Data Mining

Implementation and Applications of Databases, Spring 2019

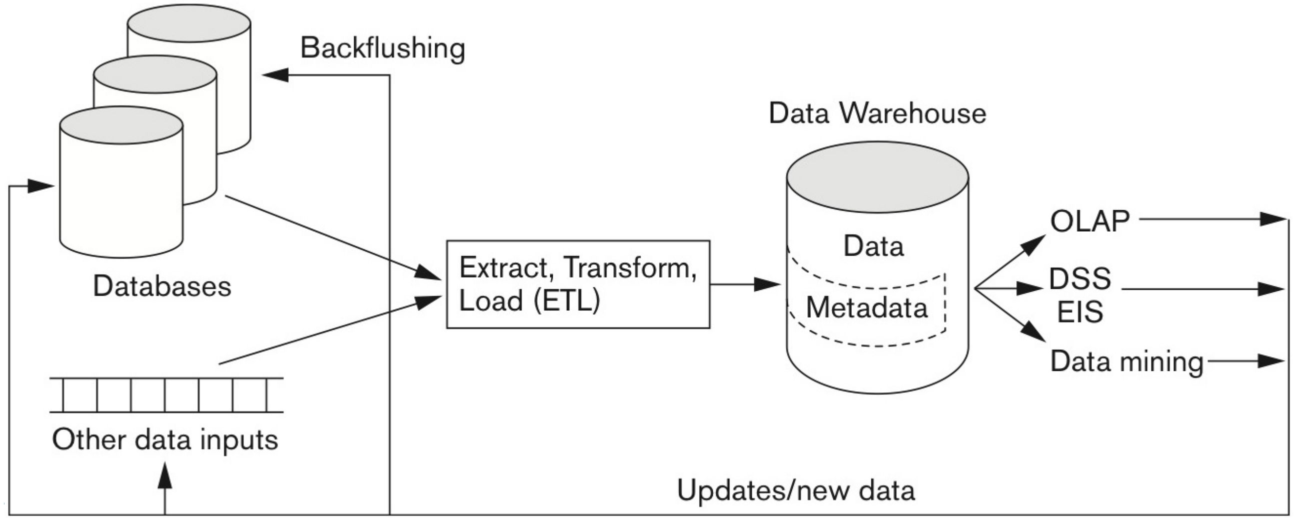
Ira Assent

Intended learning outcomes

- Be able to
  - Describe the goals and applications of common data mining approaches
  - Discuss the basic steps in k-means clustering, decision tree classification, and association rule mining

Data Warehousing recap

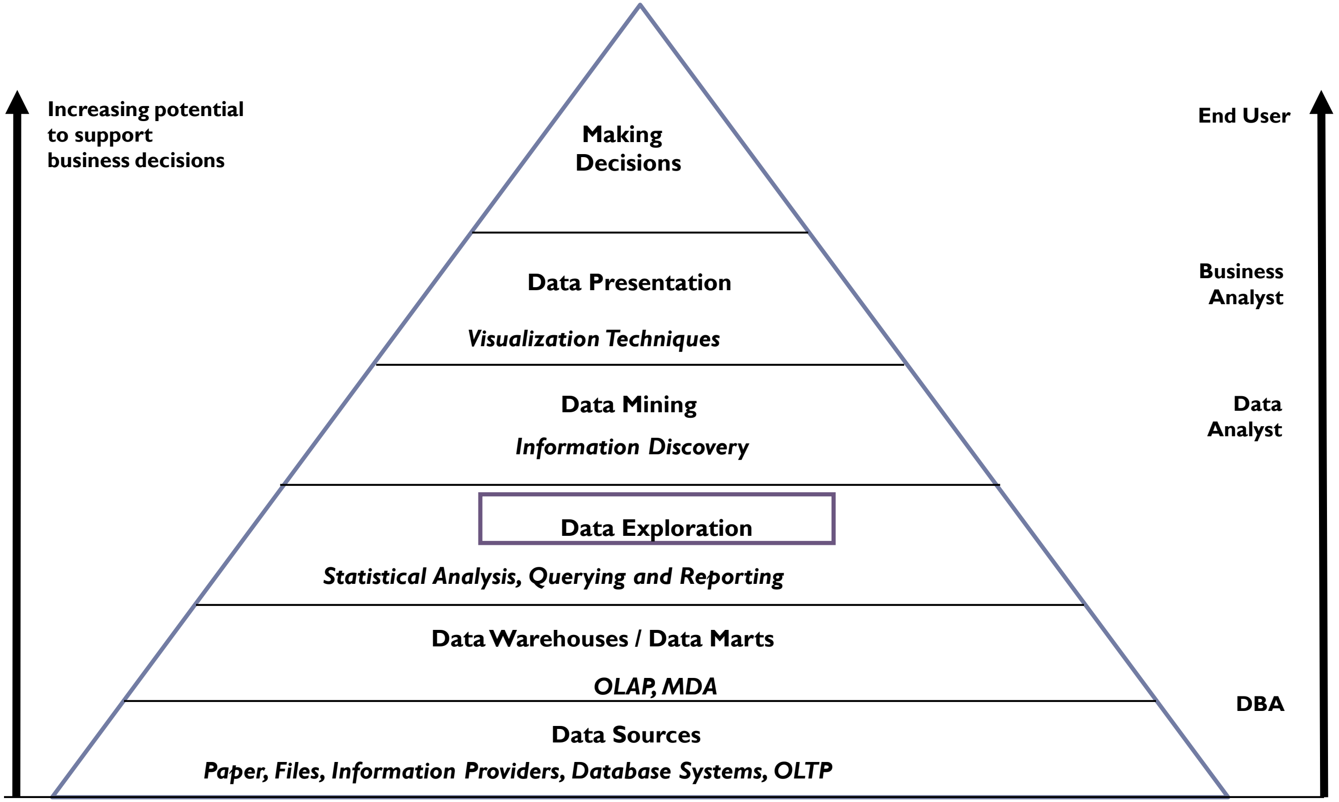
- Data Warehouse processing involves
  - Cleaning and reformatting of data
  - OLAP
  - Data Mining



