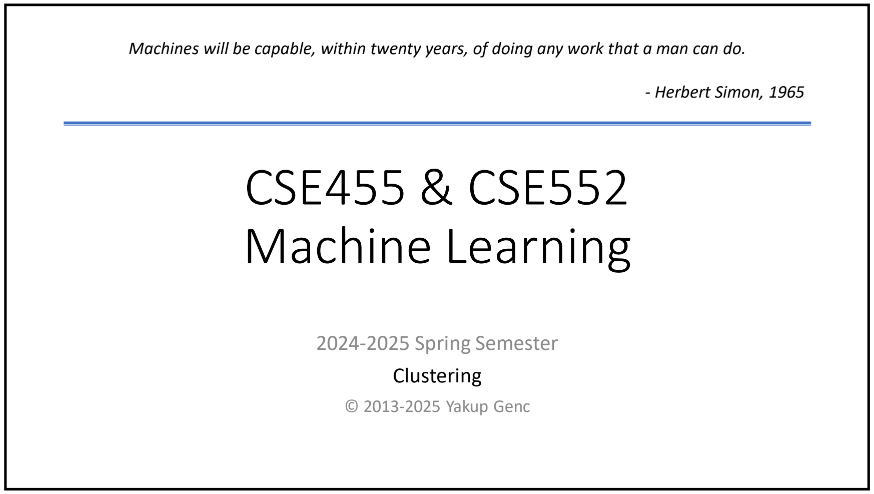


*Machines will be capable, within twenty years, of doing any work that a man can do.*

