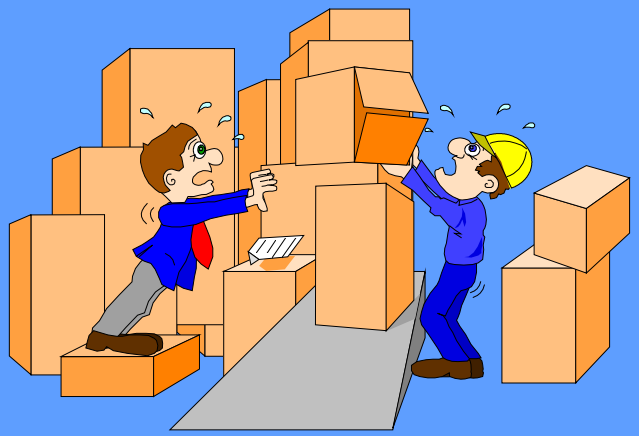
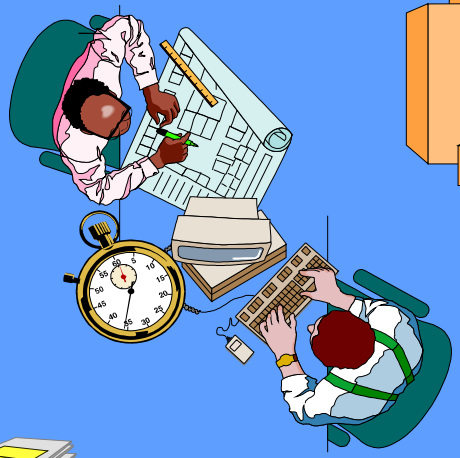
CLUSTERING LARGE DATASETS



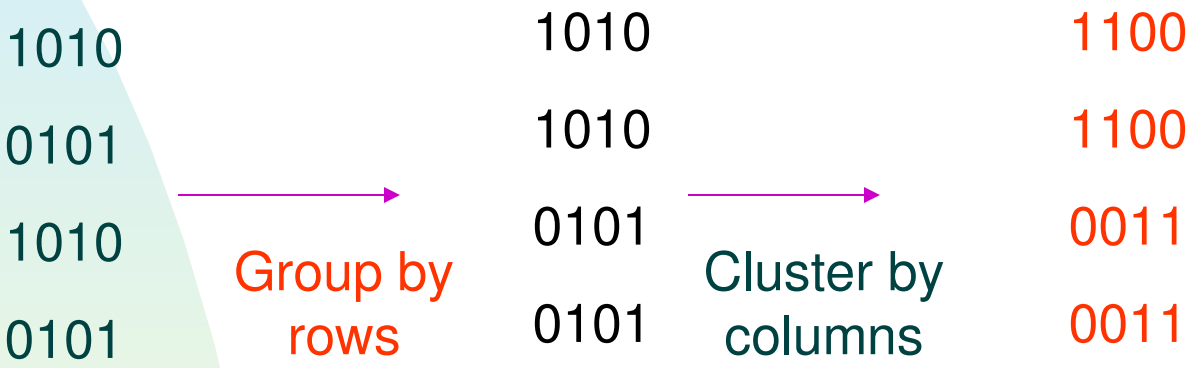


M. Narasimha Murty
Professor and Chairman, Dept. of CSA
Indian Institute of Science
Bangalore-560 012
mnm@csa.iisc.ernet.in

March 3, 2008

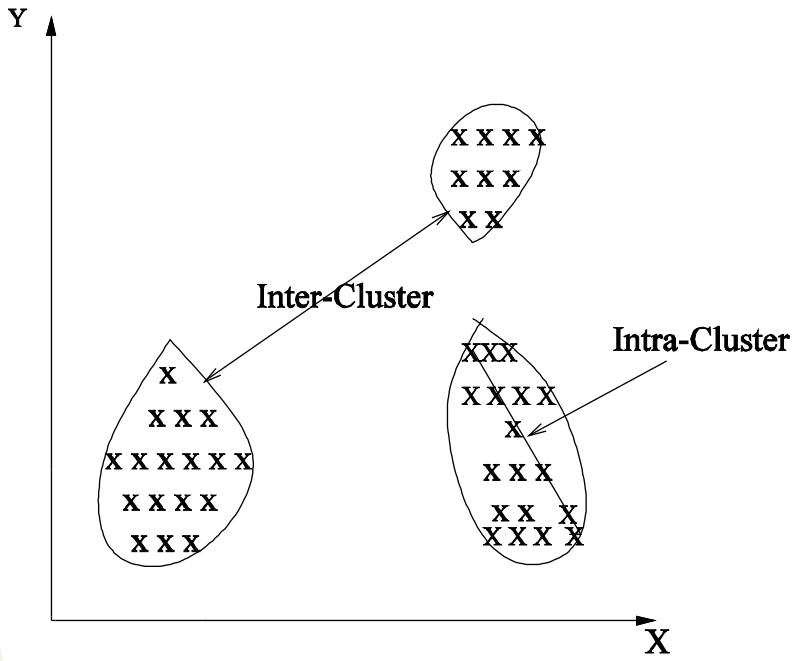
What is Clustering?

Clustering is data reorganization



- Similarity between patterns is the basis for grouping
- It reveals structure hidden in the data

What is Clustering?