What is backflushing?

A. The process of flushing log entries to the disk.
B. The process of refreshing data in the data warehouse with data from operational databases.
C. The process of updating operational databases with data warehouse data.
D. The process of sending updated metadata to all sites in a distributed database.

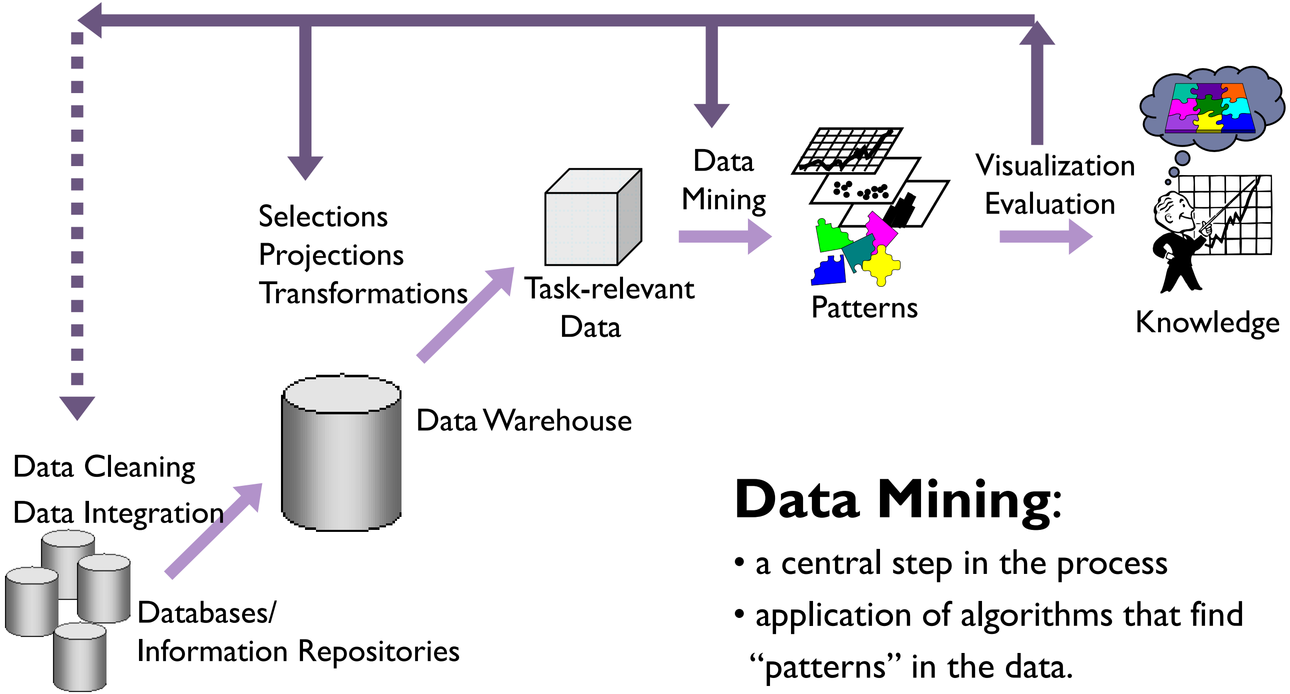
Data Mining and Business Intelligence



Definitions of Data Mining

- Discovery of new information in terms of patterns or rules from vast amounts of data
- Process of finding interesting structure in data
- Process of employing one or more computer learning techniques to automatically analyze and extract knowledge from data
- Data mining may generate thousands of patterns: not all interesting
  - Pattern is interesting if easily understood by humans, valid on new or test data with some degree of certainty, potentially useful, novel, or validates some hypothesis that a user seeks to confirm
- Objective vs. subjective interestingness measures
  - Objective: based on statistics and structures of patterns, e.g., support, confidence, etc.
  - Subjective: based on user's belief in the data, e.g., unexpectedness, novelty, actionability, etc.

KDD (knowledge discovery in databases)



Data Mining:

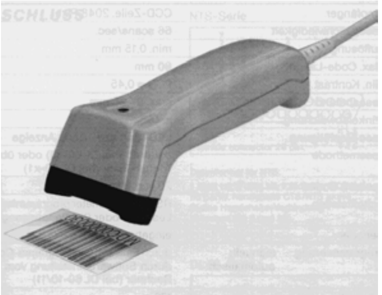
- a central step in the process
- application of algorithms that find "patterns" in the data.

Types of Discovered Knowledge

- Association Rules
- Classification Models and Predictions
- Sequential Patterns such as trends, motifs
- Clustering: groups of related objects
- …
- Applications
  - Marketing
    - Marketing strategies and consumer behavior
  - Finance
    - Fraud detection, creditworthiness and investment analysis
  - Manufacturing
    - Resource optimization