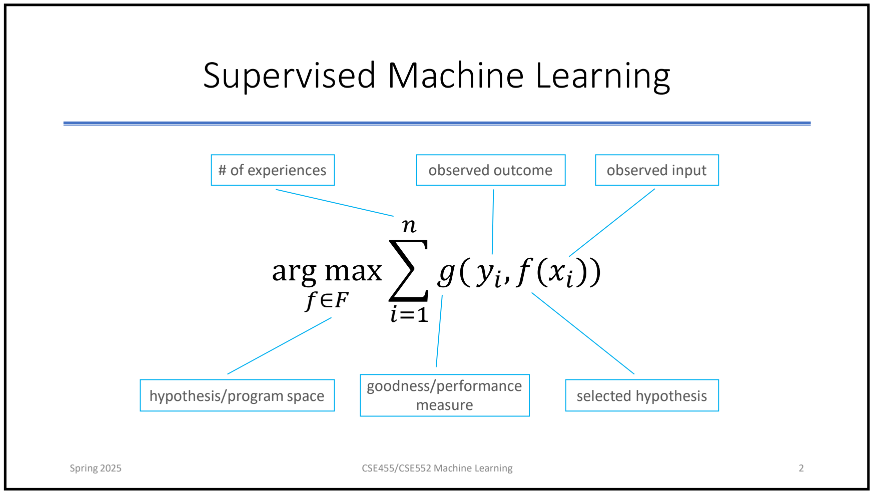
*- Herbert Simon, 1965*

# CSE455 & CSE552 Machine Learning

2024-2025 Spring Semester

Clustering

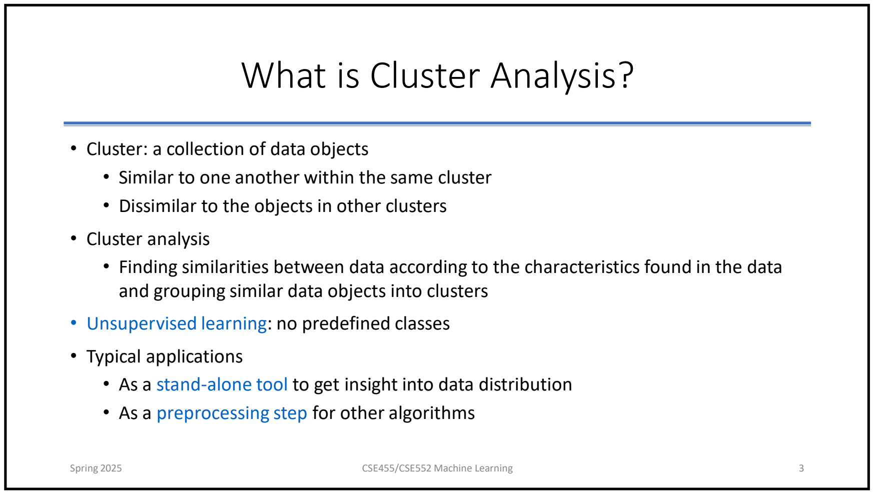
1



## Supervised Machine Learning

$$\arg\max_{f \in F} \sum_{i=1}^{n} g(y_i, f(x_i))$$

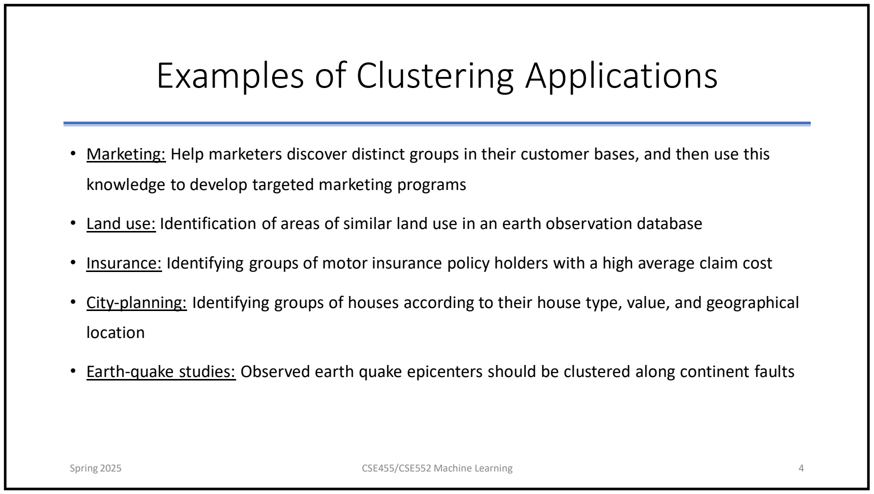
2



## 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
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

3



## Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

4

## Quality: 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

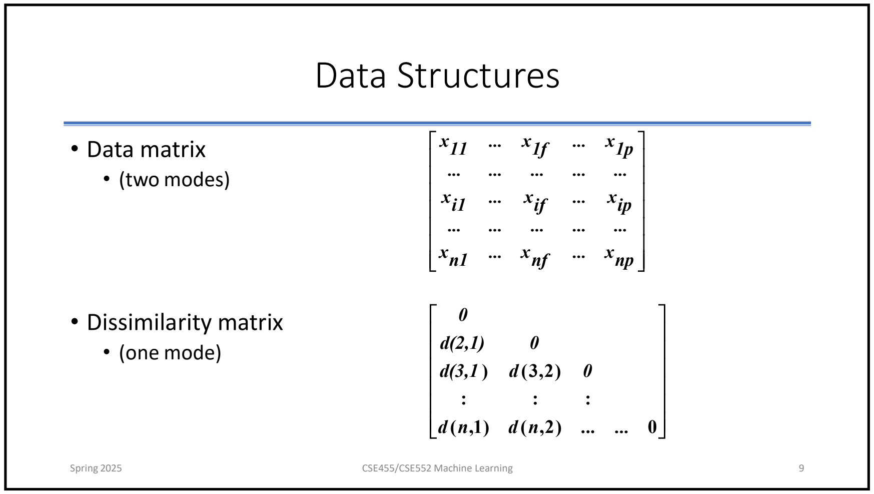
- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- 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.

## Requirements of Clustering

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- 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

## Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis ←
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



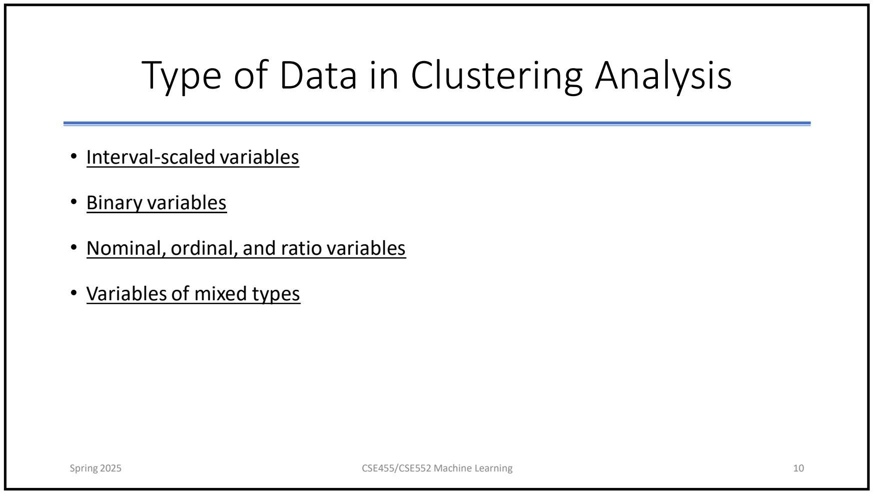
## Data Structures

- Data matrix
  - (two modes)

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

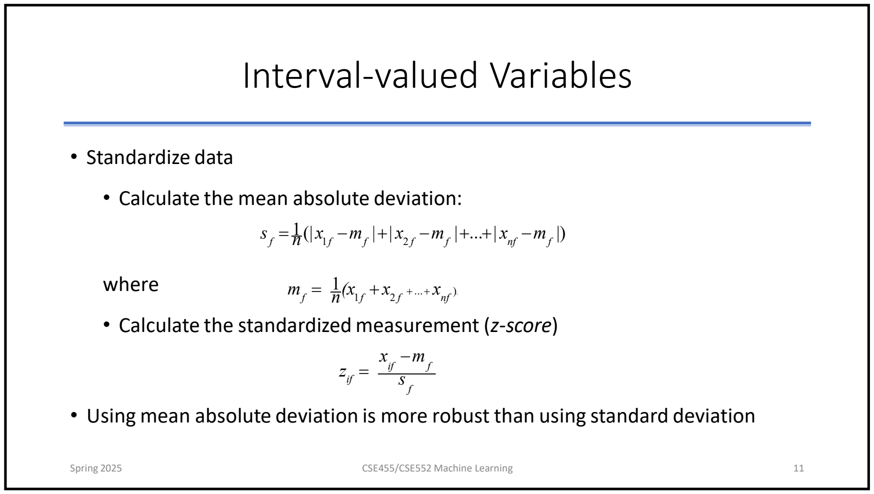
- Dissimilarity matrix
  - (one mode)

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



## Type of Data in Clustering Analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types



## Interval-valued Variables

- Standardize data
  - Calculate the mean absolute deviation:

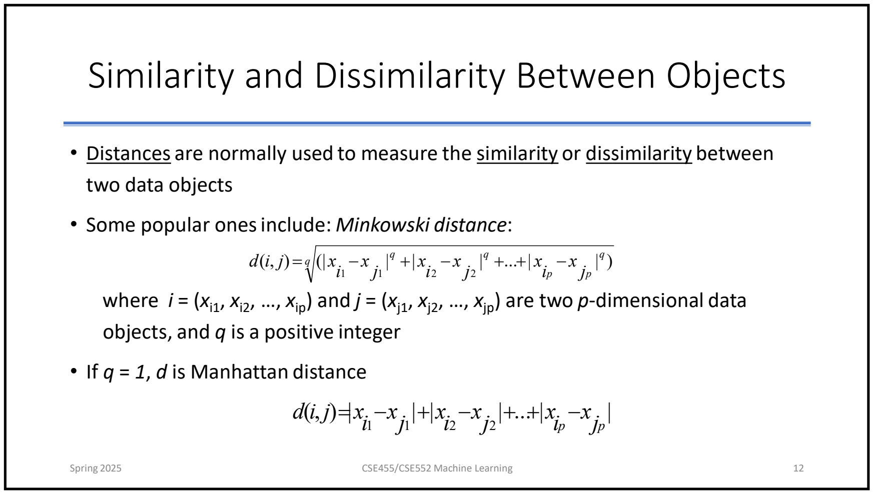
$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf})$

  - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation



## Similarity and Dissimilarity Between Objects

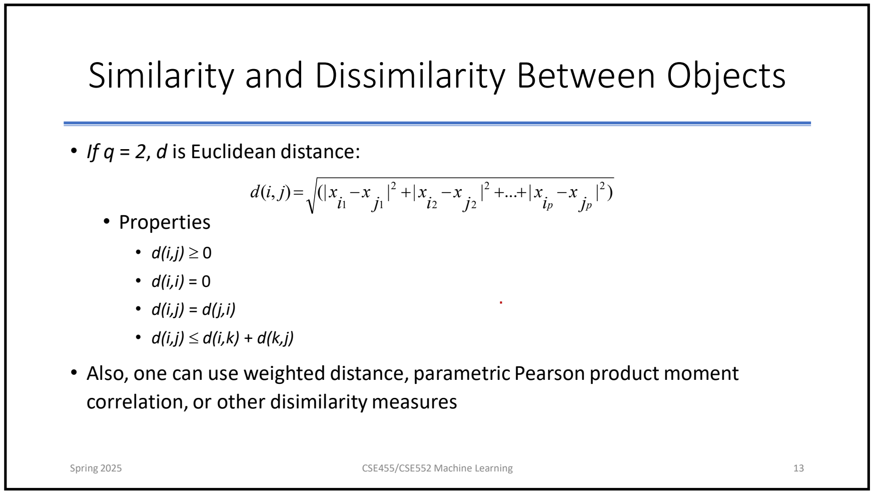
- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)}$$

  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

- If $q = 1$, $d$ is Manhattan distance

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

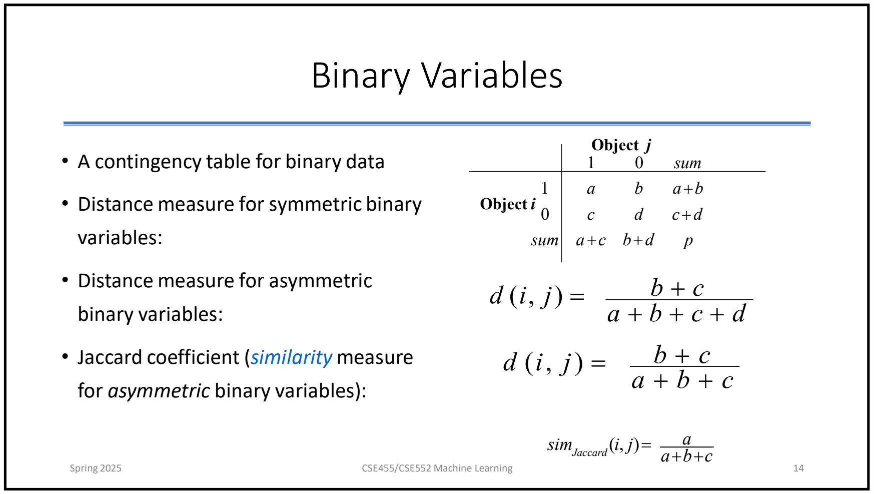


## Similarity and Dissimilarity Between Objects

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

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

  - Properties
    - *d(i,j)* ≥ 0
    - *d(i,i)* = 0
    - *d(i,j)* = *d(j,i)*
    - *d(i,j)* ≤ *d(i,k)* + *d(k,j)*
- Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures



## Binary Variables

- A contingency table for binary data

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

- Distance measure for symmetric binary variables:

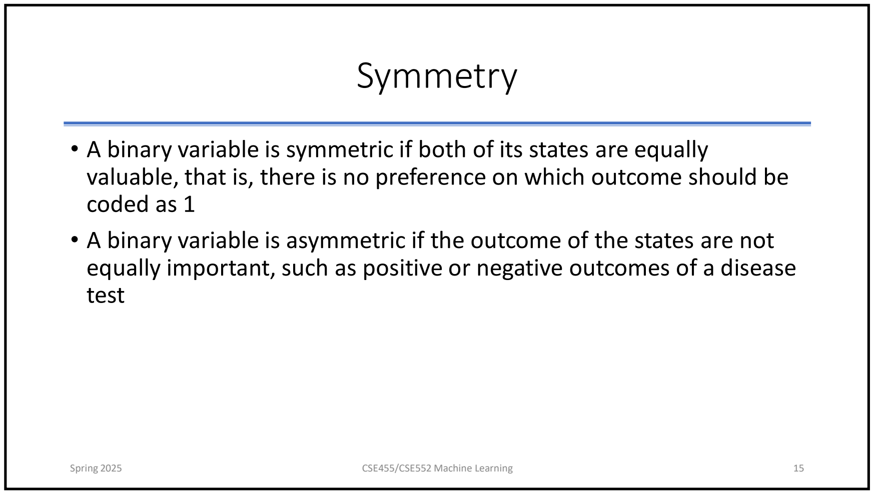
$$d(i,j)=\frac{b+c}{a+b+c+d}$$

- Distance measure for asymmetric binary variables:

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

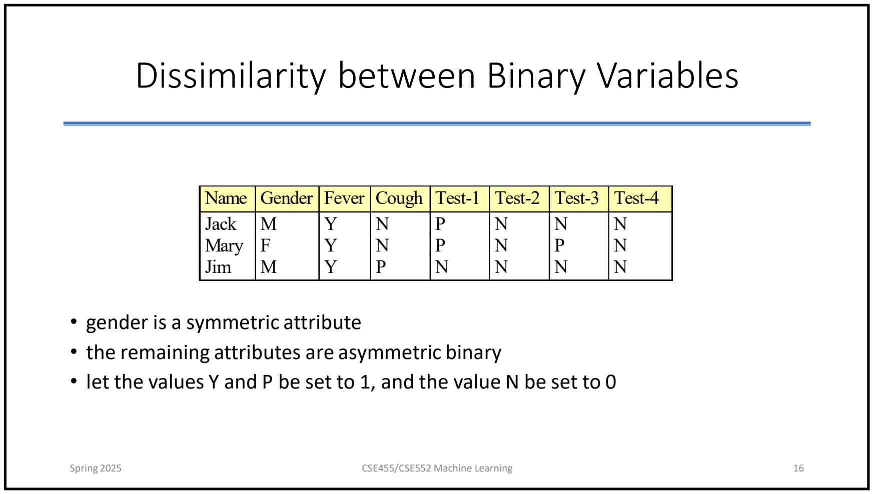
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

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



## Symmetry

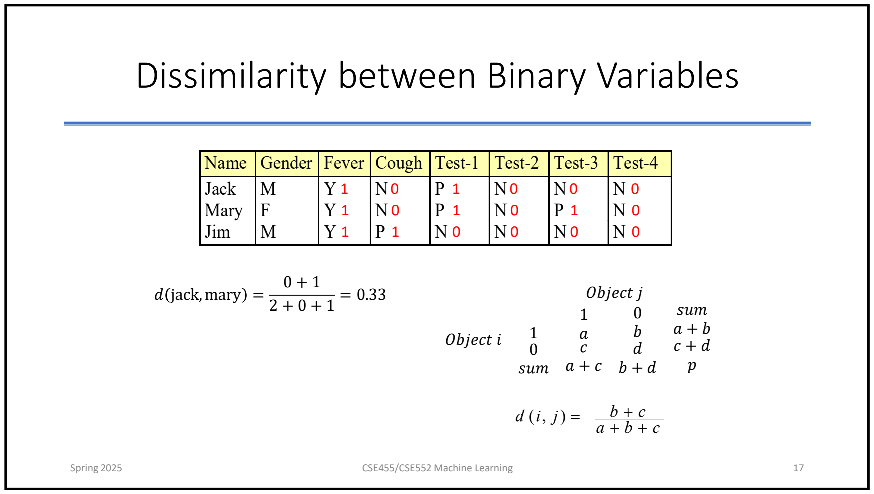
- A binary variable is symmetric if both of its states are equally valuable, that is, there is no preference on which outcome should be coded as 1
- A binary variable is asymmetric if the outcome of the states are not equally important, such as positive or negative outcomes of a disease test



## Dissimilarity between Binary Variables

| 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



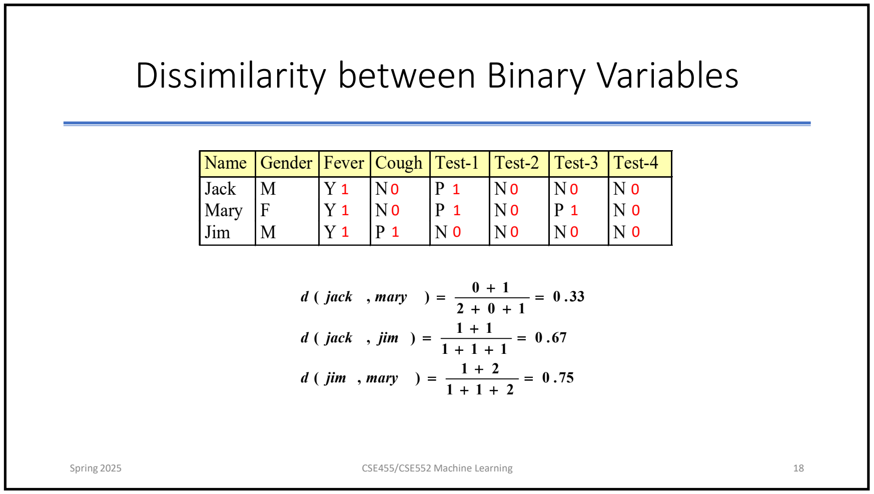
## Dissimilarity between Binary Variables

| Name | Gender | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|---|---|---|---|---|---|---|---|
| Jack | M | Y 1 | N 0 | P 1 | N 0 | N 0 | N 0 |
| Mary | F | Y 1 | N 0 | P 1 | N 0 | P 1 | N 0 |
| Jim | M | Y 1 | P 1 | N 0 | N 0 | N 0 | N 0 |