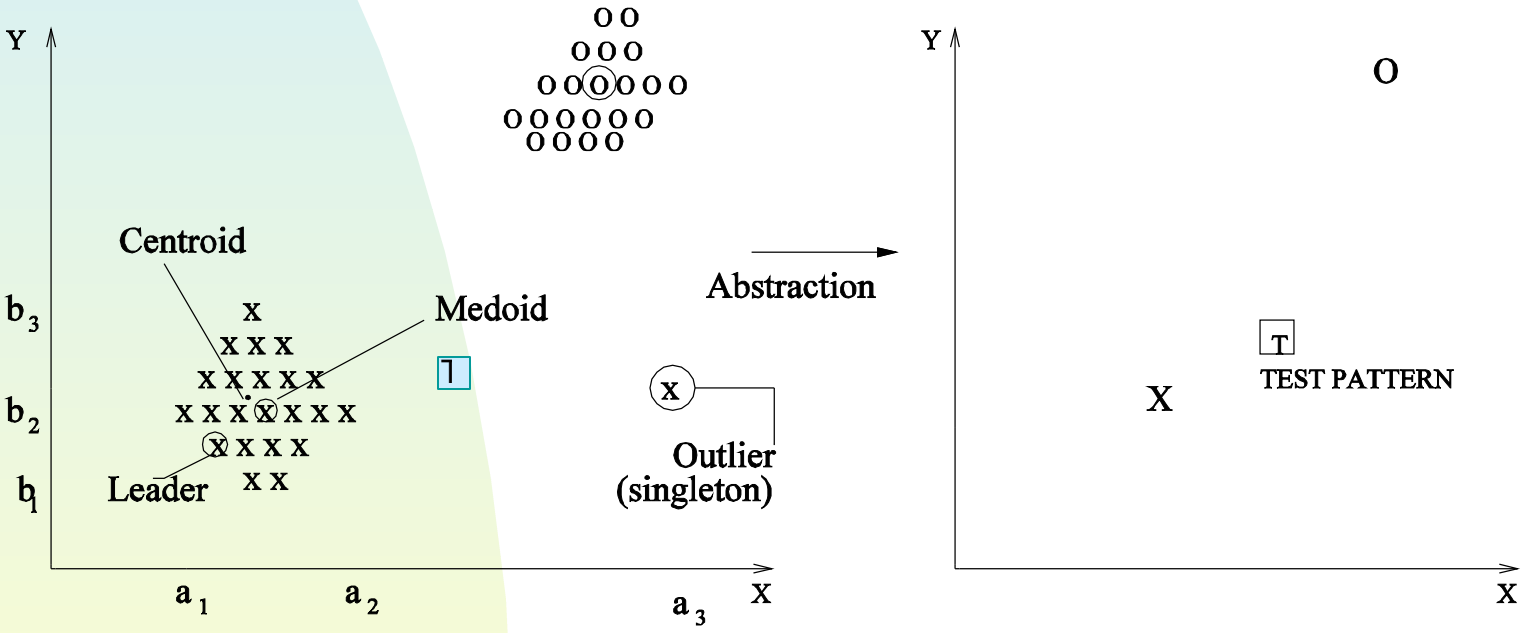


Find clusters so that the objects of each cluster are similar to each other whereas objects of different clusters are dissimilar.

Number of Partitions

- There are 2^{n-1} -1 2-partitions of n patterns
- For 3 patterns A, B, and C, we have
  - {A}, {B,C}; {B}, {A,C}; {C}, {A,B}
- There are approximately 11,259,666,000 partitions of 19 patterns clustered into 4 groups (Stirling numbers- Concrete Maths., Graham, Knuth, Patashnik)
- Exhaustive enumeration of all the partitions is not good for large n.
- So, we use some scheme to examine only a small collection of partitions and reach a good solution.

Clustering is Data Abstraction



Why Cluster?

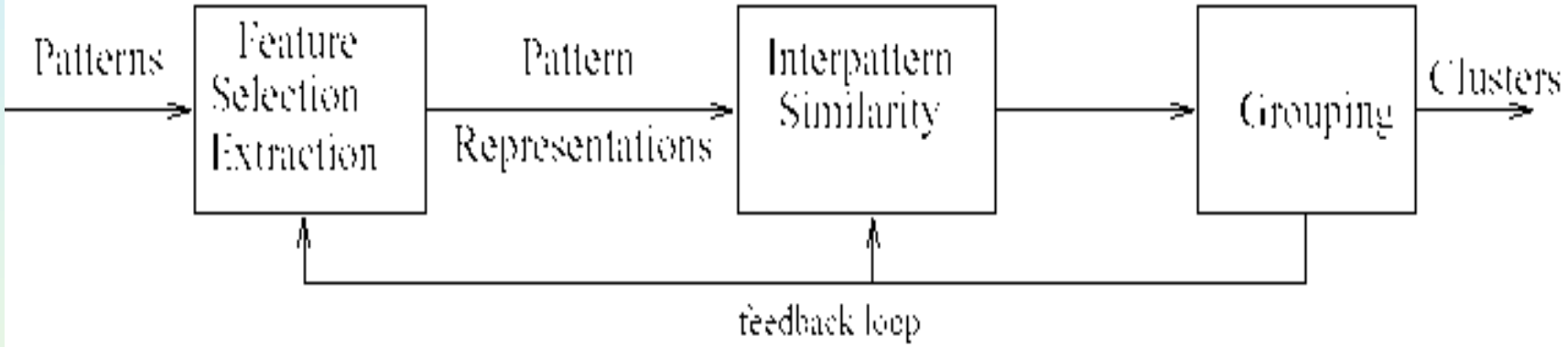
- Represent a cluster of n patterns by m points.
  - Typically, m < n leading to *data compression; can use centroids. Popularly, m = 1*
  - This would help in prototype selection for *efficient classification.*
  - *Naturally,* clustering features leads to *dimensionality reduction.*
- Clustering helps in detecting outliers; examine small size clusters.
- Even when there are no clusters, some kind of partitioning would be helpful; for example *quick sort algorithm* partitions data sets in each pass to reduce the computational complexity.

Applications

There are several real-life applications:

- Biometrics
- Bio-informatics
- Bio-medical data analysis
- Data Mining
- Document Recognition
- Indexing Documents
- Man-Machine Interfaces
- Remote Sensed Data Analysis
- Others

Stages in clustering



Pattern is a physical object (e.g. A chair) or an abstract notion (e. g. a style of writing).

For Machine recognition, we need to store the pattern representations on the machine; choice of the scheme of representation is important.

Representation of Objects

A pattern is represented as a

- Vector in a multi-dimensional space:
  For example, (30,1) represents an object with 30 units of weight and 1 unit of height
  - Statistical clustering and Neural Nets
  - Fuzzy and Rough clustering

- String of characters/primitives:
  - Description in a logic; for example
  (color = red V white)^ (make = leather)^ (shape = sphere)
  - Conceptual, Knowledge-based, itemset based and symbolic clustering

Issues in Representation

Theorem of the Ugly Duckling

It states that the number of predicates shared by any two patterns is constant when all possible predicates are used to represent patterns. So, if we judge similarity based on the number of predicates shared, then any two patterns are equally similar.

	f1	f2
P1	0	0
P2	0	1
P3	1	1
P4	1	0

Ugly Duckling Theorem

There are four object types; if the first three are considered, then there at most 8 predicates.

	f1	f2	f1^~f2	~f2	~f1^f2 (g2)	~f1 (g1)	f1v~f2	~f1vf2
P1	0	0	0	1	0	1	1	1
P2	0	1	0	0	1	1	0	1
P3	1	1	0	0	0	0	1	1

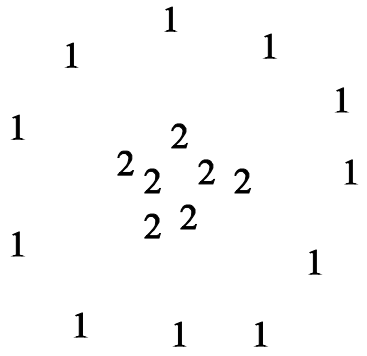
Which are primitive? f1 and f2 or g1 = ~f1 and g2 = xor(f1,f2)?

Observe that f1 = ~g1 and f2 = xor(~g1,g2)!

This means, we need extra-logical evidence to give more importance to some features over others; such knowledge is used in representation, distance/similarity computation or in the grouping phase of clustering.