  - Health
    - Image analysis, side effects of drug, and treatment effectiveness

Example: Basket Data Analysis

- Transaction database
  - {butter, bread, milk, sugar}
  - {butter, flour, milk, sugar}
  - {butter, eggs, milk, salt}
  - {eggs}
  - {butter, flour, milk, salt, sugar}



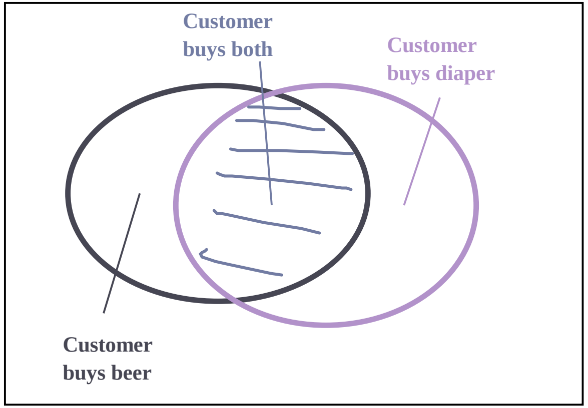
- Question of interest:
  - Which items are bought together frequently?

- Applications
  - Improved store layout
  - Cross marketing
  - Focused attached mailings / add-on sales

What Is Association Mining?

- Association rule mining
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
  - Rule form: "Body $\Rightarrow$ Head [support, confidence]"
- Applications
  - Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.
- Examples
  - buys(x, "diapers") $\Rightarrow$ buys(x, "beers") [0.5%, 60%]
    - 60% of those buying diapers also buy beers. In total, diapers and beers are bought in 0.5% of all purchases.
  - major(x, "CS") ^ takes(x, "DB") $\Rightarrow$ grade(x, "A") [1%, 75%]

Rule Measures: Support and Confidence



Find all the rules $X \& Y \Rightarrow Z$ with minimum confidence and support

- support, *s*, probability that a transaction contains {X, Y, Z}
- confidence, *c*, conditional probability that a transaction having {X, Y} also contains Z

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Let minimum support 50%, and minimum confidence 50%, then we have

- $A \Rightarrow C$ (50%, 66.6%)
- $C \Rightarrow A$ (50%, 100%)

Mining Association Rules—Example

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

- For rule $A \Rightarrow C$:
  - support = support($\{A, C\}$) = 50%
  - confidence = support($\{A, C\}$) / support($\{A\}$) = 66.6%
- Frequent items / itemsets are all those that exceed minimum support

