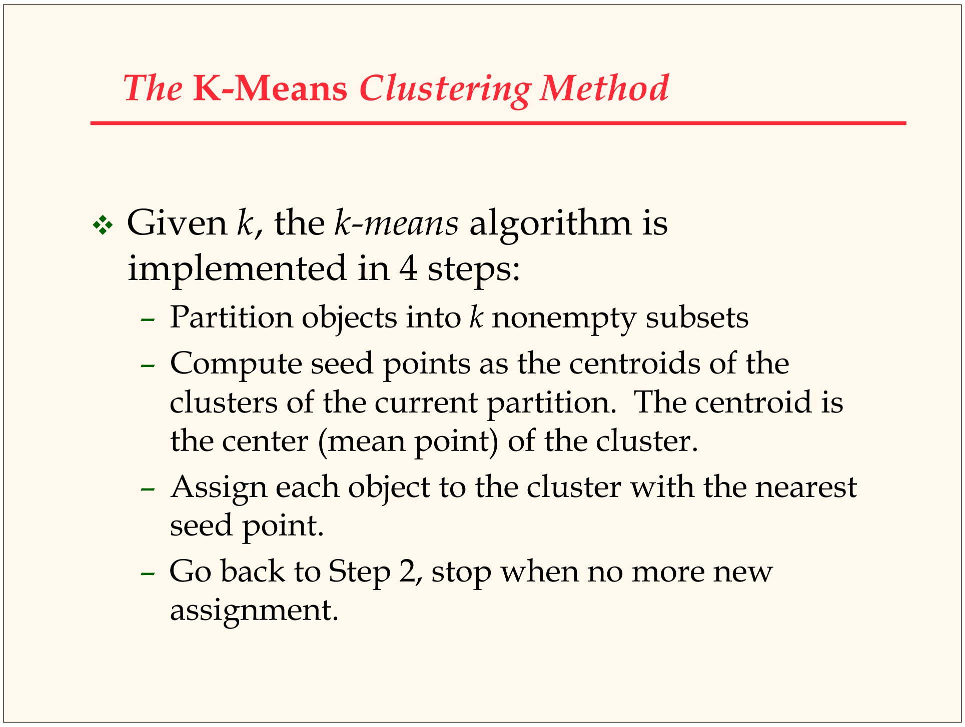
Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other



Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

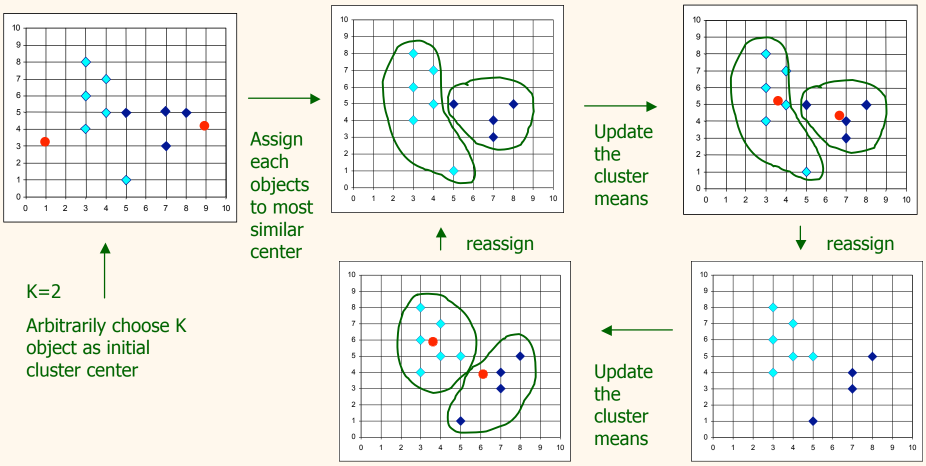


The K-Means *Clustering Method*

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignment.

The K-Means *Clustering Method*

- Example



K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

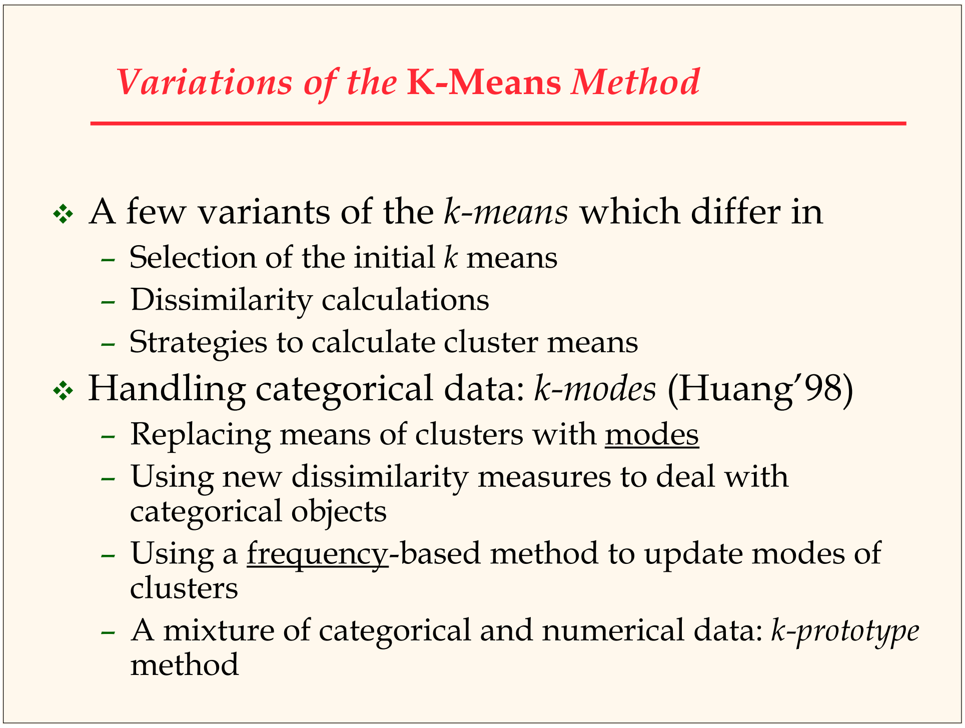
reassign

reassign

Update the cluster means

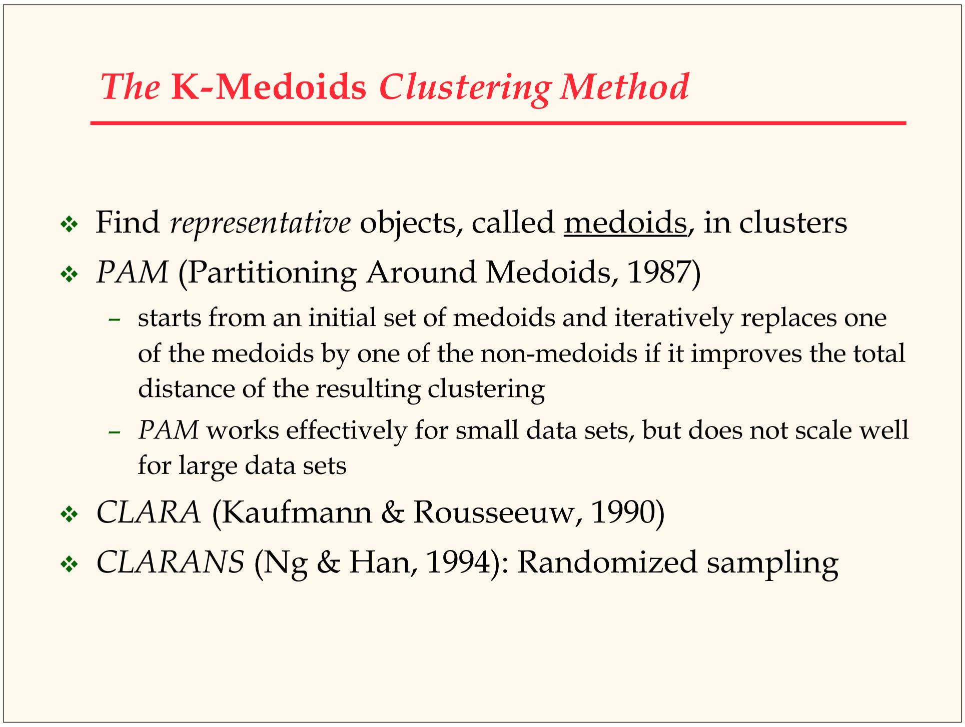
Comments on the K-Means *Method*

- <u>Strength</u>
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- <u>Weakness</u>
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*



