Data Mining
Lecture 6

CART™ Algorithm
Main Steps

- Grow the maximal tree based on the entire data set
  - A binary splitting procedure
  - Splitting rules
  - Stopping criteria
- Derive a set of pruned sub-trees
  - Create "efficiency frontier"
- Select the best tree by using validation set or cross-validation

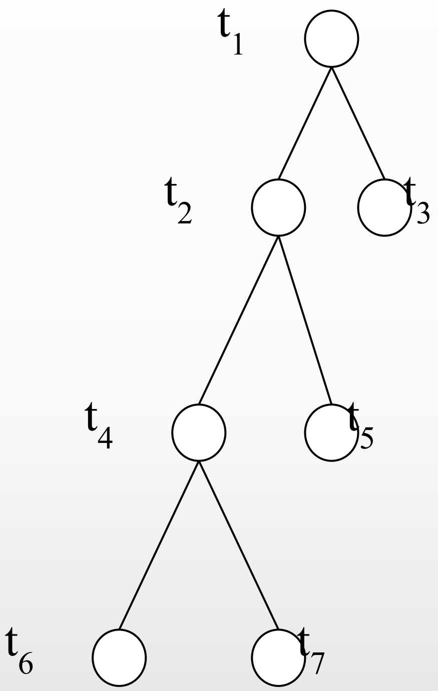
CART™ : Binary Splitting Procedure

- Continuous (Ordinal) Attributes
  - Each distinct value is considered for threshold
  - Branching rule: $x \leq C$
  - M possible splits (M - number of distinct values)
- Nominal (Categorical) Attributes
  - The branching rule is determined separately for each possible value
  - $2^{M-1} - 1$ possible splits (M - number of values)

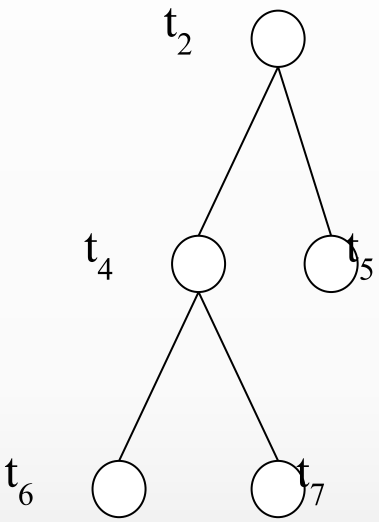
CART™ : Stopping Criteria

- Splitting is impossible
  - One case left in a node
  - All the cases in the node have the same target value
- Other reasons
  - Too few cases in the node (default = 10 cases)

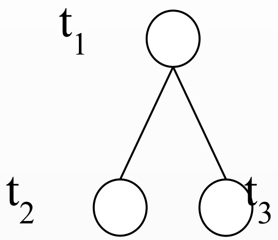
Pruning Trees



Tree T



Branch
T_{t2}



Sub-tree
$T - T_{t2}$

Deriving a set of pruned sub-trees

- Objective: minimizing the cost-complexity function

$$R_{\alpha}(T) = R(T) + \alpha \cdot \left|\widetilde{T}\right|$$

  - *T* - a tree
  - *R (T)* - the training error rate of a tree
  - R_{α} *(T)* - the cost-complexity of a tree
  - $\left|\widetilde{T}\right|$ - number of terminal nodes in a tree
  - α - complexity parameter (real number, greater than zero)

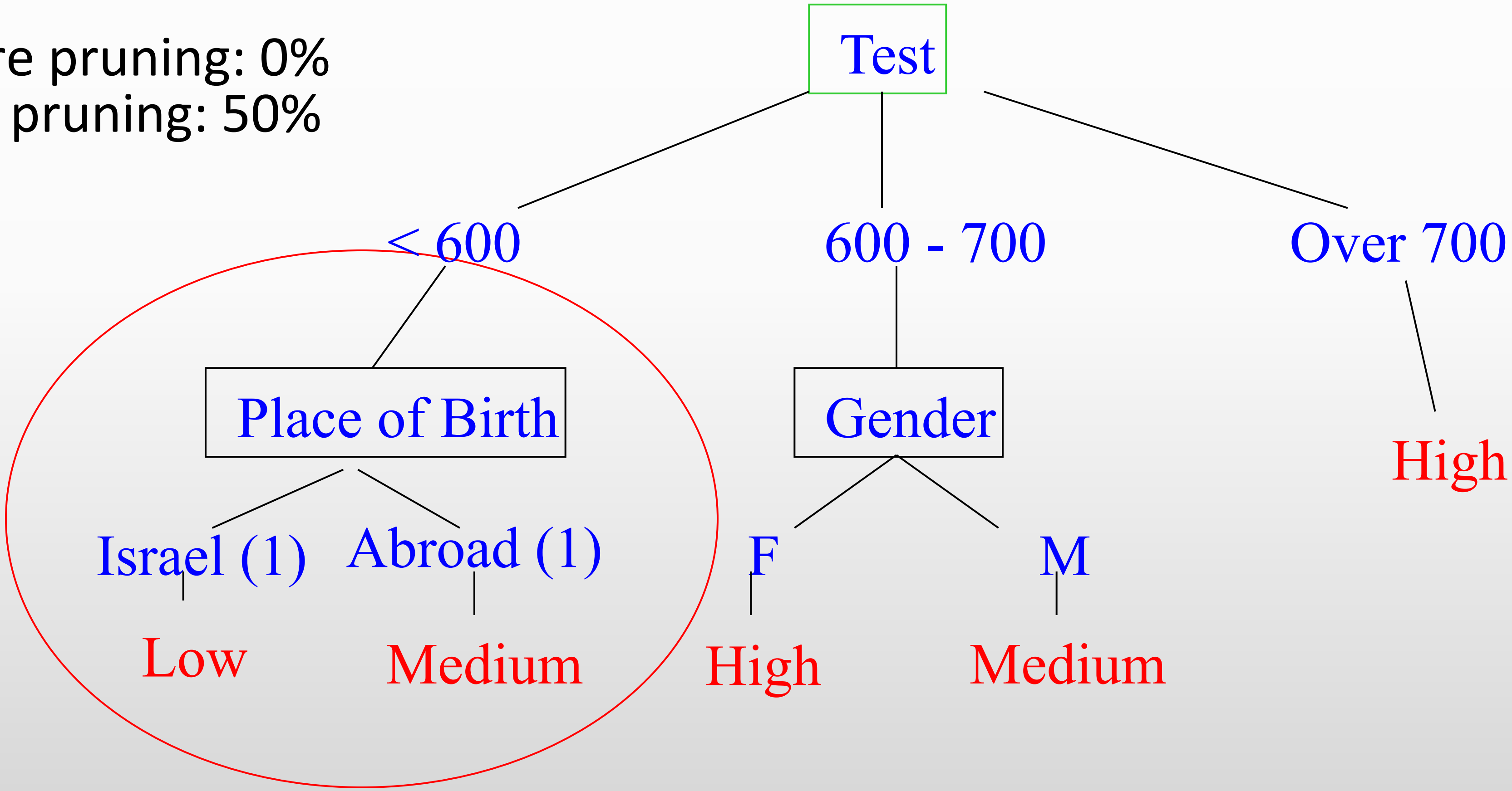
CART™ Pruning Algorithm

$$\boxed{R_\alpha(T) = R(T) + \alpha \cdot \left|\widetilde{T}\right|}$$

- Step 1 - Initialize the list of optimal trees with the maximal tree
- Step 2 - Initialize α = 0
- Step 3 - Increase α until the tree ceases to be optimal
- Step 4 - Find a new sub-tree, which is optimal with the new value of α
- Step 5 - Add the new sub-tree to the list of optimal trees.
- Step 6 - If the new sub-tree has more than one terminal node, go to Step 3. Otherwise, stop.