Mining Frequent Itemsets: Basic Idea

- Naïve Algorithm
  - count the frequency of for all possible subsets of I in the database
  - *too expensive* since there are 2^m such itemsets for $|I| = m$ items

- The ***Apriori*** principle (monotonicity):

  Any subset of a frequent itemset must be frequent

- Method based on the apriori principle
  - First count the 1-itemsets, then the 2-itemsets, then the 3-itemsets, and so on
  - When counting (k+1)-itemsets, only consider those (k+1)-itemsets where all subsets of length k have been determined as frequent in the previous step

The Apriori Algorithm

variable C_k: candidate itemsets of size k
variable L_k: frequent itemsets of size k

L_1 = {frequent items}
for ($k = 1$; $L_k \mathrel{!}= \emptyset$; k++) **do begin**
    // JOIN STEP: join L_k with itself to produce C_{k+1}
    // PRUNE STEP: discard (k+1)-itemsets from C_{k+1} that contain non-frequent k-itemsets as subsets
    C_{k+1} = candidates generated from L_k

    for each transaction t in database do
        Increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$

Generating Candidates (Join Step)

- Requirements for candidate k-itemsets C_k
  - Must contain all frequent k-itemsets (superset property $C_k \supseteq L_k$)
  - Significantly smaller than the set of all k-subsets
  - Suppose the items are sorted by any order (e.g., lexicograph.)
- Step 1: Joining
  - Consider frequent $(k-1)$-itemsets p and q
  - p and q are joined if they share the same first k – 2 items
- Step 2: Pruning
  - Remove candidate k-itemsets which contain a non-frequent $(k-1)$-subset s, i.e., s $\notin L_{k-1}$
  - Example
    - L_3 = {(1 2 3), (1 2 4), (1 3 4), (1 3 5), (2 3 4)}
    - Candidates after the join step: {(1 2 3 4), (1 3 4 5)}
    - In the pruning step: delete (1 3 4 5) because (3 4 5) $\notin L_3$, i.e., (3 4 5) is not a frequent 3-itemset; also (1 4 5) $\notin L_3$

Generating Candidates – Full Example

minsup = 2

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C3

itemset
{2 3 5}

Scan D

L3

itemset	sup
{2 3 5}	2

Scan D

C4 is empty

Generating Rules from Frequent Itemsets

- For each frequent itemset X
  - For each subset A of X, form a rule $A \Rightarrow (X - A)$
  - Delete those rules that do not have minimum confidence
- Computation of the confidence of a rule $A \Rightarrow (X - A)$

$$confidence(A \Rightarrow (X - A)) = \frac{support(X)}{support(A)}$$

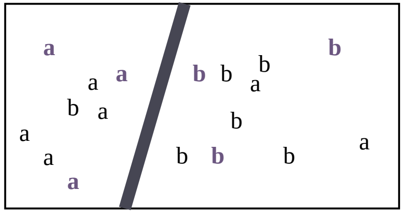
- Store the frequent itemsets and their support in a hash table in main memory → no additional database access

itemset	support
{A}	2
{B}	4
{C}	5
{A, B}	3
{A, C}	2
{B, C}	4
{A, B, C}	2

- Example: X = {A, B, C}, minConf=60%
  - conf (A $\Rightarrow$ B, C) = 1; conf (B, C $\Rightarrow$ A) = 1/2
  - conf (B $\Rightarrow$ A, C) = 1/2; conf (A, C $\Rightarrow$ B) = 1
  - conf (C $\Rightarrow$ A, B) = 2/5; conf (A, B $\Rightarrow$ C) = 2/3

Classification

- Learning a model able to describe different classes of data
- Supervised as the classes to be learned are predetermined
- Class labels are known for a small set of "training data":
  Find models/functions/rules (based on attribute values of the training examples) that
  - describe and distinguish classes
  - predict class membership for "new" objects



- Applications
  - Classify gene expression values for tissue samples to predict disease type and suggest best possible treatment
  - Automatic assignment of categories to large sets of newly observed celestial objects
  - Predict unknown or missing values (→ KDD pre-processing step)
  - . . .