Variations of the K-Means *Method*

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method



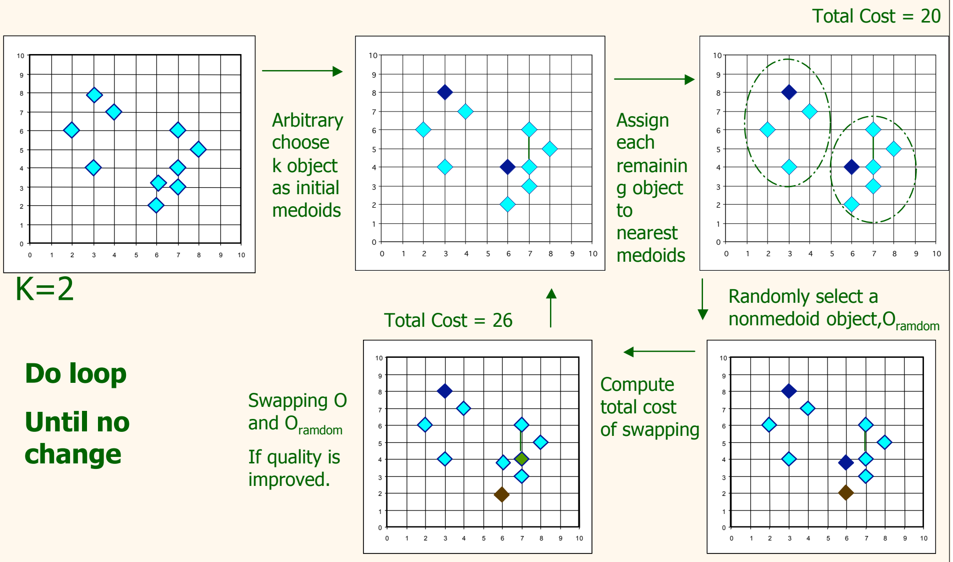
The K-Medoids *Clustering Method*

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling

k-Medoids algorithm

- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - repeat
    - Assign each remaining object to the cluster of the nearest medoid
    - Randomly select a nonmedoid object
    - Compute the total cost, S, of swapping o_j with o_{random}
    - If S < 0 then swap o_j with o_{random}
  - until there is no change

Typical k-medoids algorithm (PAM)



Total Cost = 20

Arbitrary choose k object as initial medoids

Assign each remainin g object to nearest medoids

K=2

Randomly select a nonmedoid object,O_{ramdom}

Total Cost = 26

Do loop

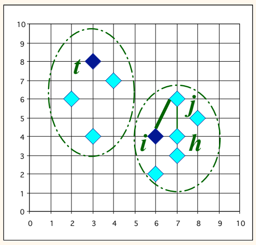
Until no change

Swapping O and O_{ramdom}

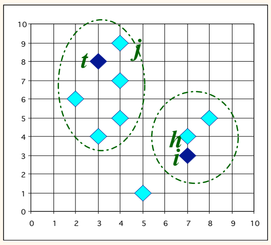
If quality is improved.

Compute total cost of swapping

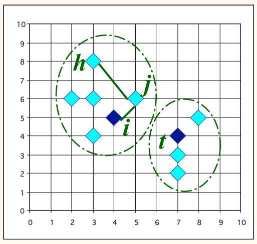
PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$



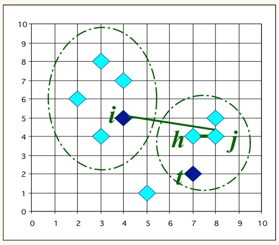
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



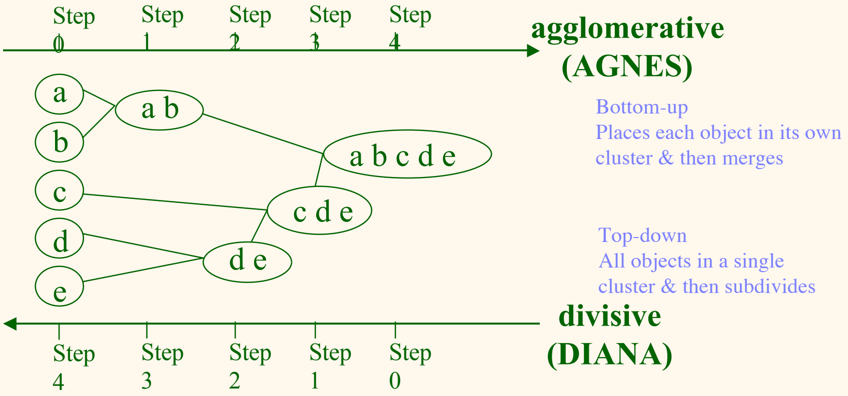
$C_{jih} = d(j, t) - d(j, i)$



$C_{jih} = d(j, h) - d(j, t)$

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



agglomerative (AGNES)

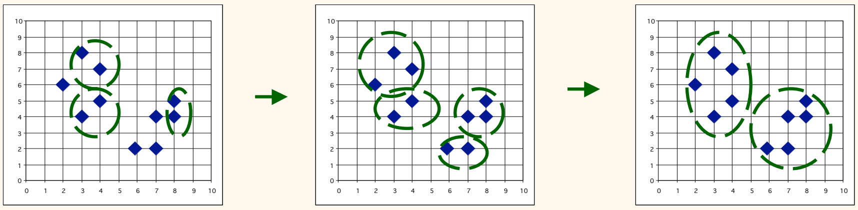
Bottom-up
Places each object in its own cluster & then merges

Top-down
All objects in a single cluster & then subdivides

divisive (DIANA)

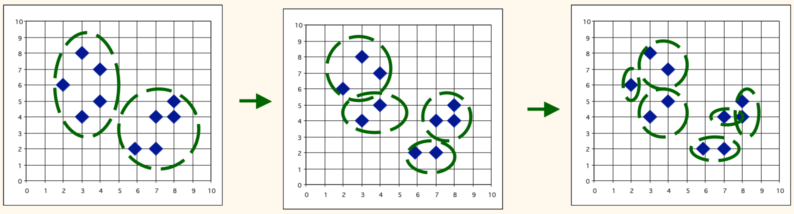
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



DIANA (Divisive Analysis)

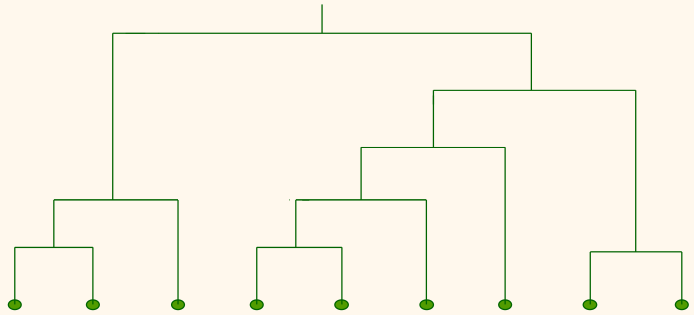
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Dendograms

Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

A <u>clustering</u> of the data objects is obtained by <u>cutting</u> the dendrogram at the desired level, then each <u>connected component</u> forms a cluster.



More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - <u>do not scale</u> well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - <u>BIRCH (1996)</u>: uses CF-tree and incrementally adjusts the quality of sub-clusters
  - <u>CURE (1998)</u>: selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - <u>CHAMELEON (1999)</u>: hierarchical clustering using dynamic modeling

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record.

