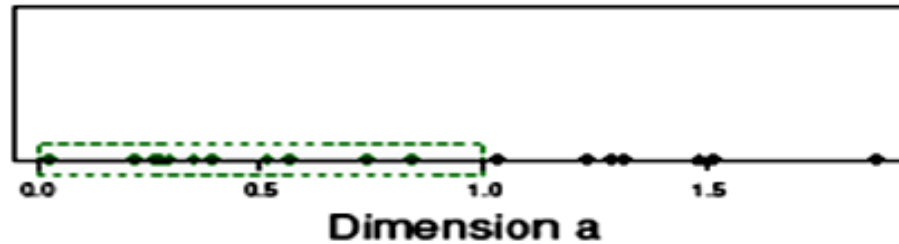


Chapter 3: Cluster Analysis

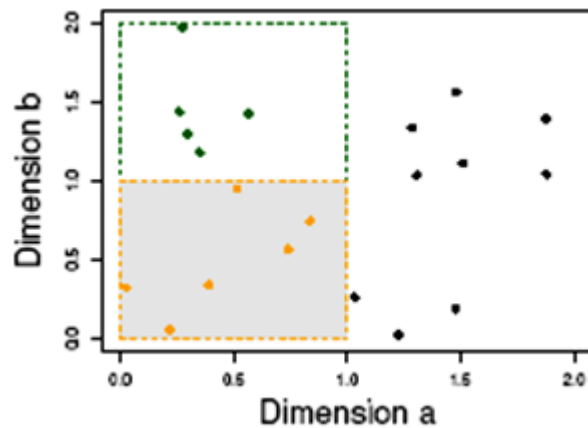
- ▶ 3.1 Basic Concepts of Clustering
- ▶ 3.2 Partitioning Methods
- ▶ 3.3 Hierarchical Methods
- ▶ 3.4 Density-based Methods
- ▶ 3.5 Clustering High-Dimensional Data
 - 3.5.1 Curse of Dimensionality
 - 3.5.2 Attribute Subset Selection
 - 3.5.3 Subspace Clustering
 - 3.5.4 CLIQUE
 - 3.5.5 Frequent Pattern-based Clustering
- ▶ 3.6 Outlier Analysis

3.5.1 The Curse of Dimensionality

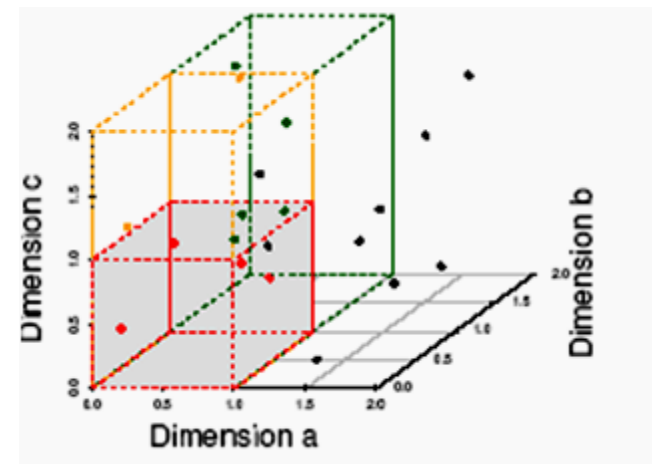
- ▶ Let's take an example of one dimensional data



We add a second dimension

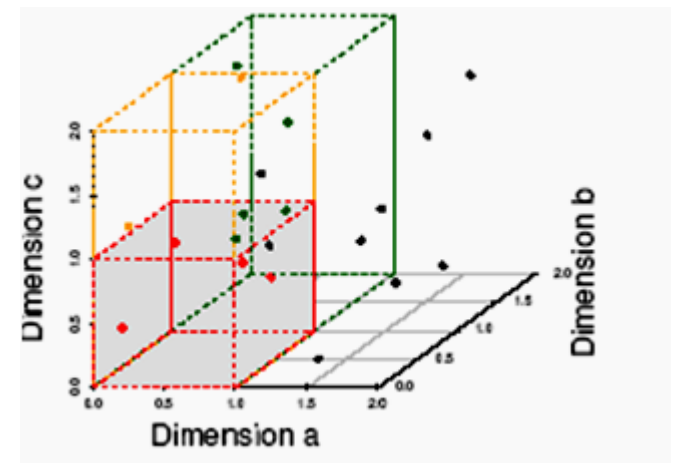
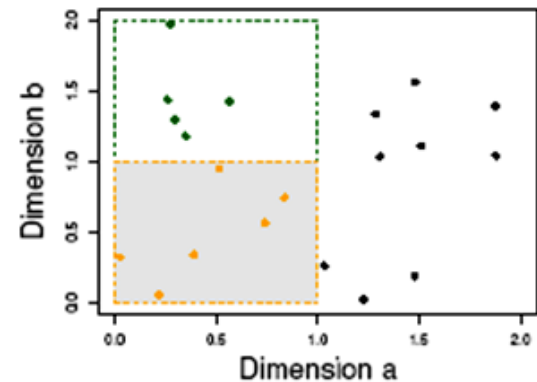
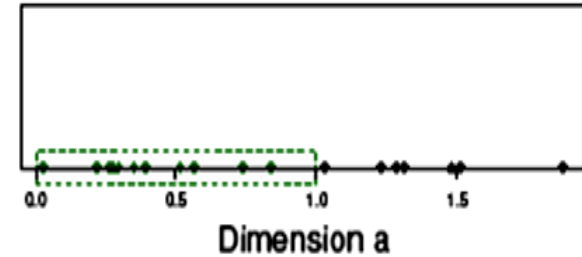