CART™ Student Example
Maximal Tree ($\alpha = 0$)

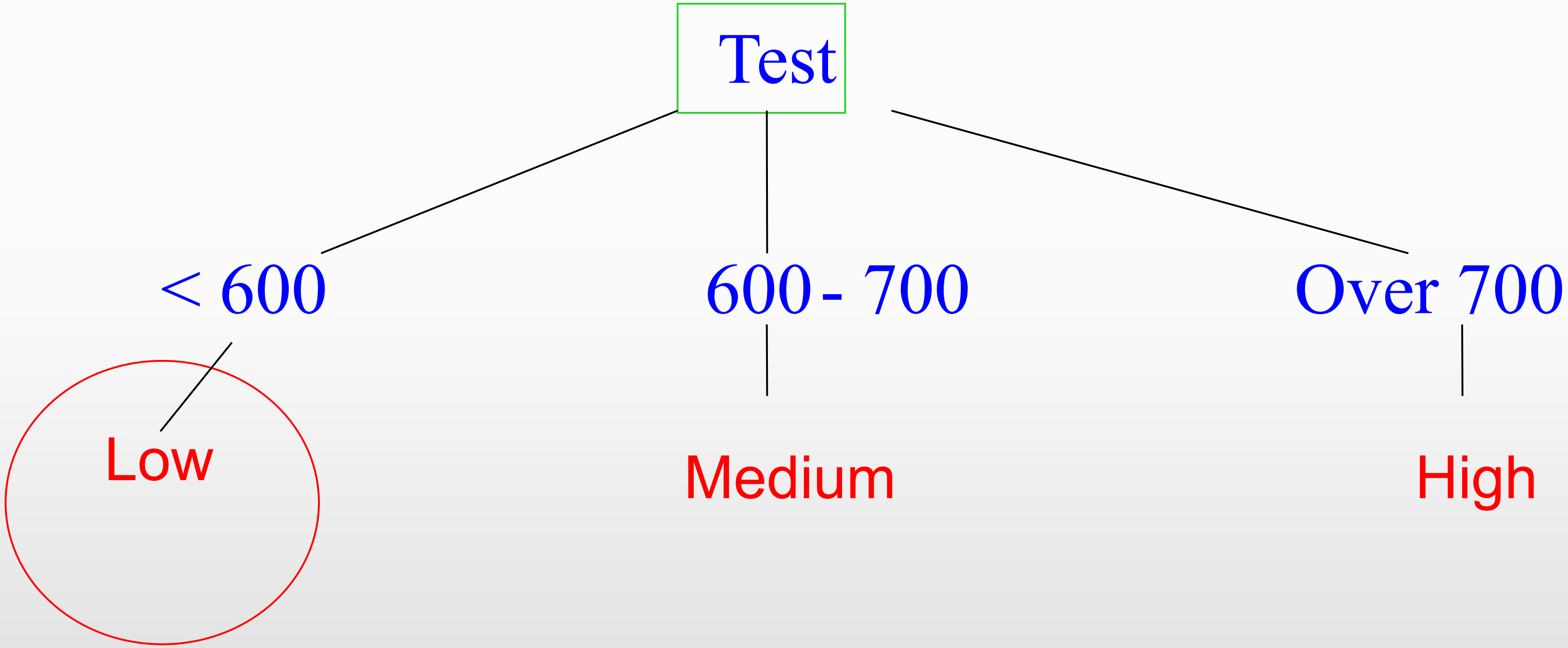
Error
Before pruning: 0%
After pruning: 50%



CART™ Student Example (cont'd) Removing *Place of Birth*

- Cost-complexity of the single node t
  - $R_\alpha(\{t\}) = R(t) + \alpha*1 = 0.50 + \alpha$
- Cost-complexity of the branch T_t
  - $R_\alpha(T_t) = R(T_t) + \alpha*|\check{T}_t| = 0 + \alpha*2$
- The critical value of α
  - $R_\alpha(\{t\}) = R_\alpha(T_t)$
  - $0.50 + \alpha = 2\alpha$
  - $\alpha = 0.50$

CART™ Student Example (cont’d)
New Sub-Tree (α = 0.50)



Lecture No. 6 – Decision Tree Learning II

- Rule Extraction
- Discretization of Continuous Attributes
- Alternative Splitting Rules
    - Information Gain Ratio
    - Gini Index
    - Twoing
- CART Overview
- Comparison of Decision Trees

Metrics for Performance Evaluation…

Confusion Matrix:

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a (TP)	b (FN)
ACTUAL CLASS Class=No	c (FP)	d (TN)

a: TP (true positive)

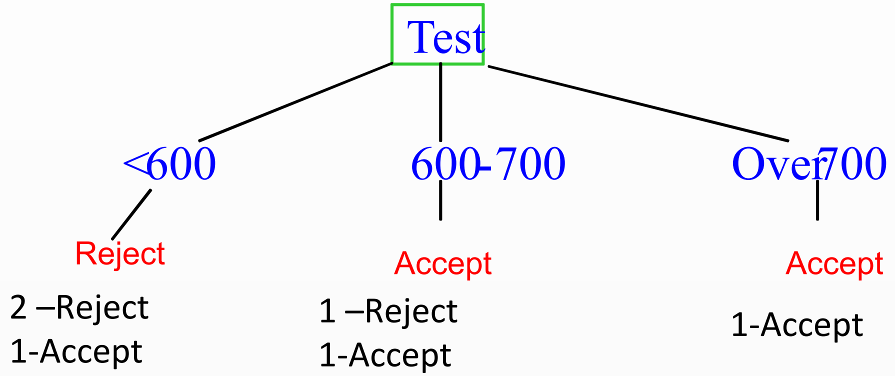
b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Confusion Matrix Example



	PREDICTED CLASS		
ACTUAL CLASS		Class=Accept	Class=Reject
	Class=Accept	a = 2 (TP)	b = 1 (FN)
	Class=Reject	c = 1 (FP)	d = 2 (TN)

Accuracy = ?

Cost-Sensitive Measures

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

	PREDICTED CLASS		
		Class=Accept	Class=Reject
ACTUAL CLASS	Class=Accept	a = 2 (TP)	b = 1 (FN)
	Class=Reject	c = 1 (FP)	d = 2 (TN)

p = ?