Axiomatic Clustering

- Kleinberg's Impossibility Theorem (f:S→)
  - Scale-Invariance: f(d) = f(ad) for a>0; d-distance
  - Richness: Range(f) is the set of all partitions of S
  - Consistency: d and d' are such that f(d) = π and for all i, j S from the same cluster of π, d'(i,j) ≤ d(i,j) and for all i, j S from different clusters of π, d'(i,j) ≥ d(i,j), then f(d') = π.

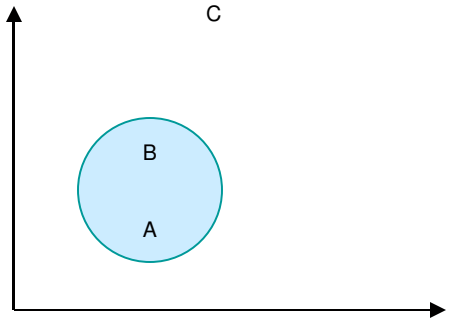


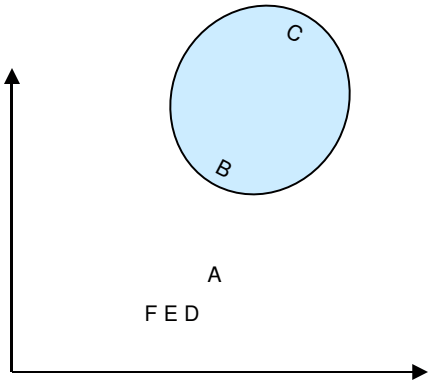
- There is no f that satisfies all the 3 props for n > 1

Similarity Measures

- Typically, *dissimilarity* between *two* patterns is computed using a distance measure.
- Distance between patterns in the same cluster is *less than that* between patterns in different clusters.
- The most popular is the *Euclidean Dist.*
  - *Invariant to translations and rotations*
  - *May vary with scaling*

Mutual Neighborhood Distance





MND(A,B) = 2

MND(B,C) = 3

MND(A,C)= 4

MND(A,B) = 5

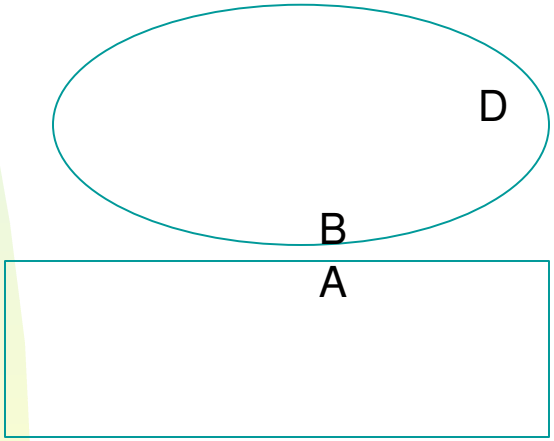
MND(B,C) = 3

	1	**2**
A	B	C
B	A	C
C	B	A

	1	2	3	4	5
A	D	E	F	B	C
B	A	C	D	E	F
C	B	A	D	E	F

Conceptual Cohesiveness

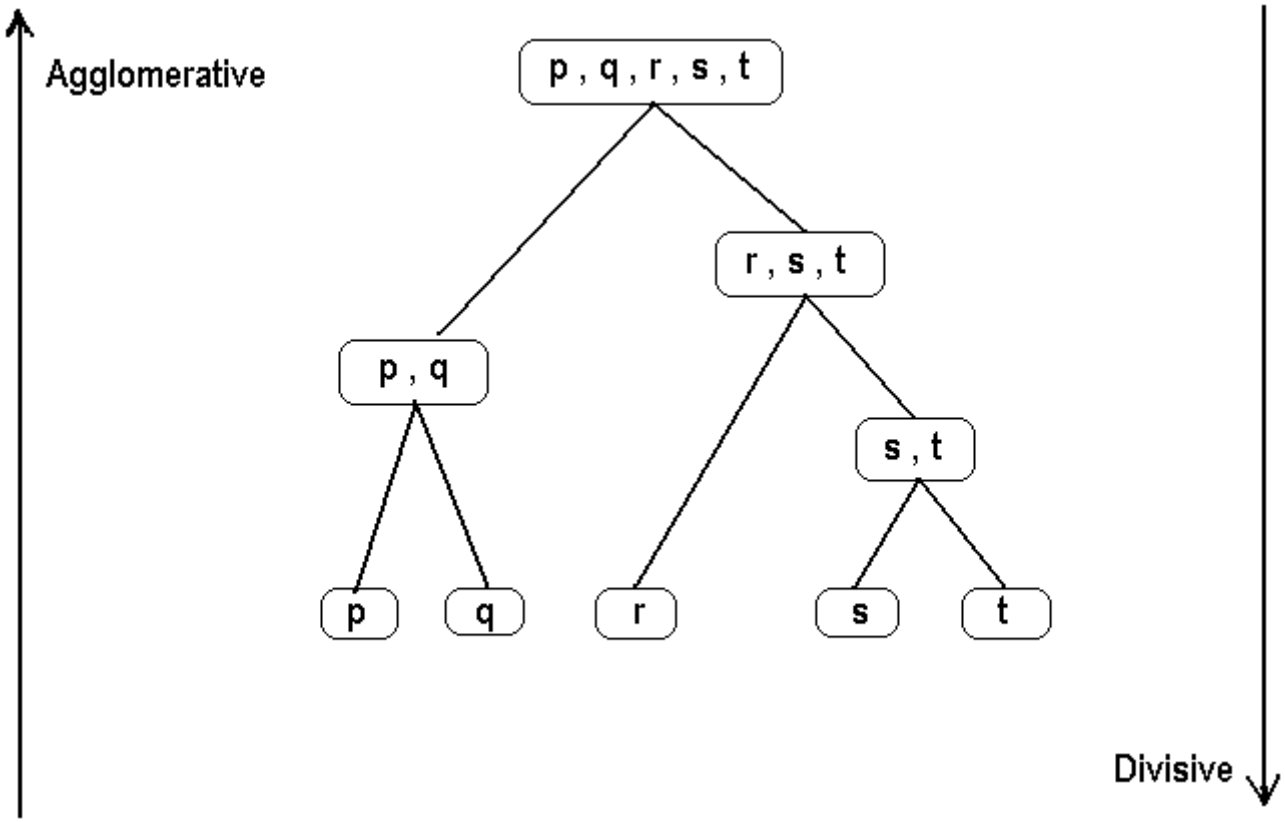
- *Conceptual Cohesiveness (A,B) = g(A, B, **E,C**), where **C is a set of pre-defined*** concepts and ***E*** is the context (surrounding points); knowledge is used in similarity computation.