We add a third dimension



The Curse of Dimensionality

- ▶ Data in only one dimension is relatively packed
- ▶ Adding a dimension “stretch” the points across that dimension, making them further apart
- ▶ Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- ▶ Distance measure becomes meaningless—due to equi-distance



Handling High Dimensionality

▶ Feature Transformation

- Transform the data onto a small space while generally preserving the original relative distance between objects
- Do not remove any of the original attributes
- Irrelevant information may mask the real clusters
- Difficult to interpret the resulting transformed attributes

▶ Feature Subset Selection

- Remove irrelevant or redundant features
- Find a subset of features that are relevant
- Evaluate subsets of features using certain criteria

3.5.2 Attribute Subset Selection

- ▶ Some attributes can be irrelevant to the mining task
- ▶ **Example**
 - Classify customers whether or not they are likely to purchase a popular new CD
 - Attributes such as customers's **phone number** are likely to be **irrelevant** unlike attributes such as age and music-taste
- ▶ **First approach**: Manual selection of attributes (by experts)
 - Difficult
 - Time consuming
 - The behavior of the data is not always well known
 - Leaving out relevant attributes and keeping irrelevant ones cause confusion
- ▶ **Second approach**: do attribute subset selection

Attribute Subset Selection (ASS)

Idea

- ▶ Find a minimum set of attributes such as the resulting probability distribution of the data classes is as close as possible to the original distribution
- ▶ Mining on a reduced set of attributes as an additional benefit
- ▶ It reduces the number of attributes appearing in discovered patterns
- ▶ Helps making the patterns easier to understand
- ▶ How can we find a “good” subset of attributes?

Attribute Subset Selection (ASS)

Idea

- ▶ For n attributes, there are 2^n possible subsets
- ▶ Exhaustive search for optimal subset of attributes can be very expensive
- ▶ Heuristic methods are needed to reduce the search space
- ▶ Use greedy methods that look for the best choice at the time
- ▶ “Best” and “Worse” attributes can be determined using
 - Statistical significance (assume independence between attributes)
 - Use evaluation measures such as information gain
 - ...

Basic Heuristic Methods

▶ Stepwise forward selection

- Start with an empty set of attributes as the reduced set
- The best of the original attributes is selected and added to the reduced set
- Iterate until a stopping condition is satisfied

Initial Attribute set
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$

Initial reduced set:

$\{\}$

$\Rightarrow \{A_1\}$

$\Rightarrow \{A_1, A_4\}$

\Rightarrow Reduced attribute set $\{A_1, A_4, A_6\}$

Basic Heuristic Methods

▶ Stepwise backward elimination

- Start with the full set of attributes
- At each step, remove the worst attribute remaining in the set
- Iterate until a stopping condition is satisfied

Initial Attribute set
 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$

⇒ $\{A_1, A_3, A_4, A_5, A_6\}$

⇒ $\{A_1, A_4, A_5, A_6\}$

⇒ Reduced attribute set $\{A_1, A_4, A_6\}$

Basic Heuristic Methods

▶ Combination of forward selection and backward elimination

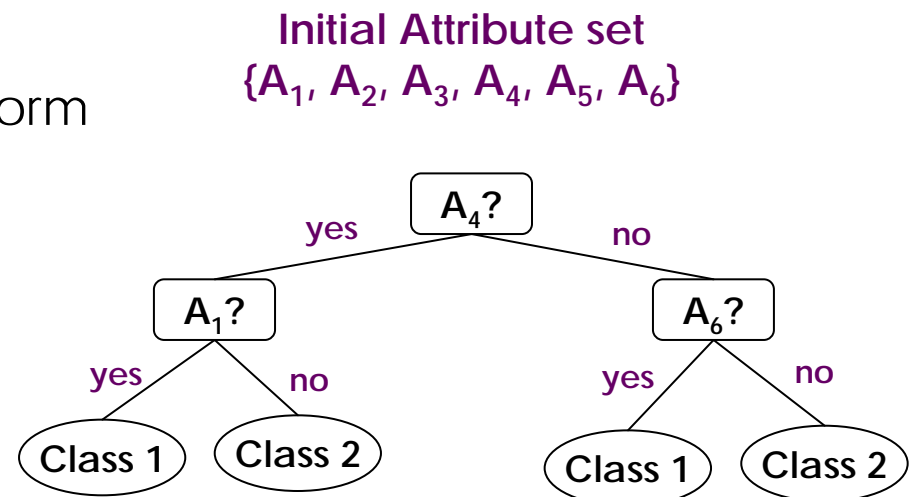
→ At each step select the best attribute and removes the worst from the remaining attributes

→ Decision Tree Induction

→ Construct a decision tree

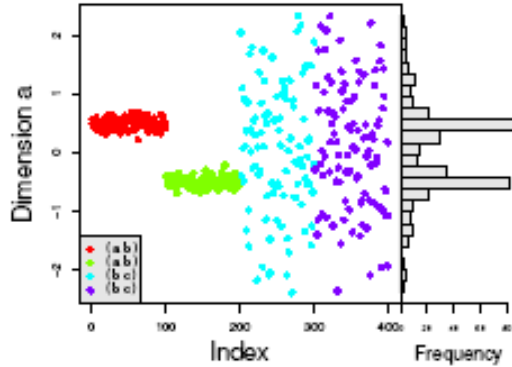
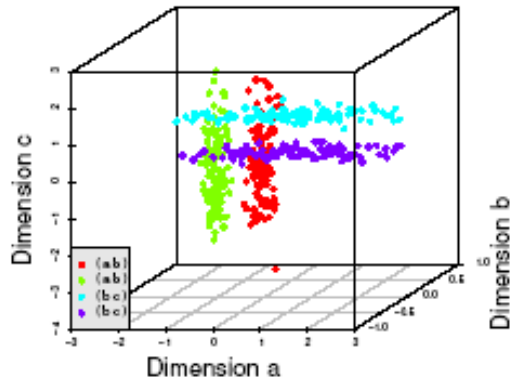
→ The attributes used in the tree form the reduced set

