Data-driven Intelligent Systems

Lecture 18
KNN, Case-based Reasoning, and Association Rules

KNOWLEDGE
TECHNOLOGY

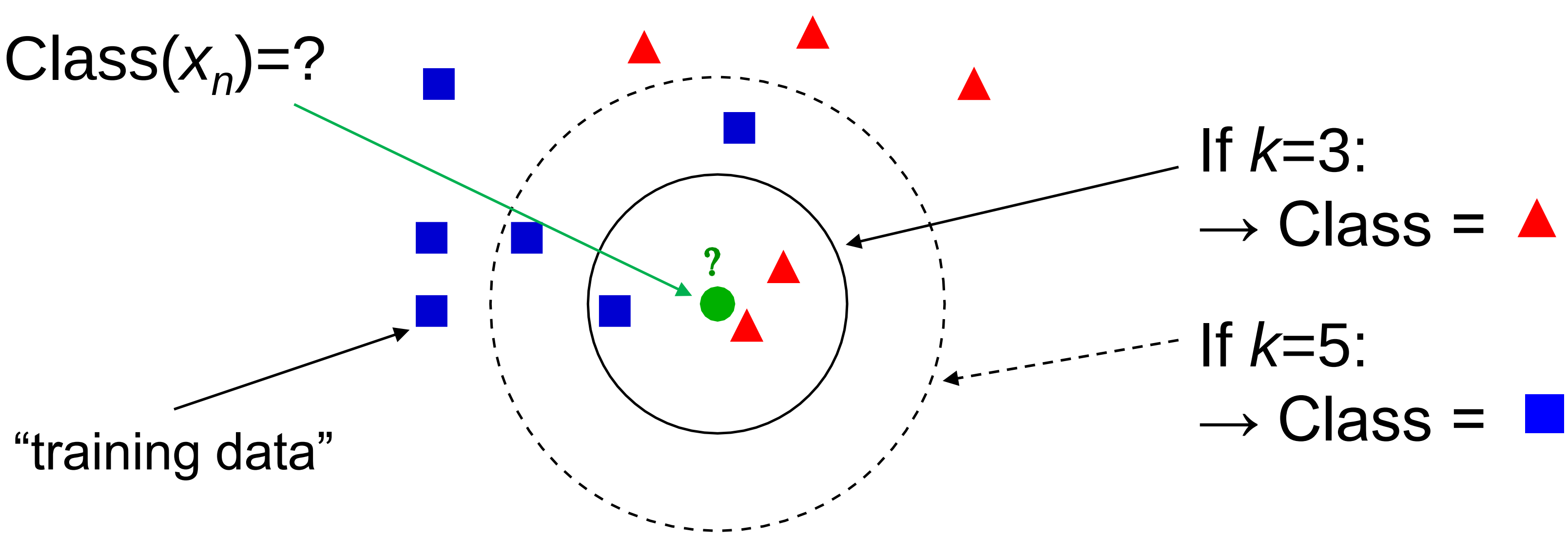
http://www.informatik.uni-hamburg.de/WTM/

Outline

1. k-Nearest Neighbors
2. Case-based Reasoning
3. Overview Predictive vs. Descriptive Modeling
4. Association Rules – Apriori algorithm

k-Nearest Neighbors (KNN)

- ***Classification without learning*** (but need to keep the data)
  - Task is to classify a new data point x_n
  - Find the k nearest points $\{x_{k'}\}$ with their class labels $\{y_{k'}\}$ (from the ~~training~~ data)
  - Assign class y_n based on the majority vote of $\{y_{k'}\}$



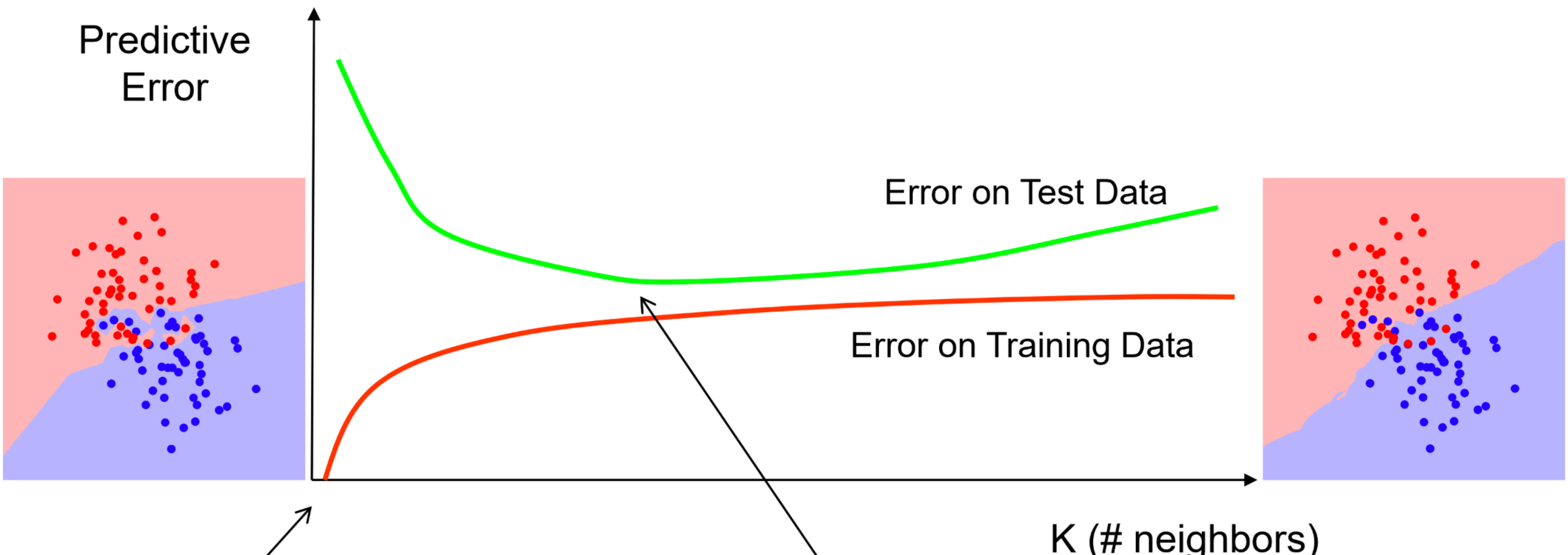
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Figure by Antti Ajanki AnAj, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2170282

k-Nearest Neighbors: Choosing *k*

k small
(overfitting)

k large
(underfitting)

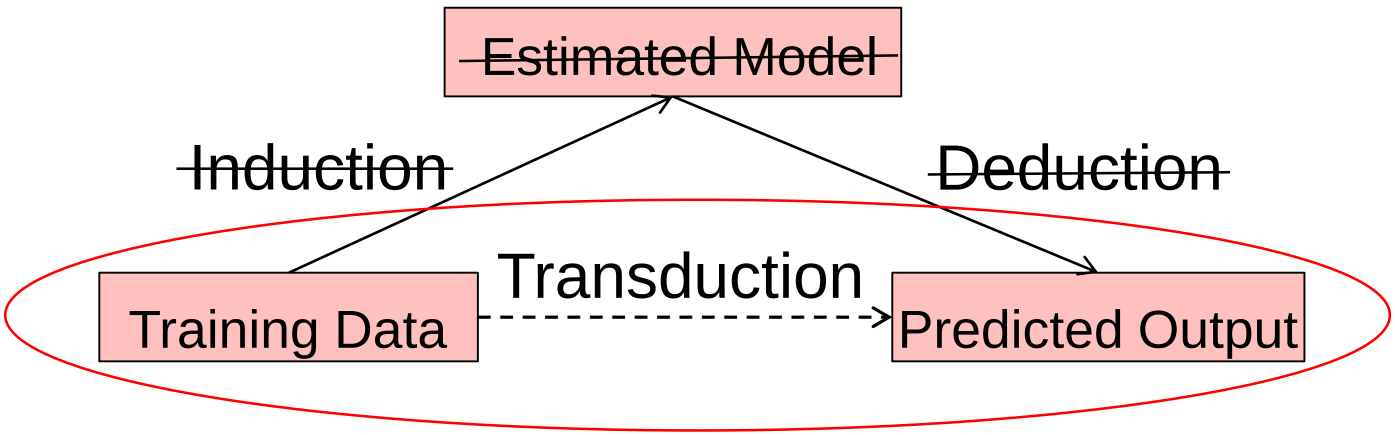


k=1:
zero error on training data

best value for *k*

k-Nearest Neighbors

- The distance measure is important, e.g. Euclidean distance
- Normalization can change the results, e.g. whitening
- KNN uses existing data *$\{x_{k'}, y_{k'}\}$* directly for decisions, does not build a model



- *k*=1: decide as in one precedence case
  → Case Based Reasoning
  (existing data are “past experience”)

Outline

1. k-Nearest Neighbors
2. Case-based Reasoning
3. Overview Predictive vs. Descriptive Modeling
4. Association Rules – Apriori algorithm