B, D are points on ellipse

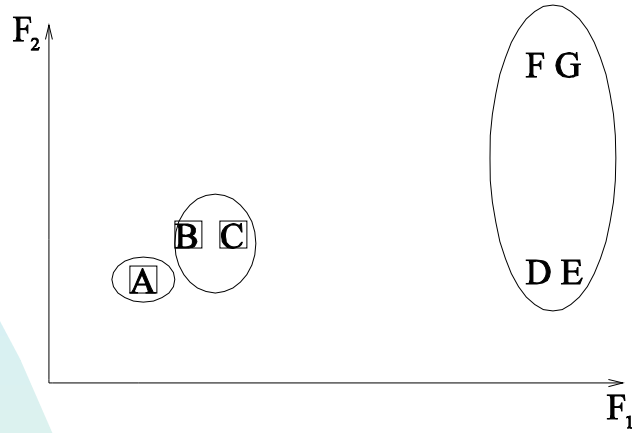
A is a point, close to B, on the rectangle

Grouping Schemes (Hierarchical Clustering)



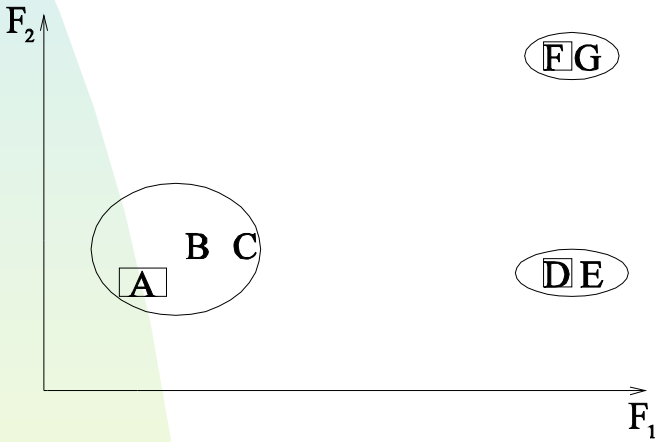
Agglomerative clustering treats each data point as a singleton cluster, and then successively merges clusters until all points have been merged into a single cluster. Multiple partitions are generated in a hierarchical manner.

Partitional Algorithms (K-Means Algorithm)



A, B, C are the initial Means

Optimal 3-partition



A, D, F are the initial Means

Nonoptimal 3-partition

Partitional algorithms generate a single partition of the dataset.

K-Means Algorithm: Variants

- Most popular because of its *simplicity and speed*
- Major problem is its convergence to a local optimum
  - Several solution directions
  - Notably randomized seeding promises O (log k) approximation algorithm for the k-means objective
  - Stochastic search algorithms: Genetic K-Means Algo
    - Uses distance-based mutation
    - K-Means operator instead of crossover
    - Converges to the Globally optimal solution
  - GA based on Tree-structured chromosomes scales up to deal with large datasets

Clustering Large Data Sets

- There are several applications where the data sets are large
  1. *Data Mining:* Both the number of patterns and features can be large
  2. *Text Processing:* The number of documents indexed is large and the number of terms also is large.
  3. *Pattern Classification:* There are several classification applications where the data sets are large.

- *Size of the data is large. So, it is not possible to fit the data in the main memory. It is stored on a disk and is transferred to memory based on need. Number of dataset scans is an important indicator of the feasibility of mining.*

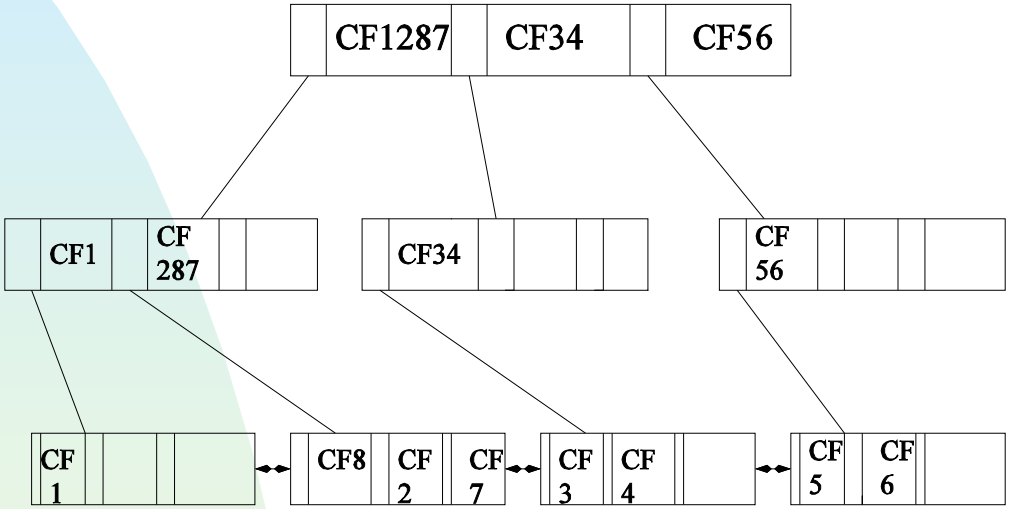
- *Random projections are useful to reduce the dimensionality.*

Algorithms

- Data is large; memory is small (GB, TB, PB)
- Solutions?
  - Incremental algorithms
  - Divide-and-conquer strategy
  - Compress the data and process
- These solution directions are used in
  - Clustering
  - Classification
  - Association Rule Mining

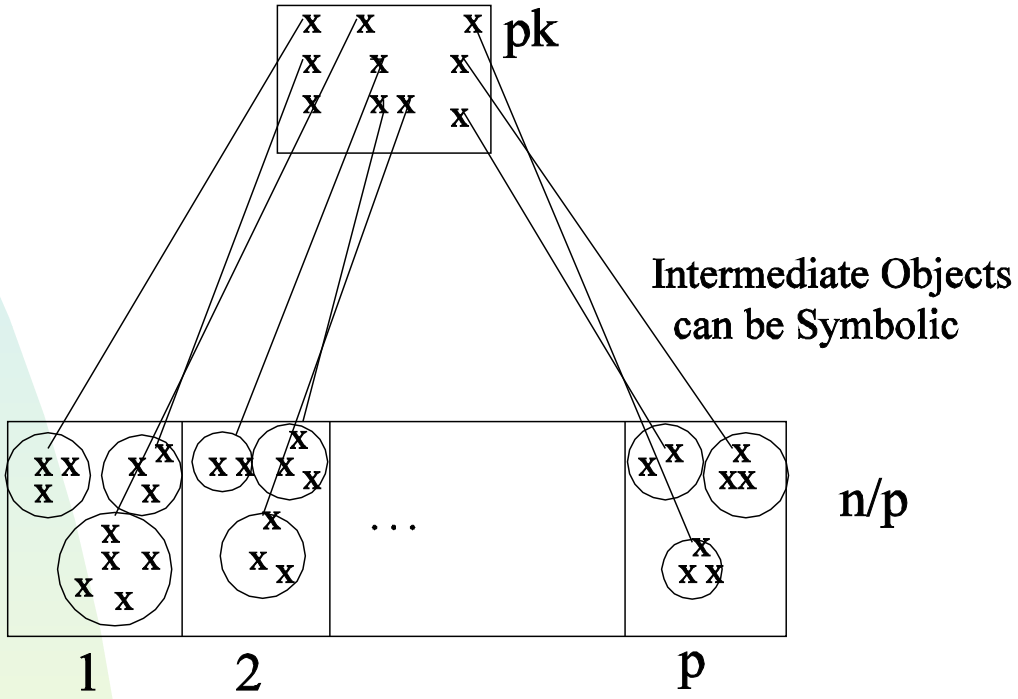
Incremental Algorithms

$S_k = \{x_1, x_2, \ldots, x_k\}$ leads to Abstraction A_k
Incremental algorithm generates A_{k+1} from A_k and X_{k+1}