▶ Most of subset selection methods are based on supervised learning

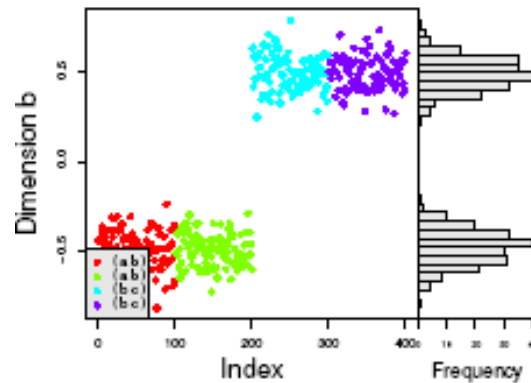


Subspace Clustering

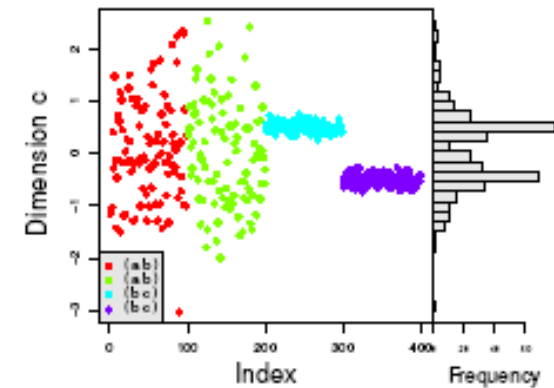
- ▶ Extension to attribute selection
- ▶ Clusters may exist only in some subspaces
- ▶ **Subspace-clustering**: find clusters in all the subspaces



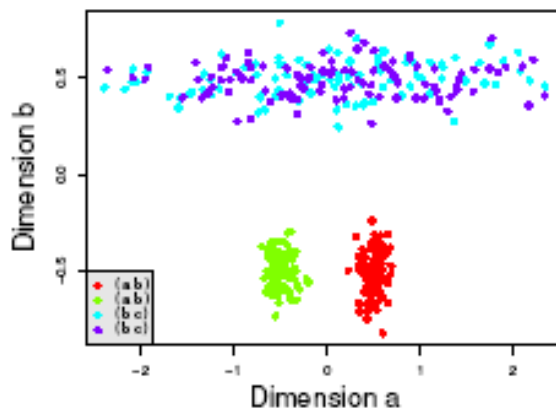
(a) Dimension a



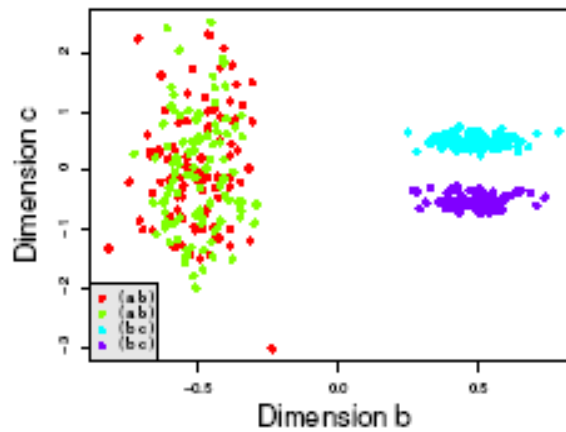
(b) Dimension b



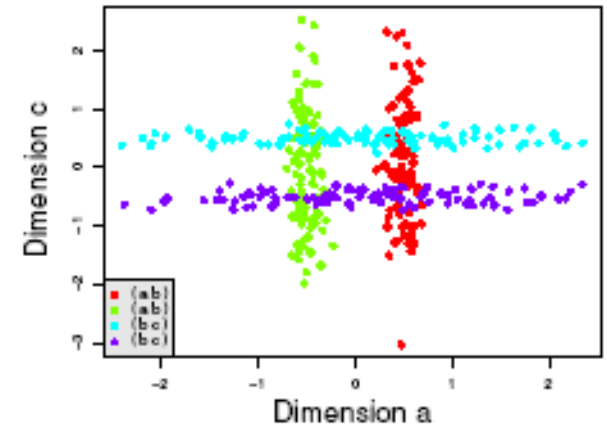
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c