CBR – A way to solve complex problems

- By remembering how we solved a similar case in the past

- This is Case Based Reasoning (CBR)
  - memory-based problem-solving
  - re-using past experiences

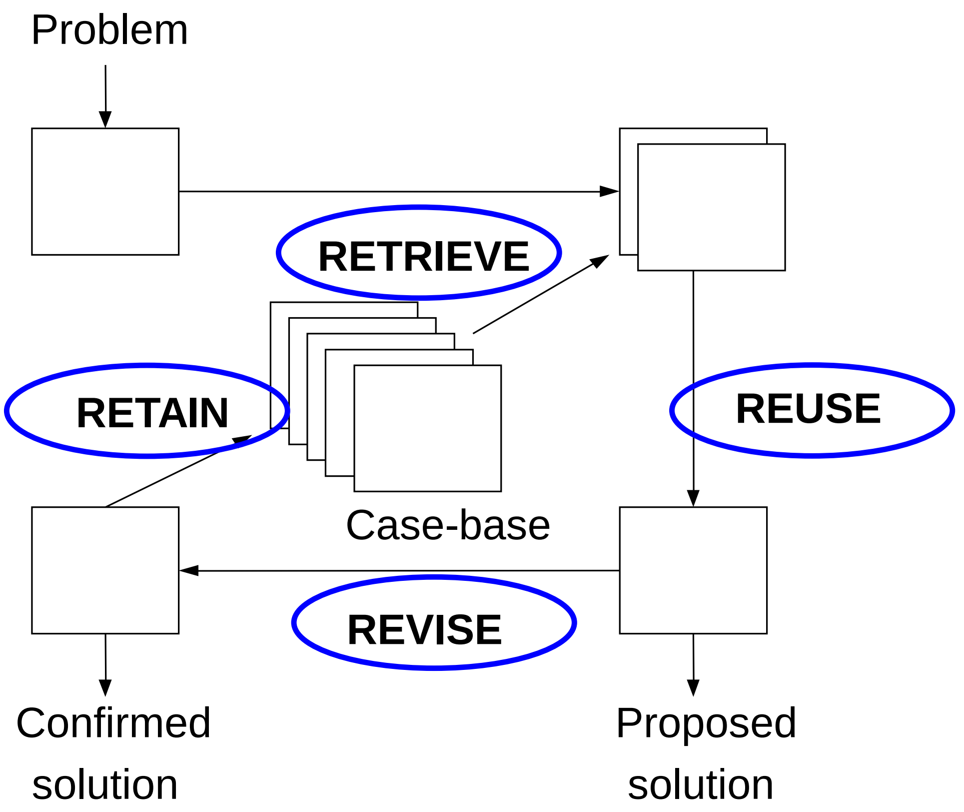
- Experts often find it easier to relate to past cases than to formulate rules about reasoning

CBR – Problems we solve this way

- Medicine
  - doctor remembers previous patients
    - especially for rare combinations of symptoms
- Law
  - law depends on precedence
  - case histories are consulted
- Management
  - decisions are often based on past rulings
- Financial
  - performance is predicted by past results
- Robotics
  - Robot soccer – imitate good moves

CBR – Overview

- CBR provides an automated method for storing experience and reusing it to make decisions in the future



CBR Prerequisites

- Expertise is embodied in a library of past cases (experiences)
- Each case typically contains
  - a *description* of the problem

    → features should allow to find similar cases
  - *goals*, and subgoals that arise in reasoning
  - *successful attempts* at achieving those goals

    → to propose solutions to new problems
  - *failed attempts*

    → to warn of possible failure

CBR Process

- Basic algorithm to solve a current problem:
  - Match the problem's features against the cases in the case base, and retrieve similar cases.
    - If multiple solutions are found then resolve any ambiguities.
  - Reuse retrieved cases to propose a solution and test for success.
  - Revise the solution, if necessary.
  - Retain the current problem as part of a new case, i.e.
    - its defining features,
    - goals & subgoals,
    - successful & failed attempts,
    - its final solution.

CBR Evaluation

- What does the CBR process depend on?
  - Appropriate methods for ***indexing cases*** using their key attributes
  - Efficient mechanisms for ***retrieving cases*** given a set of index values
  - ***Existing cases*** – a smaller case-base can be compensated for by more creativity in retrieval and revision
  - Good ***presentation*** of the information to the user

CBR Applications

- Failure prediction
  - ultrasonic non-destructive testing for Dutch railways
  - water in oil wells for Schlumberger
- Failure analysis
  - Mercedes cars for DaimlerChrysler
  - semiconductors at National Semiconductor
- Maintenance scheduling
  - Boeing 737 engines
  - TGV trains for SNCF
- Planning
  - mission planning for US navy
  - route planning for DaimlerChrysler cars

CBR in Business

“those who ignore history are doomed to repeat it“

Norwegian CBR consultant Verdande started in the oil business

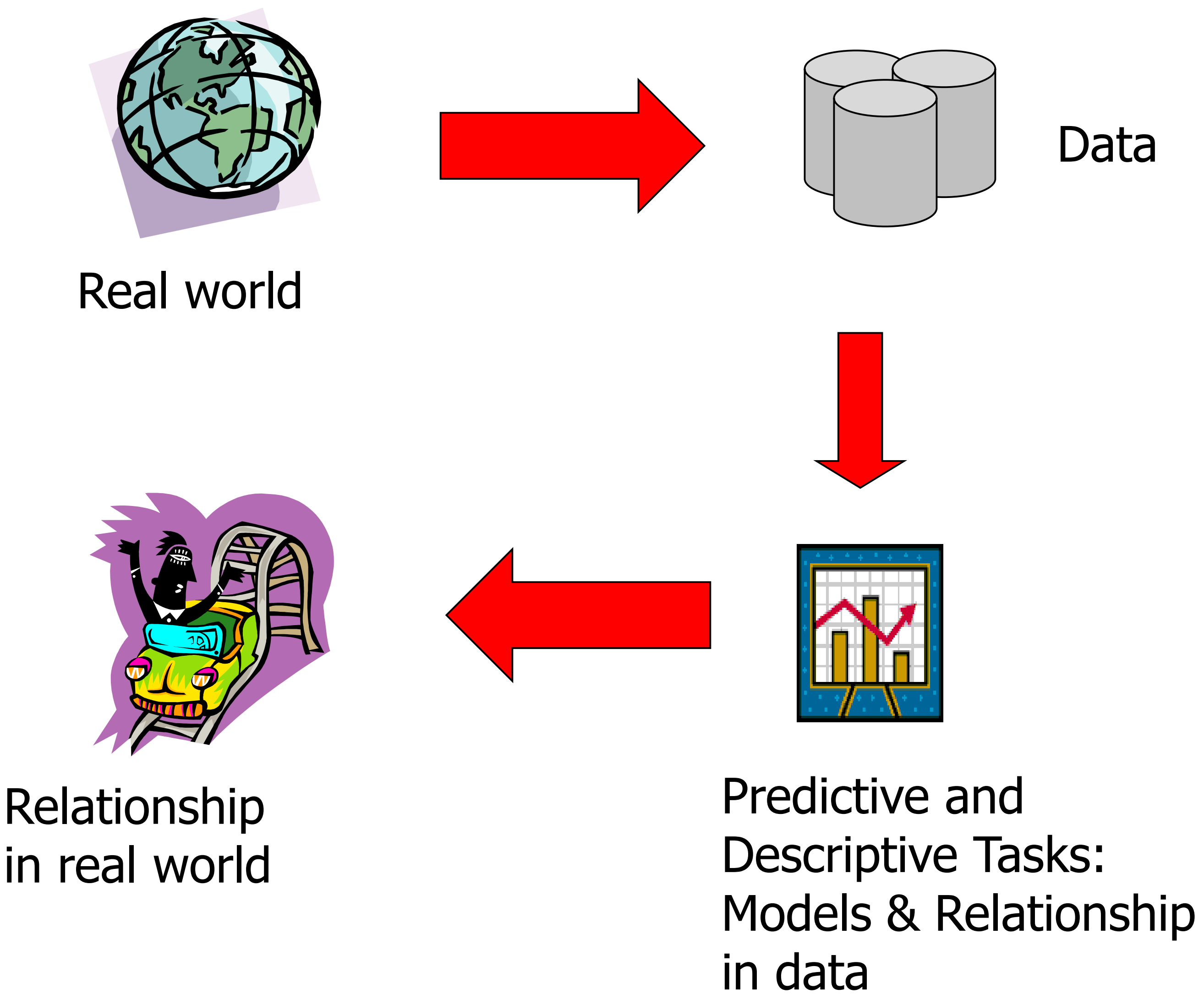
CBR as Presented by “Verdande Technology”

- ”Based on the principle that similar problems have similar solutions, CBR .. analyzes data patterns in real-time, using past events to proactively predict future problems.”
- ”harvests multiple, heterogeneous data types to index and search for those past experiences and provides organizations with the information they need”
- ”transforms big data into actionable insight”
- ”offering a realistic assessment as to whether a similar scenario is likely to occur in the future”
- ”can help reduce drilling NPT” (non-productive time) by:
  - ”Identify problem precursors.
  - Interpret and resolve the drilling situation.
  - Retrieve relevant solutions and lessons learned.”