Evaluation of Classifiers

- Classification Accuracy
  - Predict class label for each object o
  - Determine the fraction of correctly predicted class labels:

$$classification\ accuracy = \frac{count(\text{correctly predicted class label})}{count(o)}$$

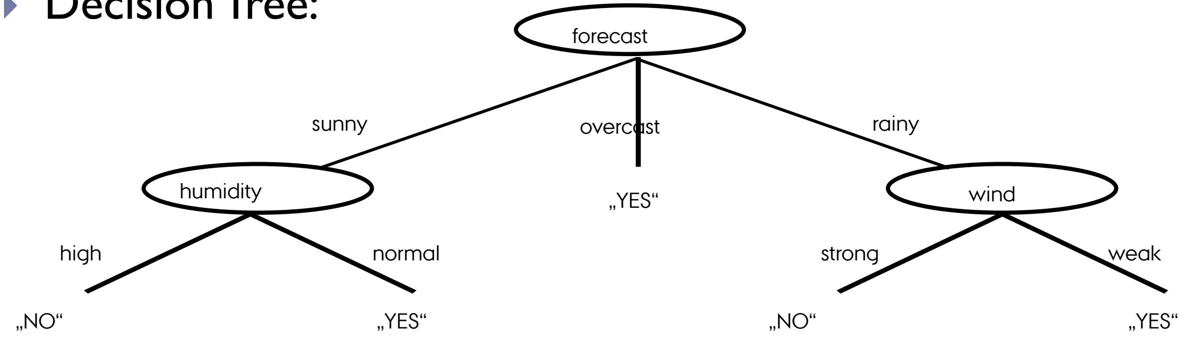
  - Classification error = 1 – classification accuracy
- Overfitting: accuracy worse on entire data than on training data
    - bad quality of training data (noise, missing values, wrong values)
    - different statistical characteristics of training data and test data
- Train-and-Test: decomposition of data set into two partitions
  - Training data to train the classifier
    - Model construction using information also class labels
  - Test data to evaluate the classifier
    - temporarily hide class labels, predict them and compare

Decision Tree Classifiers

- Are we going to play tennis?
- Training data set:

day	forecast	temperature	humidity	wind	tennis decision
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	. . .	. . .	. . .	. . .	. . .

- Decision Tree:



Which attribute should be root?

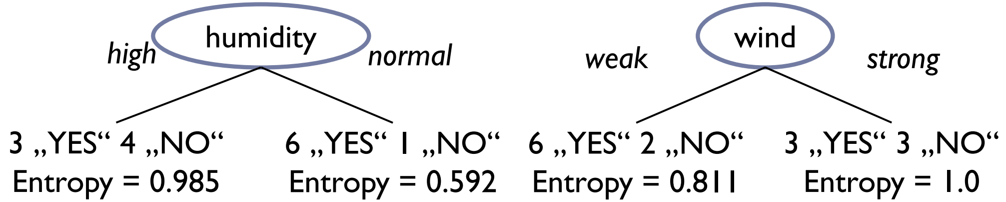
A. The one with the most frequent attribute value.
B. The one that has the attribute value with the purest class label.
C. The one that has a value distribution to balance the class label distribution.
D. The one that has a value distribution to separate the class labels.

BUILDING Decision Trees

- Tree is created top-down
- Training examples T recursively partitioned into $T_1, T_2, \ldots, T_m$
  - Entropy for k classes with frequencies p_i (Information theory: measure of uncertainty)

$$information\ gain(T, A) = entropy(T) - \sum_{i=1}^{m} \frac{|T_i|}{|T|} \cdot entropy(T_i) \qquad entropy(T) = \sum_{i=1}^{k} p_i \cdot \log_2 p_i$$

9 „YES“ 5 „NO“ Entropy = 0.940



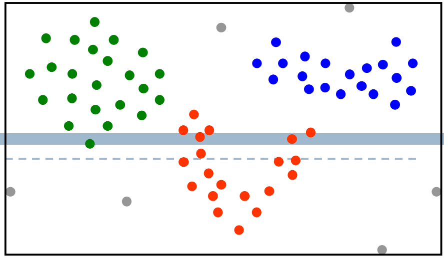
$$IG(T, hum) = 0.94 - \frac{7}{14} * 0.985 - \frac{7}{14} * 0.592 = 0.151$$

