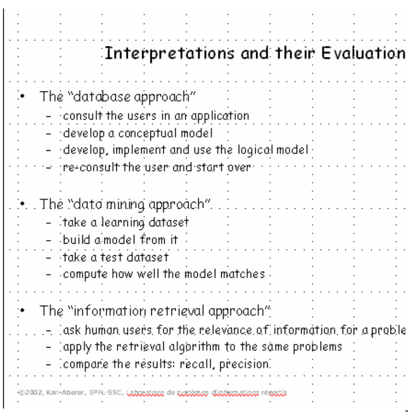
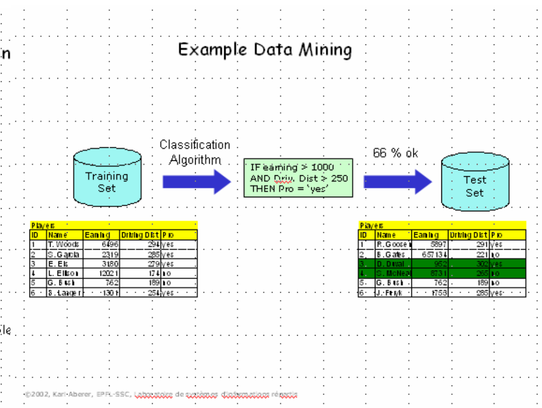
Chapter 4 – Data Mining
A Short Introduction

Today's Question

1. Data Mining Overview
2. Association Rule Mining
3. Clustering
4. Classification





1. Data Mining Overview

- Data acquisition and data storage result in huge databases
  - Supermarket and credit card transactions (market analysis)
  - Scientific data
  - Web analysis (browsing behavior, advanced information retrieval)
- Definition

  Data mining is the analysis of large observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful for the data owner
- Extraction of information from data

The wide-spread use of distributed information systems leads to the construction of large data collections in business, science and on the Web. These data collections contain a wealth of information, that however needs to be discovered. Businesses can learn from their transaction data more about the behavior of their customers and therefore can improve their business by exploiting this knowledge. Science can obtain from observational data (e.g. satellite data) new insights on research questions. Web usage information can be analyzed and exploited to optimize information access.

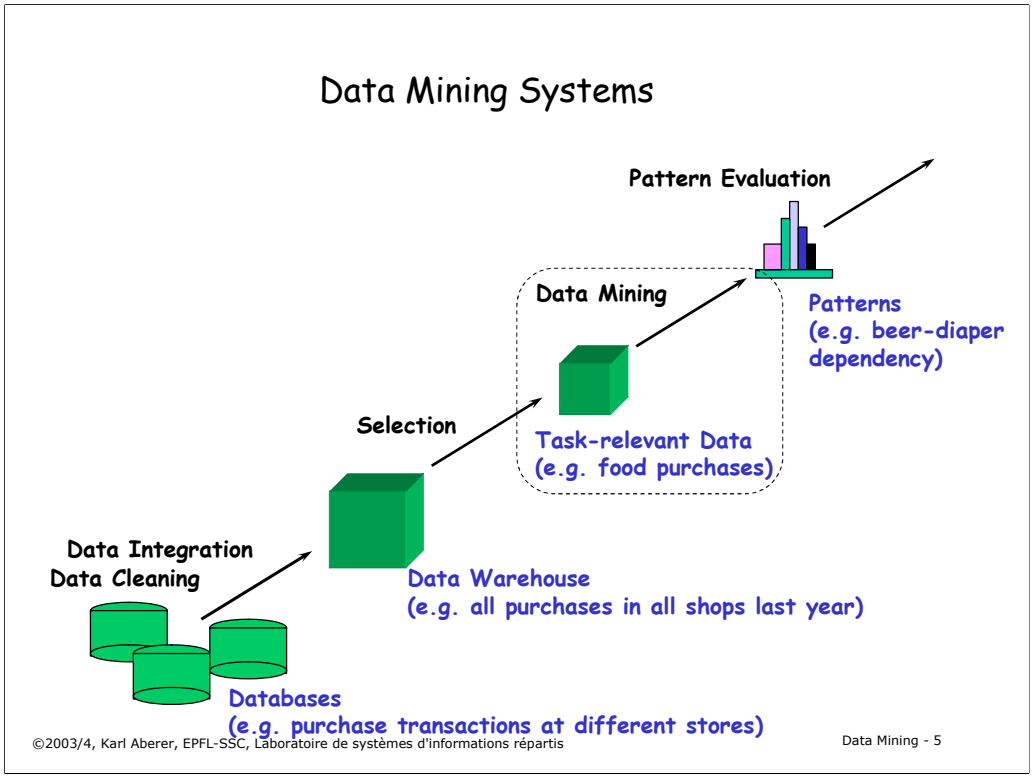
Data mining provides methods that allow to extract from large data collections unknown relationships among the data items that are useful for decision making. Thus data mining generates novel, unsuspected interpretations of data.