Clustering Feature Vector

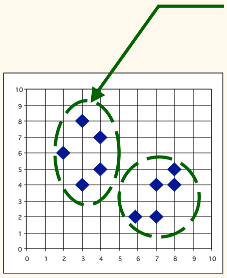
Clustering Feature: $CF = (N, \vec{LS}, SS)$

N: **Number of data points**

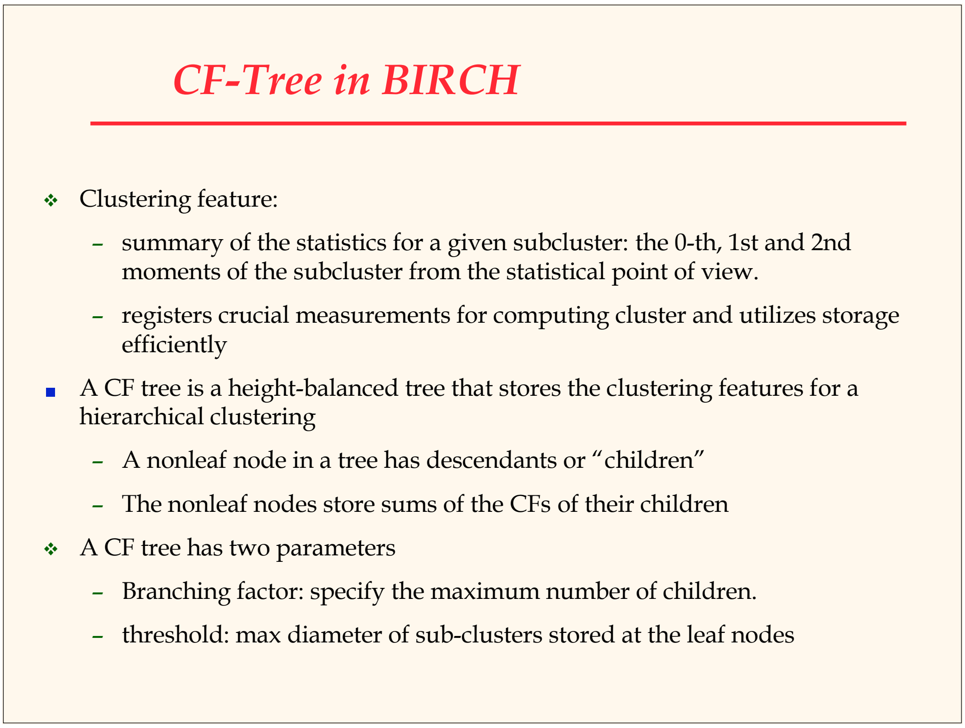
$LS: \sum^{N}_{i=1}=\vec{X}_i$

$SS: \sum^{N}_{i=1}=\vec{X}_i^2$

CF = (5, (16,30),(54,190))

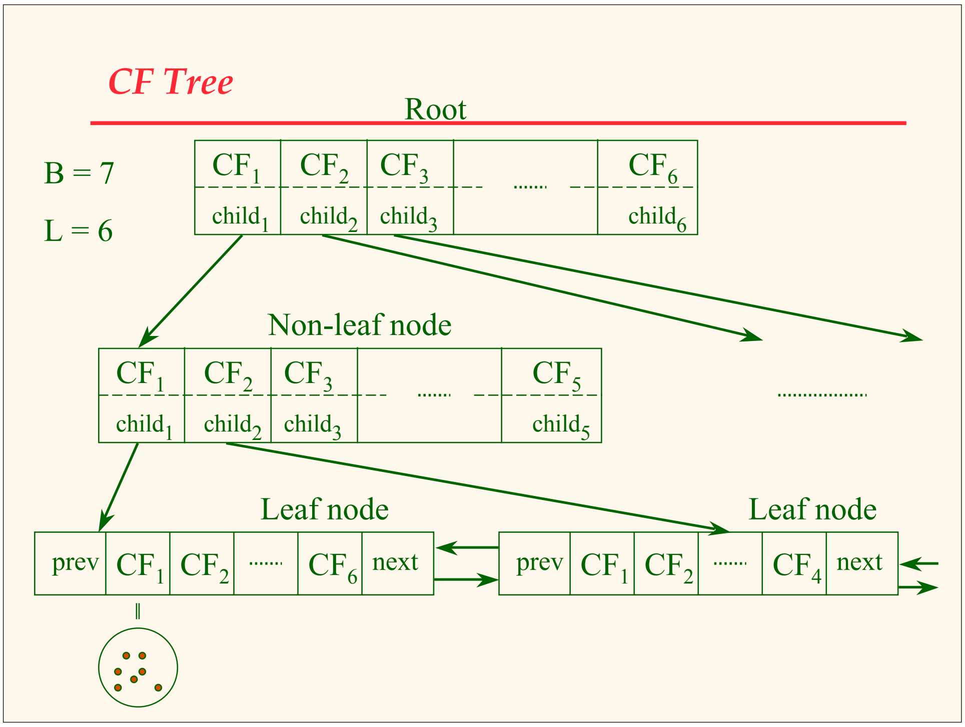


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)



CF-Tree in BIRCH

- Clustering feature:
  - summary of the statistics for a given subcluster: the 0-th, 1st and 2nd moments of the subcluster from the statistical point of view.
  - registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or "children"
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - threshold: max diameter of sub-clusters stored at the leaf nodes



CF Tree

Root

B = 7

L = 6

CF_1 CF_2 CF_3 .... CF_6

$child_1$ $child_2$ $child_3$ $child_6$

Non-leaf node

CF_1 CF_2 CF_3 .... CF_5

$child_1$ $child_2$ $child_3$ $child_5$

....

Leaf node

prev CF_1 CF_2 .... CF_6 next

Leaf node

prev CF_1 CF_2 .... CF_4 next

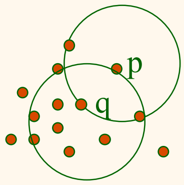
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering: Background (1)

- Two parameters*:*
  - ***Epsilon (Eps)***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Epsilon-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps, MinPts*** if
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) core point condition:

    $|N_{Eps}(q)| >= MinPts$

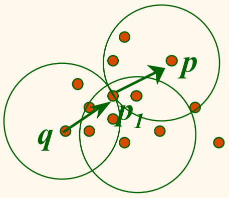


MinPts = 5

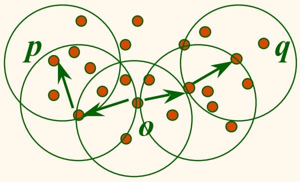
Eps = 1 cm

Density-Based Clustering: Background (2)

- Density-reachable:
  - A point p is density-reachable from a point q wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

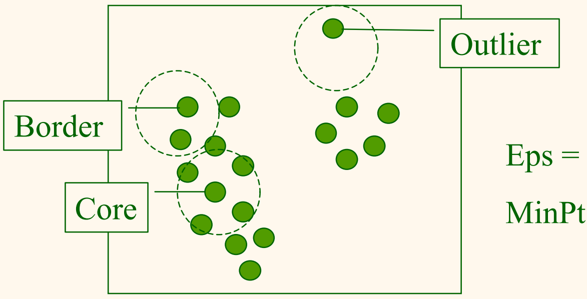


- Density-connected
  - A point p is density-connected to a point q wrt. *Eps*, *MinPts* if there is a point o such that both, p and q are density-reachable from o wrt. *Eps* and *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



Eps = 1cm

MinPts = 5

DBSCAN: The Algorithm

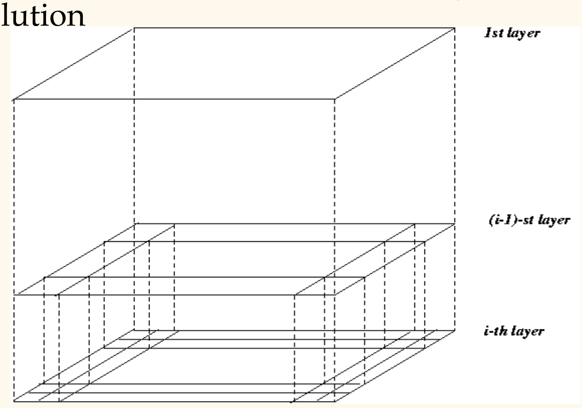
- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



STING: A Statistical Information Grid Approach (2)

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—normal, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