- The CF vector of a cluster is represented by CFv = (|C|, $\sum_{O \in C} O$, $\sum_{O \in C} \| O \|^2$)
- CF tree can be generated using a single dataset scan
- Hierarchical structure is exploited for efficient training of large-scale SVM classification
- Most popular in large-scale data mining

Divide-and-Conquer



1. Divide the set into p blocks, each of size n/p patterns.
2. Cluster patterns in each block into K clusters.
3. Collect the pK representatives and cluster them into K clusters.
4. Relabel the clusters obtained in step 2.

Divide-and-Conquer: Variations

- Optimal value of P; find out the P that minimizes the number of distance computations
- In a two-level algorithm with N patterns divided into P blocks at the first level, C clusters in each block, PC cluster centers merged into K clusters is better than a one-shot K-means if

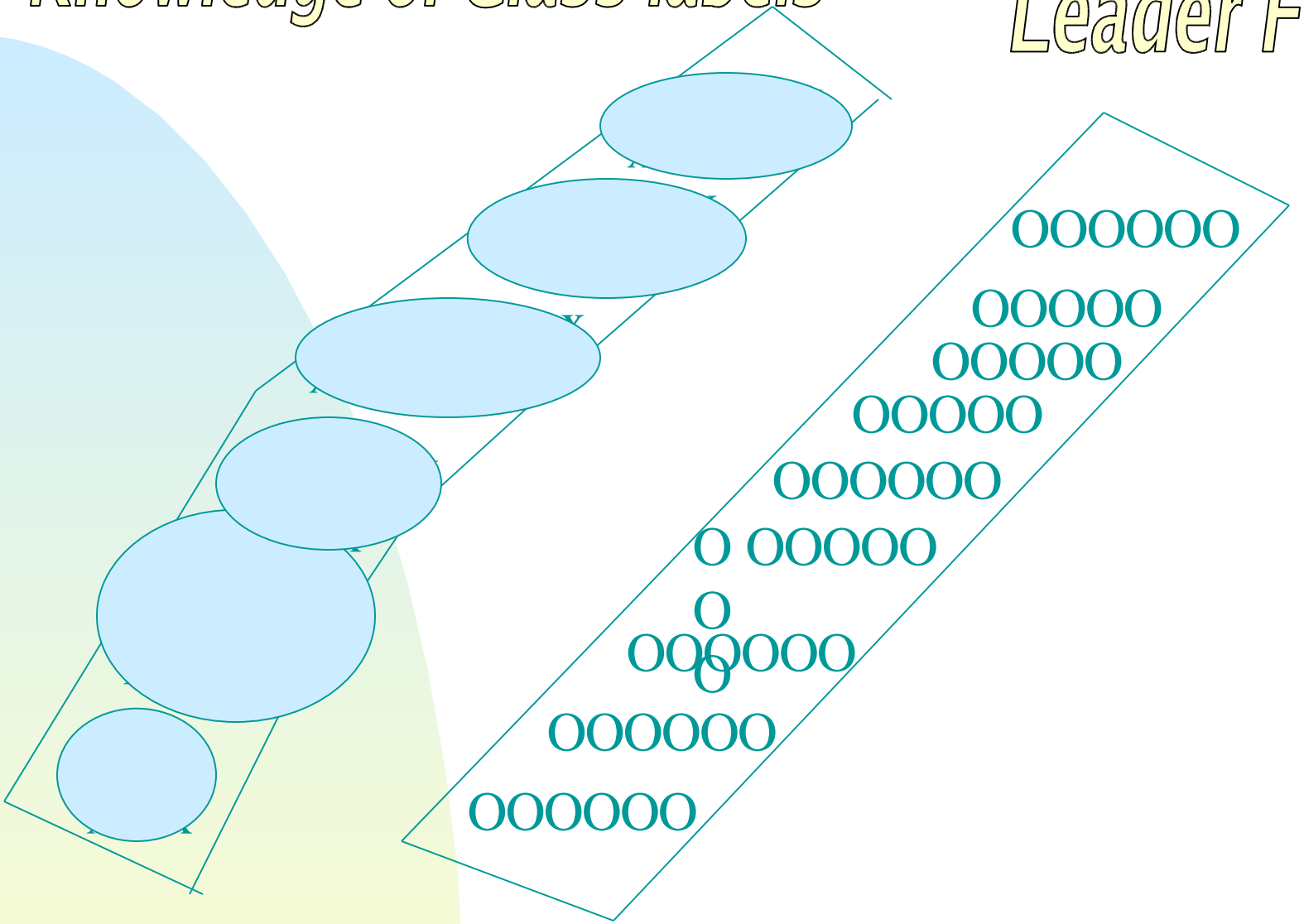
$$N > [PC(2K-C)] / 2(K-C)$$

- A 2-level algorithm with constant factor approximation is possible
- It is possible to use different algorithms at different levels; a cheaper algorithm at the first level followed by a more versatile algorithm at the second level. Such an algorithm may be called a Hybrid clustering algorithm.