The Classical Example: Association Rule Mining

- Market basket analysis
  - Given a database of purchase transactions and for each transaction a list of purchased items
  - Find rules that correlate a set of items occurring in a list with another set of items
- Example
  - 98% of people who purchase tires and auto accessories also get automotive services done
  - 60% of people who buy diapers also buy a beer
  - 90% of people who buy Neal Stephenson's "Snow Crash" at amazone also buy "Cryptonomicon"
  - etc.

The classical example of a data mining problem is "market basket analysis". Stores gather information on what items are purchased by their customers. The hope is, by finding out what products are frequently purchased jointly (i.e. are associated with each other), being able to optimize the marketing of the products (e.g. the layout of the store) by better targeting certain groups of customers. A famous example was the discovery that people who buy diapers also frequently buy beers (probably exhausted fathers of small children). Therefore nowadays one finds frequently beer close to diapers (and of course also chips close to beer) in supermarkets. Similarly, amazon exploits this type of associations in order to propose to their customers books that are likely to match their interests.

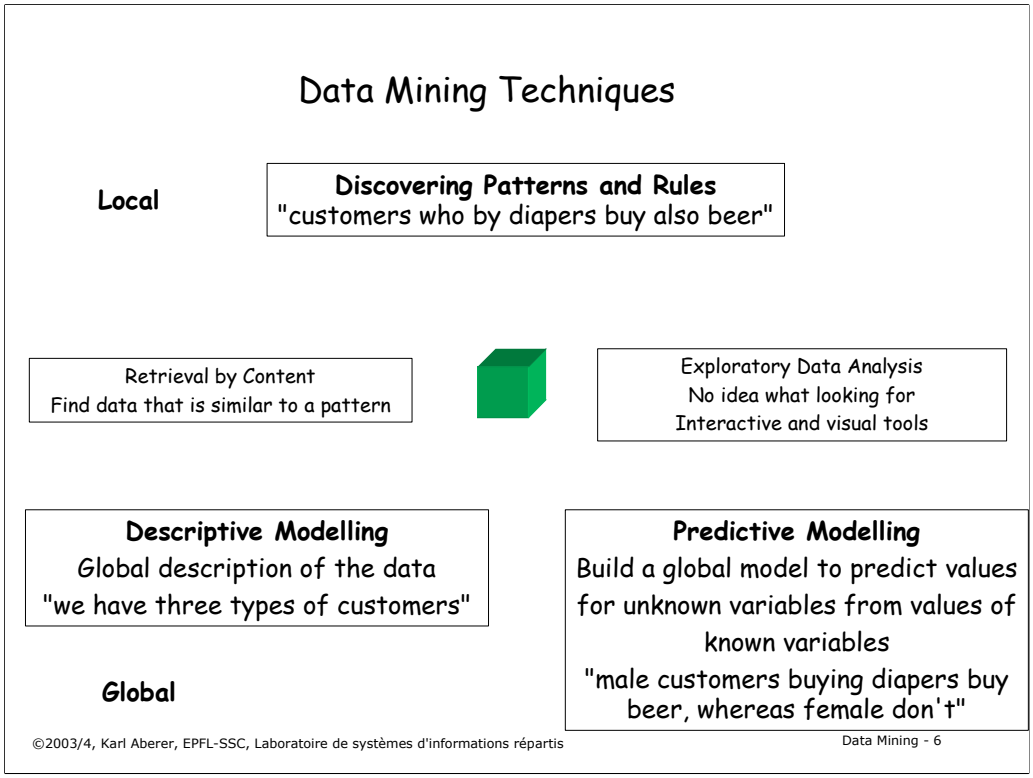
This problem was the starting point for one of the best known data mining techniques: association rule mining.



The task of discovering interesting patterns is part of a larger process supported by data mining systems.

- For being able to mine data, first the data needs to be collected from the available data sources. Since these data sources can be distributed and heterogeneous databases, database integration techniques need to be applied. The integrated data is kept in so-called data warehouses, databases replicating and consolidating the data from the various data sources. An important concern when integrating the data sources is data cleansing, i.e. removing inconsistent and faulty data as far as possible. The tasks of data integration and data cleansing are supported by so-called data warehousing systems.
- Once the data is consolidated in a data warehouse, for specific data mining tasks, i.e. tasks having a specific purpose in mind, the data can be selected from the data warehouse. This task-specific data collections are often called data-marts. The data-mart is the database to which the specific data mining task, i.e. the discovery process, is applied.
- The data mining task is the process of detecting interesting patterns in the data. This is what generally is understood as data mining in the narrow sense. We will introduce in the course examples of techniques that are used to perform this task (e.g. association rule mining).
- Once specific patterns are detected they can be further processed. Further processing can imply the evaluation of their "interestingness" for the specific problem studied and the implementation of certain actions to react on the discovery of the pattern.

Each of the steps described can influence the preceding steps. For example, patterns or outliers detected during data mining may indicate the presence of erroneous data rather than of interesting features in the source databases. This may imply adaptations of the data cleansing during data integration.



Data mining techniques can be classified according to the goals they pursue and the results they provide.