$$IG(T, wind) = 0.94 - \frac{8}{14} * 0.811 - \frac{6}{14} * 1.0 = 0.048$$

Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

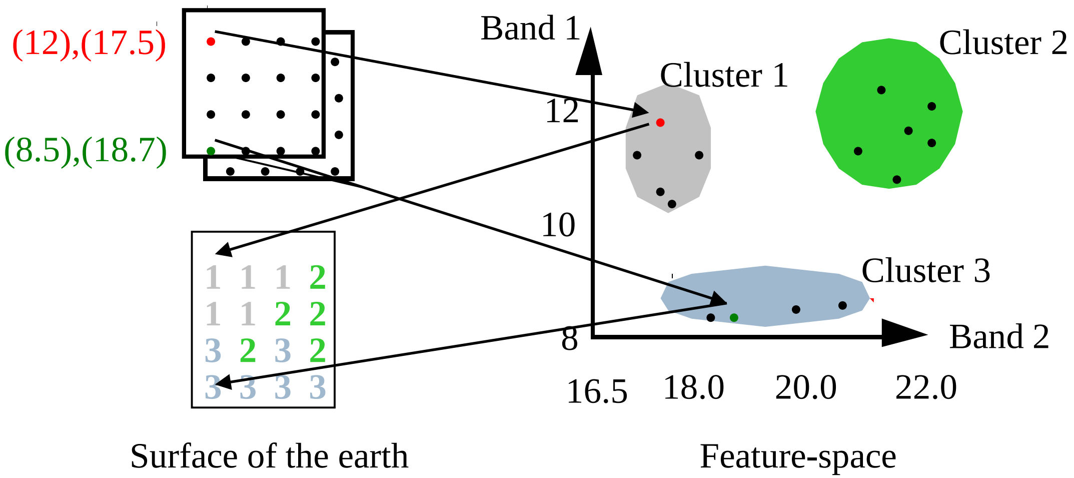
Clustering

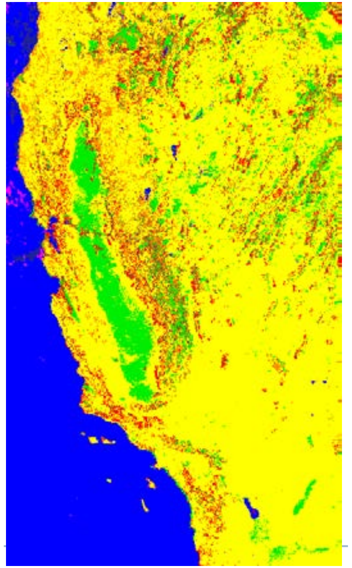


- Class labels are unknown:
  Group objects into sub-groups (clusters)
  - Similarity function (or dissimilarity fct. = distance) to measure similarity between objects
  - Objective: "maximize" intra-class similarity and "minimize" interclass similarity
- Clustering = ***unsupervised classification*** (no predefined classes)
- Typical usage
  - As a *stand-alone tool* to get insight into data distribution
  - As a *preprocessing step* for other algorithms
- Applications
  - Customer profiling/segmentation
  - Document or image collections
  - Web access patterns
  - …

A Typical Application: Thematic Maps

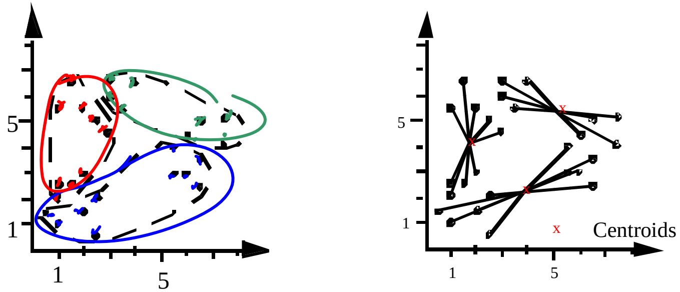
- Satellite images of a region in different wavelengths
  - Different land-uses reflect and emit light of different wavelengths in characteristic way
  - Each point on surface $p = (x_1, \ldots, x_d)$ has d values x_i of recorded intensity in band i



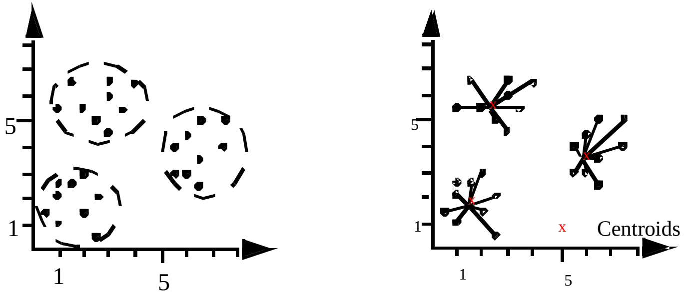


K-Means Clustering: Basic Idea

- Objective: For a given *k*, form *k* groups so that the sum of the (squared) distances between the mean of the groups and their elements is minimal.
  - Poor Clustering



  - Optimal Clustering



K-Means Clustering: Algorithm

Given *k*, the *k*-means algorithm is implemented in 4 steps:

1. Partition the objects into *k* nonempty subsets
2. Compute the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
3. Assign each object to the cluster with the nearest representative.
4. Go back to Step 2, stop when representatives do not change.