Subspace Clustering

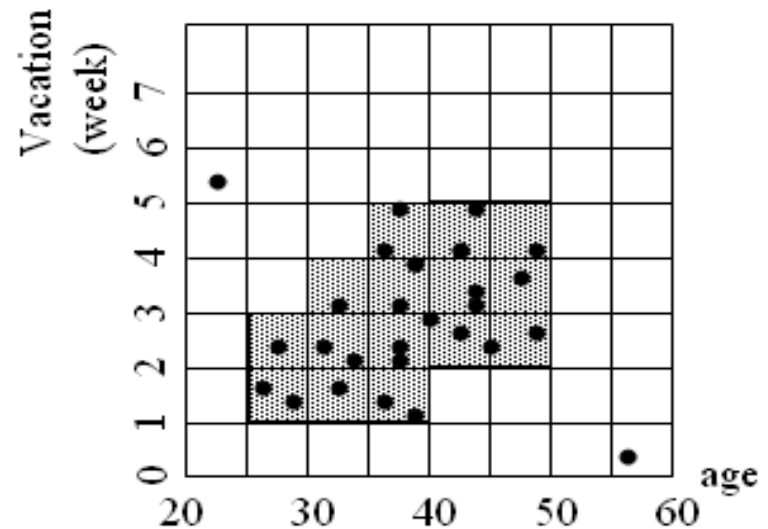
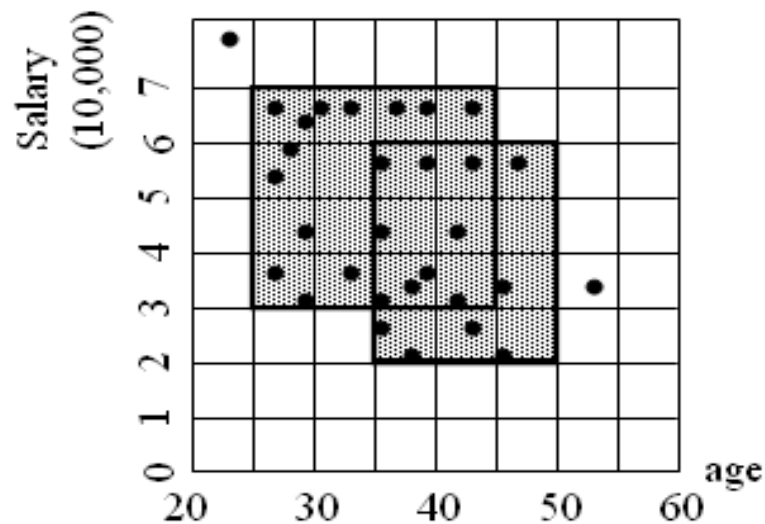
- ▶ How to find subspace clusters effectively and efficiently?
- ▶ We are going to see two approaches
 - Dimension-growth subspace clustering
 - Frequent pattern-based clustering

3.5.4 CLIQUE

- ▶ **CLIQUE (CLustering in QUES)** was the first algorithm proposed for dimension **growth subspace clustering** in high-dimensional space
- ▶ Start at single-dimensional subspaces and grow upward to higher dimensional ones
- ▶ CLIQUE partitions each dimension like a grid structure and determines whether a cell is dense based on the number of points it contains
- ▶ CLIQUE is an integration of grid-based and density-based methods

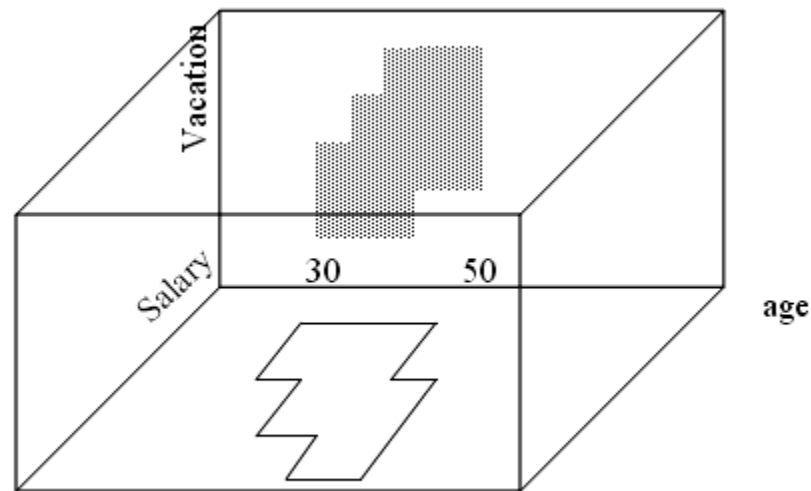
CLIQUE

- Partition the d-dimensional data space into non overlapping rectangular units (done in 1-D for each partition)
- Identify dense units
- A unit is dense if the fraction of total data points contained in it exceeds an input model parameter



CLIQUE

- The subspaces representing dense regions are intersected to form a **candidate** search space in which dense units of higher dimensionality may exist



- Why does CLIQUE confine its search for dense units of higher dimensionality to the intersection of the dense units in the subspaces?

CLIQUE

- ▶ The property adapted by CLIQUE states:
 - If a k -dimensional unit is dense, then so are its projections in $(k-1)$ dimensional space
- ▶ Generate potential or candidate dense units in k -dimensional space from dense units found in $(k-1)$ dimensional space
- ▶ The resulting space searched is much smaller than the original space
- ▶ The dense units are then examined to determine clusters

CLIQUE

▶ Strength

- Automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- Insensitive to the order of records in input and does not presume some canonical data distribution
- Scales linearly with the size of input and has good scalability as the number of dimensions in the data increases

▶ Weakness

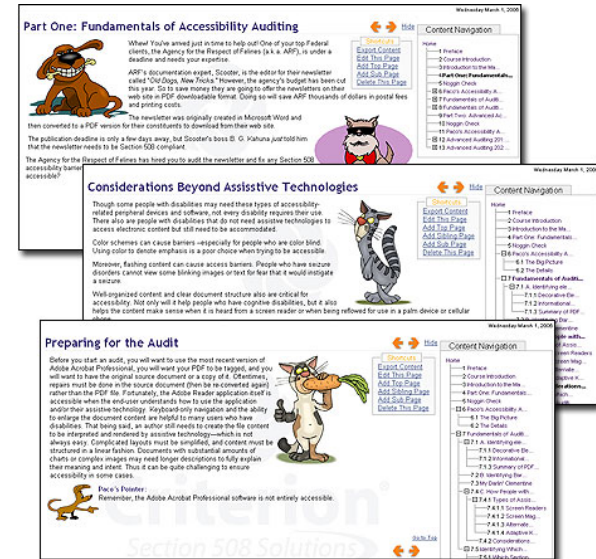
- The accuracy of the clustering result may be degraded at the expense of simplicity of the method

3.5.5 Frequent Pattern-based Clustering

- ▶ **Frequent pattern** mining leads to the discovery of interesting associations and correlations among data objects
- ▶ The frequent patterns discovered may also indicate clusters
- ▶ Well suited for high dimensional data, **however boundaries of different dimensions are not obvious**
 - Rather than growing clusters dimension by dimension, we grow sets of frequent items
 - Lead to clusters descriptions