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

$$\begin{array}{cc|ccc} & & \multicolumn{3}{c}{Object\ j} \\ & & 1 & 0 & sum \\ \hline Object\ i & 1 & a & b & a+b \\ & 0 & c & d & c+d \\ & sum & a+c & b+d & p \end{array}$$

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



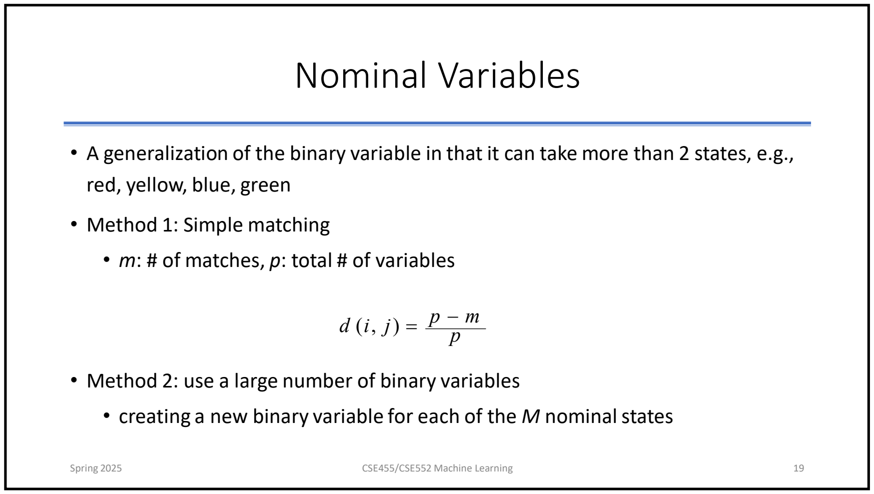
## Dissimilarity between Binary Variables

| Name | Gender | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|---|---|---|---|---|---|---|---|
| Jack | M | Y 1 | N 0 | P 1 | N 0 | N 0 | N 0 |
| Mary | F | Y 1 | N 0 | P 1 | N 0 | P 1 | N 0 |
| Jim | M | Y 1 | P 1 | N 0 | N 0 | N 0 | N 0 |

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

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

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

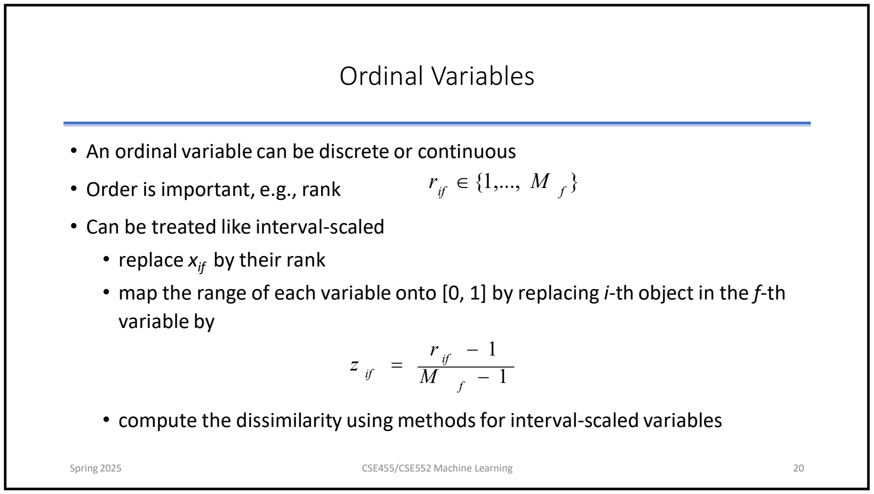


## Nominal Variables

- 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 $r_{if} \in \{1, \ldots, M_f\}$
- Can be treated like interval-scaled
  - replace $x_{if}$ by their rank
  - 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

## Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as $Ae^{Bt}$ or $Ae^{-Bt}$
- 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

21

## 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 $d_{ij}^{(f)} = 1$ otherwise
  - $f$ 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

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

22

## Vector Objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biologic taxonomy, etc.
- Cosine measure

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{|\vec{X}||\vec{Y}|},$$

$\vec{X}^t$ is a transposition of vector $\vec{X}$, $|\vec{X}|$ is the Euclidean normal of vector $\vec{X}$,

- A variant: Tanimoto coefficient

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

23

## Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

24

## 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, ROCK, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue

## Major Clustering Approaches (II)

- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE
- 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: pCluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering

## Calculating 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

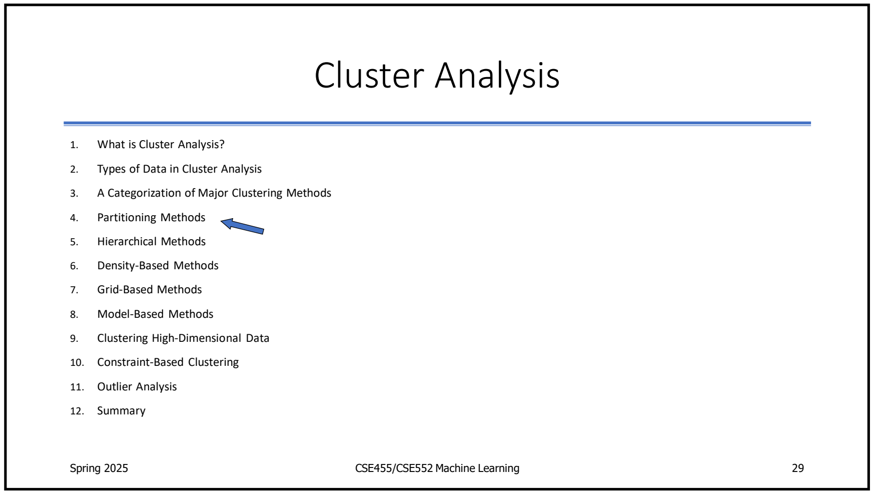
$$C_m = \frac{\Sigma_{i=1}^{N}(t_{ip})}{N}$$