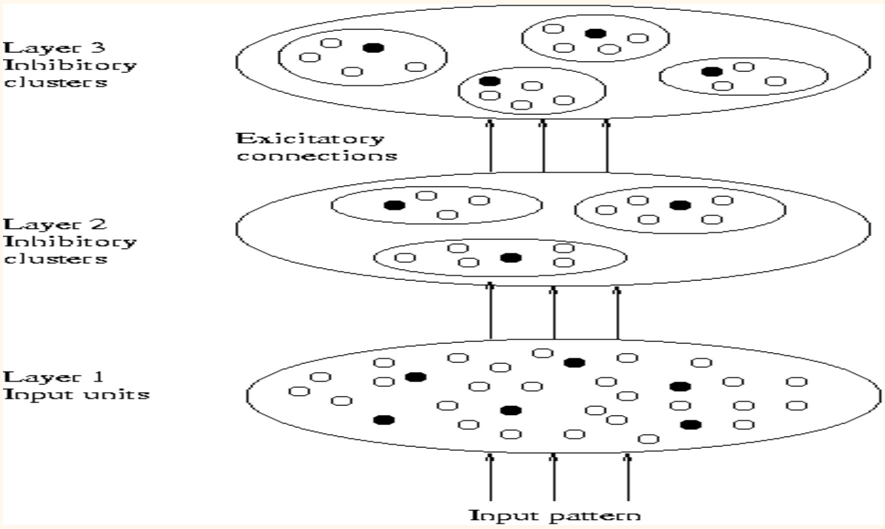
STING: A Statistical Information Grid Approach (3)

- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K),* where *K* is the number of grid cells at the lowest level
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

Model-Based Clustering Methods

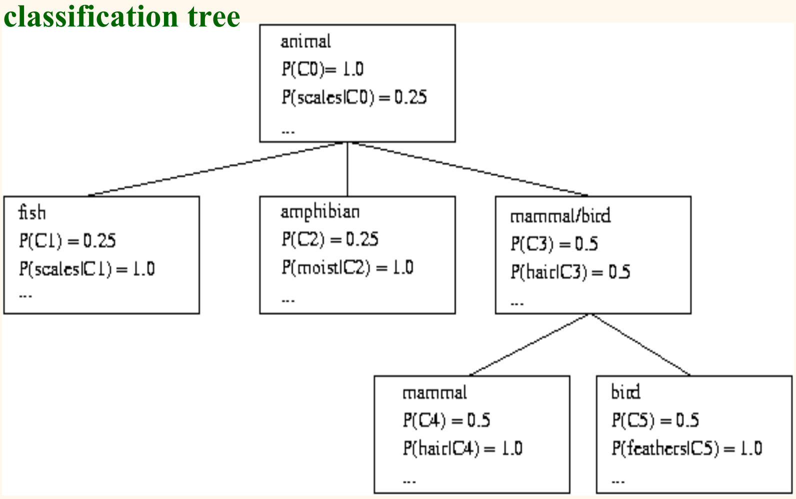
- Attempt to optimize the fit between the data and some mathematical model
- Statistical and AI approach
  - Conceptual clustering
    - A form of clustering in machine learning
    - Produces a classification scheme for a set of unlabeled objects
    - Finds characteristic description for each concept (class)
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a classification tree
    - Each node refers to a concept and contains a probabilistic description of that concept

Model-Based Clustering Methods



COBWEB Clustering Method

A classification tree



animal
P(C0)= 1.0
P(scales|C0) = 0.25
...

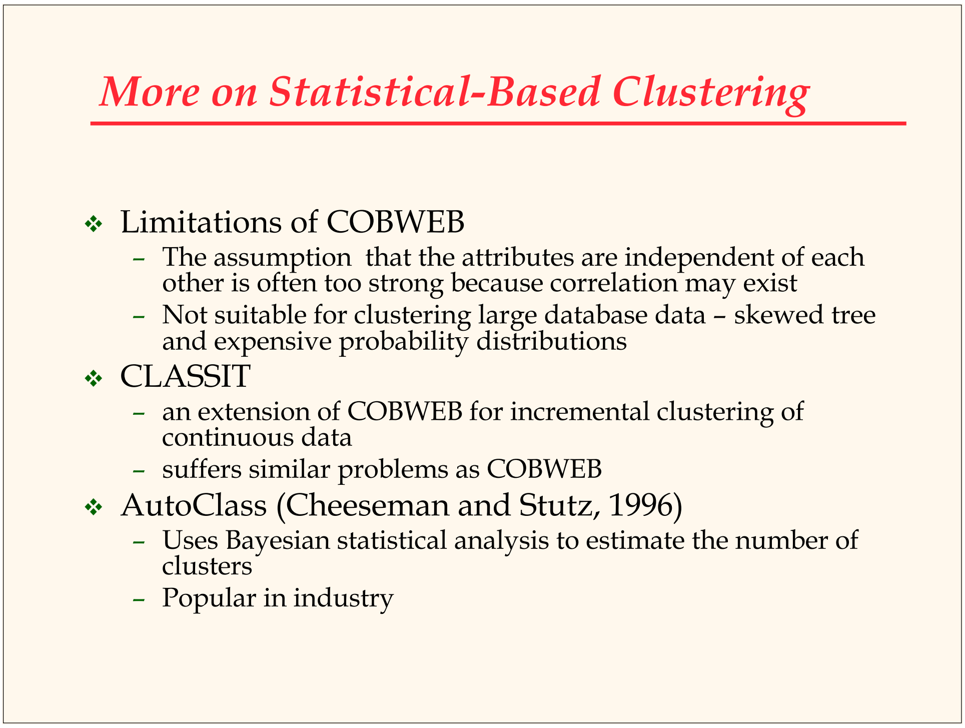
fish
P(C1) = 0.25
P(scales|C1) = 1.0
...

amphibian
P(C2) = 0.25
P(moist|C2) = 1.0
...

mammal/bird
P(C3) = 0.5
P(hair|C3) = 0.5
...

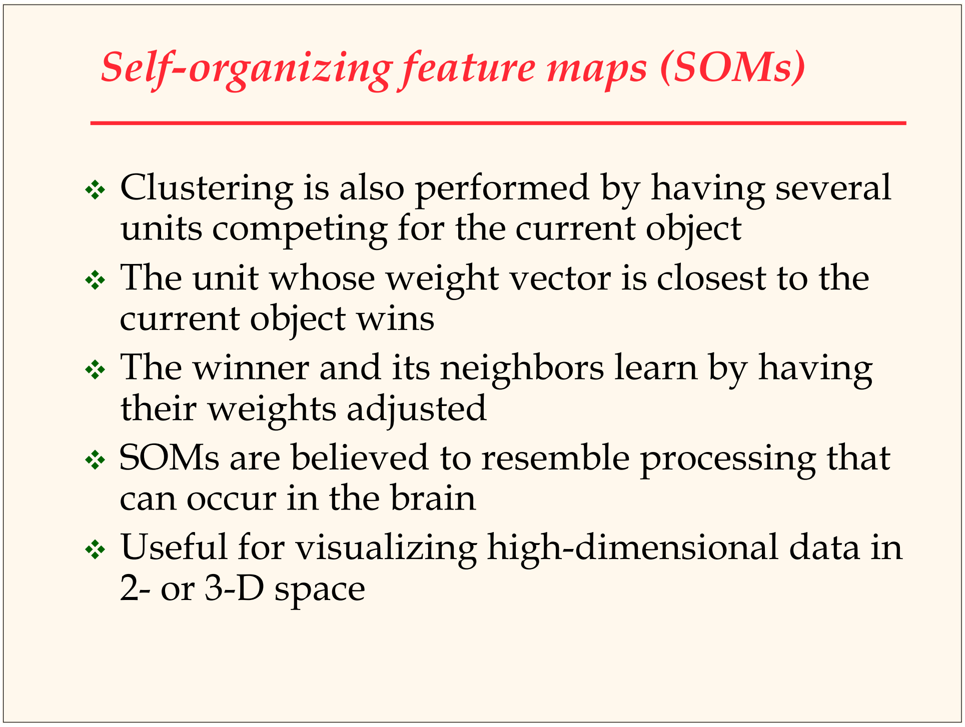
mammal
P(C4) = 0.5
P(hair|C4) = 1.0
...

bird
P(C5) = 0.5
P(feathers|C5) = 1.0
...