Example: Frequent term-based text

- ▶ Documents contain terms
- ▶ Extract terms
 - Parsing
 - Stemming
- ▶ Each document can be represented as a set of terms
- ▶ Consider each term as a dimension
- ▶ The dimension space will be very high
- ▶ The dimension space can be referred as: **term vector space**



Example: Frequent term-based text

- ▶ Documents are clustered based on the frequent terms they contain
- ▶ Consider only the low-dimensional frequent term sets as “cluster candidates”
- ▶ Frequent term set is not a cluster but a description of a cluster
- ▶ A cluster consists of documents containing all the terms of the frequent term set

News, education
, sport



Cluster1

Science, Computer



Cluster2

Example: Frequent term-based text

- ▶ How to select a good subset of the set of all frequent term sets?
- ▶ Let
 - F_i be a set of frequent term sets
 - $\text{Cov}(F_i)$ be the set of documents covered by F_i
- ▶ Find a well-selected subset F_1, F_2, \dots, F_k , of all frequent term sets

▶ Principle

- (1) the selected subset should cover all the documents to be clustered

$$\sum_{i=1}^k \text{cov}(F_i) = D$$

- (2) the overlap between any two partitions F_i and F_j for ($i \neq j$) should be minimized (e.g., using entropy)
- ▶ This approach automatically generates cluster description, In traditional methods, an additional step is required to describe the resulting clusters.

Chapter 3: Cluster Analysis

- ▶ 3.1 Basic Concepts of Clustering
- ▶ 3.2 Partitioning Methods
- ▶ 3.3 Hierarchical Methods
- ▶ 3.4 Density-Based Methods
- ▶ 3.5 Clustering High-Dimensional Data
- ▶ 3.6 Outlier Analysis

3.6.1 Definition

3.6.2 Statistical-Based Methods

3.6.3 Distance-Based Methods

3.6.4 Density-Based Local Methods

3.6.5 Deviation-Based Methods

3.6.1 Definition

- ▶ **Outliers:** data objects that do not comply with the general behavior or model of the data
- ▶ Outlier detection or analysis is referred to as **Outlier Mining**
- ▶ Outlier mining has different applications
 - Fraud detection
 - Detecting unusual usage of telecommunication services
 - Identifying the spending behavior of costumers with extremely low or extremely high incomes
 - Finding unusual responses to various medical treatments
 - Etc.

Outlier Mining

- ▶ Given a set of n data objects and k expected number of outliers
- ▶ Find the **top k objects** that are considerably
 - **Dissimilar**
 - **Exceptional**
 - **Inconsistent** with respect to the remaining data
- ▶ The outlier mining problem can be seen as two sub-problems
 - **1)** Define what data can be considered as inconsistent in a given data set
 - **2)** Find an efficient method to mine the outliers so defined
- ▶ Data visualization methods are weak in detecting data with many categorical attributes or data of high dimensionality
- ▶ Investigate computer-based techniques to detect outliers

3.6.2 Statistical Distribution-Based Methods

- ▶ Assume a distribution model for the given data set (e.g., Normal)
- ▶ Identify outliers w. r. t the model using a **discordancy test**
- ▶ **How does it work?**

- ▶ Examine two hypothesis
 - **working hypothesis**
 - **alternative hypothesis**

- ▶ A working hypothesis **H** is a statement that the entire data set of n objects comes from an initial distribution model **F** that is:

$$H: o_i \in F, \quad \text{where } i=1,2,\dots,n$$

- ▶ The hypothesis **H is retained** if there is **no** statistically significant **evidence** supporting its rejection

Discordancy Test

- ▶ Verifies whether an object o_i is significantly large (or small) in relation to the distribution F
- ▶ **Principle**
 - Choose some statistic T for discordancy testing
 - The value of the statistic for object o_i is v_i
 - If **significance probability** $SP(v_i) = P(T > v_i)$ is **sufficiently small**
 - o_i is discordant
 - The working hypothesis is rejected
 - An **alternative hypothesis** $\neg H$ which says that o_i comes from a another distribution model G is **adopted**
- ▶ The result depends on how the model F is chosen because o_i may be an outlier under one model and perfectly valid value under another

Discordancy Test: Example

- ▶ Let o_1, \dots, o_n represent the data objects
- ▶ Compute the sample mean μ and the standard deviation σ
- ▶ If an object o_i is suspected to be an outlier
 - Compute the test statistic T

$$T = \frac{|\mu - o_i|}{\sigma}$$

- If T exceeds some critical value, then o_i is an outlier

Alternative Distributions

Inherent Alternative Distribution

- ▶ The working hypothesis that all objects come from distribution **F** is rejected
- ▶ Alternative hypothesis assume that all objects come from another distribution **G**

$$\neg H: o_i \in G, \quad \text{where } i=1,2,\dots,n$$

- ▶ **F** and **G**: different distributions
- ▶ **F** and **G** : the same distribution but with different parameters
- ▶ Distribution **G** must have the potential to produce outliers (a different mean, or dispersion, or a longer tail)

Alternative Distributions

Mixture Alternative Distribution

- ▶ The discordant values are not outliers in F population but contaminants from some other population G
- ▶ The alternative hypothesis is

$$\neg H: o_i \in (1-\lambda) F + \lambda G, \quad \text{where } i=1,2,\dots,n$$

Slippage Alternative Distribution

- ▶ All objects (except a small number) are from initial model F, with its given parameters
- ▶ The remaining objects are from a modified version of F in which the parameters have been shifted