A basic distinction is made among techniques that provide a global statement on the data, in the form of summaries and globally applicable rules, or that provide local statements only, in the form of rare patterns or exceptional dependencies in the data.

The example of association rule mining is a typical case of discovering local patterns. The rules obtained are unexpected (unprobable) patterns and typically relate to small parts of the database only.

Techniques for providing global statements on the data are further distinguished into techniques that are used to "simply" describe the data and into techniques that allow to make predictions on data if partial data is known. Descriptive modeling techniques provide compact summaries of the databases, for example, by identifying clusters of similar data items. Predictive modeling techniques provide globally applicable rules for the database, for example, allowing to predict properties of data items, if some of their properties are known.

Information retrieval is usually also considered as a special case of data mining, where given patterns are searched for in the database. Finally, exploratory data analysis is used when no clear idea exists, what is being sought for in the database. It may serve as a preprocessing step for more specific data mining tasks.

Components of a Data Mining Algorithm

- Model or pattern structure
  - Which kind of global model or local pattern is searched
  - Vector representation of documents
- Score function
  - Determine how well a given data set fits a model or pattern
  - Similarity of vectors
- Optimization and search method
  - Finding best parameters for a global model: optimization problem
  - Finding data satisfying a pattern: search problem
  - Search the k nearest neighbors
- Data management strategy
  - Handling very large datasets
  - Inverted files

Each data mining method can be characterized in terms of four aspects:

- The models or patterns that are used to describe what is searched for in the data set. Typical models are dependency rules, clusters and decision trees.
- The scoring functions that are used to determine how well a given dataset fits the model. This is comparable to the similarity functions used in information retrieval.
- The method that is applied in order to find data in the dataset that scores well with respect to the scoring function. Normally this requires efficient search algorithms that allow to identify those models that fit the data well according to the scoring functions.
- Finally the scalable implementation of the method for large datasets. Here indexing techniques and efficient secondary storage management are applied.

In particular the last two issues differentiate data mining from related areas like statistics and machine learning: scalability for large databases is a key problem in data mining and only statistical and machine learning techniques that scale well are applicable for data mining.

For illustration we identify the components of information retrieval, when looked at as data mining method.

Summary

- What is the purpose of data mining ?
- Which preprocessing steps are required for a data mining task ?
- Which are the four aspects that characterize a data mining method ?
- What is the difference between classification and prediction ?
- Explain of how information retrieval can be characterized as a data mining method ?

2. Association Rule Mining

- Search patterns given as association rules of the form

  Body ⇒ Head [support, confidence]

  Body: property of an object x e.g. a transaction, a person
  Head: property probable to be implied by Body
  support, confidence: measures on validity of the rule

- Examples
  - buys(x, "diapers") ⇒ buys(x, "beers") [0.5%, 60%]
  - major(x, "CS") ∧ takes(x, "DB") ⇒ grade(x, "A") [1%, 75%]

- Problem: Given
  (1) database of transactions
  (2) each transaction is a list of items
  Find: all rules that correlate the presence of one set of items with that of another set of items

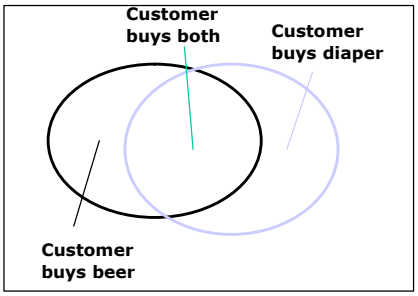
Association rule mining is a technique for discovering unsuspected data dependencies and is one of the best known data mining techniques. The basic idea is to identify from a given database, consisting of itemsets (e.g. shopping baskets), whether the occurrence of specific items, implies also the occurrence of other items with a relatively high probability. In principle the answer to this question could be easily found by exhaustive exploration of all possible dependencies, which is however prohibitively expensive. Association rule mining thus solves the problem of how to search efficiently for those dependencies.

Single vs. Multidimensional Association Rules

- Single-dimensional rules
  buys(X, "milk") Þ buys(X, "bread")
- Multi-dimensional rules: more than 2 dimensions or predicates
  age(X,"19-25") Ù buys(X, "popcorn") Þ buys(X, "coke")
- Transformation into single-dimensional rules:
  use predicate/value pairs as items
  customer(X, [age, "19-25"]) Ù customer(X, [buys, "popcorn"])
  Þ customer(X, [buys,"coke"])
- Simplified Notation for single dimensional rules
  {milk} Þ {bread}
  {[age, "19-25"], [buys, "popcorn"]} Þ {[buys,"coke"]}

In the "logical" notation we have used before in order to express association rules, it was possible to establish dependencies among different types of predicates applied to the items. These general types of rules are called multi-dimensional association rules. However, it is straightforward to transform multi-dimensional association rules into single-dimensional rules, by considering different predicates applied to the same items as different items. Therefore in the following we will only consider single-dimensional association rules.