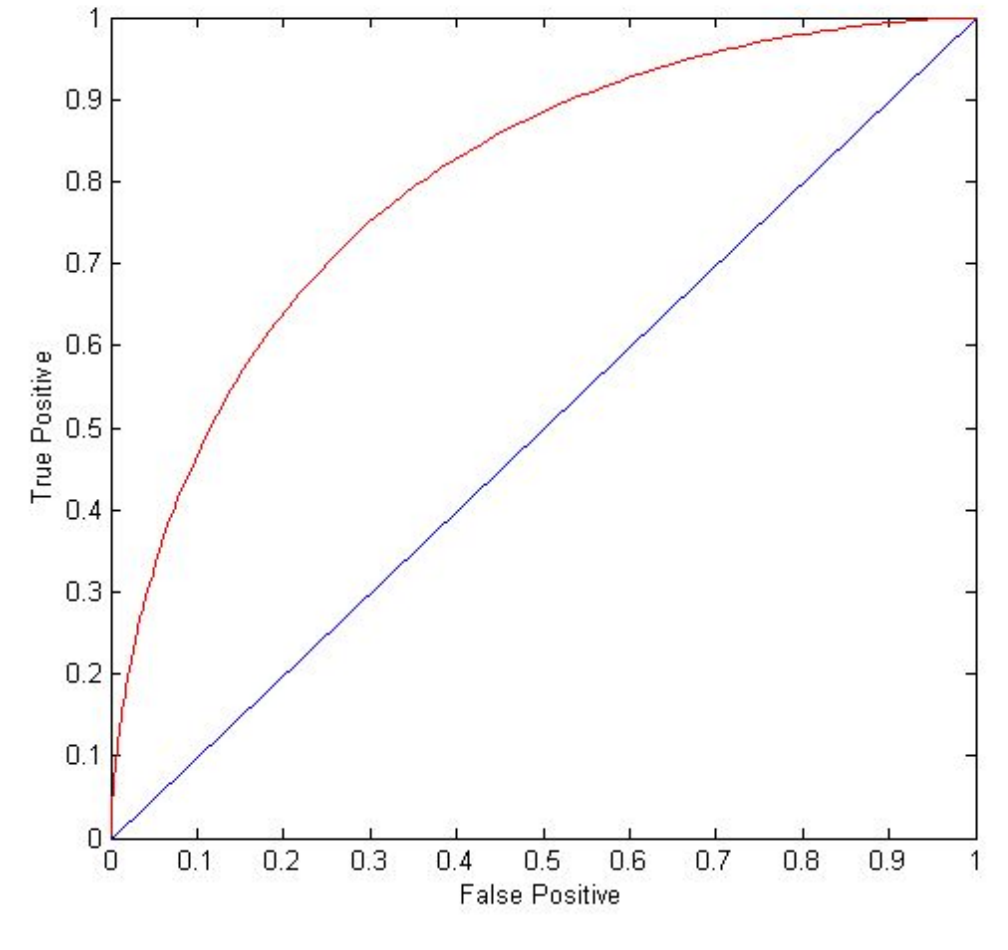
r = ?

F = ?

ROC Curve

TP Rate = TP / P

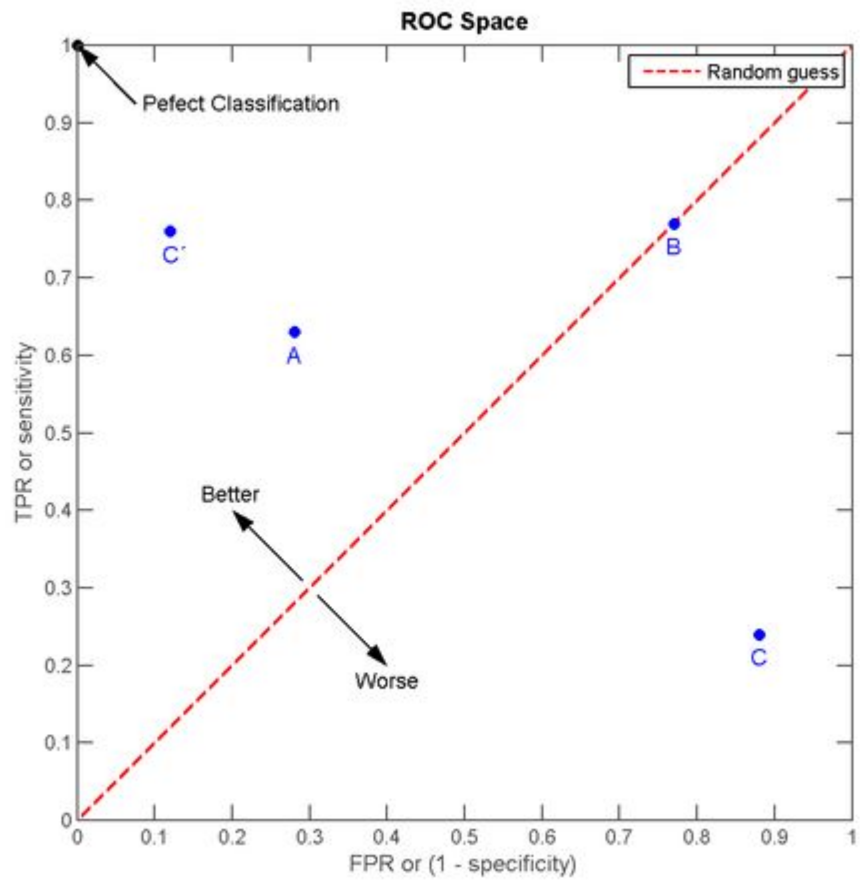
FP Rate = FP / N



(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess: ניחוש אקראי לחלוטין
    - Area = 0.5 (diagonal line)



FP(1)	TP
TN	FN (2)

בדוגמה הבאה מוצגים ארבעה סיווגים שונים ממדגם שבו 100 פרטים חיוביים ו-100 שליליים:

A	B	C	C'
63	28	91 37	72	109 100	100	200	77	77	154 23	23	46 100	100	200	24	88	112 76	12	88 100	100	200	76	12	88 24	88	112 100	100	200
שח"א = 0.63	שח"א = 0.77	שח"א = 0.24	שח"א = 0.76
שח"כ = 0.28	שח"כ = 0.77	שח"כ = 0.88	שח"כ = 0.12
PPV = 0.69	PPV = 0.50	PPV = 0.21	PPV = 0.86
F1 = 0.66	F1 = 0.61	F1 = 0.22	F1 = 0.81
נכונות = 0.68	נכונות = 0.50	נכונות = 0.18	נכונות = 0.82

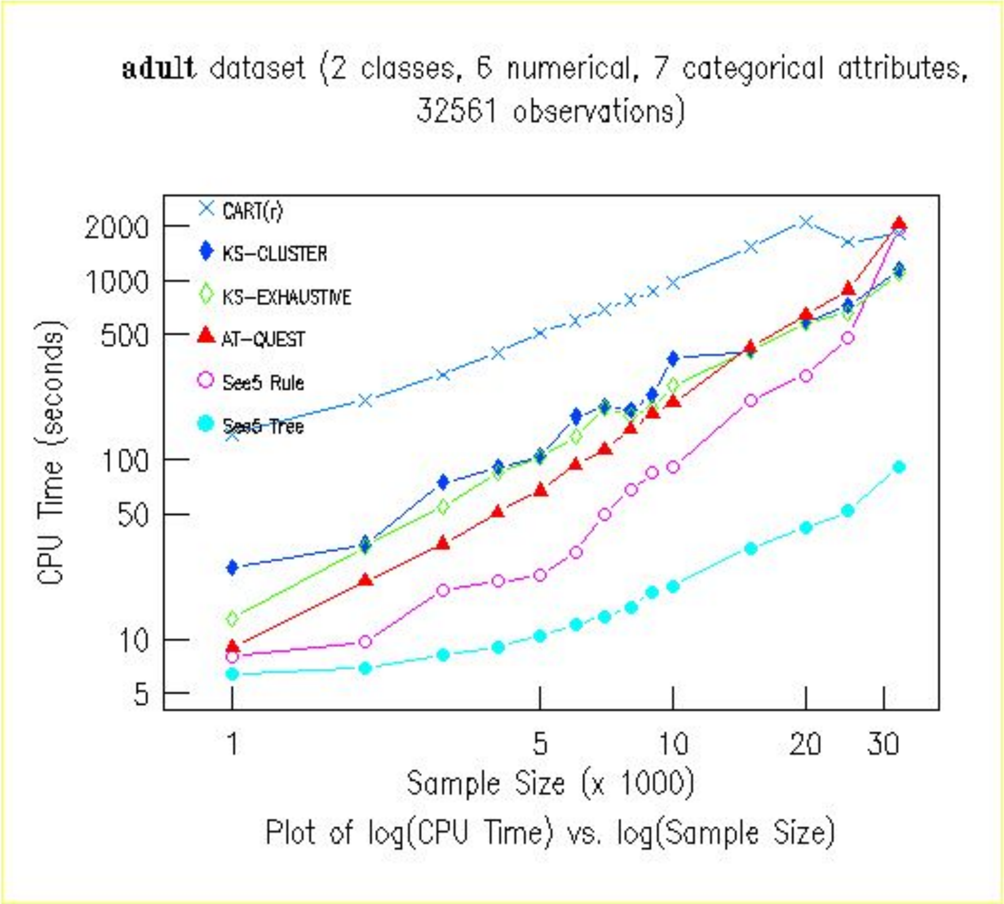
ארבעת המסווגים מיוצגים על ידי נקודת במרחב ROC שבתמונה. התמונה מראה בבירור שמסווג A הוא המוצלח ביותר מבין A, B ו-C. תוצאת הסיווג של B נמצאת על קו הניחוש האקראי (האלכסון), וניתן לראות גם שהנכונות של B היא 50%.