Characteristics of Statistical-Based Methods

- ▶ Tests are for single attributes
- ▶ Need to find outliers in multidimensional space
- ▶ Statistical approaches require knowledge about parameters of the data set
- ▶ Statistical methods do not guarantee that all outliers will be found
 - No specific test was developed
 - The distribution cannot be adequately modeled with any standard distribution

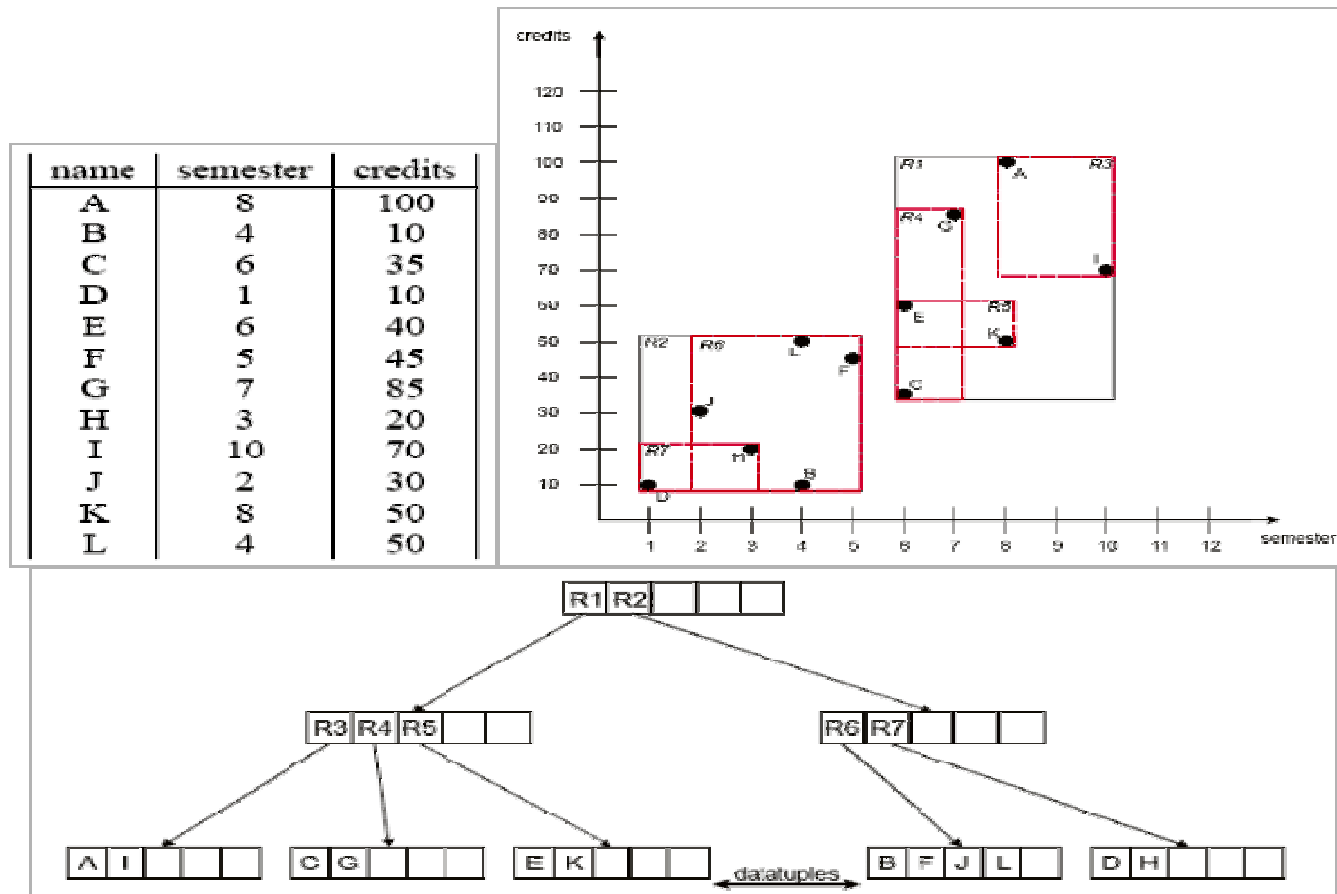
3.6.2 Distance-Based Methods

- ▶ Generalize the test-based techniques
- ▶ Distance-based outliers are those objects that do not have “enough” neighbors
- ▶ **Formally**
 - Define **DB(pct, dmin)-outlier**: a distance based outlier with parameters **pct** and **dmin**
 - An object **o** is **DB(pct, dmin)-outlier** if at least a fraction **pct** of the objects lie at a distance greater than **dmin** from **o**
- ▶ Avoids excessive computation related to fitting the observed data into some standard distribution and selecting discordancy tests

Distance-Based Algorithms

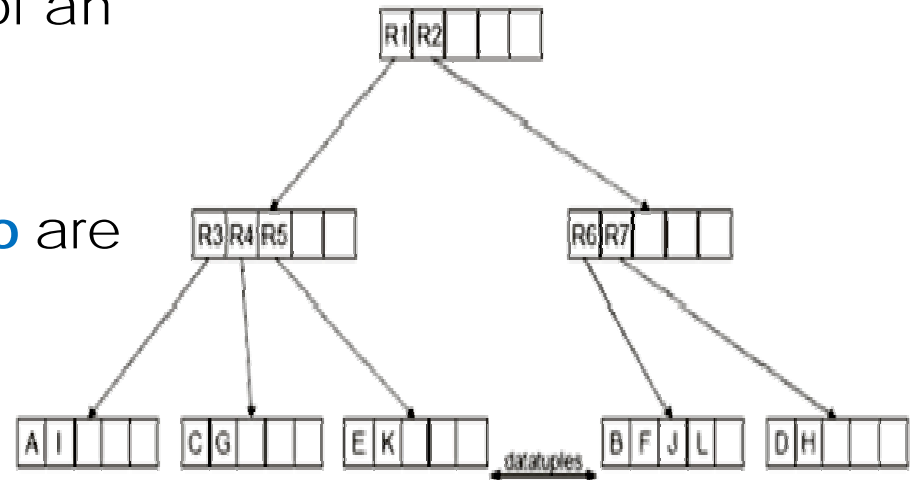
Index-based algorithms

- Use multidimensional indexing structures such as R-trees or k-d trees to search for neighbors of each object o