Support and Confidence



Transaction ID	Items Bought
2000	beer, diaper, milk
1000	beer, diaper
4000	beer, milk
5000	milk, eggs, apple

support = probability that body and head occur in transaction
confidence = probability that if body occurs also head occurs

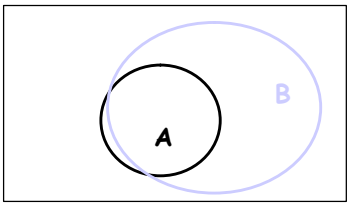
Let minimum support 50%
and minimum confidence 50%

{beer} ⇒ {diaper} [50%, 66.6%]
{diaper} ⇒ {beer} [50%, 100%]

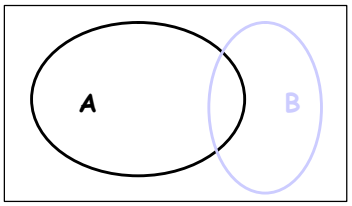
This example illustrates the basic notions used in association rule mining: transactions, itemsets, support and confidence. Transaction consist of a transaction identifier and an itemset. The itemset is the set of items that occur jointly in a transactions (e.g. the items bought). Support is the number of transactions in which the association rule holds, i.e. in which all items of the rule occur (e.g. both beer and diaper). If this number is too small, probably the rule is not important or accidentally true. Confidence is the probability that in case the head of the rule (the condition) is satisfied also the body of the rule (the conclusion) is satisfied. This indicates to which degree the rule is actually true, in the cases where the rule is applicable.

Support and Confidence: Possible Situations

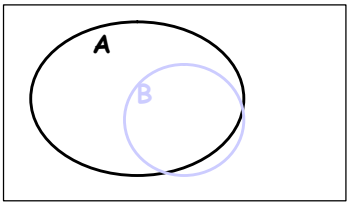
- Assume support for $A \cup B$ is high (above threshold)



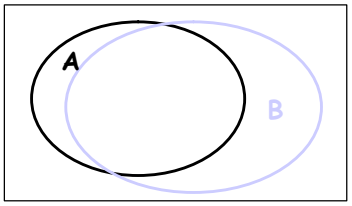
Conf(A ⇒ B) high
Conf(B ⇒ A) low



Conf(A ⇒ B) low
Conf(B ⇒ A) low



Conf(A ⇒ B) low
Conf(B ⇒ A) high



Conf(A ⇒ B) high
Conf(B ⇒ A) high

This figure illustrates the meaning and importance of the "directionality" of association rules. We assume that in all cases the intersection areas (i.e. the support) are above the required threshold. Then four cases are possible as shown. Thus association rules not only express a high probability of co-occurrence of items, such as in the last case, but also conditional dependencies among the occurrences of items (or inclusion relationships).

Definition of Association Rules

Terminology and Notation

- Set of all items I, subset of I is called itemset
- Transaction (tid, T), $T \subseteq I$ itemset, transaction identifier tid
- Set of all transactions D (database), Transaction $T \in D$

Definition of Association Rules $A \Rightarrow B$ [s, c]

- A, B itemsets ($A, B \subseteq I$)
- $A \cap B$ empty
- support s = probability that a transaction contains $A \cup B$
  $= P(A \cup B)$
- confidence c = conditional probability that a transaction having A also contains B
  $= P(B|A)$

Example: Items I = {apple, beer, diaper, eggs, milk}
Transaction (2000, {beer, diaper, milk})
Association rule {beer} $\Rightarrow$ {diaper} [0.5, 0.66]

This is a summary of the basic notations and notions used in association rule mining.

Frequent Itemsets

Transaction ID	Items Bought
2000	beer, diaper, milk
1000	beer, diaper
4000	beer, milk
5000	milk, eggs, apple

Frequent Itemset	Support
{beer}	0.75
{milk}	0.75
{diaper}	0.5
{beer,milk}	0.5
{beer,diaper}	0.5

1. $A \Rightarrow B$ can only be an association rule if $A \cup B$ is a **frequent itemset**
   - search for frequent itemsets !
2. Any subset of a frequent itemset is also a frequent itemset (apriori property -> apriori algorithm)
   - e.g., if {beer,diaper} is a frequent itemset, both {beer} and {diaper} are frequent itemset
3. Therefore iteratively find frequent itemsets with increasing cardinality from 1 to k **(k-itemsets)**
   - reduces number of possible candidates in search for larger frequent itemsets

Here we summarize the most important ideas that will allow to search for association rules efficiently.

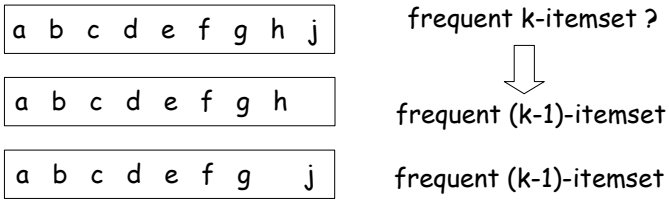
First, a necessary condition for finding an association rule of form A->B is sufficiently high support. Therefore, for finding such rules, we have first to find itemsets within the transactions that occur sufficiently frequent. These are called *frequent itemsets*.

Second we can observe that any subset of a frequent itemset is necessarily also a frequent itemset (this is called the apriori property).