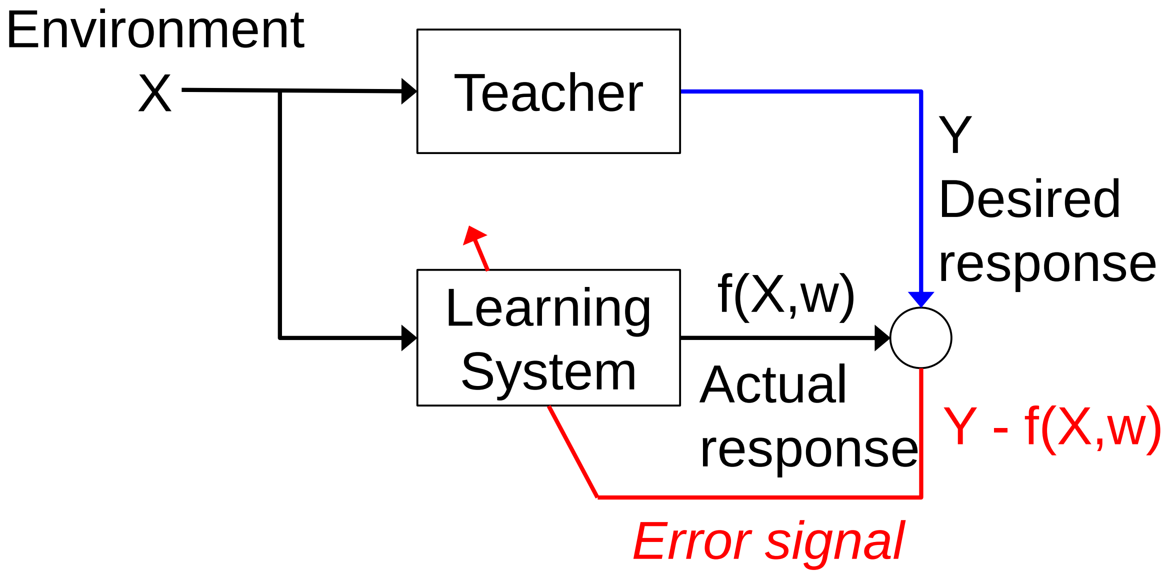
Outline

1. k-Nearest Neighbors
2. Case-based Reasoning
3. Overview Predictive vs. Descriptive Modeling
4. Association Rules – Apriori algorithm

Turning Data into Knowledge

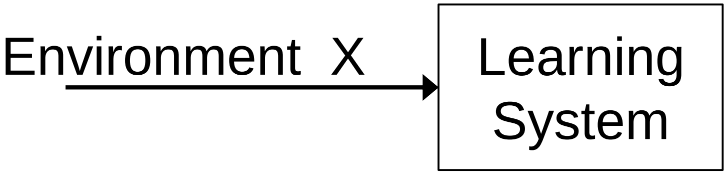


Main Types of Inductive Learning



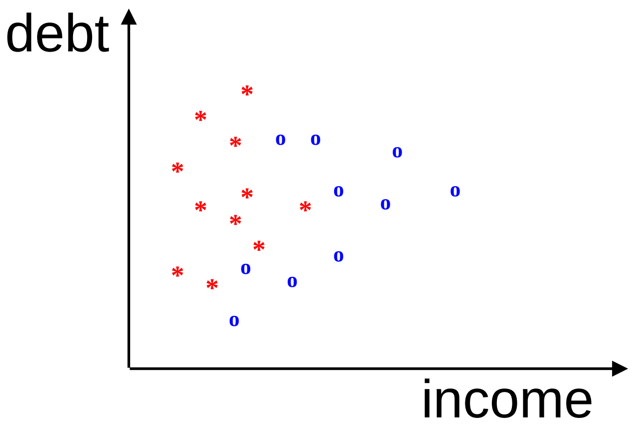
Supervised learning
(with a teacher)

Unsupervised learning:
(without teacher)

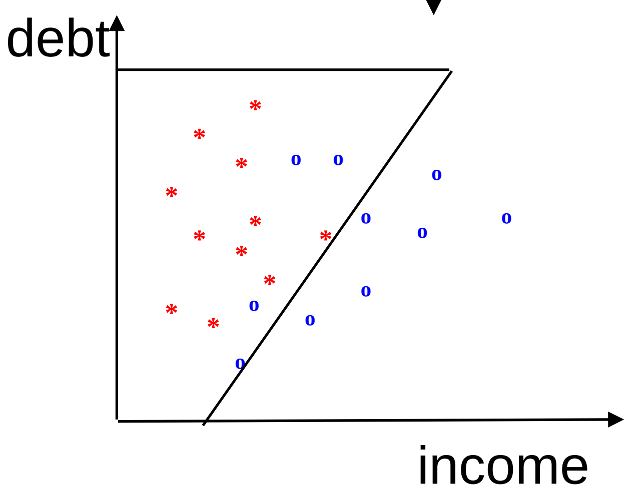


- goal is to discover “natural” structure in the data,
- requires task-independent measure of quality of representation

Example – Supervised Learning



Supervised
Learning



Problem: bank approval of credit (1)

Available data:

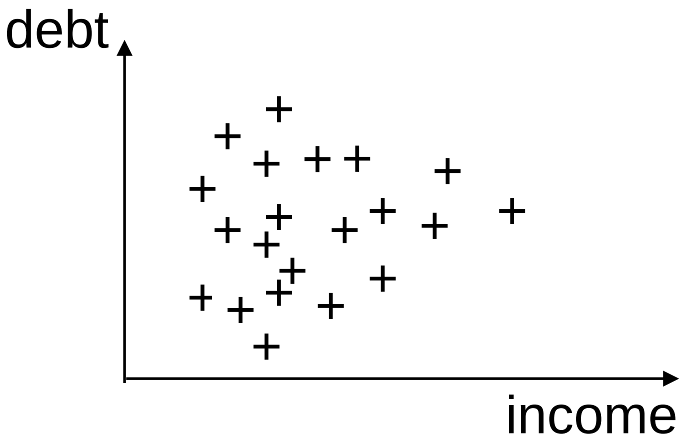
- * Rejected
- o Approved by expert

Previous customers with or without approval.

Learning: Linear classification function:

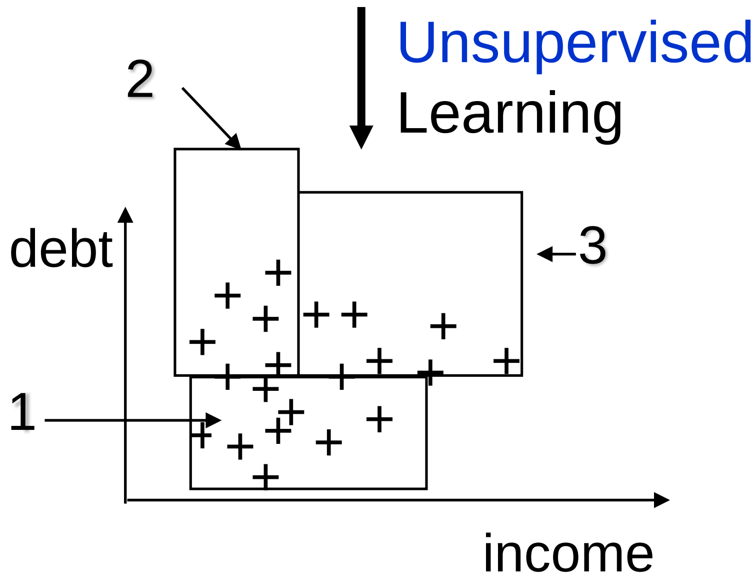
1. above – reject
2. below – accept

Example – Unsupervised Learning



Problem: bank approval of credit (2)

Available data: + Unlabeled



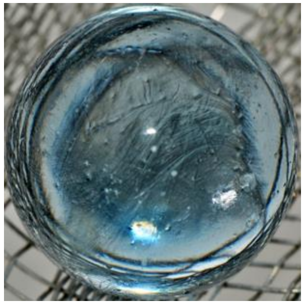
Approval unknown for previous customers.