More on Statistical-Based Clustering

- Limitations of COBWEB
  - The assumption that the attributes are independent of each other is often too strong because correlation may exist
  - Not suitable for clustering large database data – skewed tree and expensive probability distributions
- CLASSIT
  - an extension of COBWEB for incremental clustering of continuous data
  - suffers similar problems as COBWEB
- AutoClass (Cheeseman and Stutz, 1996)
  - Uses Bayesian statistical analysis to estimate the number of clusters
  - Popular in industry

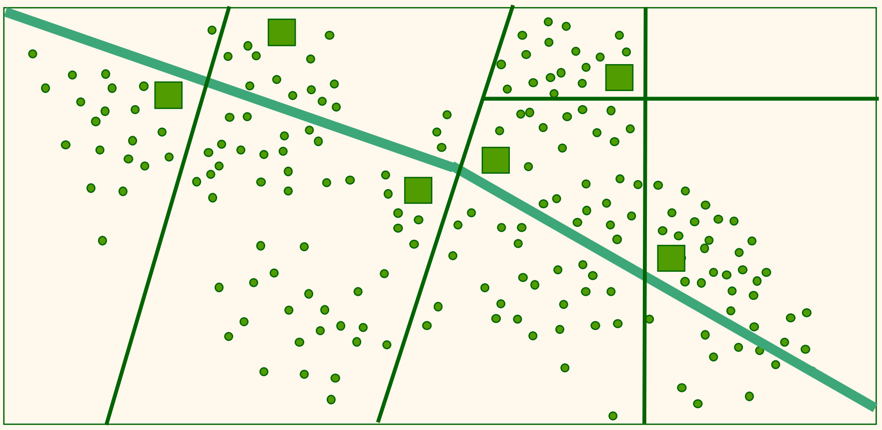


Self-organizing feature maps (SOMs)

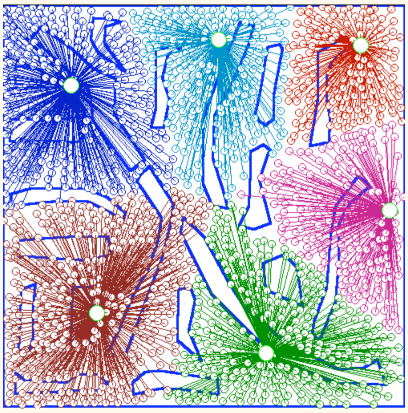
- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

Constraint-Based Clustering Analysis

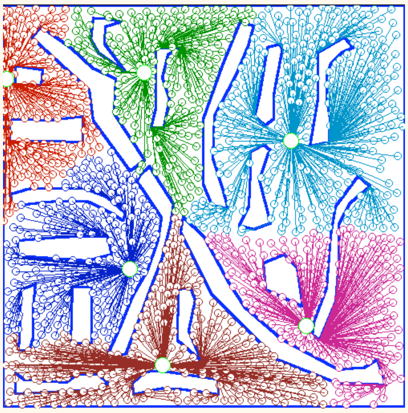
- Clustering analysis: less parameters but more user-desired constraints, e.g., an ATM allocation problem



Clustering With Obstacle Objects



Not Taking obstacles into account



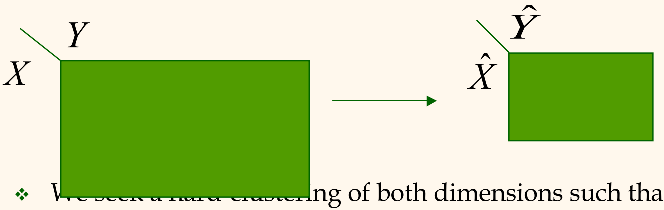
Taking obstacles into account

Co-clustering

- Given a multi-dimensional data matrix, co-clustering refers to **simultaneous** clustering along multiple dimensions
- In a two-dimensional case it is simultaneous clustering of rows and columns
- Most traditional clustering algorithms cluster along a single dimension
- Co-clustering is more robust to sparsity

Co-clustering and Information Theory

- View (scaled) co-occurrence matrix as a joint probability distribution between row & column random variables



- We seek a hard clustering of both dimensions such that loss in "Mutual Information"

$$I(X,Y) - I(\hat{X},\hat{Y})$$

is minimized given a fixed no. of row & col. clusters (similar framework as in Tishby, Pereira & Bialek(1999), Berkhin & Becher(2002))

Information Theory Concepts

- Entropy of a random variable X with probability distribution p(x):

$$H(p) = -\sum_{x} p(x) \log p(x)$$

- The Kullback-Leibler(KL) Divergence or "Relative Entropy" between two probability distributions p and q:

$$KL(p,q) = \sum_{x} p(x) \log(p(x)/q(x))$$

- Mutual Information between random variables X and Y:

$$I(X,Y) = \sum_{x} \sum_{y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Jensen-Shannon Divergence

- Jensen-Shannon(JS) divergence between two probability distributions:

$$\begin{aligned} JS_{\Pi}(p_1,p_2) &= \pi_1 KL(p_1, \pi_1 p_1 + \pi_2 p_2) + \pi_2 KL(p_2, \pi_1 p_1 + \pi_2 p_2) \\ &= H(\pi_1 p_1 + \pi_2 p_2) - \pi_1 H(p_1) - \pi_2 H(p_2) \end{aligned}$$

where $\pi_1, \pi_2 \geq 0, \pi_1 + \pi_2 = 1$

- Jensen-Shannon(JS) divergence between a finite number of probability distributions:

$$\begin{aligned} JS_{\Pi}(\{p_1, \ldots.., p_n\}) &= \sum_{i} \pi_i KL(p_i, \pi_1 p_1 + \ldots.. + \pi_n p_n) \\ &= H\left(\sum_{i} \pi_i p_i\right) - \sum_{i} \pi_i H(p_i) \end{aligned}$$

Information-Theoretic Clustering:

Preserving mutual information

- (Lemma) The loss in mutual information equals:

$$I(X,Y) - I(X,\hat{Y}) = \sum_{j=1}^{k} \pi(\hat{y}_j) JS_{\pi^*}(\{p(x \mid y_t) : y_t \in \hat{y}_j\})$$

- Interpretation: Quality of each cluster is measured by the Jensen-Shannon Divergence between the individual distributions in the cluster.
- Can rewrite the above as:

$$I(X,Y) - I(X,\hat{Y}) = \sum_{j=1}^{k} \sum_{y_t \in \hat{y}_j} \pi_t KL(p(x \mid y_t), p(x \mid \hat{y}_j))$$

- Goal: Find a clustering that minimizes the above loss

Information Theoretic Co-clustering

Preserving mutual information

- (Lemma) Loss in mutual information equals

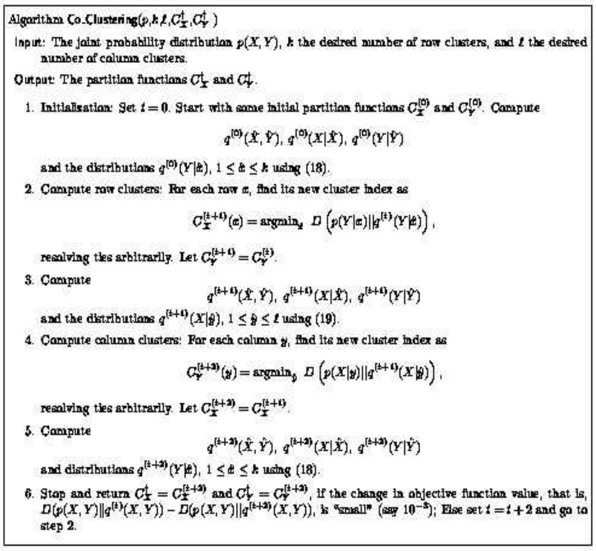
$$I(X,Y) \ - \ I(\overset{\surd}{X},\overset{\surd}{Y}) \ = \ KL(p(x,y) \ \| \ q(x,y))$$

where $$= H(\overset{\surd}{X},\overset{\surd}{Y}) + H(X \mid \overset{\surd}{X}) + H(Y \mid \overset{\surd}{Y}) - H(X,Y)$$

$$q(x,y) = p(\hat{x},\hat{y})p(x \mid \hat{x})p(y \mid \hat{y}), \quad \text{where} \quad x \in \hat{x}, y \in \hat{y}$$

- Can be shown that *q(x,y)* is a "maximum entropy" *approximation* to *p(x,y)*.
- *q(x,y)* preserves marginals : *q(x)=p(x)* & *q(y)=p(y)*