Distance-Based Algorithms

- ▶ Find neighbors of object **o** within a radius **d_{min}**
- ▶ **M** is the maximum number of objects within the **d_{min}**-neighborhood of an outlier
- ▶ Once **M+1** neighbors of object **o** are found, then **o is not an outlier**
- ▶ Complexity of **$O(n^2k)$**
 - N: number of objects
 - K: dimensionality
- ▶ Complexity is in search time. Building the index can be computationally very expensive



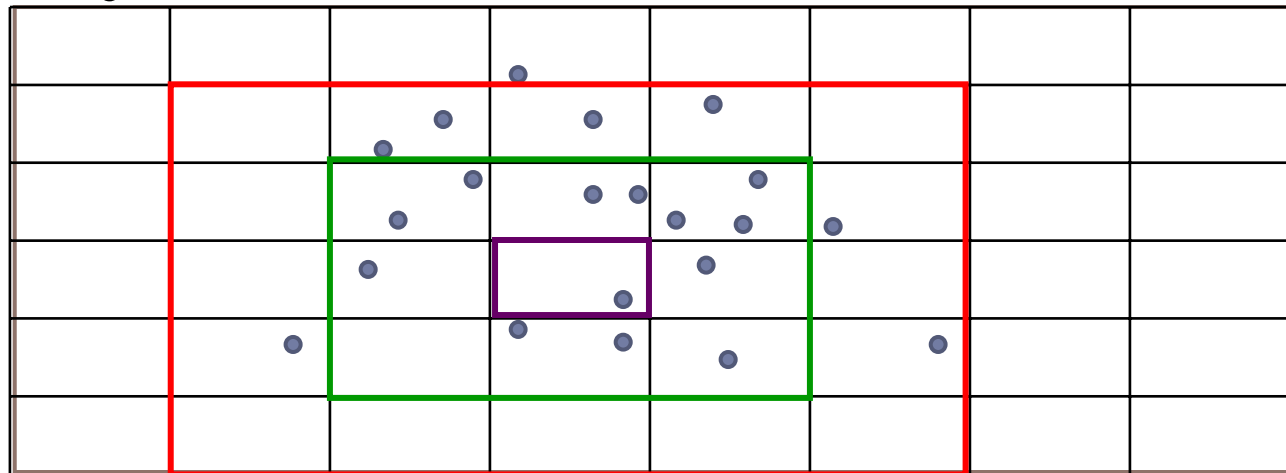
Distance-Based Algorithms

Cell-based algorithms

- ▶ The data space is partitioned into cells with a side length equal to

$$\frac{d_{min}}{2\sqrt{k}}$$

- ▶ **d_{min}**: radius around objects
- ▶ **K**: dimensionality

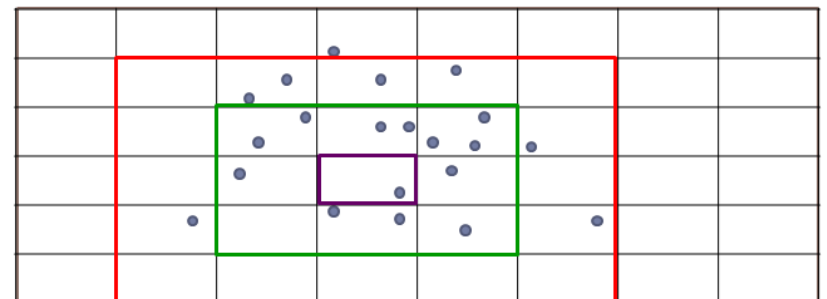


- ▶ Each cell has two layers surrounding it
 - First layer is 1-cell thick
 - Second layer is $2\sqrt{k} - 1$ thick, rounded up to the closest integer

Distance-Based Algorithms

Cell-based algorithms

- ▶ Count outliers on a **cell-by-cell** rather than **object-by-object** basis
- ▶ For a given cell, the algorithm accumulates three counts
 - The number of objects on the cell **C**
 - The number of objects in the cell and the first layer **C+1**
 - The number of objects in the cell and the second layer **C+2**
- ▶ How to determine outliers with these counts?