Third, we can exploit this observation in order to reduce the number of itemsets that need to be considered in the search. Once frequent itemsets of lower cardinality are found, only itemsets of larger cardinality need to be considered that contain one of the frequent itemsets already found. This allows to reduce the search space drastically as we will see.

Exploiting the Apriori Property (1)

- If we know the frequent (k-1)-itemsets, which are candidates for being frequent k-itemsets ?



- If we know all frequent (k-1)-itemsets Lk-1, then we can construct a candidate set Ck for frequent k-itemsets by joining two frequent (k-1)-itemsets that differ by exactly 1 item: **join step**
  - only these itemsets CAN BE frequent k-itemsets

Assume that we know frequent itemsets of size k-1. Considering a k-itemset we can immediately conclude that by dropping two different items we have two frequent (k-1) itemsets. From another perspective this can be seen as a possible way to construct k-itemsets. We take two (k-1) item sets which differ only by one item and take their union. This step is called the join step and is used to construct POTENTIAL frequent k-itemsets.

Algorithm for Creating Candidates

- Suppose the items in L_{k-1} are increasingly sorted in a list, then C_k is created through the following SQL statement

insert into C_k

SELECT **p.item$_1$, p.item$_2$, …, p.item$_{k-1}$, q.item$_{k-1}$**
FROM **L_{k-1} p, L_{k-1} q**
WHERE **p.item$_1$=q.item$_1$, …, p.item$_{k-2}$=q.item$_{k-2}$, p.item$_{k-1}$ < q.item$_{k-1}$**

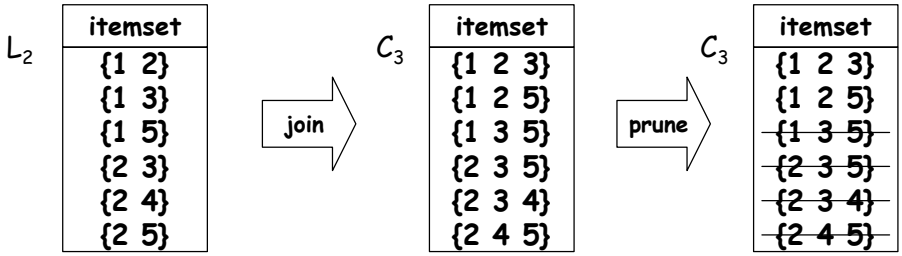
item1	item2	…	itemk-2	itemk-1
i1,1	i2,1	…	ik-2,1	ik-1,1
i1,1	i2,1	…	ik-2,1	ik-1,2
…	…	…	…	…
i1,1	i2,1	…	ik-2,1	ik-1,p1
i1,2	i2,2		ik-2,2	ik-1,p1+1
i1,2	i2,2		ik-2,2	ik-1,p1+2
…	…	…	…	…
i1,2	i2,2		ik-2,2	ik-1,p2

take all possible pairs

We may express the step of creating all possible combinations of (k-1) itemsets by the SQL query shown above, assuming the itemsets are stored as relations with attributes item$_1$,…,item$_k$. The table illustrates of how these combinations are obtained. For each subset of items that share the first k-2 items, we construct all possible k-itemsets, by taking all possible, ordered pairs from the last column.

Exploiting the Apriori Property (2)

- A candidate itemset still not necessarily satisfies the apriori property



- After generating the candidate set C_k eliminate all itemsets for which not ALL (k-1)-itemsets are elements of L_{k-1}, i.e. are frequent (k-1) itemsets: **prune step**

- Only then count the remaining candidate k-itemsets in the database and eliminate those that are not frequent.

The k-itemsets constructed in the join step not necessarily are frequent k-itemsets. One possible reason is that they contain some subset of items which is not frequent. These are eliminated in a prune step, by considering all itemsets of lower cardinality that have been constructed earlier.

After that, it is still possible that among the remaining k-itemsets some are not frequent when determining their frequency in the database. The important point is that this last check, which is expensive as it requires access to the complete database, needs to be performed for much fewer itemsets, since many possibilities have been eliminated in the join and prune step.

Generating Frequent Itemsets: The Apriori Algorithm

k:=1; L_k := {frequent items in D};

while L_k !=∅
{

C_{k+1} := candidates generated from L_k by joining and pruning;