Co-Clustering Algorithm

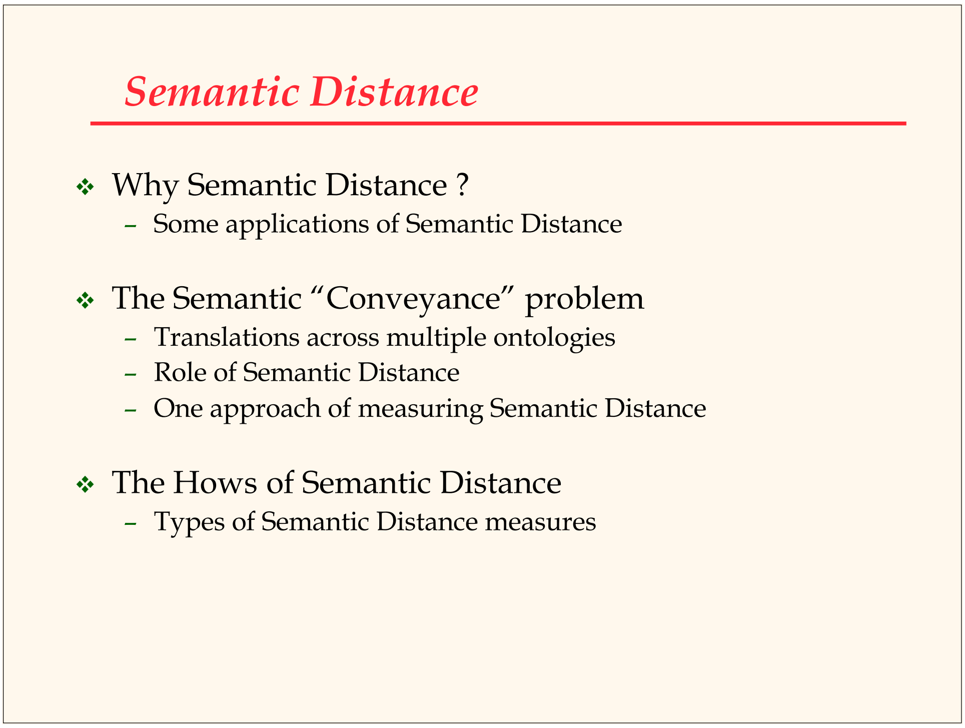


```
Algorithm Co_Clustering($p,k,\ell,C_X^\dagger,C_Y^\dagger$)
Input: The joint probability distribution $p(X,Y)$, $k$ the desired number of row clusters, and $\ell$ the desired
    number of column clusters.
Output: The partition functions $C_X^\dagger$ and $C_Y^\dagger$.
  1. Initialization: Set $t=0$. Start with some initial partition functions $C_X^{(0)}$ and $C_Y^{(0)}$. Compute
         $q^{(0)}(\hat{X},\hat{Y}),\ q^{(0)}(X|\hat{X}),\ q^{(0)}(Y|\hat{Y})$
     and the distributions $q^{(0)}(Y|\hat{x})$, $1 \le \hat{x} \le k$ using (18).
  2. Compute row clusters: For each row $x$, find its new cluster index as
         $C_X^{(t+1)}(x) = \mathrm{argmin}_{\hat{x}}\ D\left(p(Y|x)||q^{(t)}(Y|\hat{x})\right)$,
     resolving ties arbitrarily. Let $C_Y^{(t+1)} = C_Y^{(t)}$.
  3. Compute
         $q^{(t+1)}(\hat{X},\hat{Y}),\ q^{(t+1)}(X|\hat{X}),\ q^{(t+1)}(Y|\hat{Y})$
     and the distributions $q^{(t+1)}(X|\hat{y})$, $1 \le \hat{y} \le \ell$ using (19).
  4. Compute column clusters: For each column $y$, find its new cluster index as
         $C_Y^{(t+2)}(y) = \mathrm{argmin}_{\hat{y}}\ D\left(p(X|y)||q^{(t+1)}(X|\hat{y})\right)$,
     resolving ties arbitrarily. Let $C_X^{(t+2)} = C_X^{(t+1)}$.
  5. Compute
         $q^{(t+2)}(\hat{X},\hat{Y}),\ q^{(t+2)}(X|\hat{X}),\ q^{(t+2)}(Y|\hat{Y})$
     and distributions $q^{(t+2)}(Y|\hat{x})$, $1 \le \hat{x} \le k$ using (18).
  6. Stop and return $C_X^\dagger = C_X^{(t+2)}$ and $C_Y^\dagger = C_Y^{(t+2)}$, if the change in objective function value, that is,
     $D(p(X,Y)||q^{(t)}(X,Y)) - D(p(X,Y)||q^{(t+2)}(X,Y))$, is "small" (say $10^{-3}$); Else set $t=t+2$ and go to
     step 2.
```

Figure 1: Information theoretic co-clustering algorithm that simultaneously clusters both the rows and columns

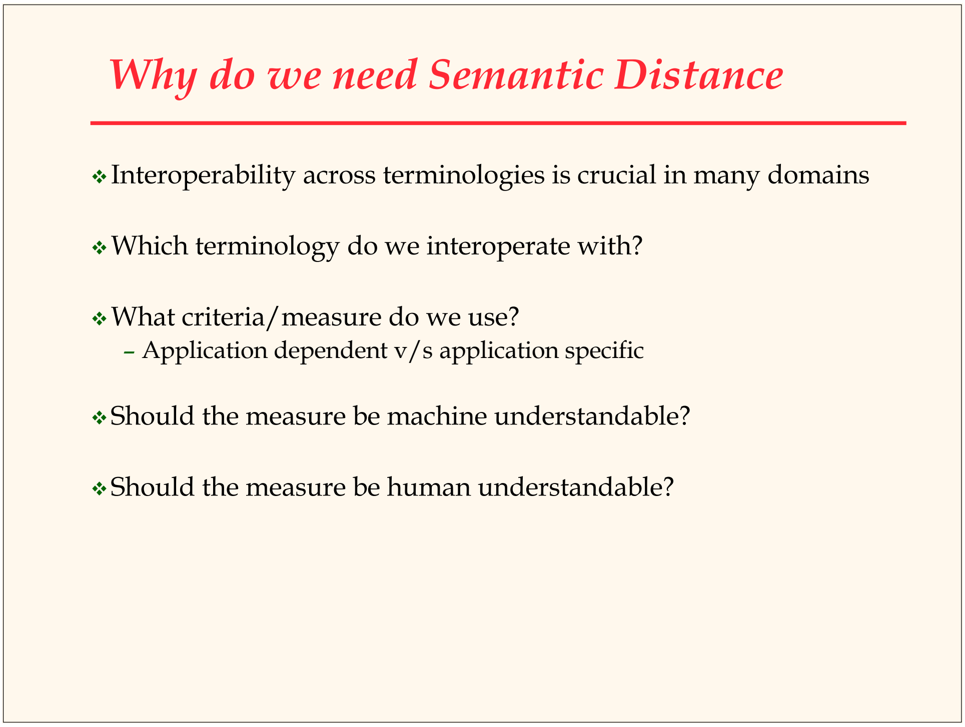
Properties of Co-clustering Algorithm

- **Theorem**: The co-clustering algorithm monotonically decreases loss in mutual information (objective function value)
- Marginals *p(x)* and *p(y)* are preserved at every step (*q(x)=p(x)* and *q(y)=p(y)*)
- Can be generalized to higher dimensions



Semantic Distance

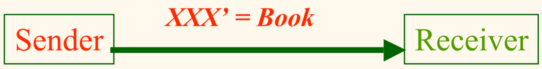
- Why Semantic Distance ?
  - Some applications of Semantic Distance
- The Semantic "Conveyance" problem
  - Translations across multiple ontologies
  - Role of Semantic Distance
  - One approach of measuring Semantic Distance
- The Hows of Semantic Distance
  - Types of Semantic Distance measures



Why do we need Semantic Distance

- Interoperability across terminologies is crucial in many domains
- Which terminology do we interoperate with?
- What criteria/measure do we use?
  - Application dependent v/s application specific
- Should the measure be machine understandable?
- Should the measure be human understandable?