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

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

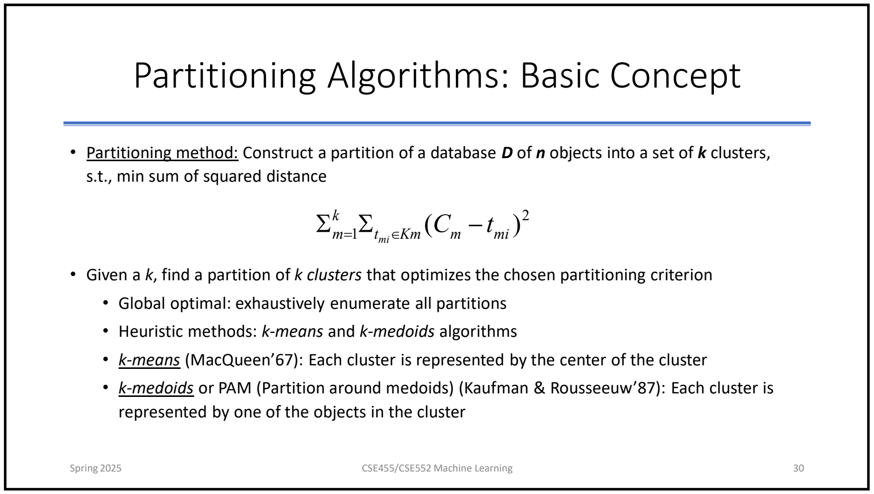
- Diameter: square root of average mean squared distance between all pairs of points in the cluster

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



## Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

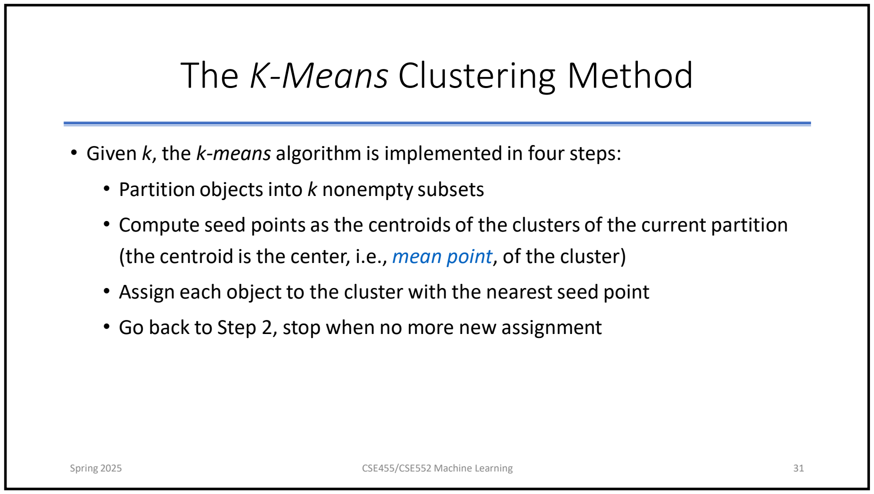


## Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, s.t., min sum of squared distance

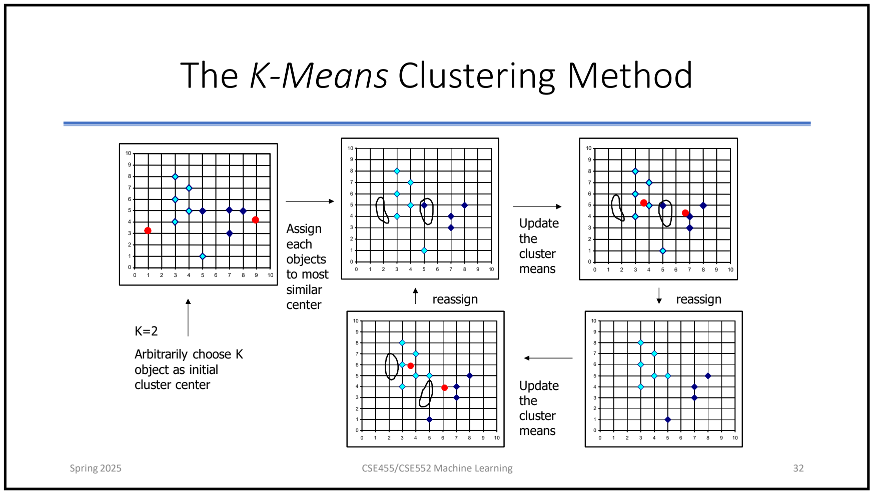
$$\Sigma_{m=1}^{k}\Sigma_{t_{mi}\in Km}(C_m - t_{mi})^2$$

- 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 four 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, 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 no more new assignment



## The *K-Means* Clustering Method

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

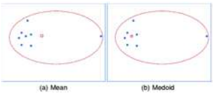
- Strength: *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
    - Comparing: PAM: O(k(n-k)$^2$ ), CLARA: O(ks$^2$ + k(n-k))
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
    - 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*