→ Three classes of customers:

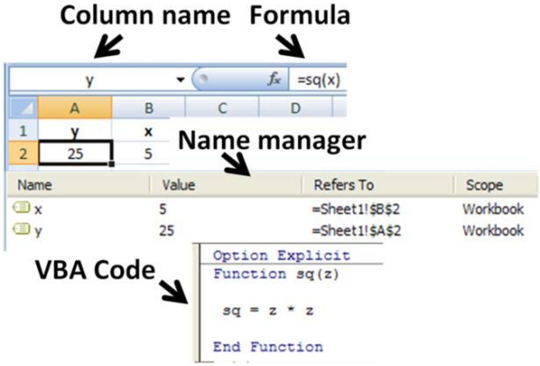
1. Low debt – approved
2. High debt +Low income – reject
3. High debt +High income – additional analysis

Data Mining Tasks Overview (1)

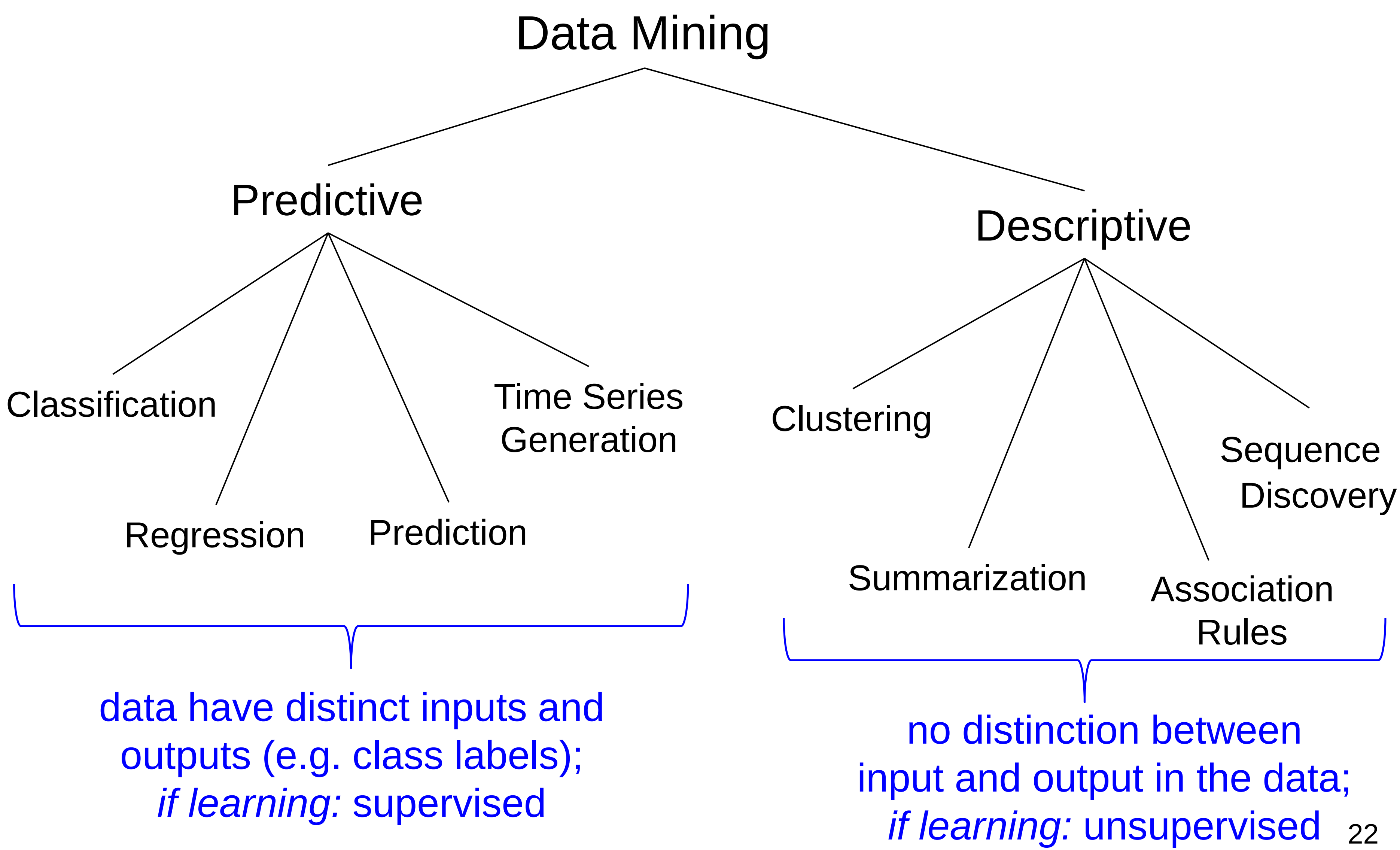
- ***Prediction*** tasks
  - Use some variables to predict unknown or future values of other variables.
    - Produce as a result the ***model***.
    - **Examples**: classification & regression with decision tree, artificial neural network, …



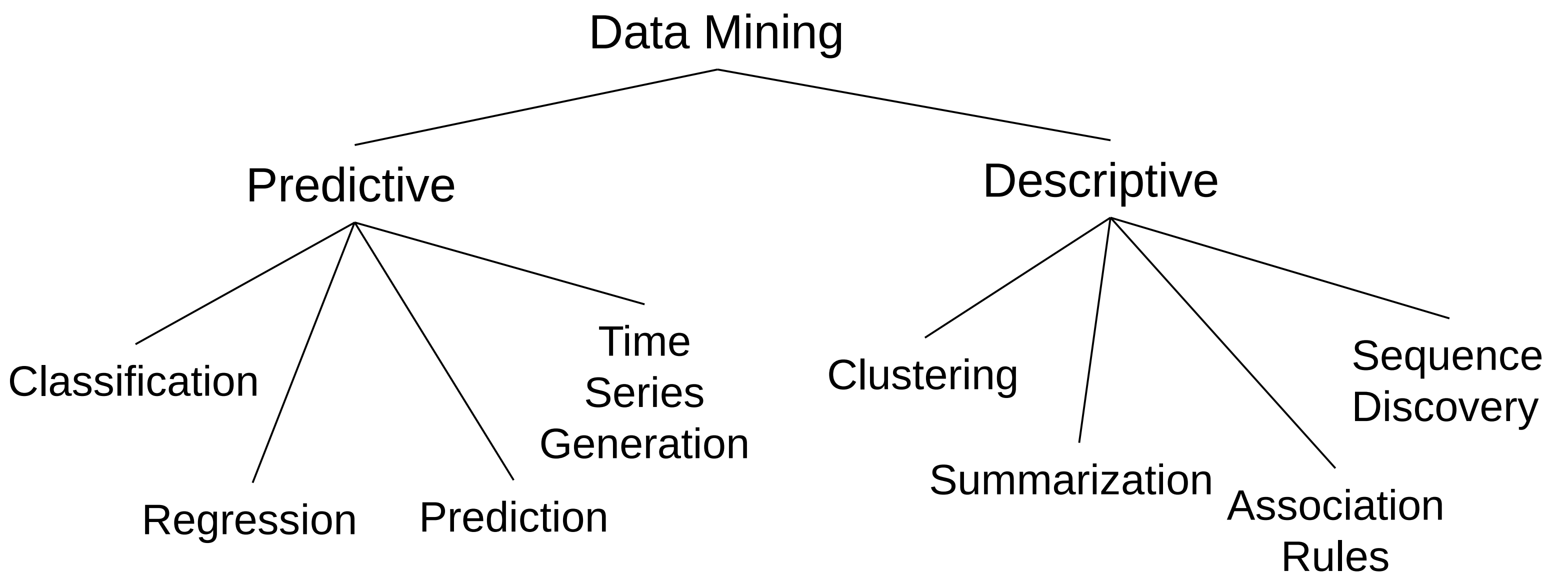
- ***Description*** tasks
  - Find human-interpretable patterns that describe the data.
    - Produce as a result ***information***.
    - **Examples**: rule, graph, summary, …



Data Mining Tasks Overview (2)



Data Mining Tasks Overview (3)



How about Reinforcement Learning?

- *uses* regression (a predictive task) for value estimation
- learns an action strategy to seek rewards
- agent-environment interaction – “data” not of interest

→ beyond data mining, own category

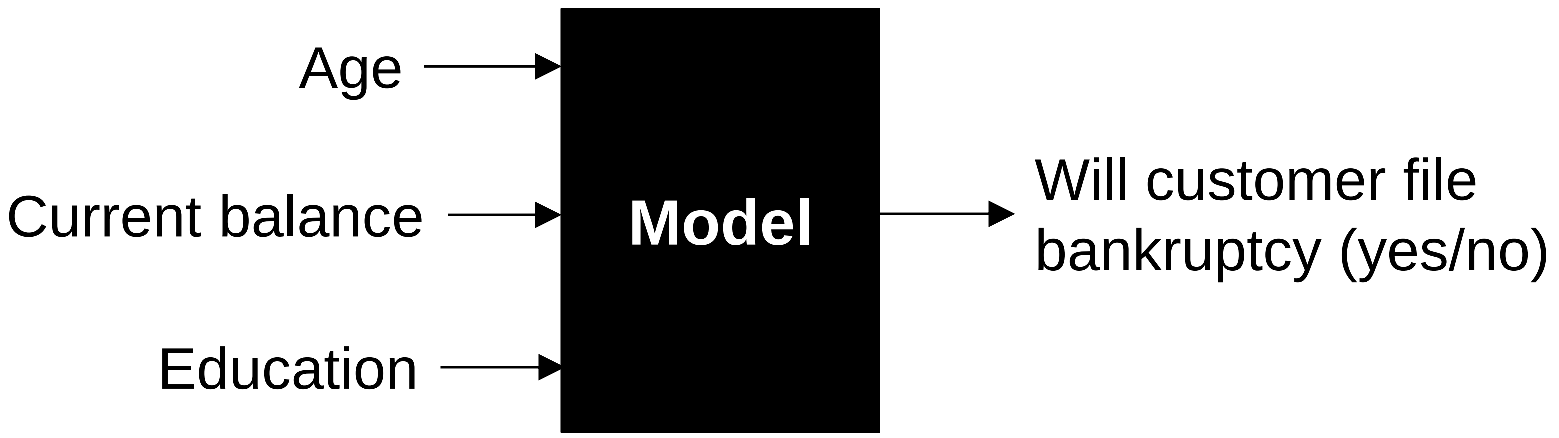
Data Mining Algorithms

A data mining algorithm is a well-defined procedure that takes data as input and produces output in the form of models or patterns.