K-Means Clustering: Basic Notions

- Objects $p = (x^p{}_1, \ldots, x^p{}_d)$ are points in a d-dimensional vector space (the mean of a set of points must be defined)
- *Centroid* μ_C: Mean of all points in a cluster C, $$\mu_C = \frac{1}{|C|} \sum_{x_i \in C} x_i$$
- Measure for the compactness („Total Distance“) of a **cluster** C_j:

$$TD(C_j) = \sqrt{\sum_{p \in C_j} dist(p, \mu_{C_j})^2}$$

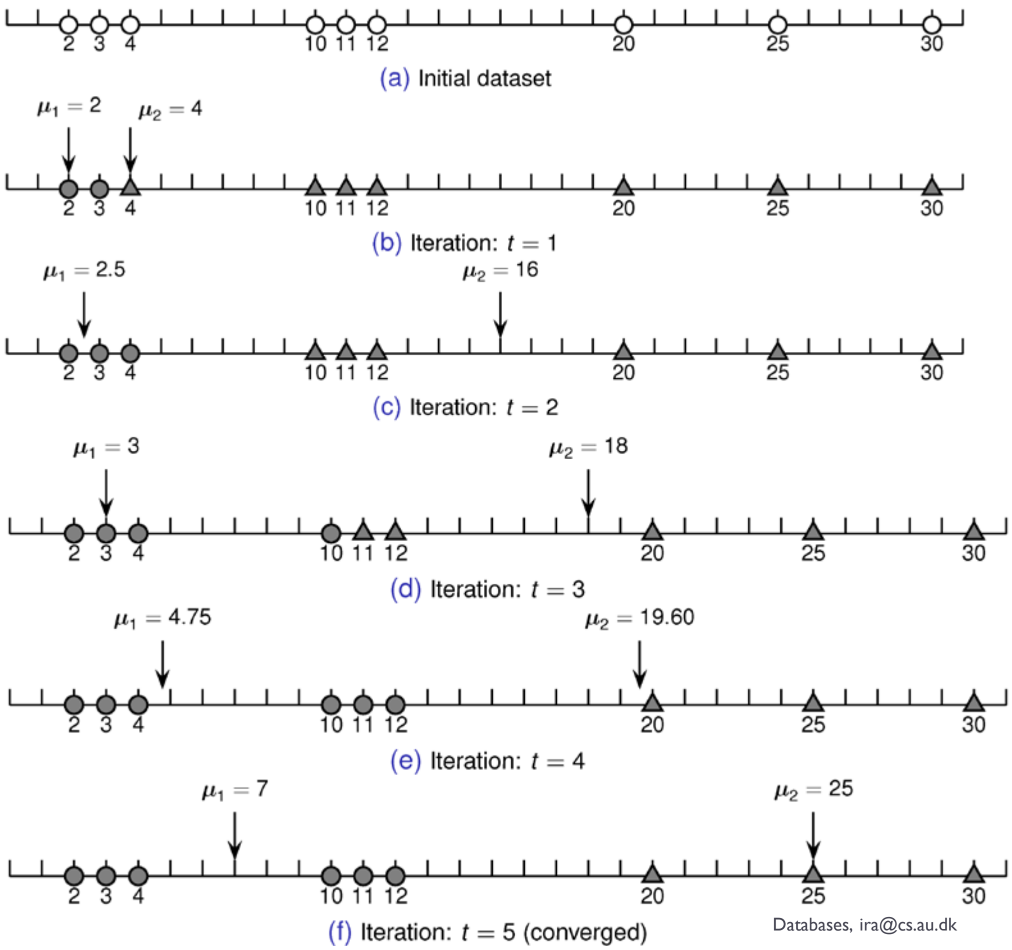
- Measure for the compactness of a **clustering**