עם זאת, אם נשקף את נקודה C דרך הנקודה (0.5,0.5), התוצאה המתקבלת, 'C היא מוצלחת אפילו יותר מ-A. 'C פשוט מחליפה בין הקטגוריות של C (כאשר C מסווגת "שלילי", 'C מסווגת חיובי, ולהפך).

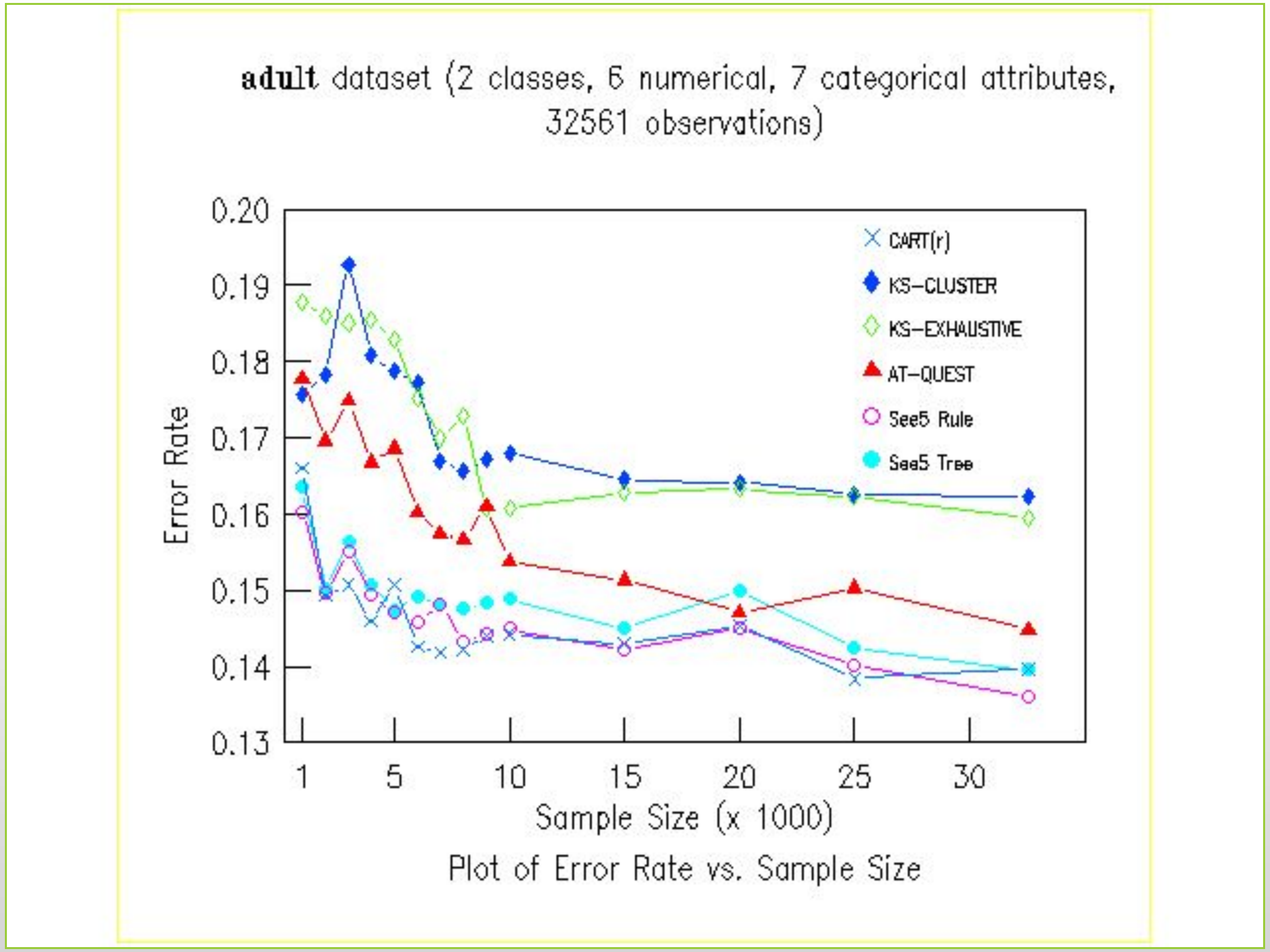
PPV= שח"א/(שח"א+שח"כ). PPV מייצג את שיעור החיובים האמיתיים מתוך כלל החיוביים.

https://he.wikipedia.org/wiki/%D7%A2%D7%A7%D7%95%D7%

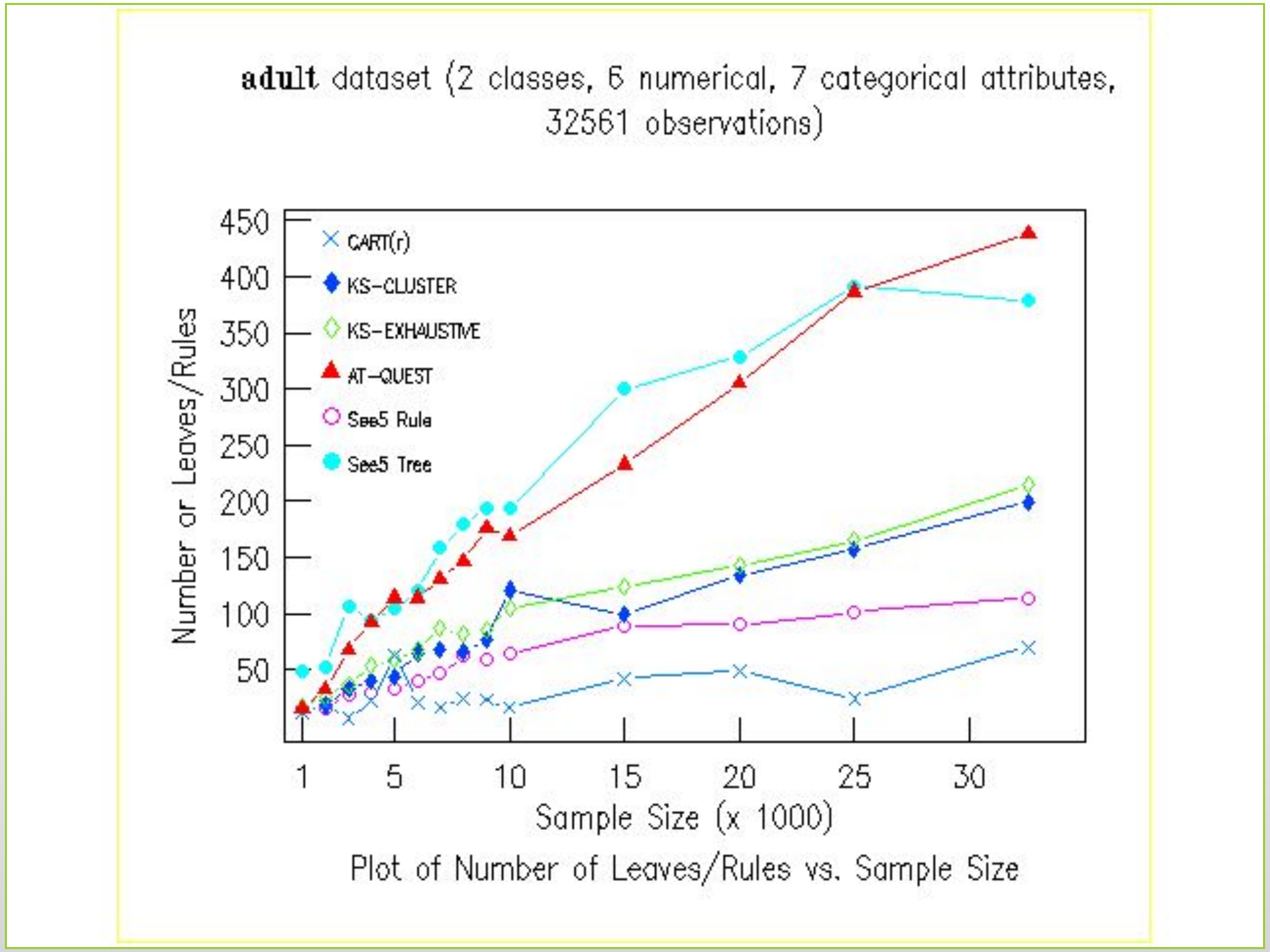
Comparison of Decision Trees
(based on Lim *et al*., Machine Learning, 40, 203–228, 2000)
Computational Complexity