- ***Well-defined:*** can be encoded in software
- ***Algorithm***: must terminate after some finite number of steps

Predictive Modeling (1)

- A ***black box*** that makes predictions about the future based on information from the past and present.



Predictive Modeling (2)

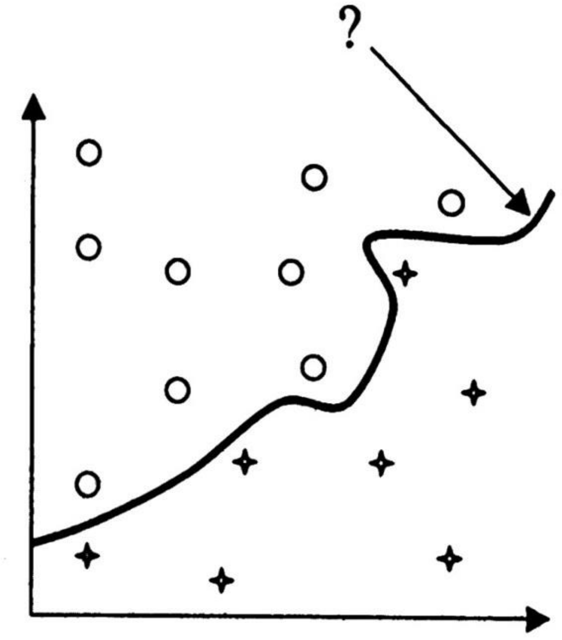
- Predict one variable *Y* given a set of other variables *X*
  - Here *X* could be a n-dimensional vector
- ***Classification***: *Y* is categorical
- ***Regression***: *Y* is real-valued
  - Y may be a m-dimensional vector

$$y = f(x)$$

- This is function approximation: learning the relationship f between Y and X
- Many algorithms in statistics and machine learning
- Often
  - emphasis on predictive accuracy (~ERM)
  - less emphasis on the model itself (~SRM)

Predictive Modeling (3)

- ***Classification*** is a learning function that ***classifies*** a sample into one of several ***predefined*** classes.
  - Given a collection of samples
  - Find a ***model for the class attribute*** as a function of the other attributes
  - Goal: previously unseen samples should be assigned a class as accurately as possible (*test set*)

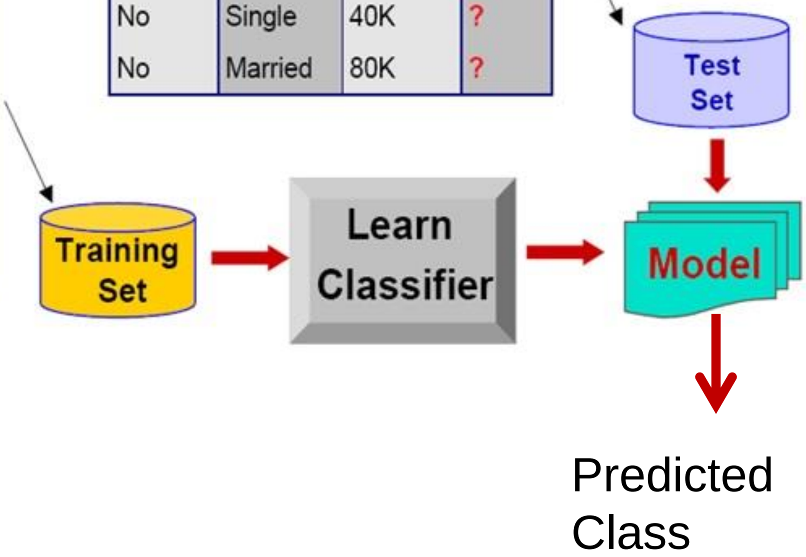


Classification Example

categorical categorical continuous class

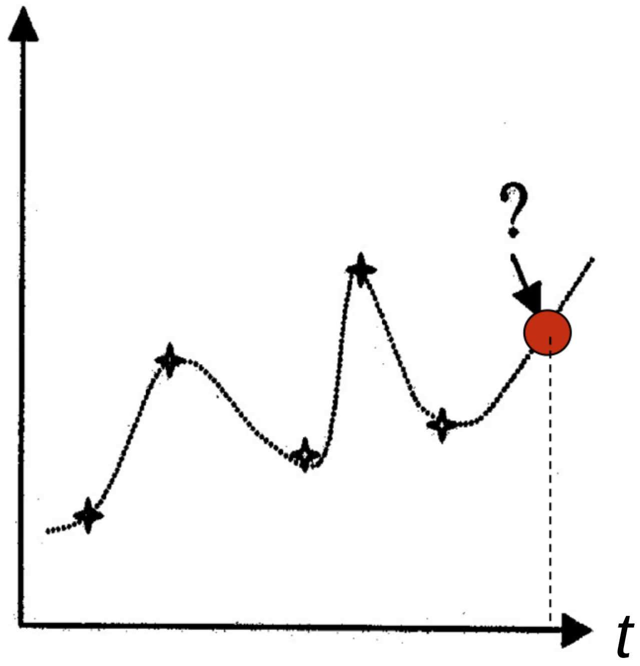
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



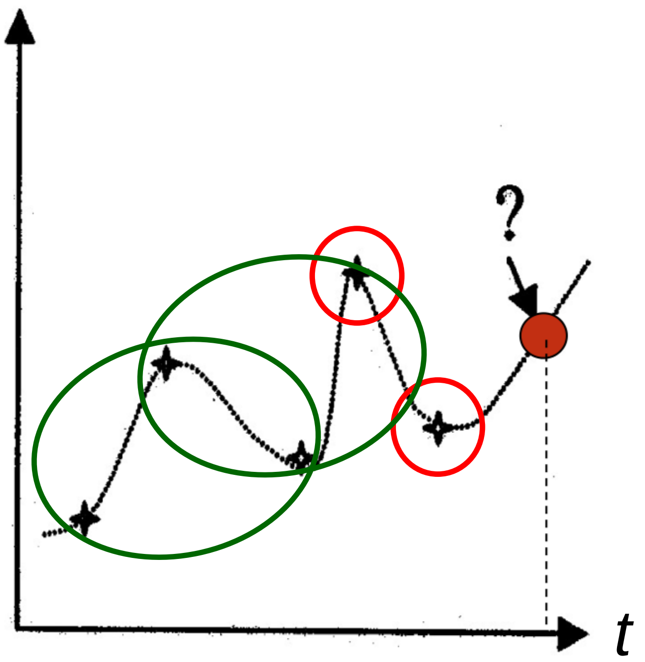
Predictive Modeling (4)

- ***Prediction*** is a learning function that ***maps*** a sample to a real valued ***predicted attribute***.
  - Given a collection of samples
  - Find *a* ***model for the predicted attribute*** as a function of the other attributes
  - Goal: previously unseen sample should be assigned a value as accurately as possible (*test set*)



Predictive Modeling (4)

- ***Prediction*** is a learning function that ***maps*** a sample to a real valued ***predicted attribute***.
  - Often learnt like regression:
  - Inputs are sequence values from time steps: *t-n … t-1*
  - Output is sequence value at time step: *t*



→ learning ***time series***

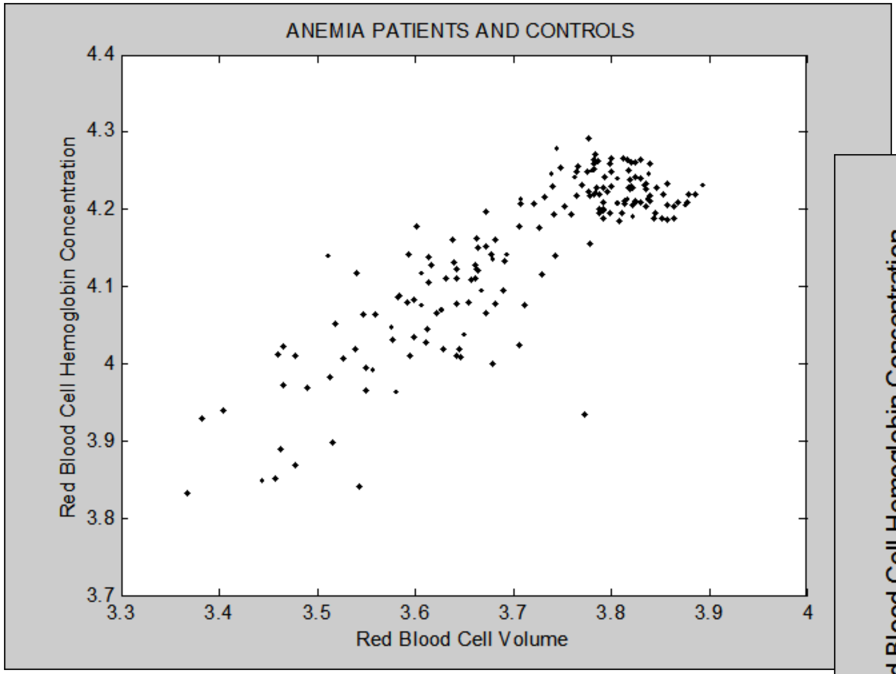
- Apply model recursively to generate the learnt time series