$$TD = \sqrt{\sum_{j=1}^{k} TD^2(C_j)}$$

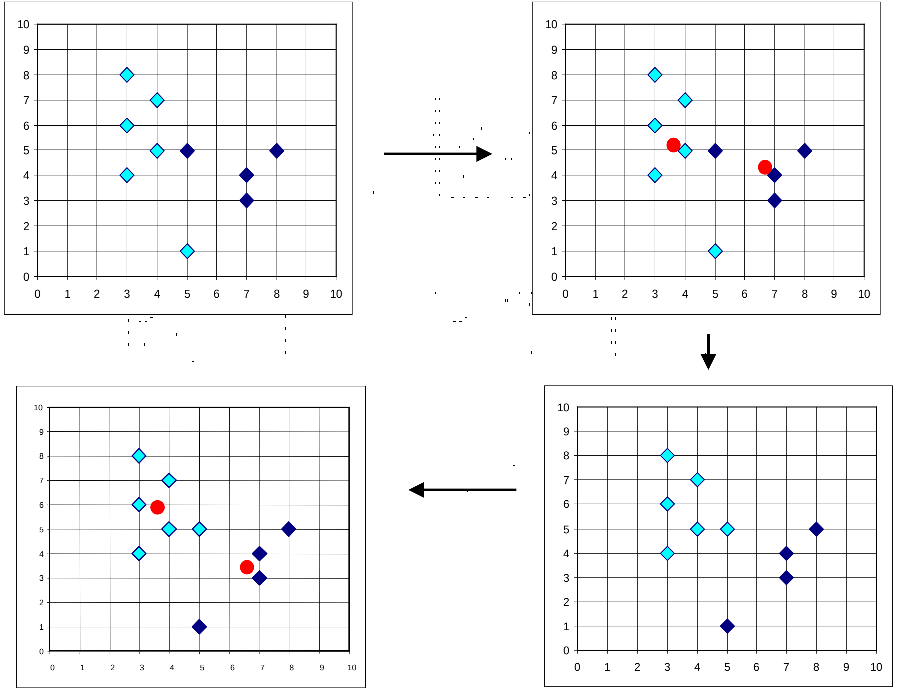
What is a good k-means clustering?

A. TD is low.
B. TD is high.
C. The centroid is stable.
D. The centroid is equally far from all points in the cluster.

K-Means example in one dimension



K-Means Clustering: Example



Further Data Mining / Machine Learning Methods

- Sequential pattern analysis
- Time Series Analysis
- Regression
- Neural Networks
- Genetic Algorithms

- Machine Learning course (Bachelor, 3rd year)
- Advanced Data Management and Analysis course (Master)

Sequential Pattern Analysis

- Transactions ordered by time of purchase
  - form sequence of **itemsets**
- Goal: find all **subsequences** from a given set of sequences that exceed minimum support
  - Sequence S_1, S_2, S_3, .. predictor that a customer purchasing itemset S_1 is likely to buy S_2, and then S_3, and so on
  - Temporal order relevant
    - E.g. buy baby milk, then buy children's food; not so much the other way around

Time Series Analysis

- Time series sequences of values
  - Example: closing price of a stock every week day
- Time series analysis
  - Identify the price trends of a stock or mutual fund
  - Generally temporal trends of values
  - Extended functionality of temporal data management

Regression Analysis

- A regression equation estimates a dependent variable using a set of independent variables and a set of constants
  - Independent and dependent variables all numeric
  - written in the form $Y=f(x_1,x_2,\ldots,x_n)$ where Y is dependent variable
  - If f is linear in the domain variables x_i, the equation is called a linear regression equation

Neural Networks

- A neural network is a set of interconnected nodes inspired by the human brain (not a model of the brain!)
- Node connections have weights which are modified during the learning process
- Neural networks can be used for supervised learning and unsupervised clustering
  - Recently dramatic improvements in performance
    - Big Data: lots of training data
    - New training methods and network architectures
- The output of a neural network is quantitative and not easily understood

Genetic Learning

- Genetic learning based on the theory of evolution
  - Initial population of several candidate solutions provided to the learning model
  - Fitness function defines which solutions survive from one generation to the next
  - Crossover, mutation and selection used to create new population elements

What is supervised learning?

VOTE

A. Clustering based on total distance.
B. Association rule mining with fixed minimum support.
C. Model creation based on class label information.
D. Learning with user input.

Intended learning outcomes

- Be able to
  - Describe the goals and applications of common data mining approaches
  - Discuss the basic steps in k-means clustering, decision tree classification, and association rule mining

What was this all about?

Guidelines for your own review of today's session

- In data mining, the goal is to…
  - The KDD process involves…
- Clustering is also called…
  - The learning goal is to…
- Classification is also called…
  - The learning goal is to…
  - Overfitting is the problem of… and can be addressed using…
- Association rule mining tries to…
  - The general idea in apriori makes use of…
- Other data mining tasks are…