Comparison of Decision Trees
Error Rate



Comparison of Decision Trees
Tree Size

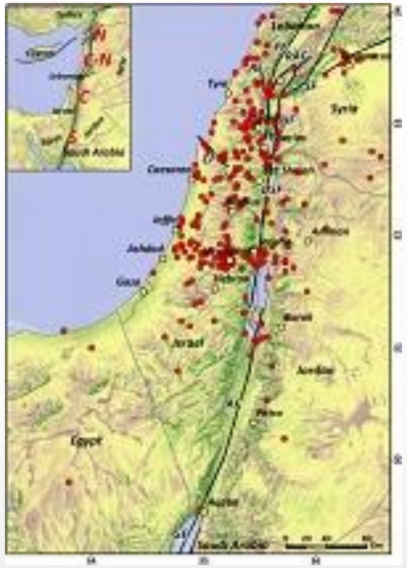


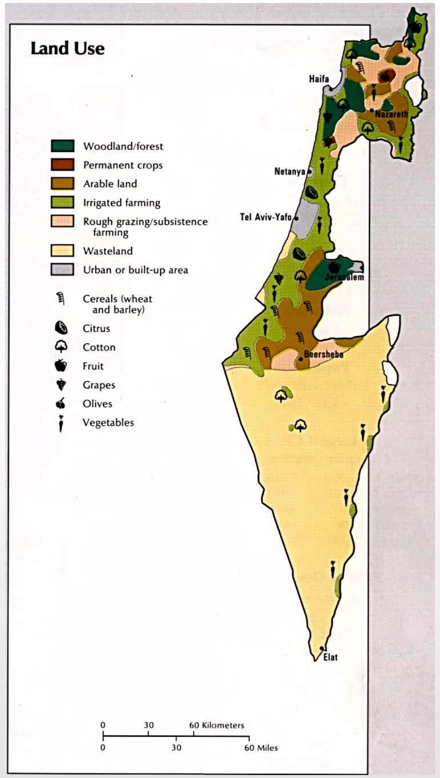
What is Cluster Analysis?

- 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

Examples of Clustering Applications

- **Information retrieval**: document clustering
- **Land use**: Identification of areas of similar land use in an earth observation database
- **Marketing**: Develop targeted marketing programs
- **City planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Earthquake studies**: Observe earthquake epicenters
- **Climate**: understanding earth climate, find patterns of atmospheric and ocean





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.

Requirements of Clustering in Data Mining

- Scalability (Big Data)
- 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
- Incremental clustering and insensitivity to input order
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Lesson 12. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Data Structures

- Data matrix
  - *p* – number of variables
  - *n* – number of objects
  - x_{if} - value of variable *i* in record *f*

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

- Dissimilarity matrix
  - *d (i,j)* – distance between objects *i* and *j*

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

Attribute Types in Clustering Analysis

- **Nominal:** categories, states, or "names of things"
  - *Hair_color* = {*auburn, black, blond, brown, grey, red, white*}
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - <u>Symmetric binary</u>: both outcomes equally important
    - e.g., gender
  - <u>Asymmetric binary</u>: outcomes not equally important.
    - e.g., medical test (positive vs. negative), TF
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size* = {*small, medium, large*}, grades, army rankings

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval**
  - Measured on a scale of **equal-sized units**
  - Values have order
    - E.g., *temperature in C˚or F˚, calendar dates*
  - No true zero-point
- **Ratio**
  - Inherent **zero-point**
  - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚).
    - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
  - Requirements
    - *Non-negativity: d(i,j)* ≥ 0
    - *Distance to itself: d(i,i)* = 0
    - *Symmetry: d(i,j) = d(j,i)*
    - *Triangular inequality: d(i,j) ≤ d(i,k) + d(k,j)*
- Some popular ones include: Minkowski distance:

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+\ldots+|x_{i_p}-x_{j_p}|^q)}$$

  - where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two *p*-dimensional data objects, and *q* is a positive integer

Similarity and Dissimilarity Between Objects (Cont.)

- If q = 1, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

- *If q = 2, d* is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

- Other dissimilarity measures are available

Binary Variables

- A contingency table for binary data
  - *p* – number of variables
- Distance measure for <u>symmetric</u> binary variables:
- Distance measure for <u>asymmetric</u> binary variables:
- Jaccard coefficient (*similarity* measure for <u>*asymmetric*</u> binary variables):

		Object *j*		
		1	0	*sum*
Object *i*	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

$$d(i,j) = \frac{b+c}{a+b+c}$$

$$sim_{Jaccard}(i,j) = \frac{a}{a+b+c}$$

Dissimilarity between Binary Variables

- Example

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
- the remaining attributes are asymmetric binary
- let the values *Y* and *P* be set to 1, and the value *N* be set to 0

$$d(i,j) = \frac{b+c}{a+b+c}$$

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

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

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

		Jack 1	Jack 0	*sum*
Mary	1	2	1	3
Mary	0	0	3	3
	sum	2	4	6

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
- Can be treated like interval-scaled
  - replace x_{if} by their 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

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

    - Example: academic ranks in Israel
  - compute the dissimilarity using methods for interval-scaled variables

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
  - Examples: salaries, web links
- Methods:
  - treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
  - apply logarithmic transformation

$$y_{if} = \log(x_{if})$$

  - treat them as continuous ordinal data, treat their rank as interval-scaled

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

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

  - $\delta_{ij}(f)$ – binary indicator ($\delta_{ij}(f) = 0$ if a variable f should be skipped)
  - $d_{ij}^{(f)}$ – contribution of variable f to dissimilarity between i and j
  - *Variable f* is binary or nominal:

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

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

Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches (II)