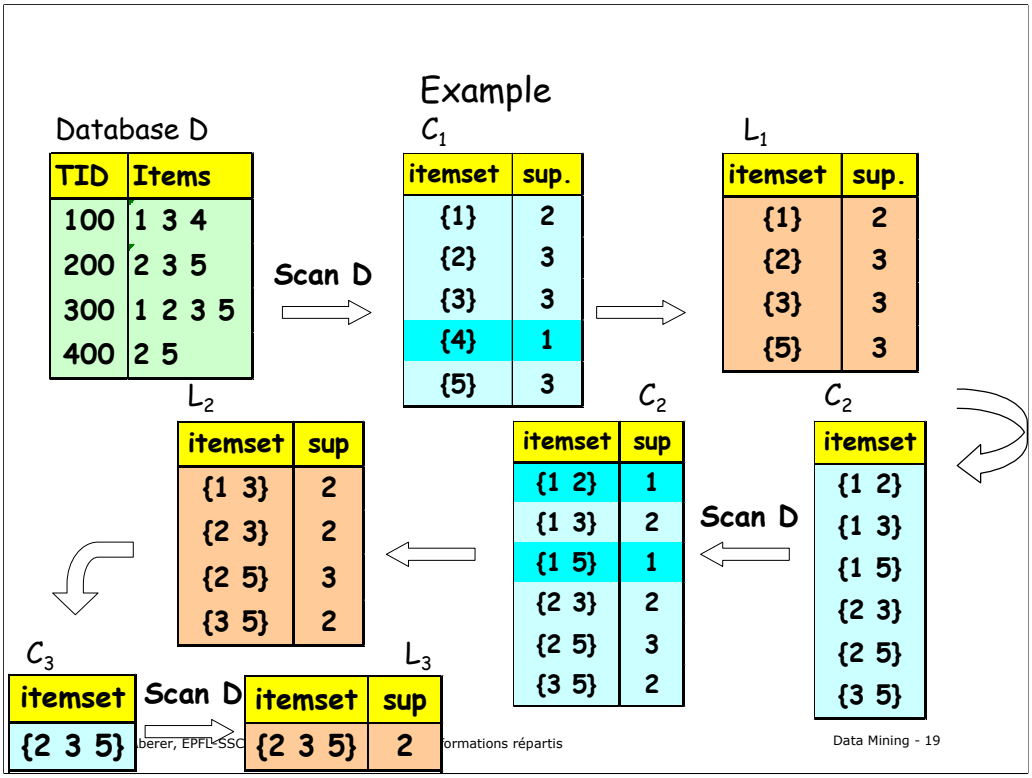
for each transaction T in the database
increment the count of all candidate item sets in C_{k+1}
that are contained in T;

L_{k+1} := candidates in C_{k+1} with min_support;

k := k+1; }

return $\cup_k L_k$;

This is the complete apriori algorithm for determining frequent itemsets.



Notice in this example of how the scan steps (when determining the frequency with respect to the database) eliminates certain items. Notice that in this example pruning does not apply.

Generating Association Rules from Frequent Itemsets

For each frequent itemset L generate all non-empty subsets S

For every nonempty subset S output the rule $S \Rightarrow L \setminus S$ if

$$\frac{sc(L)}{sc(S)} \geq min\_conf$$

sc = support count,

min_conf = minimal confidence

since

$$confidence(A \Rightarrow B) = P(B \mid A)$$

$$= \frac{sc(A \cup B)}{sc(A)} \geq min\_conf$$

Once the frequent itemsets are found the derivation of association rules is straightforward: one checks for every frequent itemset whether there exists a subset S that can occur as the head of a rule. For doing that, the support count, i.e. the frequency of the itemset in the database, which was obtained during the execution of the apriori algorithm, is used to compute the confidence (as a conditional probability). Note that also L\S is a frequent itemset, and therefore the support count is available for that set from the apriori algorithm.

Example

- Assume minimal confidence (min_conf) =0.75

  L={2, 3, 5}, S={3, 5}, confidence(S ⇒ L\S)= sc(L)/sc(S)=2/2
  therefore {3,5} ⇒ {2} ok

  L={2, 3, 5}, S={2}, confidence(S ⇒ L\S)= sc(L)/sc(S)=2/3
  therefore {2} ⇒ {3,5} not ok

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

frequent itemsets

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

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

Improving Apriori's Efficiency

- Transaction reduction
  - A transaction that does not contain any frequent k-itemset is useless in subsequent scans
- Partitioning
  - Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
- Sampling
  - mining on a subset of given data, lower support threshold + a method to determine the completeness
- Many further advanced techniques

Though the basic apriori algorithm is designed to work efficiently for large datasets, there exist a number of possible improvements:

•Transactions in the database that turn out to contain no frequent k-itemsets can be omitted in subsequent database scans.

•One can try to identify first frequent itemsets in partitions of the database. This methods is based on the assumption that if an itemset is not frequent in one of the partitions at least (local frequent itemset) then it will also not be frequent in the whole database.

•The sampling method selects samples from the database and searches for frequent itemsets in the sampled database using a correspondingly lower threshold for the support.

Mining Quantitative Association Rules

- Categorical Attributes
  - finite number of possible values, no ordering among values
- Quantitative Attributes
  - numeric, implicit ordering among values

- Quantitative attributes are transformed into categorical attributes by
  - Static discretization of quantitative attributes
    - Quantitative attributes are statically discretized by using predefined concept hierarchies.
  - Dynamic discretization
    - Quantitative attributes are dynamically discretized into "bins" based on the distribution of the data.

For quantitative attributes the situation is more complex. A simple approach is to statically or dynamically discretize them into categorical attributes.

However, the rules that can be found depend on the discretization chosen. It may happen that the bins are for example too fine-grained, and a rule that could be more efficiently be expressed at a coarser granularity is split into multiple rules.