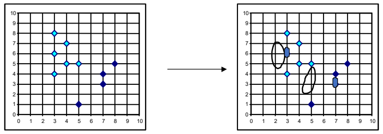
## Variations of the *K-Means* Method

- 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

## What Is the Problem of the 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.





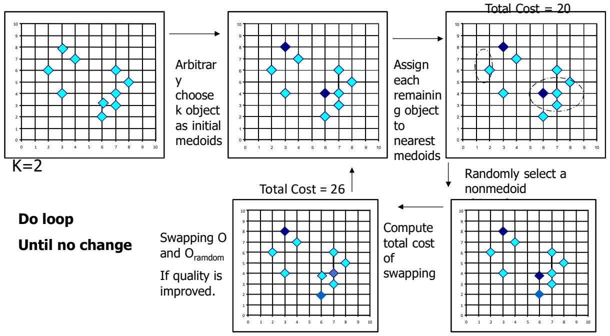
## Example: Hand Image vs Background

## Example: Text Documents

## 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
- Focusing + spatial data structure (Ester et al., 1995)

## A Typical K-Medoids Algorithm (PAM)



K=2

Arbitrary choose k object as initial medoids

Assign each remaining object to nearest medoids

Total Cost = 20

Randomly select a nonmedoid

Compute total cost of swapping

Total Cost = 26

Swapping O and $O_{ramdom}$

If quality is improved.

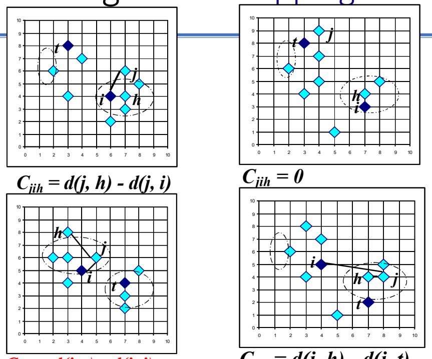
**Do loop**

**Until no change**

## PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost $TC_{ih}$
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih}$ < 0, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

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



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

$C_{jih} = 0$

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

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

# What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - O(k(n-k)$^2$ ) for each iteration

    where n is # of data,k is # of clusters

➔Sampling based method,

CLARA(Clustering LARge Applications)

# *CLARA* (Clustering Large Applications)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

# *CLARANS* ("Randomized" CLARA)

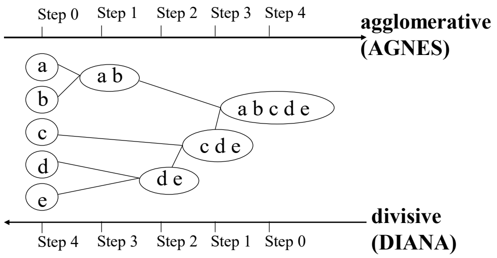
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*
- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95)

# Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. 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



Step 0 Step 1 Step 2 Step 3 Step 4

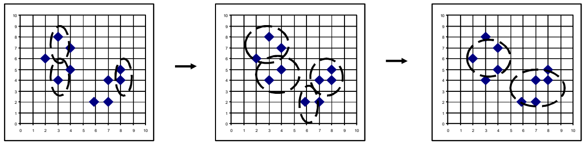
**agglomerative (AGNES)**

Step 4 Step 3 Step 2 Step 1 Step 0

**divisive (DIANA)**

# AGNES (Agglomerative Nesting)

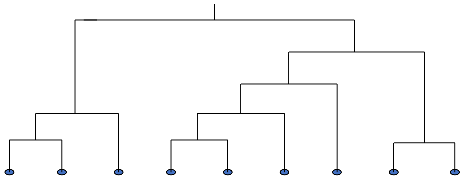
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster

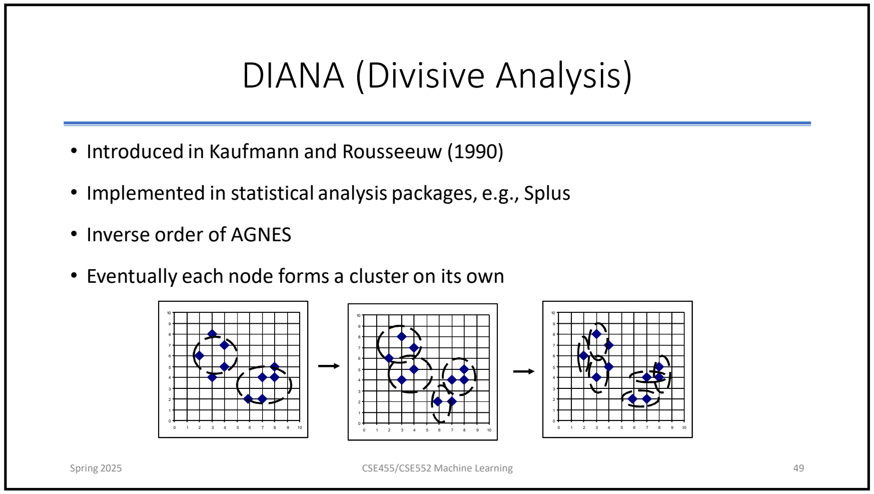


# Dendrogram: Shows How the Clusters are Merged

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

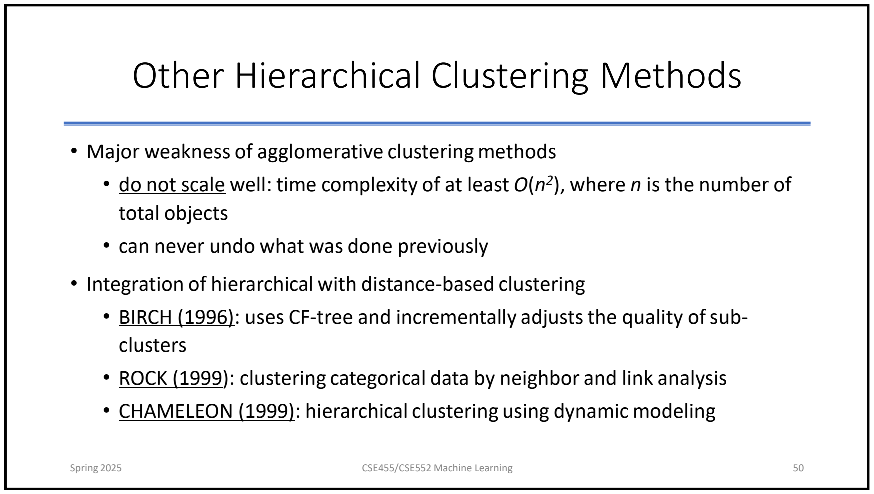
A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.