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

Typical Alternatives to Calculate the Distance between Clusters

- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e.,

$$dis(K_i, K_j) = min(t_{ip}, t_{jq})$$

- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e.,

$$dis(K_i, K_j) = max(t_{ip}, t_{jq})$$

- **Average**: avg distance between an element in one cluster and an element in the other, i.e.,

$$dis(K_i, K_j) = avg(t_{ip}, t_{jq})$$

- **Centroid**: distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$
- **Medoid**: distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Lesson 12. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Partitioning Algorithms: Basic Concept

- <u>Partitioning method:</u> Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

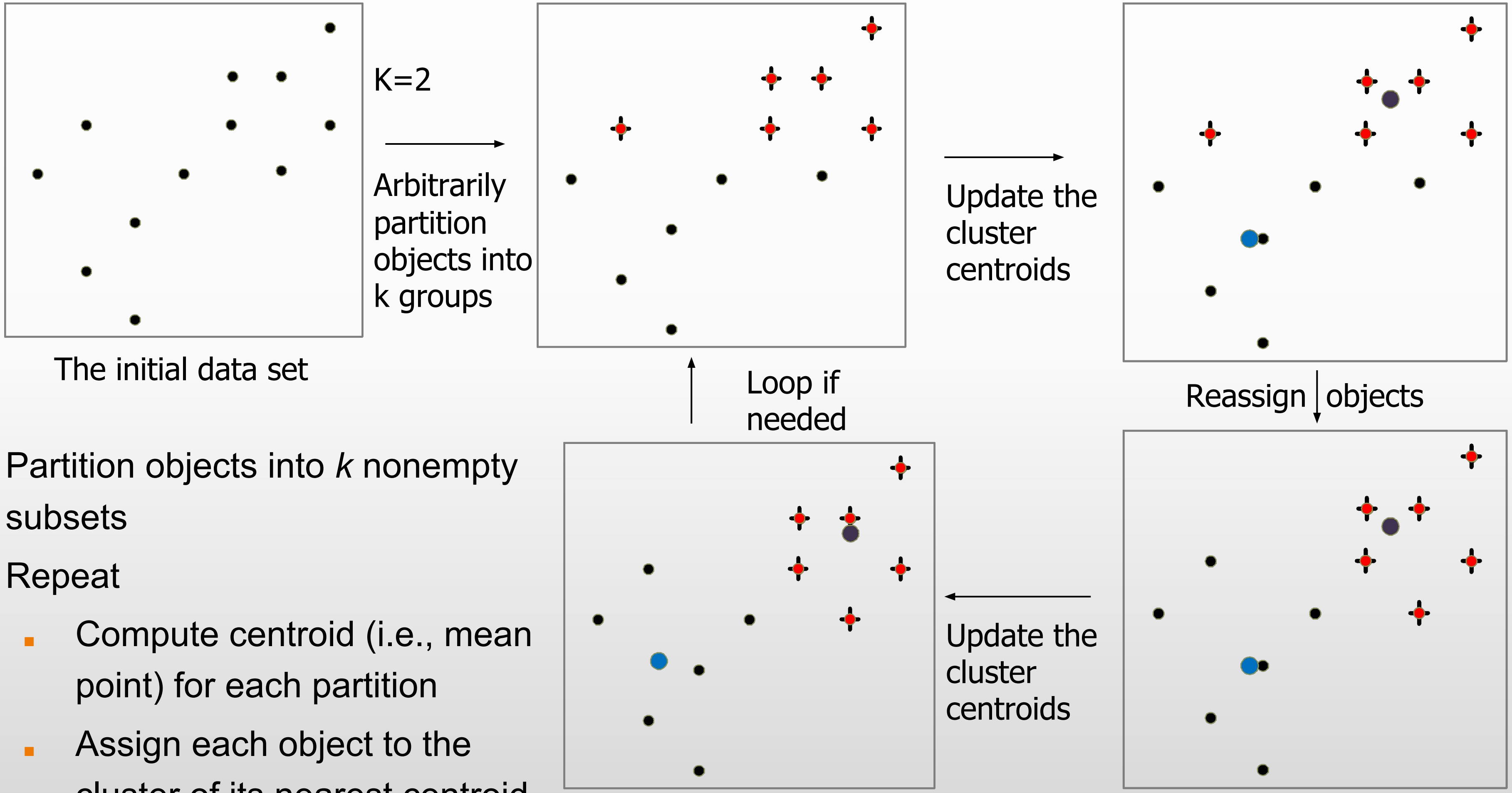
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *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
  - <u>*k-means*</u> (MacQueen’67, Lloyd’57/’82): Each cluster is represented by the center of the cluster
  - <u>*k-medoids*</u> 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 four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

An Example of *K-Means* Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

K-Means Example (k = 2)

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

Iteration 1

Cluster 1

Rec No	X1	X2	X3	X4	X5
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	1	1
Mean	0.33	0.33	0.33	0.33	0.33

Cluster 2

Rec No	X1	X2	X3	X4	X5
4	1	0	1	0	1
5	0	1	0	1	0
6	1	0	1	1	1
7	1	1	1	1	1
Mean	0.75	0.5	0.75	0.75	0.75

Objects Re-assignment

Rec No	Old	to Cluster 1	to Cluster 2	Min	New
1	1	0.745	1.581	0.745	1
2	1	0.745	1.581	0.745	1
3	1	1.491	0.707	0.707	2
4	2	1.247	1.000	1.000	2
5	2	1.106	1.414	1.106	1
6	2	1.374	0.707	0.707	2
7	2	1.491	0.707	0.707	2

K-Means Example (k = 2)

Iteration 2

Cluster 1

Rec No	X1	X2	X3	X4	X5
1	0	0	0	0	0
2	0	0	0	0	0
5	0	1	0	1	0
Mean	0.00	0.33	0.00	0.33	0.00

Cluster 2

Rec No	X1	X2	X3	X4	X5
3	1	1	1	1	1
4	1	0	1	0	1
6	1	0	1	1	1
7	1	1	1	1	1
Mean	1	0.5	1	0.75	1

Objects Re-assignment

Rec No	Old	to Cluster 1	to Cluster 2	Min	New
1	1	0.471	1.953	0.471	1
2	1	0.471	1.953	0.471	1
3	2	1.972	0.559	0.559	2
4	2	1.795	0.901	0.901	2
5	1	0.943	1.820	0.943	1
6	2	1.886	0.559	0.559	2
7	2	1.972	0.559	0.559	2

Comments on the *K-Means* Method

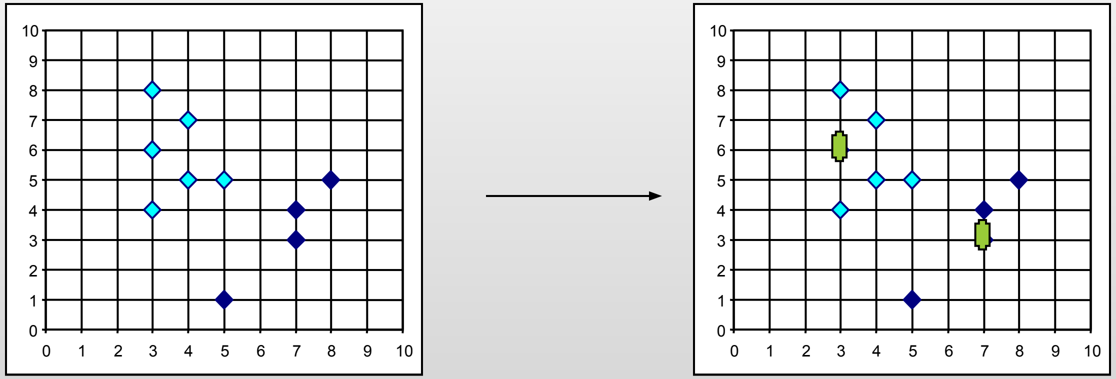
- <u>Strength:</u> *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- <u>Comment:</u> Often terminates at a *local optimal*.
- <u>Weakness</u>
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