Semantic "Conveyance"

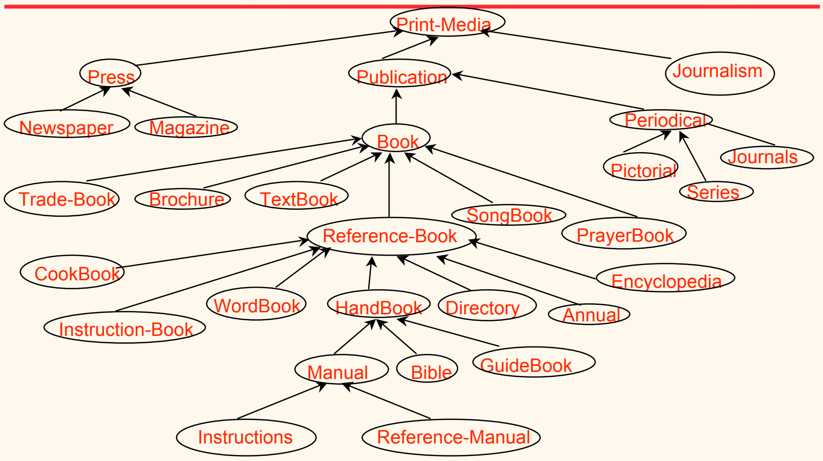

XXX' = Book

Sender

Receiver

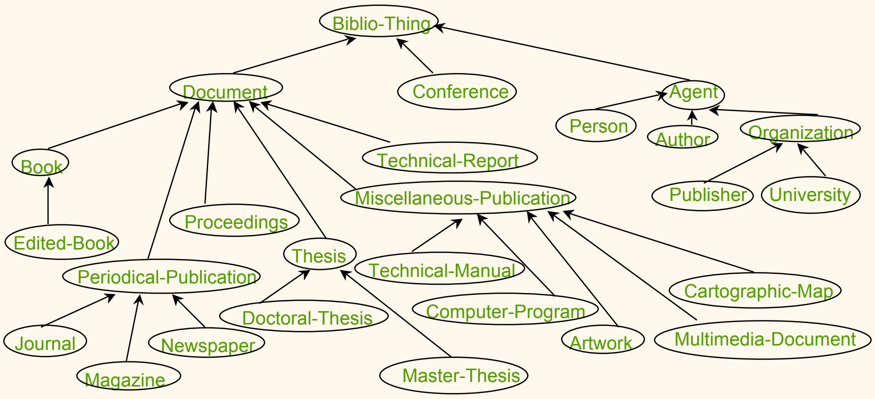
- The Sender has his own ontology (The Red Ontology)
- The Receiver has his own (The Blue Ontology)
- For communication to take place,
  - The receiver should translate the message (content) from the Red Ontology to the Blue Ontology
- Questions:
  - Is it always possible?
  - How many candidate possibilities are there?
  - How do we choose from them, Semantic Distance?

Terminology 1: The Red Terminology



http://www.cogsci.princeton.edu/~wn/w3wn.html

Terminology 2: The Blue Terminology



http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/bibliographic-data/

Inter-terminological relationships: Typically represented in the UMLS Metathesaurus

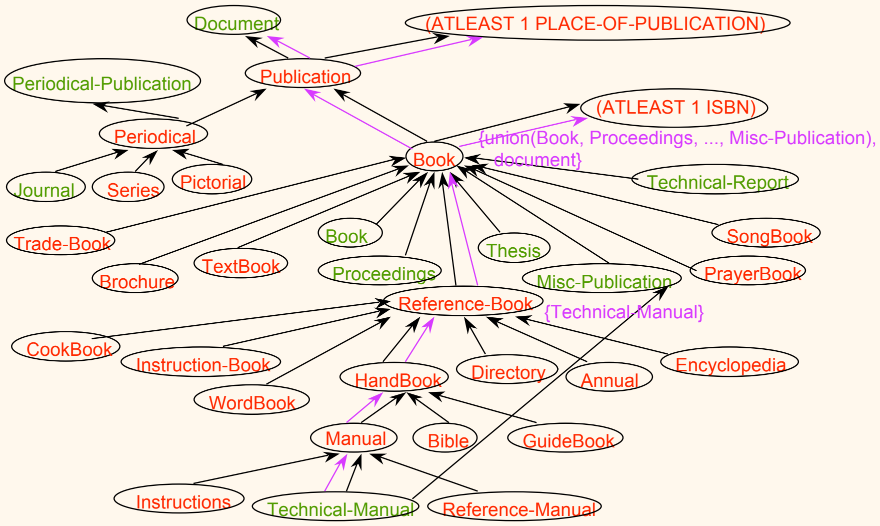
❖Synonyms
- semantics preserving

❖Hyponyms/Hypernyms
- semantics altering
- typically results in loss of information

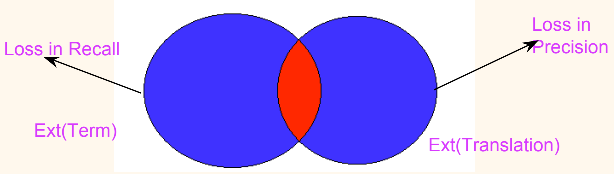
❖List of Hyponyms

– technical-manual	***hyponym***	manual
– book	***hyponym***	book
– proceedings	***hyponym***	book
– thesis	***hyponym***	book
– misc-publication	***hyponym***	book
– technical-reports	***hyponym***	book
– press publication	***hyponym***	periodical-
– periodical publication	***hyponym***	periodical-

Translations across multiple terminologies



Proposal for Semantic Distance: Extensional Measure



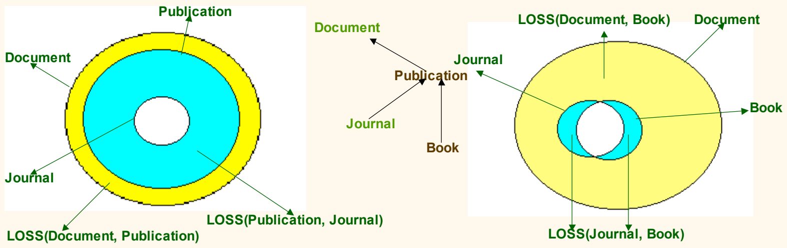
$$\text{Precision} = \frac{|\,\text{Ext(Term)} \cap \text{Ext(Translation)}|}{|\text{Ext(Translation)}|} \qquad \text{Recall} = \frac{|\,\text{Ext(Term)} \cap \text{Ext(Translation)}|}{|\text{Ext(Term)}|}$$

$$\text{Percentage Loss} = \frac{|\,\text{Ext(Term)}\ \Delta\ \text{Ext(Translation)}|}{|\text{Ext(Term)}| + |\text{Ext(Translation)}|}$$

$$= 1 - \frac{1}{1/2(1/\text{Precision}) + 1/2(1/\text{Recall})}$$

$$\Rightarrow 1 - \frac{1}{(\text{alpha})(1/\text{Precision}) + (1-\text{alpha})(1/\text{Recall})} \quad 0 < \text{alpha} < 1$$

Choosing an optimal translation



- Local Decision Making
  - LOSS(Publication, Journal) > LOSS(Document, Publication)
  - Document is chosen as the translation
  - But LOSS(Book, Document) > LOSS(Book, Journal) !!
- Global Decision Making
  - Both translations {Document, Journal} are passed on to the next level
  - Journal is chosen as the appropriate translation

Proposal for Semantic Distance: Intensional Measure

- Difference in Translation:
  - **Book ⇒ union(Book, Thesis, Proceedings, Technical-Manual, Misc-Publication)**
- Terminological Difference
  - **Book ⊆ (AND Publication (ATLEAST 1 ISBN))**
  - **Publication ⊆ (AND document (ATLEAST 1 PLACE-OF-PUBLICATION))**
  - **Book ⊆ (AND document (ATLEAST 1 ISBN) (ATLEAST 1 PLACE-OF-PUBLICATION))**
- Loss of Information:
  - **(-) union(Trade-Book, Brochure, SongBook, PrayerBook, TextBook)**
    - **information related to trade books, brochures, song books, prayer books and text books is lost**
  - **(+) (AND (ATLEAST 1 ISBN) (ATLEAST 1 PLACE-OF-PUBLICATION))**
    - **spurious documents that don’t have an ISBN number and a place of publication are gained**