Descriptive Modeling

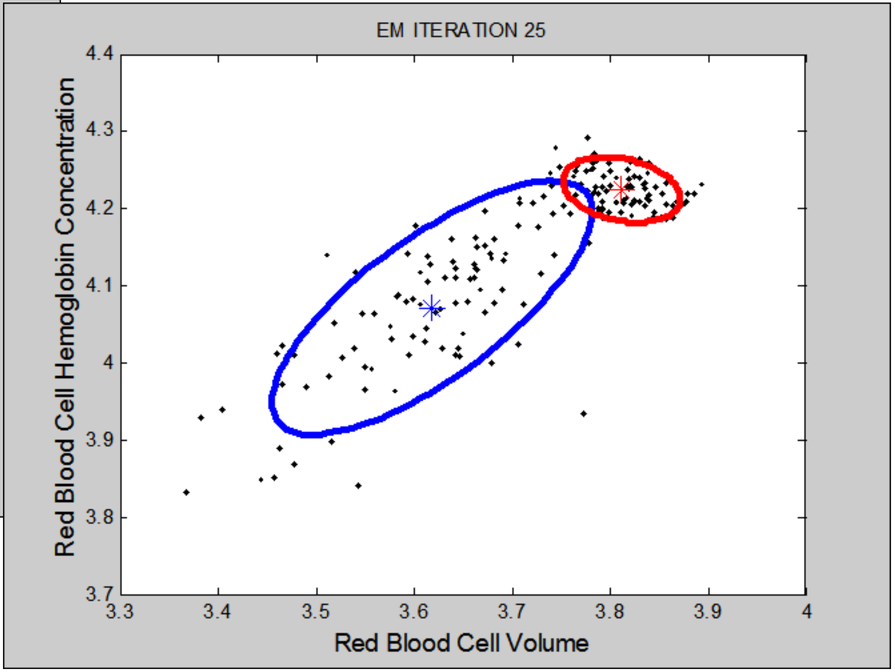
- Goal is to build a ***generative*** or ***descriptive*** model
  - E.g., a model that could simulate the data helping to understand basic characteristics of the process.

- **Examples:**
  - ***Density estimation***:
    - Estimate the joint distribution $P(x_1, \ldots\ldots x_p)$
  - ***Cluster analysis***:
    - Find natural groups in the data and describe them
  - ***Dependency models*** among variables
    - Learn a Bayesian network for the data

Example of Descriptive Modeling



Control Data

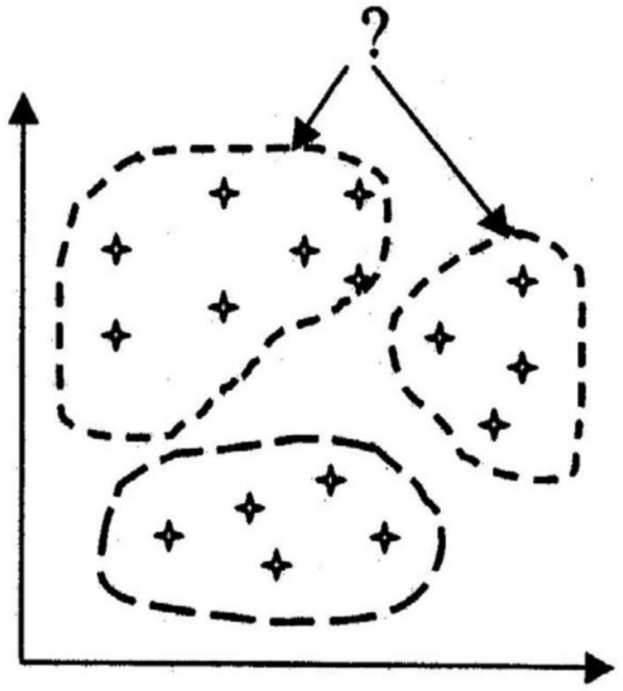


Main Groups

Describe data with elliptic clusters

Descriptive Modeling: Clustering

- Clustering is a ***descriptive*** task where one seeks to identify a finite ***set of categories*** (or clusters) to describe the data
  - Given a collection of samples
  - Find *a* ***model as a function*** of all attributes



Pattern Discovery is a Descriptive Task

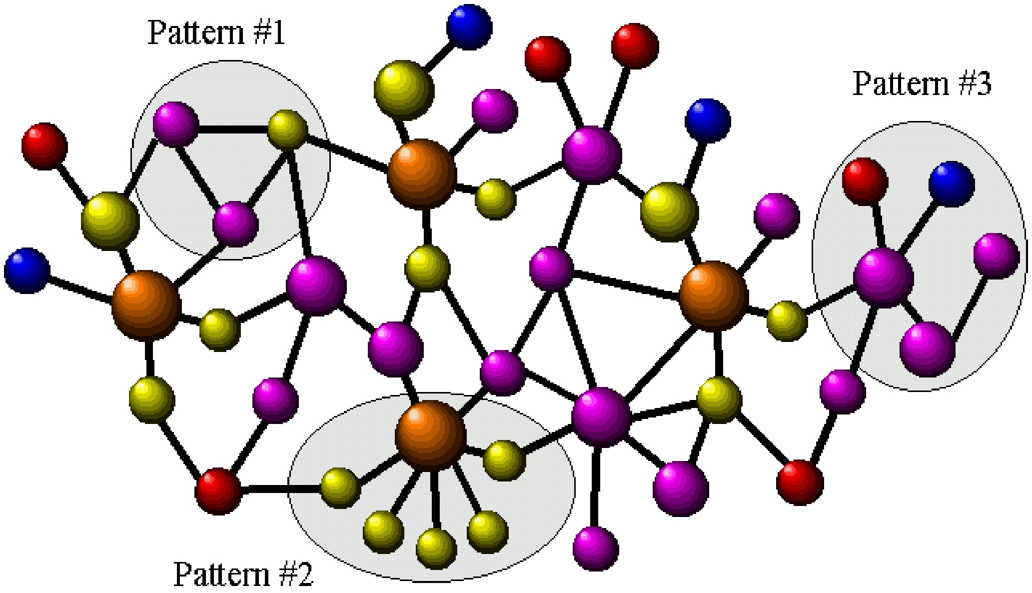
- Gene Analysis Example:

ADACABDABAABBDDBCADDDDBCDDBCCBBCCDADADAADABDBBDA
BABBCDDDCDDABDCBBDBDBCBBABBBCBBABCBBACBBDBAACCAD
DADBDBBCBBCCBBBDCABDDBBADDBBBBCCACDABBABDDCDDBBA
BDBDDBDDBCACDBBCCBBACDCADCBACCADCCCACCDDADCBCADA
DBAACCDDDCBDBDCCCCACACACCDABDDBCADADBCBDDADABCCA
BDAACABCABACBDDDCBADCBDADDDDCDDCADCCBBADABBAAADA
AABCCBCABDBAADCBCDACBCABABCCBACBDABDDDADAABADCDC
CDBBCDBDADDCCBBCDBAADADBCAAAADBDCADBDBBBCDCCBCDC
DCCADAADACABDABAABBDDBCADDDDBCDDBCCBBCCDADADACCC
DABAABBCBDBDBADBBBBCDADABABBDACDCDDDBBCDBBCBBCCD
ABCADDADBACBBCCCDBAAADDDBDDCABACBCADCDCBAAADCADD
ADAABBACCBB