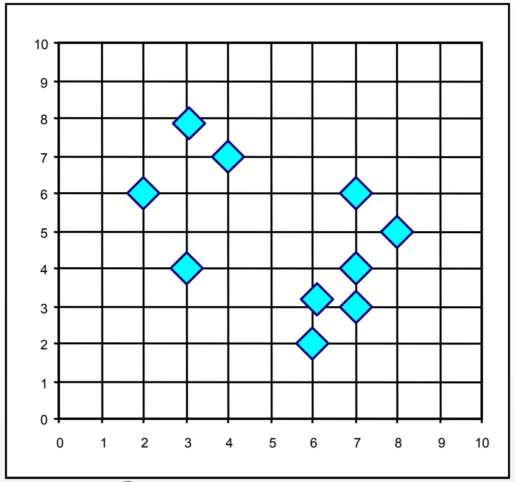
- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to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

What is the problem of k-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

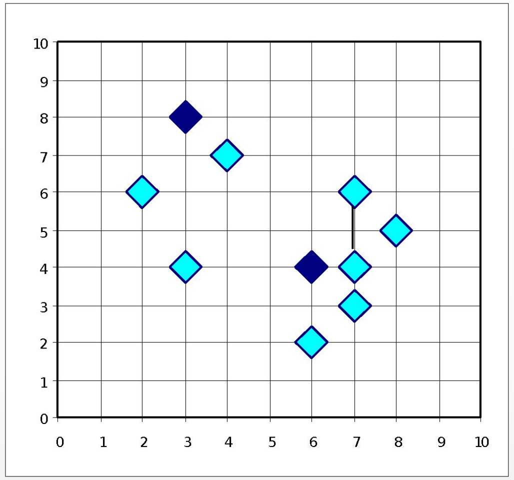


PAM: A Typical K-Medoids Algorithm



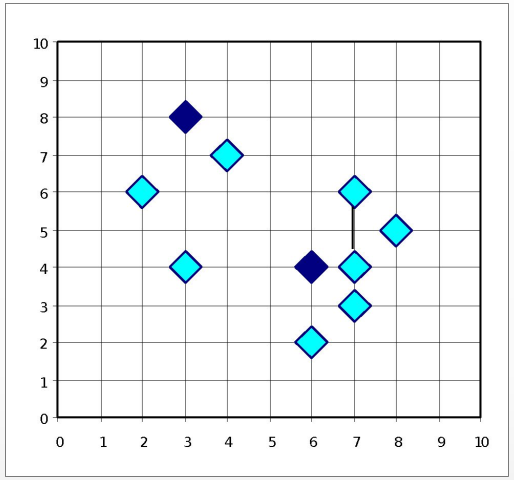
K=2

Arbitrary choose *k* objects as initial medoids

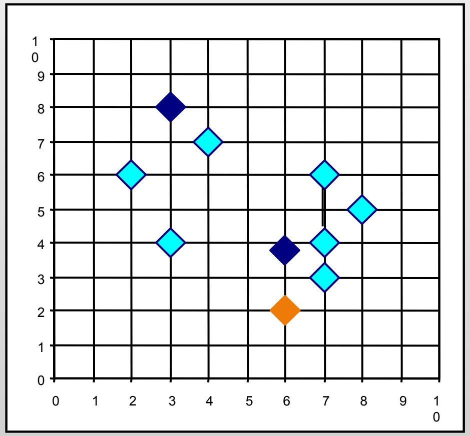


Assign each remaining object to nearest medoids

Total Cost = 20

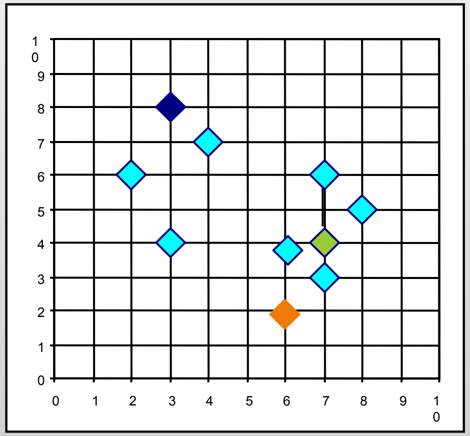


Randomly select a nonmedoid object, O_{random}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{random}

If quality is improved.

Do loop

Until no change

The K-Medoid Clustering Method

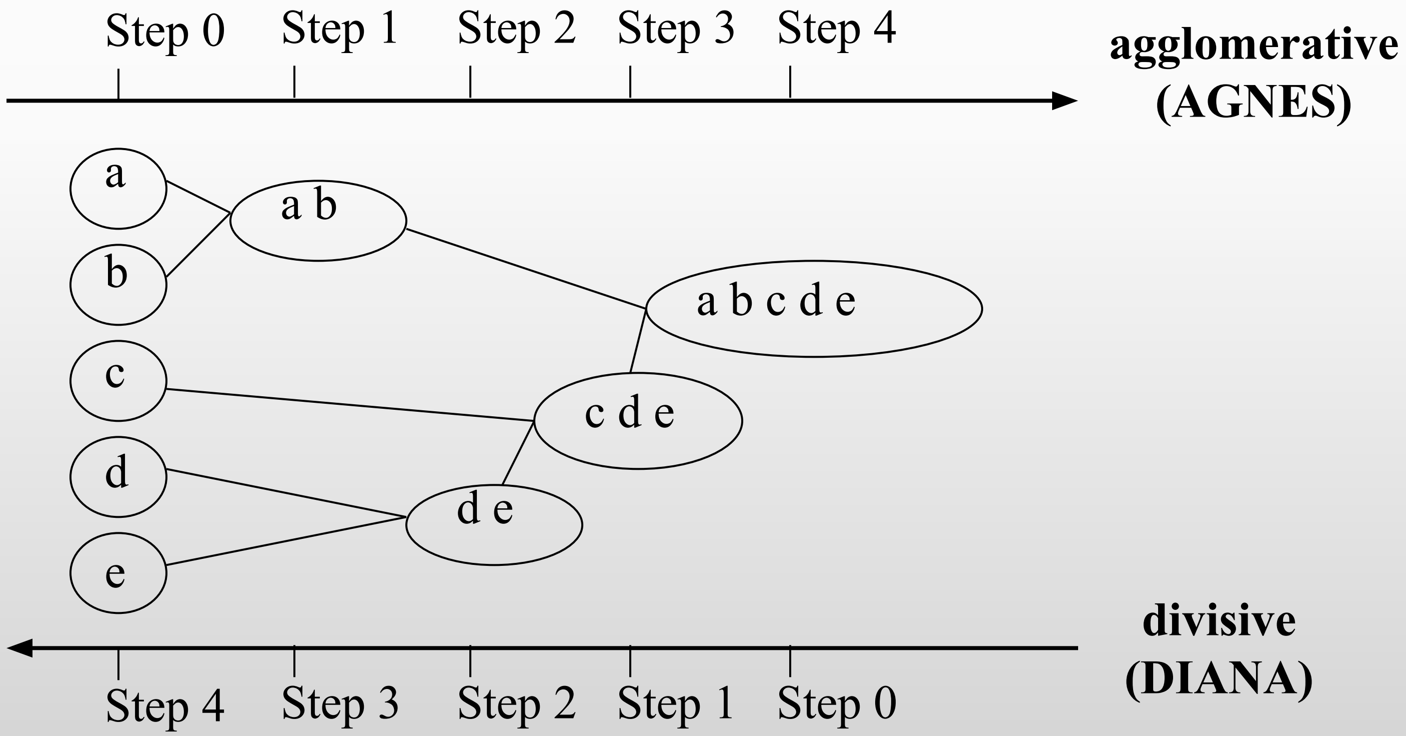
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the randomly selected non-medoids if it decreases the total distance of the resulting clustering
      - Each object is assigned to a cluster represented by the nearest medoid
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