Measures for Semantic Distance: Pros and Cons

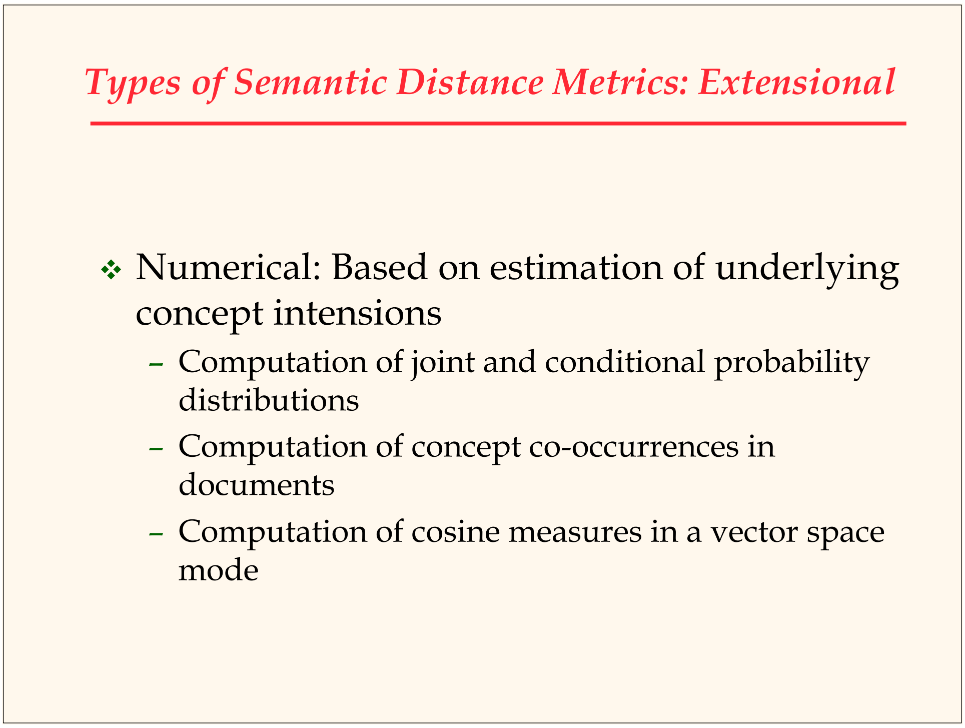
- Intensional Measure:
  - May not make sense as it mixes two vocabularies,
    - **e.g., does Book - Book make any sense ?**
  - The problem becomes worse if the two terminologies are in different languages
  - Makes it hard for the system to differentiate between the various alternatives

- Extensional Measure:
  - Based on Standard Information Retrieval Measures (F-measure)
  - Can be tailored to reflect change in semantic distance for different applications
  - However:
    - Probability distributions of various terms need to be estimated
    - An information loss interval doesn't make much sense to the user.

Types of Semantic Distance Metrics: Intensional

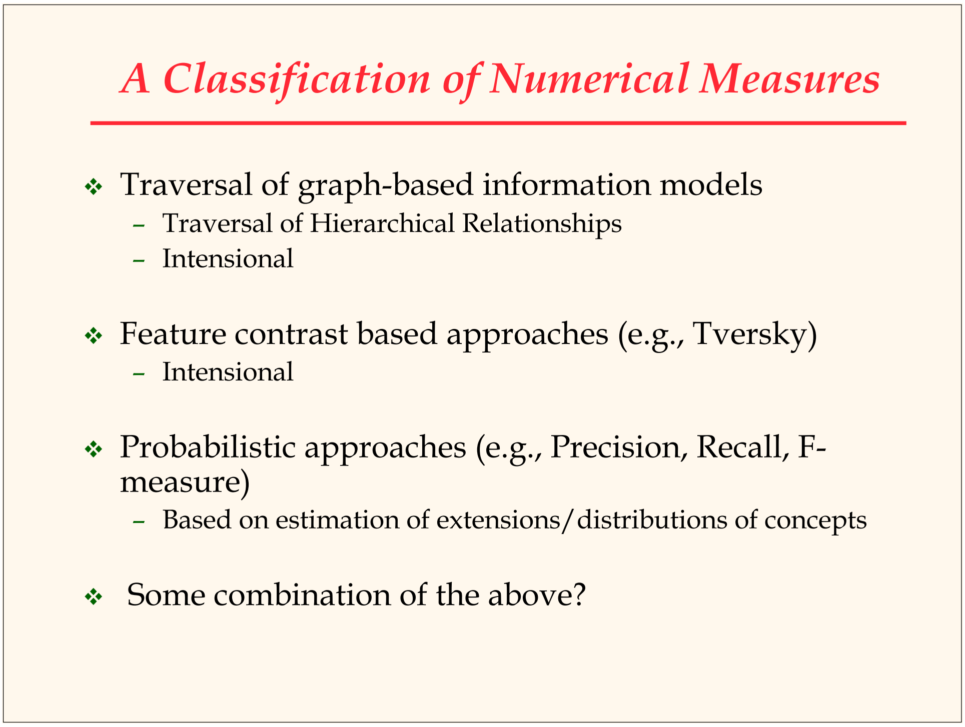
- Numerical
  - Based on features (e.g., Tversky's measure)
  - Based on traversal of specific conceptual relationships (is-a, part-of) and arbitrary domain specific relationships

- Non-numerical
  - Based on semantic concept differences, e.g. a book without a publication date
  - Important for human understandability



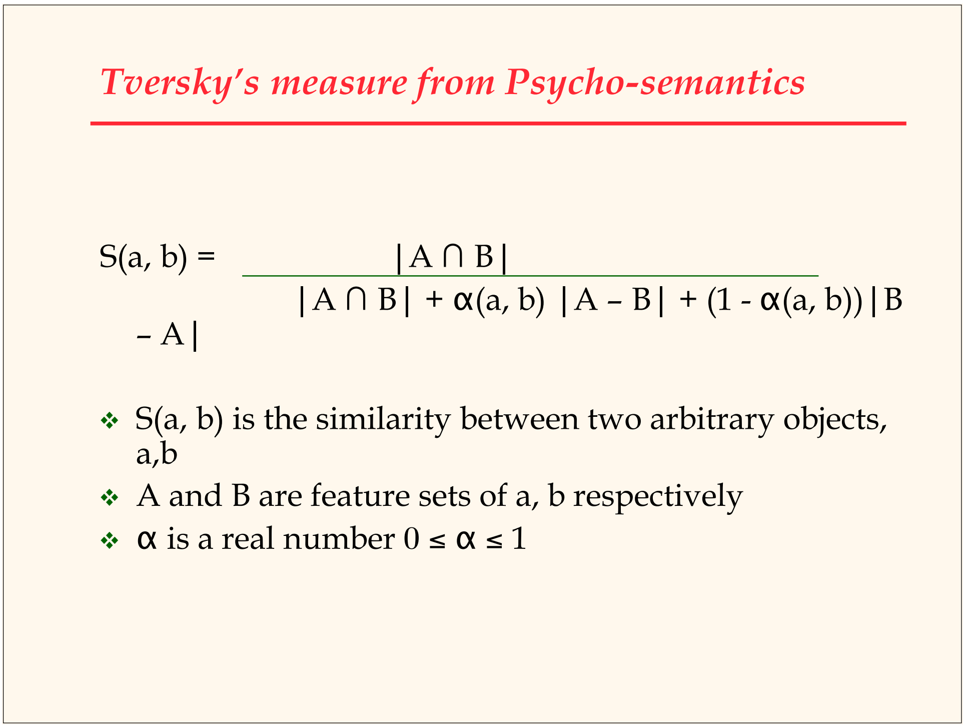
Types of Semantic Distance Metrics: Extensional

- Numerical: Based on estimation of underlying concept intensions
  - Computation of joint and conditional probability distributions
  - Computation of concept co-occurrences in documents
  - Computation of cosine measures in a vector space mode



A Classification of Numerical Measures

- Traversal of graph-based information models
  - Traversal of Hierarchical Relationships
  - Intensional
- Feature contrast based approaches (e.g., Tversky)
  - Intensional
- Probabilistic approaches (e.g., Precision, Recall, F-measure)
  - Based on estimation of extensions/distributions of concepts
- Some combination of the above?



Tversky's measure from Psycho-semantics

$$S(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a, b)\ |A - B| + (1 - \alpha(a, b))|B - A|}$$

- S(a, b) is the similarity between two arbitrary objects, a,b
- A and B are feature sets of a, b respectively
- α is a real number $0 \leq \alpha \leq 1$