For example: if age is discretized into steps of 2 years we would probably find rules

Age(X, 18..19) and lives(X, Lausanne) -> profession(X, student)

Age(X, 20..21) and lives(X, Lausanne) -> profession(X, student)

Could be also expressed as a rule

Age(X, 18..21) and lives(X, Lausanne) -> profession(X, student)

which is more compact but requires a different discretization. There exist specialized techniques to deal with this problem (e.g. ARCS).

Components of a Data Mining Algorithm

- Model or pattern structure
  - Which kind of global model or local pattern is searched
  - Vector representation of documents
  - Association Rules
- Score function
  - Determine how well a given data set fits a model or pattern
  - Similarity of vectors
  - Support and confidence
- Optimization and search method
  - Finding best parameters for a global model: optimization problem
  - Finding data satisfying a pattern: search problem
  - Search the k nearest neighbors
  - Joining and pruning
- Data management strategy
  - Handling very large datasets
  - Inverted files
  - Sampling, partitioning and transaction elimination

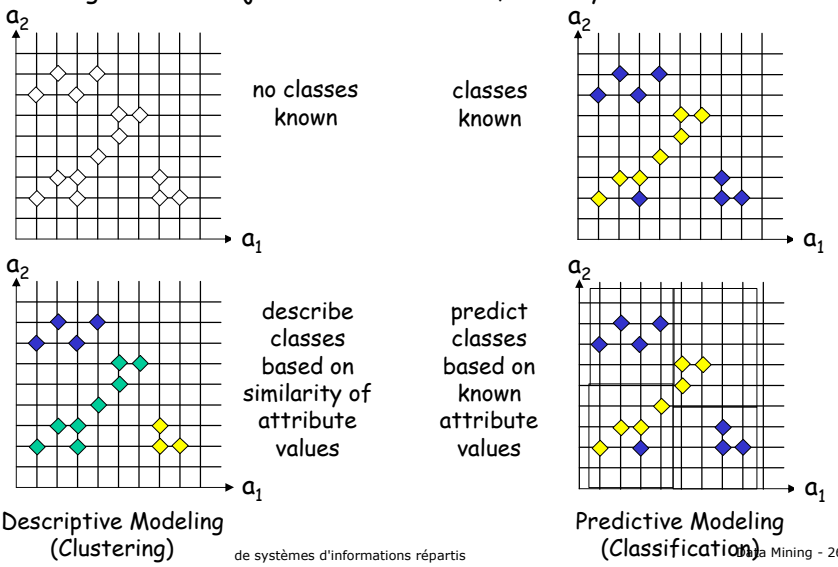
We illustrate here of how the four main components of data mining algorithms, are instantiated with association rule mining. Compare also to the corresponding methods used for vector space retrieval.

Summary

- What is the meaning of support and confidence for an association rule ?
- Is a high support for $A \cup B$ a sufficient condition for A->B or B->A being an association rule ?
- Which properties on association rules and itemsets does the apriori algorithm exploit ?
- Which candidate itemsets can in the apriori algorithm be eliminated in the pruning step and which during the database scan ?
- How often is a database scanned when executing apriori ?
- How are association rules derived from frequent itemsets ?

3. Clustering - Descriptive vs. Predictive Modeling

- Problem: given data objects with attributes, classify them



For establishing global models of data collections there exist two different approaches: descriptive and predictive modeling. We illustrate their difference by an example. We assume that a set of data items with two attributes a1 and a2 is given. Assume the global model we are interested in is a classification of the data items.

In descriptive modeling we just know the data items, as indicated by points in the 2-dimensional grid. A descriptive modeling technique, such as clustering, produces classes (or categories), which are not known in advance. For doing this it relies on some criteria that specify when two data items probably belong to the same class. Such a criteria is usually given as a similarity measure.

A predictive modeling technique, such as classification, starts from a given classification of the data items. From that it derives conditions on the properties of the data objects, that allow to predict the membership to a specific class. For example, the prediction could be based on a partitioning of the attribute values along each dimension, as shown in the figure to the right. There, first attribute a1 is partitioned into two intervals, and for each of the intervals a different partitioning of the attribute a2 is used to determine the regions corresponding to classes. Misclassifications may occur as seen in the example.

Clustering

- Model: Clusters of objects
- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Clustering is unsupervised classification
  - no predefined classes
- Typical use
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms
- Typical applications
  - WWW
    - Document classification
    - Cluster Weblog data to discover groups of similar access patterns
  - Economic Science (especially market research)
  - Pattern Recognition, Spatial Data Analysis, Image Processing

Both clustering and classification aim at partitioning a dataset into subsets that bear similar characteristics. Different to classification clustering does not assume any prior knowledge, which are the classes/clusters to be searched for. There exist no class label attributes, that would tell which classes exist. Thus clustering serves in particular for exploratory data anlaysis with little or no prior knowledge.