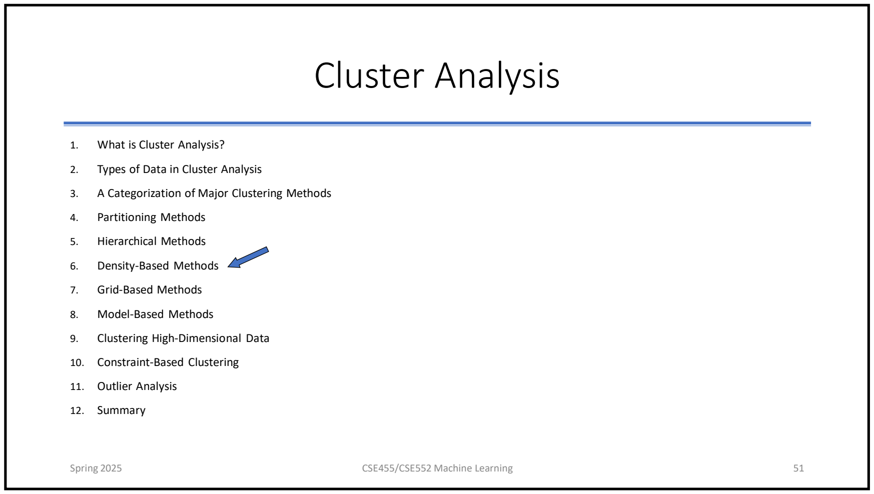
## DIANA (Divisive Analysis)

- 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



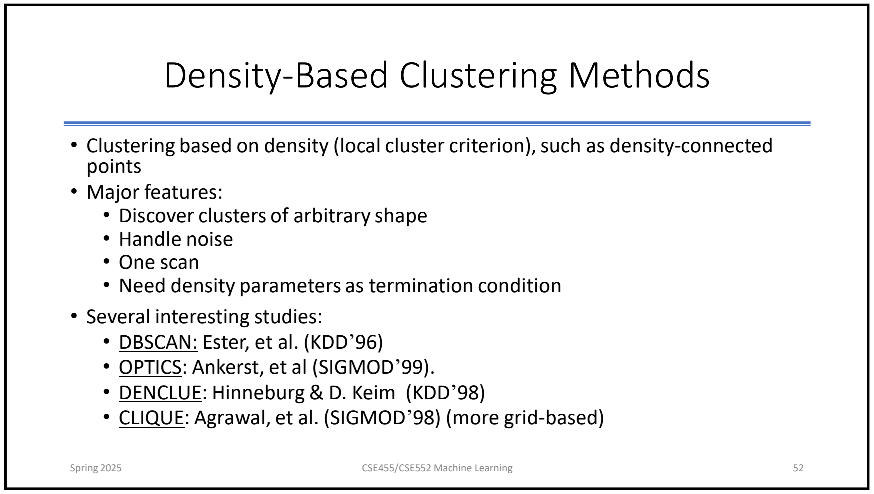
## Other Hierarchical Clustering Methods

- 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
  - ROCK (1999): clustering categorical data by neighbor and link analysis
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling



## Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



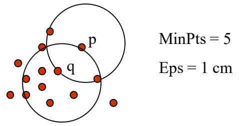
## Density-Based Clustering Methods

- 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) (more grid-based)

## Density-Based Clustering: Basic Concepts

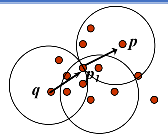
- Two parameters*:*
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: *{q belongs to D | dist(p,q) <= Eps}*
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*,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

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

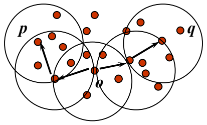


## Density-Reachable and Density-Connected

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *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 $p_i$

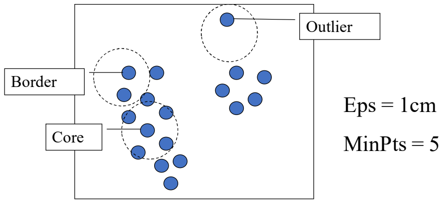


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



## 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



## DBSCAN: The Algorithm

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *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.

## DBSCAN: Sensitive to Parameters

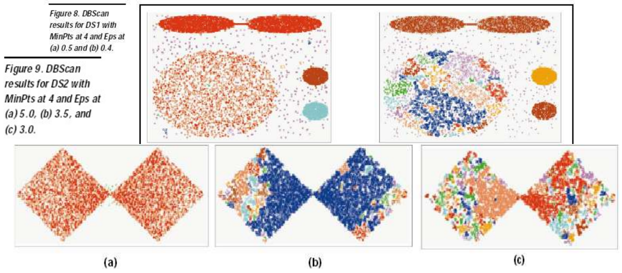


*Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.*

*Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.*

## Thanks for listening!