same: **CBBCC**   similar: **CCBBCD**, **CCBCD**

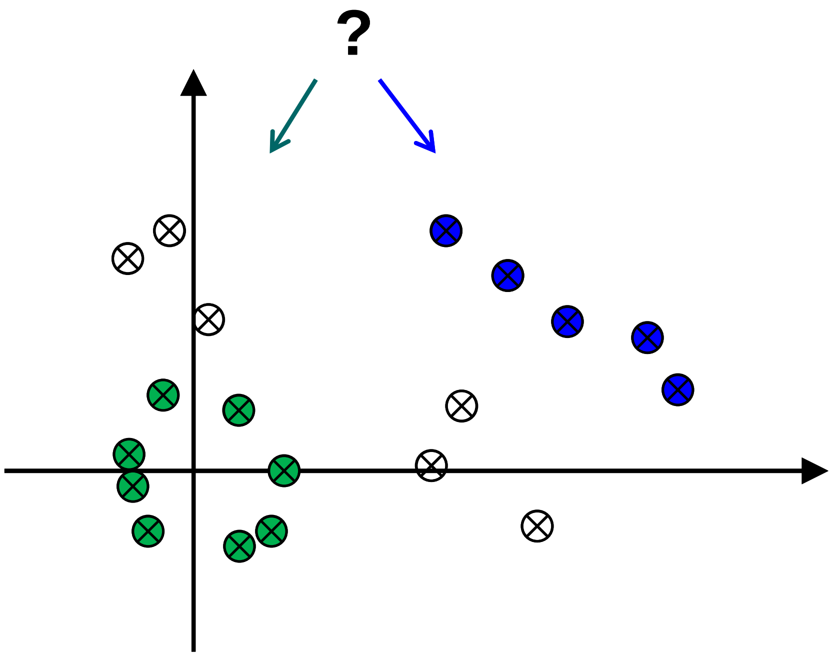
Another Example of Descriptive Modeling

- Learning Directed Graphical Models



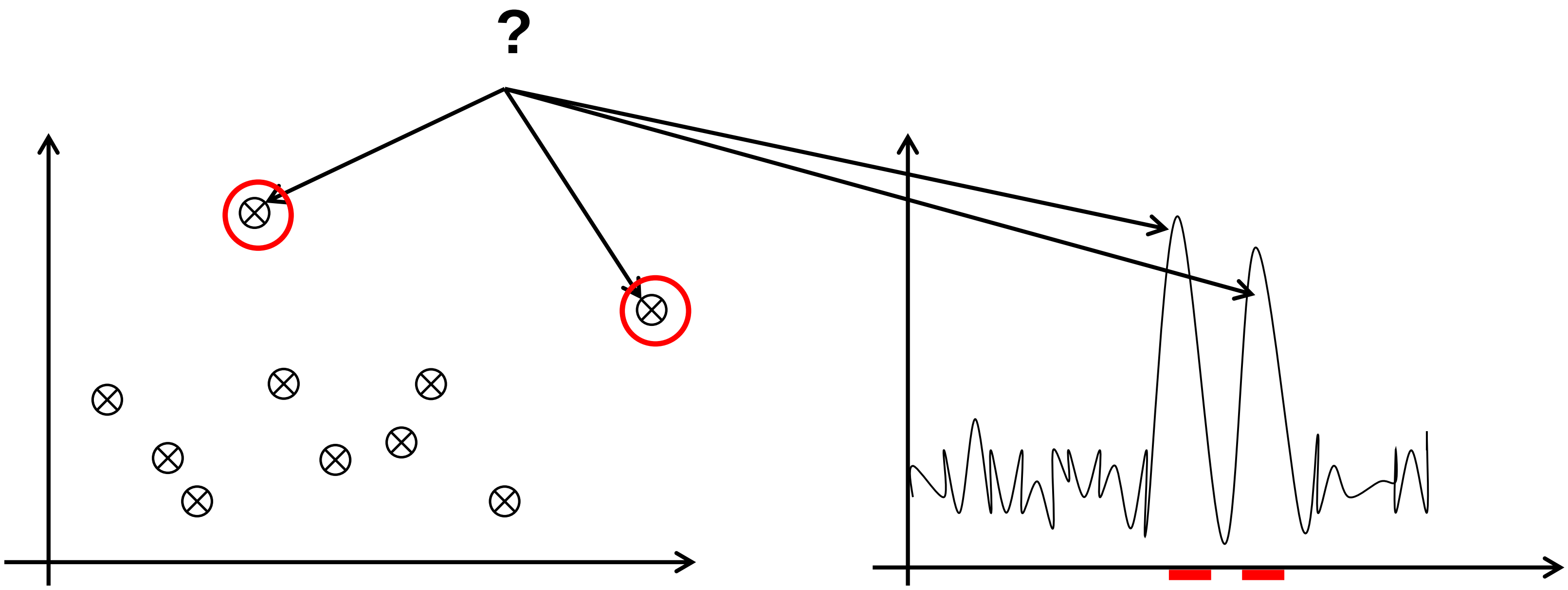
Descriptive Modeling: Dependency Modeling

- The task consists of finding a model that describes ***significant dependency*** in a set (subset) of samples.
  - Given a collection of samples
  - Find significant ***local models*** for subsets of samples



Change- & Deviation Detection

- Focuses on methods for discovering the most ***significant changes*** in large data sets.

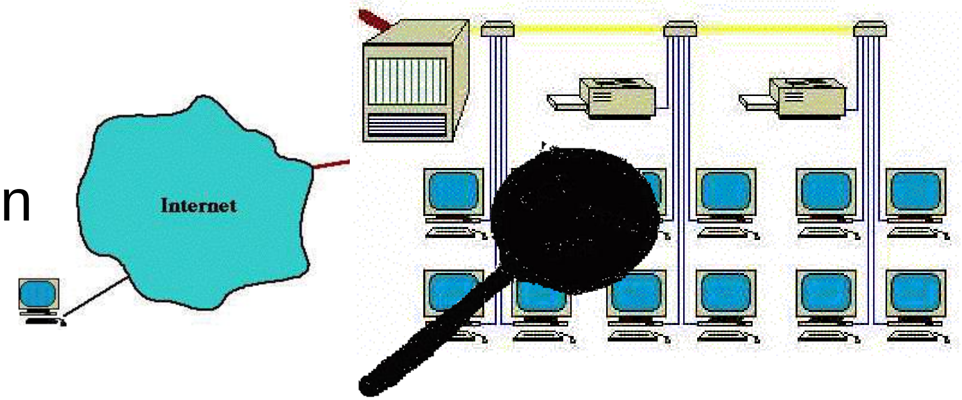


~ outlier detection

Deviation Detection Example

- Detect significant deviations from normal behavior
- Applications:

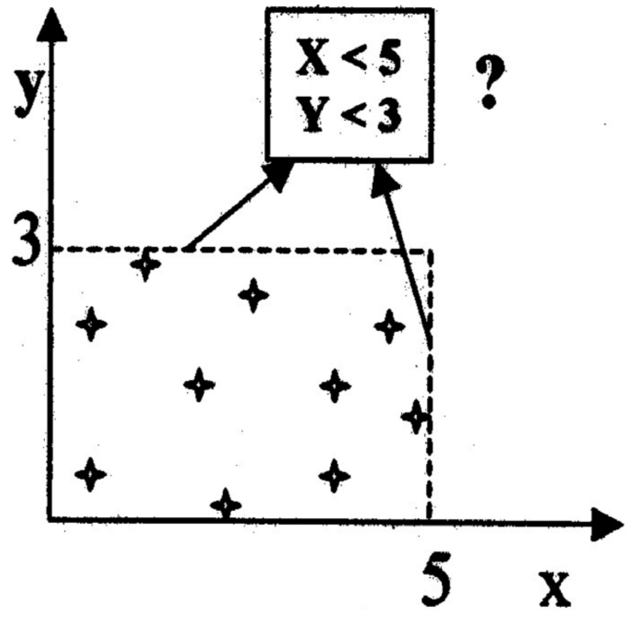
  - Credit card fraud detection
  - Network intrusion detection



  - Early detection of malfunction in mechanical equipment

Descriptive Modeling: Summarization

- Summarization involves methods for finding *a* ***complete description*** for a set of samples.
  - Given a collection of samples
  - Find *a* short, ***simple descriptive model*** for samples as a function of all attributes.



Descriptive Modeling: Summarization

- Special case: ***Text summarization***

- A crude approach is simple:
  1. Identify significant words in the text
     - These are frequent, but generally not frequent in other texts
  2. Select sentences with many of these important words
  3. Done. Possible post-processing:
     - Reduce the extracted sentences, removing unimportant words
     - Adapt sentences to each other for fluent reading

Summarizing: Frequent Pattern Analysis as Association Rules

- ***Frequent pattern***: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of ***frequent itemsets*** and ***association rule mining***
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?
  - What are the subsequent purchases after buying, e.g. a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, DNA sequence analysis.

Outline

1. k-Nearest Neighbors

2. Case-based Reasoning

3. Overview Predictive vs. Descriptive Modeling

▶ Association Rules – Apriori algorithm

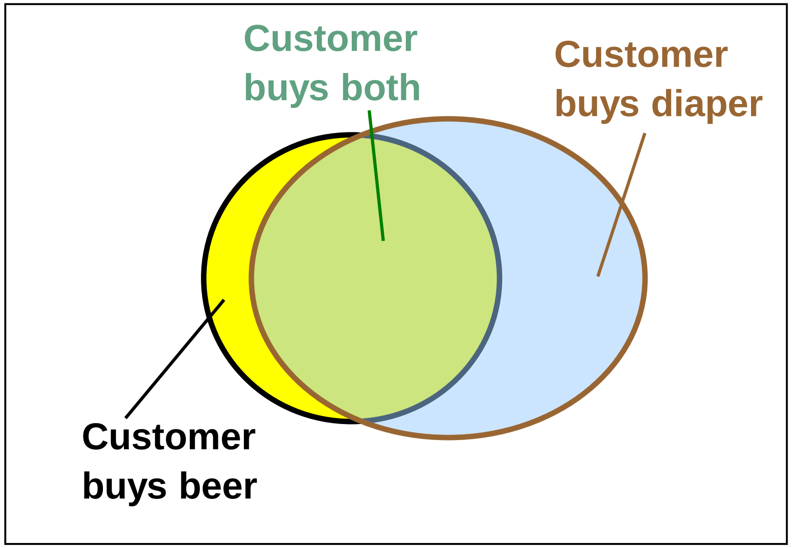
Why is freq. Pattern Mining important?