Knowledge of Class labels

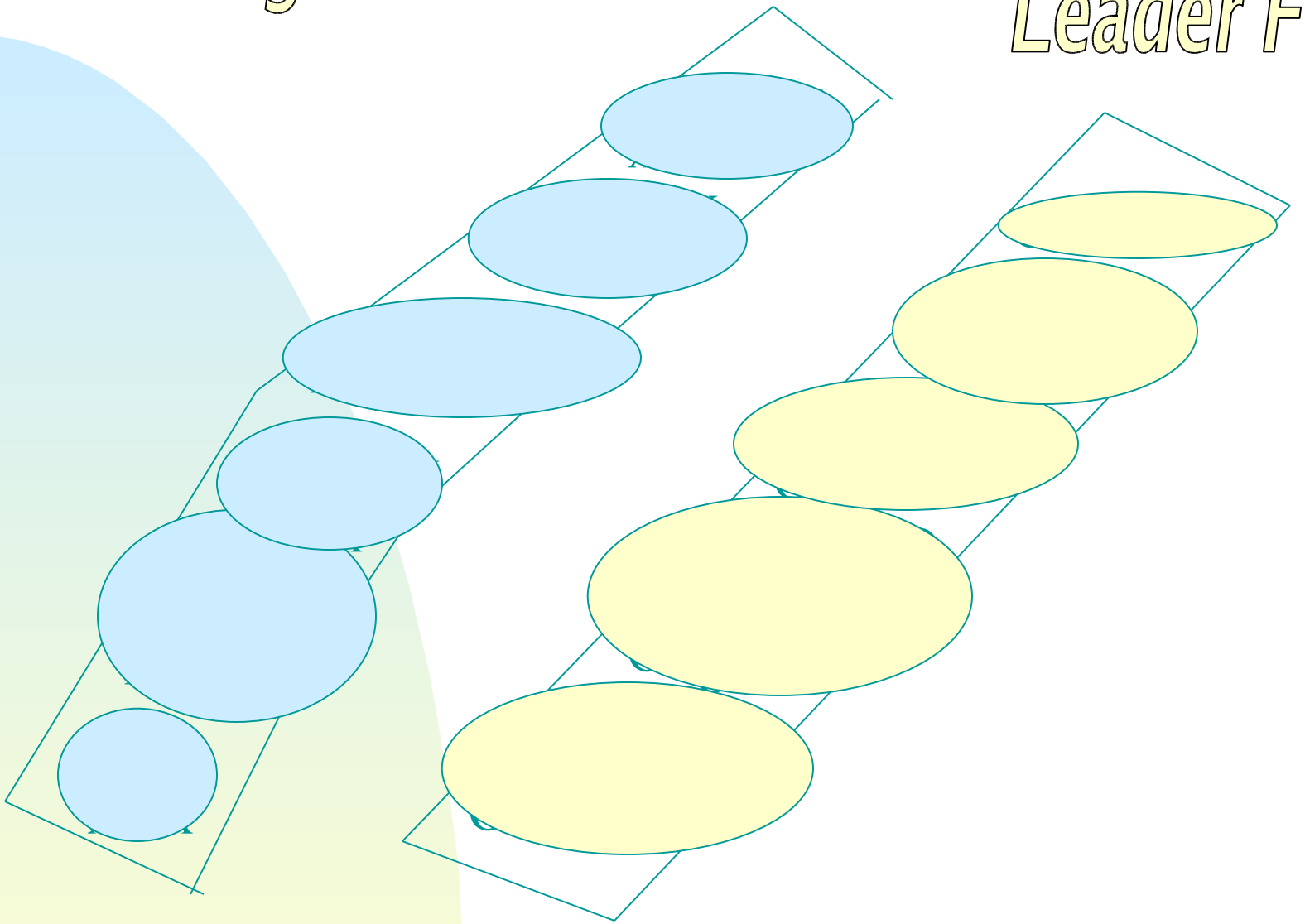
Knowledge of Class labels

Leader Follower



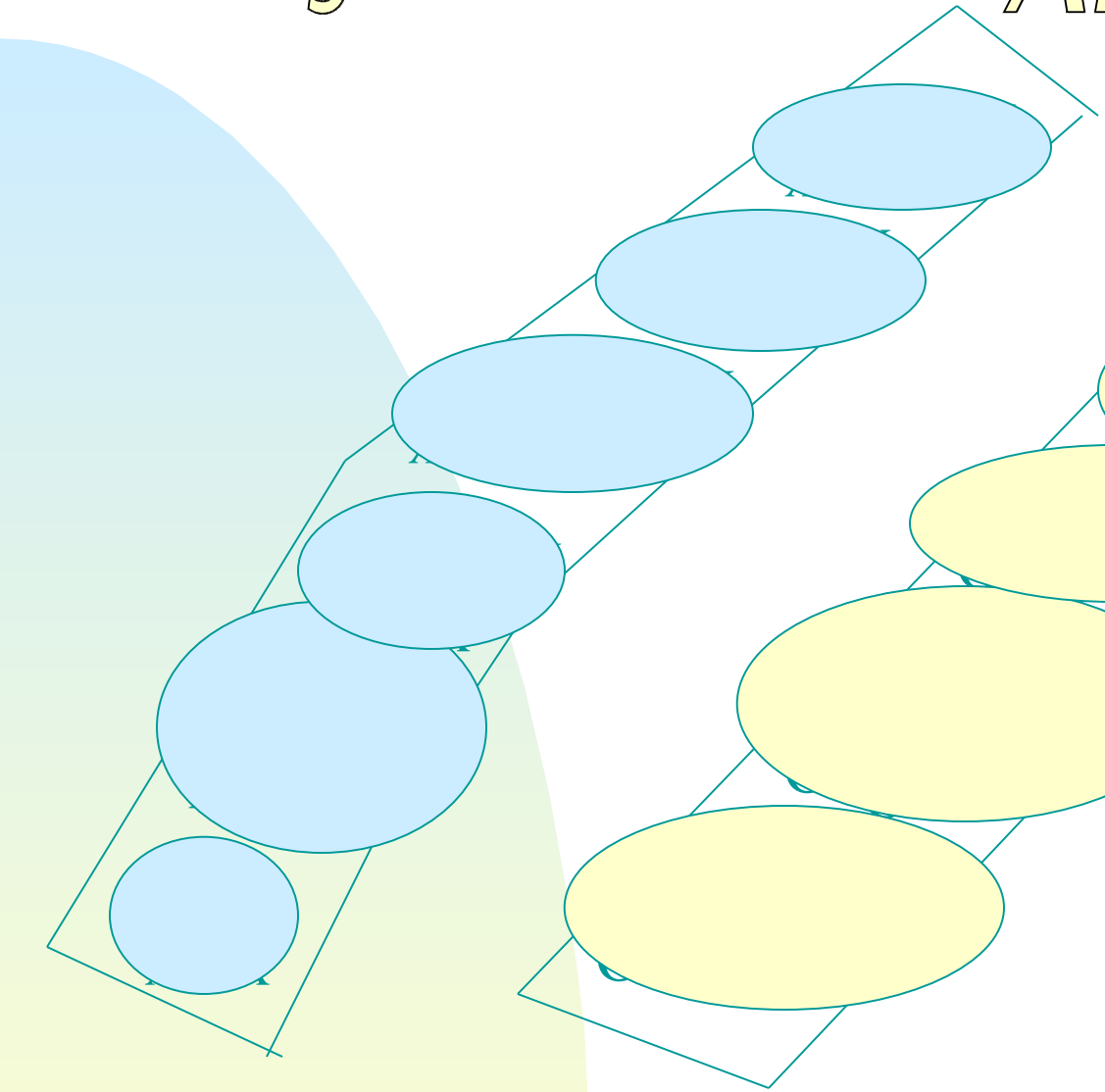
Knowledge of Class labels

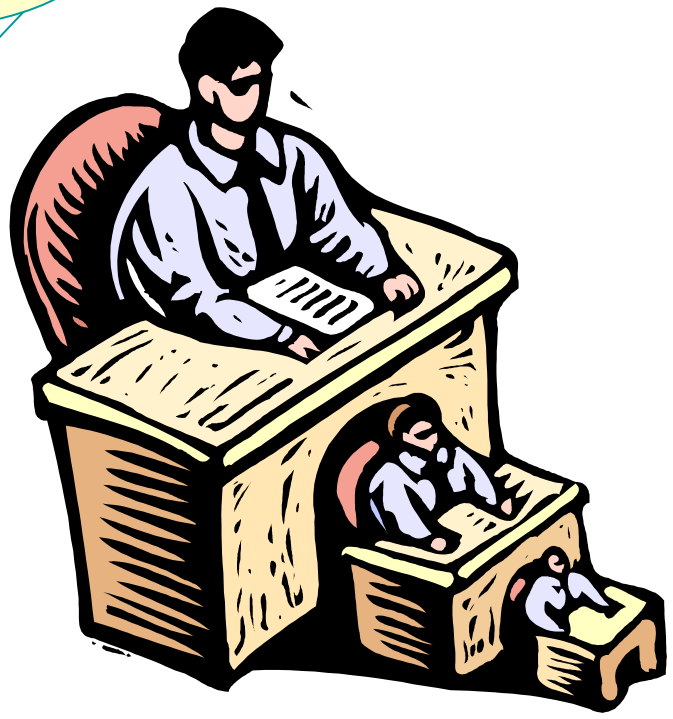
Leader Follower



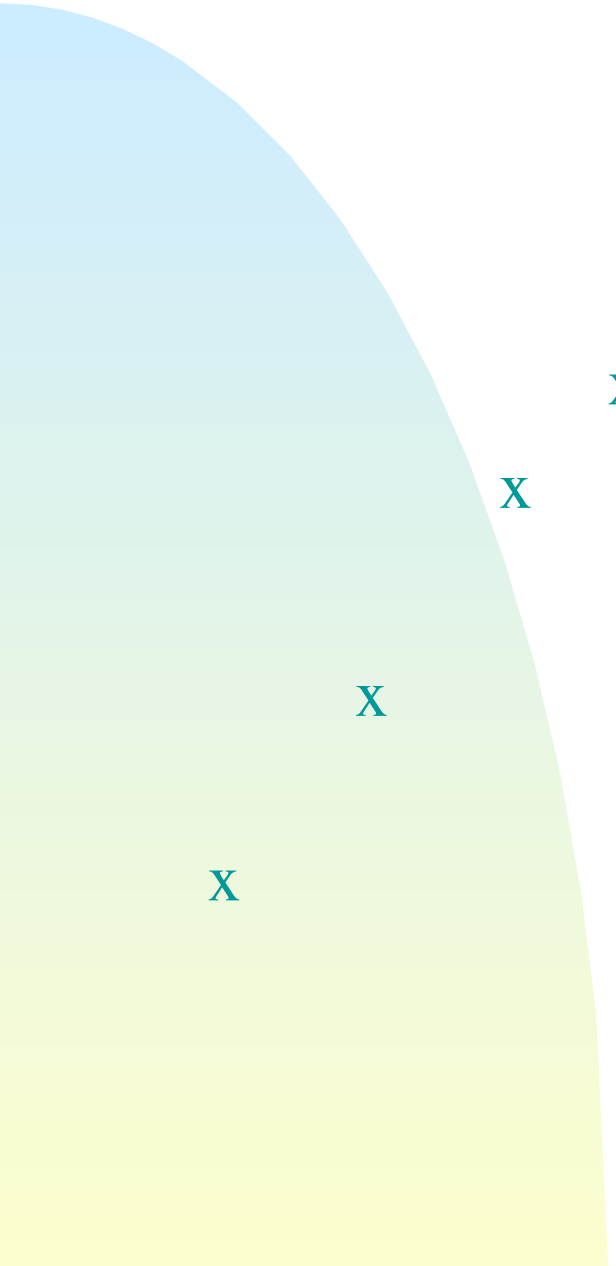
Knowledge of Class labels

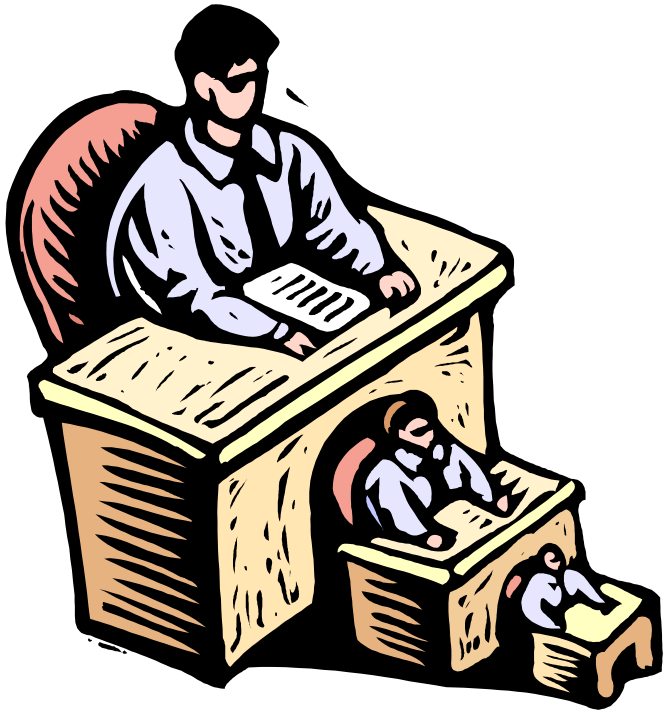
Abstraction



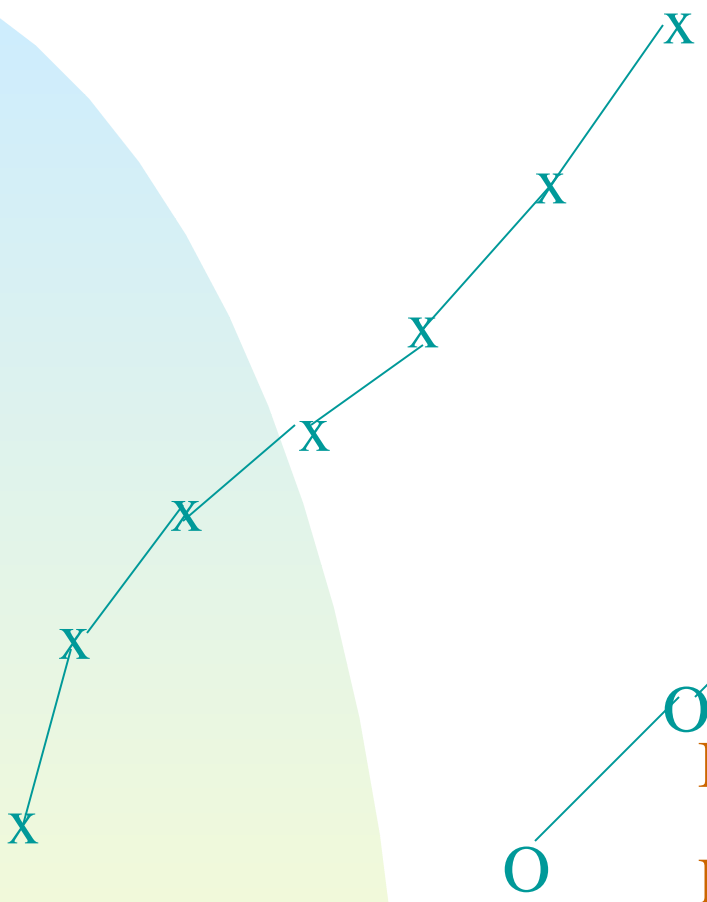


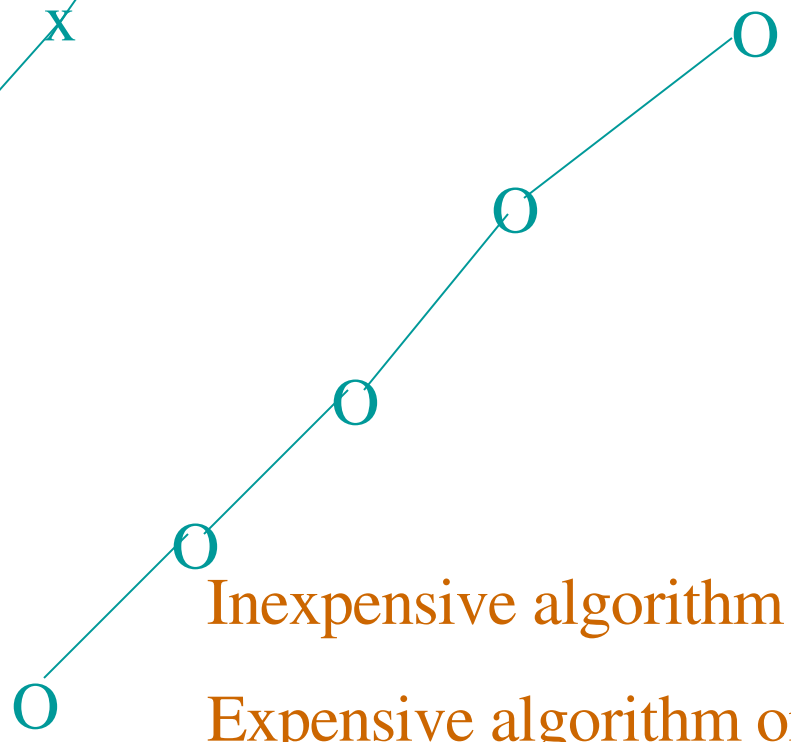
Cluster Representatives





Singe-Link





Inexpensive algorithm at level 1

Expensive algorithm on reduced data

(at level 2)

Using an Intermediate Representation

1. Start with an initial representation. Consider each pattern and suitably modify the representation.
2. Use the abstraction obtained in step 1 further.
   - Important Properties of this class of algorithms:
     - They require only one scan of the database.
     - The resulting structure may be used for synthesis also.