Distance-Based Algorithms

Cell-based algorithms

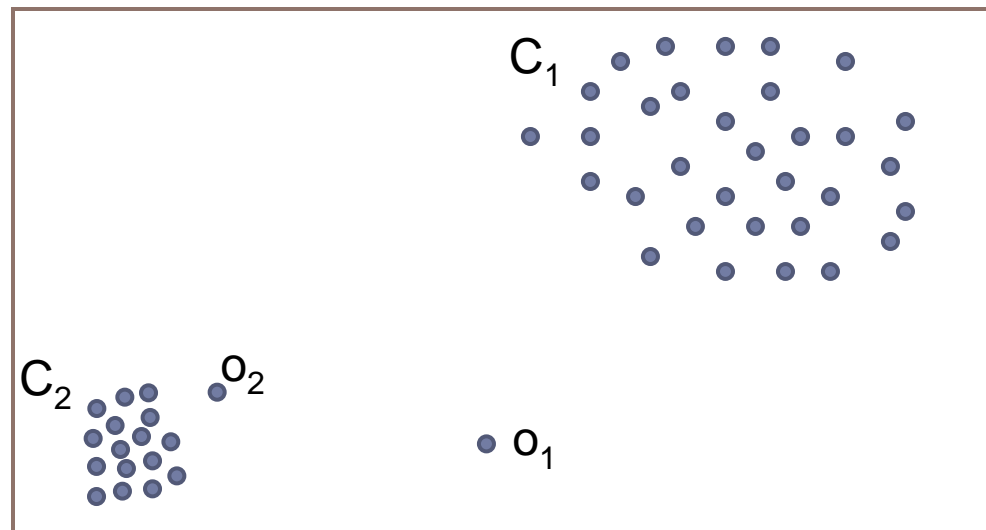
- ▶ Assume M to be a threshold used to detect outliers
- ▶ An object o is considered as an **outlier** if $C+1 < M$, else all the objects in the cell are considered as **non outliers**
- ▶ If $C+2 < M$, all the objects in the cell are considered **outliers**
- ▶ If $C+2 > M$, it is **possible** that some objects in the cell are **outliers**
 - do object-by-object processing to detect outliers
 - only objects that have less than M objects in their d_{min} -neighborhood are outliers
 - the d_{min} -neighborhood consist of the object's cell, all of its first layer and some of its second layer

Characteristics of Distance-Based Methods

- ▶ Avoid $O(n^2)$ computational complexity
- ▶ Its complexity is $O(c^k+n)$
 - c is a constant depending on the number of cells
 - k the dimensionality
 - n number of objects
- ▶ Developed for memory-resident data sets
- ▶ Requires the user to set both d_{min} and pct
- ▶ Finding suitable settings for these parameters can involve much trial and error

3.6.3 Density-Based Methods

- ▶ Statistical and distance-based methods depend on the overall “global” distribution of data
- ▶ Data are usually not uniformly distributed
- ▶ Data can have different density distributions



Density-Based Methods

- ▶ Define **Local Outliers**

- An object is a local outlier if it is outlying relative to its local neighborhood (w. r. t the density of the neighborhood)

- ▶ Does not consider being an outlier as a binary property

- Asses the degree to which an object is an outlier

- The degree of the “outlierness” is computed as the **Local Outlier Factor(LOF)** of an object

- The degree depends on how isolated the object is with respect to the surrounding neighborhood

- ▶ Detect global and local outliers

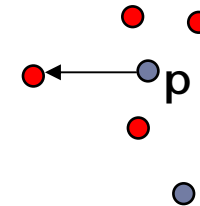
Density-Based Methods

- ▶ To define the local outlier factor of an object, the following concepts should be introduced
 - K-distance
 - K-distance neighborhood
 - Reachability distance
 - Local reachability distance

K-distance & K-distance neighborhood

- ▶ The **k-distance** of an object **p** is the maximal distance that **p** gets from its **k-nearest neighbors**

→ Denoted **k-distance(p)**

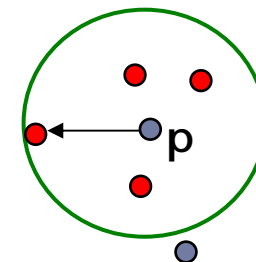


→ How k is determined?

→ LOF method sets k to the parameter **MinPts** used in the density-based clustering (e.g., **Minpts=4**) [**MinPts-distance**]

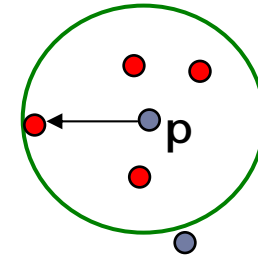
- ▶ **K-distance neighborhood** of an object **p** contains the MinPts-nearest neighbors of p

→ Denoted $N_{k\text{-distance}}(P)$ or $N_k(P)$, also N_{MinPts}



Reachability distance

- ▶ The **reachability-distance** of an object **q** with respect to object **o** (where **o** is within the MinPts-nearest neighbors of **P**) is denoted **reach_distMinPts(p,o)**



- ▶ $\text{Reach_distMinPts}(p,o) = \max\{\text{MinPts_distance}(o), d(p,o)\}$
- ▶ If **p** is far away from **o**, the reachability distance between the two is simply their actual distance
- ▶ If they are close, then the actual distance is replaced by the MinPts_distance of **o**

Local Outlier Factor (LOF)

- ▶ The **local reachability density** of p is the inverse of the average reachability density based on the MinPts-nearest neighbors of p

$$lrd_{MinPts}(p) = \frac{|N_{MinPts}(P)|}{\sum_{o \in N_{MinPts}(p)} reach_dist_{MinPts}(p, o)}$$

- ▶ The **local outlier factor (LOF)** of p captures the degree to which we call p an outlier

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(P)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(P)}}{|N_{MinPts}(P)|}$$

3.6.4 Deviation-Based Methods

- ▶ Identify outliers by examining the main characteristics of objects in a group
- ▶ Objects that deviate from this description are outliers
- ▶ The term **deviation** is used to refer to **outliers**
- ▶ Two main methods
 - Sequential Exception Technique
 - OLAP Data Cube Technique

Summary of Chapter 3

- ▶ A **cluster** is a collection of data objects that are similar within the same cluster and dissimilar to the objects on other clusters
- ▶ Clustering can be used as
 - a **main task** to gain insights about the data
 - a **preprocessing** step for other data mining algorithms
- ▶ **Several applications**
 - Market segmentation
 - Pattern recognition
 - Biological studies
 - Spatial data analysis
 - Web document classification, etc.

Summary of Chapter 3

- ▶ The **quality** of clustering can be **assessed** based on **dissimilarity** of objects
- ▶ Many techniques have been developed
 - Partitioning Methods
 - Hierarchical methods
 - Density-based methods
 - Grid-based methods
 - Model-based methods
 - Clustering high dimensional data
 - Constrained-based methods