Broad Applications of Frequent Pattern Mining

- Frequent pattern: An intrinsic property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Finding and analyzing sequential, structural (e.g., sub-graph) patterns
    - E.g. in spatiotemporal, multimedia, time-series, and stream data
  - Discriminate data based on frequent patterns
    - Clustering
    - Classification (if labels available)
  - Semantic data compression
  - Data warehousing
    - E.g. access data based on frequent patterns

Basic Concepts: Frequent Patterns

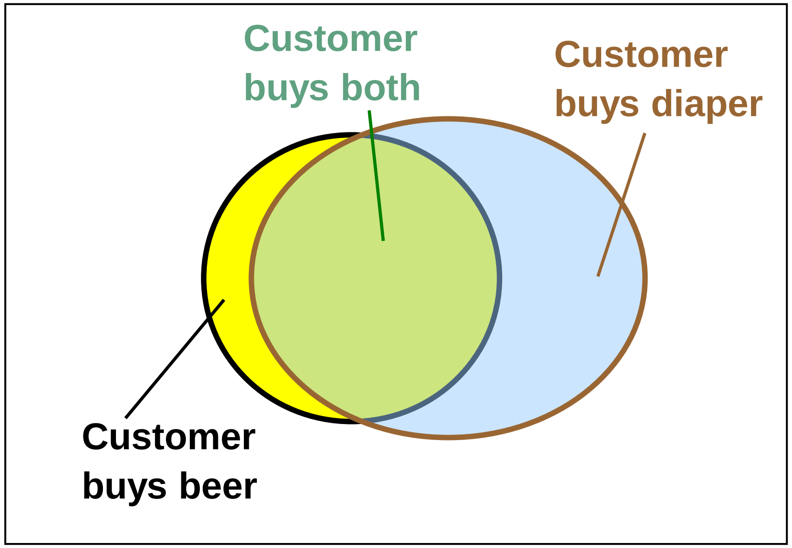
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- ***itemset***: a set of one or more items
- ***k-itemset*** $X = \{x_1, \ldots, x_k\}$
- ***(absolute) support***, *or,* ***support count*** of X: frequency of occurrence of an itemset X
- ***(relative) support*** of X: fraction of transactions that contains X (~ the probability that a transaction contains X)
- An itemset X is ***frequent*** if its support is no less than a *minsup* threshold

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



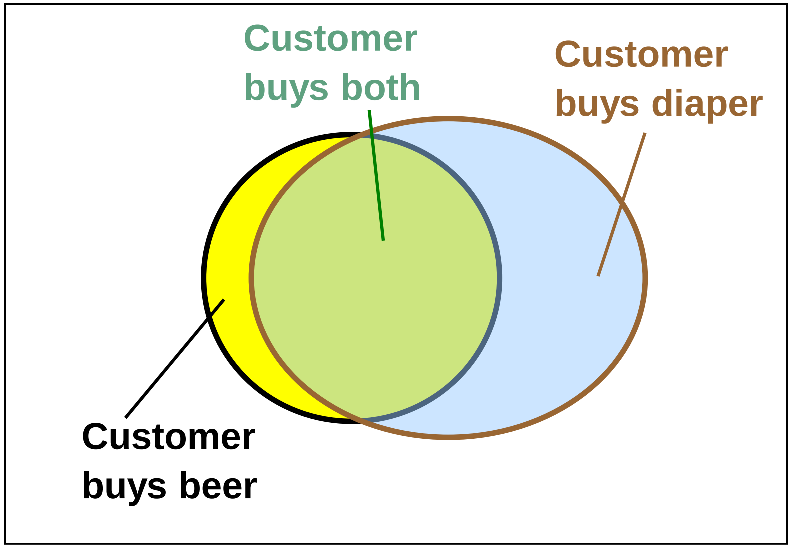
is associated with

- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - ***Support*** s: probability that a transaction contains an itemset
  - ***Confidence*** c: conditional probability that a transaction having X also contains *Y*

$$Confidence(X \rightarrow Y) = Support(X \cup Y) / Support(X)$$

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- *Let minsup = 50%, minconf = 50%*
- *Freq. Patterns:*
  - Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3, {Beer, Diaper}: 3
- Association rules:
  - Beer → Diaper (s=60%, c=100%)
  - Diaper → Beer (s=80%, c=75%)
  - many more …

Interpreting Association Rules

assume:
conf=0.7

1. {Milk, Bread} → Eggs
2. {Milk, Bread} → Beer

is not the same as $\not\Downarrow$ $\Uparrow$

3. {Milk, Bread} → {Eggs, Beer}

However, from 3. follows: $\Downarrow$ $\not\Uparrow$

4. {Milk, Bread, Eggs} → Beer
5. {Milk, Bread, Beer} → Eggs

Downward Closure Property and Scalable Mining Methods

- The ***downward closure*** property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - I.e., every transaction having **{beer, diaper, nuts}** also contains **{beer, diaper}**
- *Scalable* mining methods: Three major approaches
  - **Apriori** (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: a Candidate Generation & Test Approach

- Apriori ***pruning principle***: If there is any itemset which is infrequent, its superset should not be generated/tested! [Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94]
- Method:

  according to parameter: *minsup*

  - Initially, scan DB once to get ***frequent*** 1-itemsets
  - From length k ***frequent*** itemsets, ***generate*** length (k+1) ***candidate*** itemsets
  - ***Test*** the candidates against DB
  - Terminate when no frequent or candidate set can be generated
- ***Apriori*** name: use of prior knowledge of frequent itemset

The Apriori Algorithm (Pseudo-Code)

```
Ck: Candidate itemset of size k
Lk: Frequent itemset of size k

L1 = {frequent items};
for (k = 1; Lk !=∅; k++) do begin
    Ck+1 = candidates generated from Lk;
    for each transaction t in database do
      increment the count of all candidates in Ck+1
      that are contained in t
    end
    Lk+1  = candidates in Ck+1 with min_support
end
return ∪k Lk;
```

The Apriori Algorithm – an Example

$sup_{min} = 2$

Transaction DB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan for count →

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→ L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan →

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

→ L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan →

L_3

Itemset	sup
{B, C, E}	2

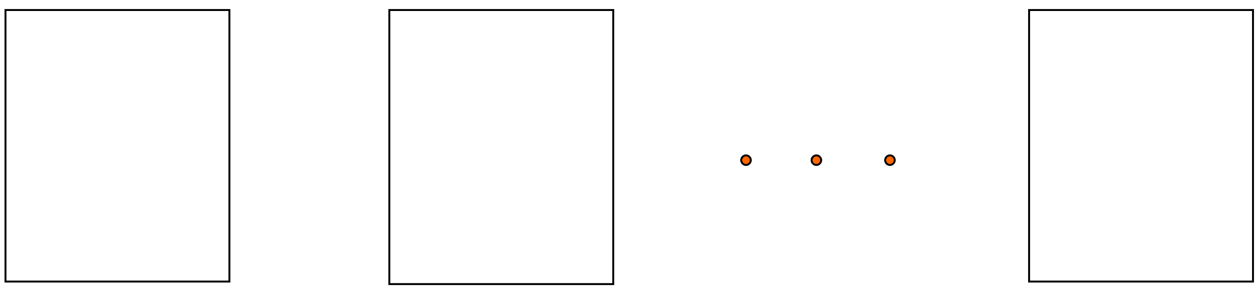
Further Improvement of the Apriori Method

- Major computational challenges
  - Multiple scans of transaction database
  - Huge number of candidates:
    - minsup low → potentially an exponential number of frequent itemsets
    - Worst case: M^N

      where M: # distinct items; N: max length of transactions
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

Partition: Local & Global Scan of Database

- Any itemset that is potentially frequent with sup σ in DB must be frequent with sup σ/*k* in at least one of *k* partitions of DB
  - Scan 1: partition database and find *local* frequent patterns



$$DB_1 + DB_2 + \cdots + DB_k = DB$$

$$sup_1(i) \quad sup_2(i) \quad sup_k(i) \quad sup(i)$$

  - Scan 2: consolidate *global* frequent patterns

[A. Savasere, E. Omiecinski and S. Navathe, *VLDB'95*]

Summary

- k-Nearest Neighbors & Case-Based Reasoning
  - Prediction without learning
- Tasks in data mining
  - Predictive tasks, e.g. via supervised learning
  - Descriptive tasks, e.g. via unsupervised learning
- Frequent pattern mining: Apriori algorithm
  - Descriptive task without learning