     - Abstraction may be used for other data mining tasks.
     - The abstraction may fit in the main memory even when the data set does not!
   - Centroids, medoids, leaders, CF-Tree are example abstractions.

Frequent Itemsets

ID	Items
1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

Support	Itemsets
100%(6)	C
83%(5)	CW
67%(4)	CD, CT, ACW
50%(3)	CDW, ACTW

Size	Frequent Itemsets
1	A, C, D, T, W
2	AC, AD, AT, AW, CD, CT, CW, DT, DW, TW
3	ACT, ACW, ATW, CDW, CTW
4	ACTW

Clustering using Frequent Itemsets

Consider *maximal frequent itemsets*: ACTW (3) and CDW (3)

Generate clusters having the itemset considered:

Cluster1 = { ACTW, ACTW, ACDTW}

Cluster2 = { CDW, ACDW, ACDTW}

A Hard Partition can be generated:

Cluster1 = Cluster1 - Cluster2

Cluster2 = { CDW, ACDW}

A Priori takes L database scans, where

L = maxi |Si|; (L = 4 in the example)

This gives a flavor of logic-based clustering; we get not only clusters but their descriptions as well.

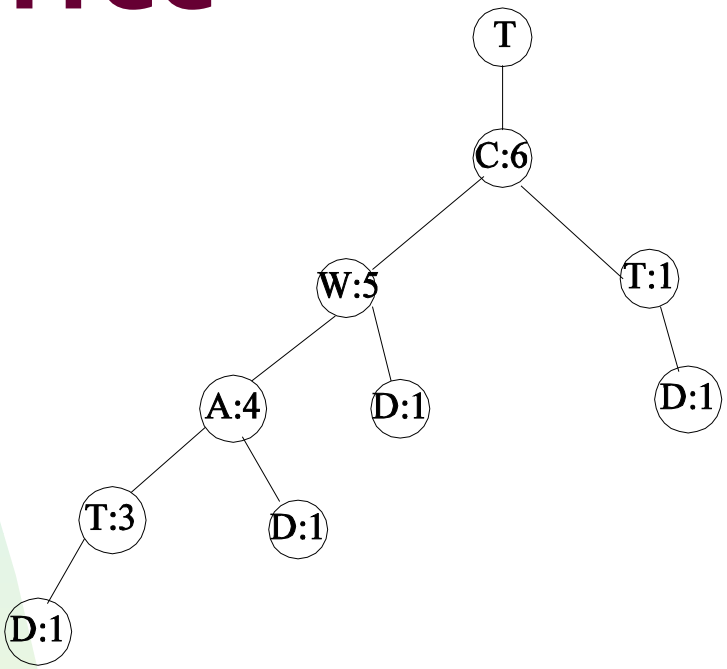
Frequent-Pattern Tree

Frequencies of items are:

C – 6; W – 5; A – 4; T – 4; D - 3

id	Original	Frequency-based
1	ACTW	CWAT
2	CDW	CWD
3	ACTW	CWAT
4	ACDW	CWAD
5	ACDTW	CWATD
6	CDT	CTD

FP-Tree



1. FP-Tree is a compact representation of the data.
2. It requires 2 db scans to construct the FP-Tree.
3. It can be used for frequent itemset generation, association rules, and classification.
4. It can be used for clustering.

Summary

- Pattern and cluster representation schemes play a major role.
- Similarity measures/functions should be appropriately chosen.
- For data mining purposes, single scan algorithms are meaningful.
- Divide-and-Conquer strategy is useful in handling large data sets successfully.
- Intermediate representations can be useful for clustering large data sets. Same structure may be used for other data mining tasks.
- Compressing the data and clustering the compressed data directly is a good direction to explore.