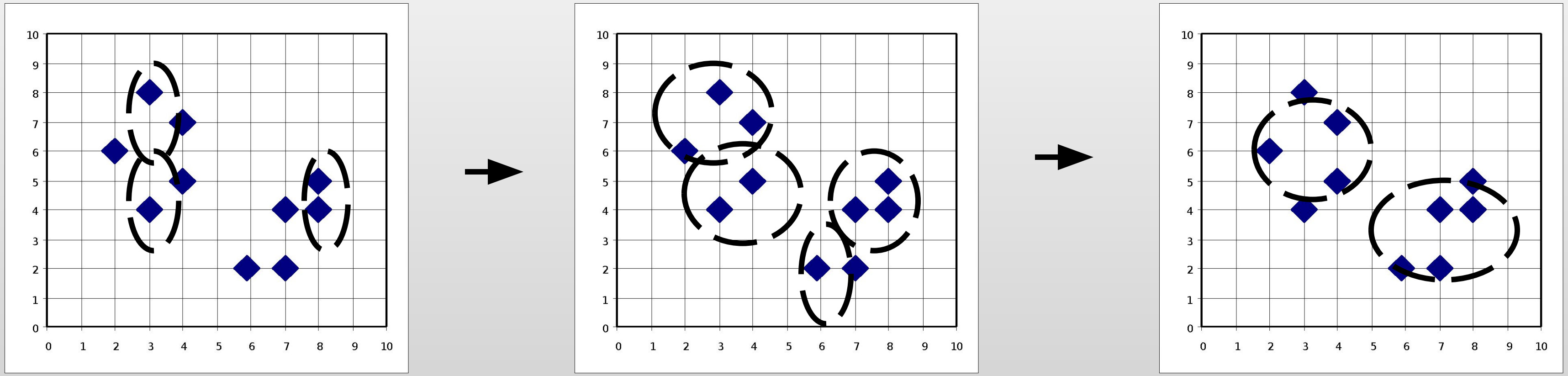
Hierarchical Clustering

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

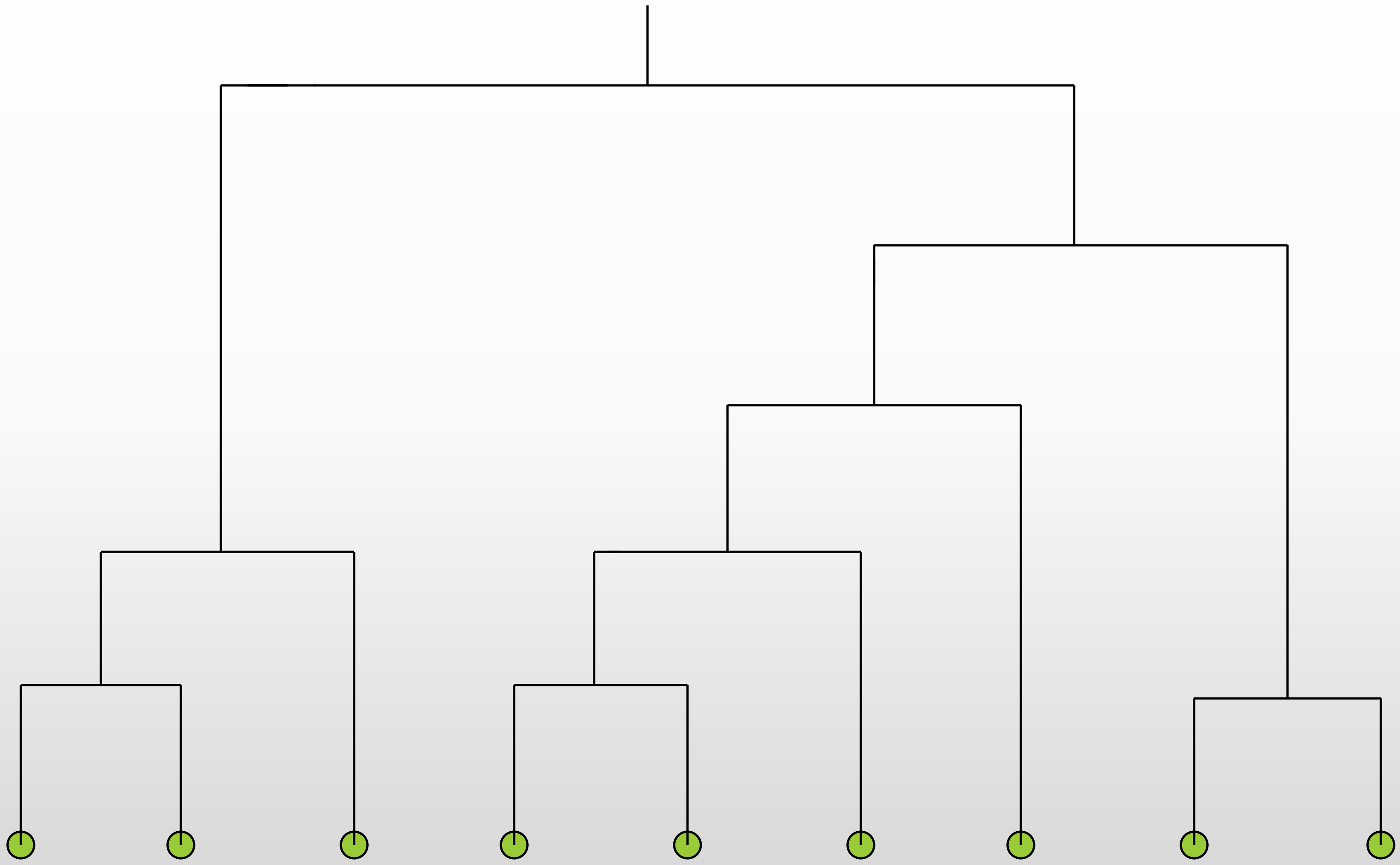


AGNES (Agglomerative Nesting)

- 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



Dendrogram: Shows How the Clusters are Merged



Agglomerative Clustering Example

:Data Objects

Rec No	X1	X2	X3	X4	X5
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	1	1
4	1	0	1	0	1
5	0	1	0	1	0
6	1	0	1	1	1
7	1	1	1	1	1

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$$

Dissimilarity Matrix (Euclidean :(Distances

Rec No	Dist1	Dist2	Dist3	Dist4	Dist5	Dist6	Dist7
1							
2	0.000						
3	2.236	2.236					
4	1.732	1.732	1.414				
5	1.414	1.414	1.732	2.236			
6	2.000	2.000	1.000	1.000	2.000		
7	2.236	2.236	0.000	1.414	1.732	1.000	

More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale 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
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as efficient clustering in big data environment

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD’99.
- Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB’98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

References (2)

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB’98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD’98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB’98.

References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB’98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles, ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, SIGMOD’02
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB’97
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96
- X. Yin, J. Han, and P. S. Yu, “LinkClus: Efficient Clustering via Heterogeneous Semantic Links”, VLDB'06