One important application of clustering we have in fact already introduced in information retrieval.The basic problem of information retrieval, i.e. find a set of documents matching a query, can be interpreted as a clustering problem, where the goal is to find two clusters of documents, namely the cluster of relevant ones and the cluster of non-relevant ones. In the tf-idf scheme in fact the tf-measure served to measure intra-cluster similarity for the two document clusters, whereas the idf-measure served to measure inter-cluster dissimilarity of the document clusters.

Clustering has important applications on the Web in order to extract information from large data collections, both document collections and transactional data. Clustering is also an important tool in scientific data analysis and has, for example, a long tradition in image processing and related areas. Data mining frequently adopts techiques from these areas and extends them to make them applicable for analysing large data sets.

Clustering Problem

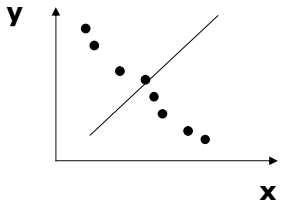
- Given: database D with N d-dimensional data items
- Find: partitioning into k clusters and noise
- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation

In its simplest formulation the clustering problem can be described in a way analogous to the vector space retrieval model. Given a database of data items that are represented by d-dimensional vectors (feature vectors), then partition the database into k clusters.

Popular similarity measures include Euclidean distance and Manhattan distance.

Criteria for Clustering Methods

- Quantitative Criteria
  - Scalability: number of data objects N
  - High dimensionality
- Qualitative criteria
  - Ability to deal with different types of attributes
  - Discovery of clusters with arbitrary shape
- Robustness
  - Able to deal with noise and outliers
  - Insensitive to order of input records
- Usage-oriented criteria
  - Incorporation of user-specified constraints
  - Interpretability and usability



Clearly, clustering methods have to work efficiently for large datasets. Another scalability problem clustering methods have to deal with is however dimensionality: the problem is that in data sets with high dimensionality (large d) it becomes increasingly difficult to find clusters, as the occurence of clusters is highly sensitive on the dimensions that are selected to project the data into a low-dimensional space. Without selecting specific dimensions the data would be too sparse in the high-dimensional space in order to find clusters. The figure illustrates how this problem occurs already in 2 dimensions: Only by choosing the right plane for projecting onto a single dimension we will observe a cluster. If we would project only on the x or y-axis we would not recognize any clustering effect. Thus, when projecting the choice of the subspaces used for projection is crucial. The number of choices for projection dimensions grows combinatorially.

Qualitative criteria address the ability of dealing with continuous as well as categorical attributes, and the type of clusters that can be found. Many clustering methods can detect only very simple geometrical shapes, like spheres, hyperplanes etc.

Clustering methods can be sensitive both to noisy data and the order of how the records are processed. In both cases it would be undesireable to have a dependency of the clustering result on these aspects which are unrelated to the nature of data in question.

Finally, an important criterion is the ability of how well a clustering method can incorporate user requirements both in terms of information that is provided from the user to the clustering method (in terms of constraints), which can guide the clustering process, and in terms of what information is provided from the method to the user.

Partitioning Methods

- Partitioning method
  - Construct a partition of a database **D** of **n** objects into a set of **k** clusters
- Given k, find a partition of k clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Heuristic methods: k-means and k-medoids algorithms
  - k-means: each cluster is represented by the center of the cluster
  - k-medoids or PAM (Partition around medoids): each cluster is represented by one of the objects in the cluster

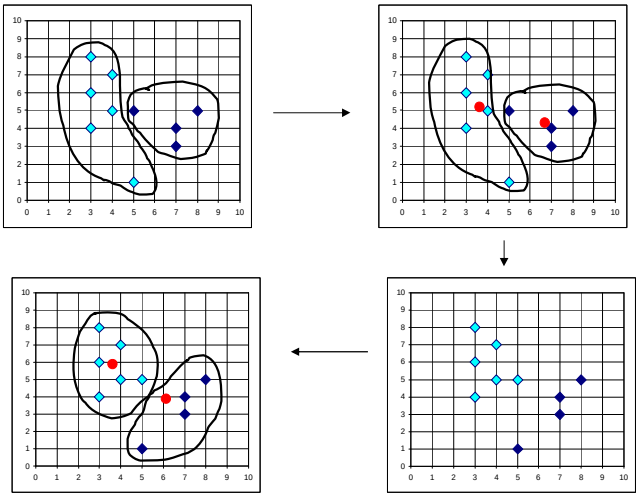
Partitioning methods are a basic approach to clustering. Partitioning methods attempt to partition the data set into a given numberk of clusters optimizing intra-cluster similarity and inter-cluster dissimilarity. Since an exhaustive enumeration for finding the optimal partitioning is not practical various heuristic methods have been proposed.

The k-Means Partitioning Method

- Assume objects are characterized by a d-dimensional vector
- Given k, the k-means algorithm is implemented in 4 steps
  - Step 1: Partition objects into k nonempty subsets
  - Step 2: Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster
  - Step 3: Assign each object to the cluster with the nearest seed point
  - Step 4: Stop when no new assignment occurs, otherwise go back to Step 2

In k-Means, the centroids are computed as the arithmetic mean of the cluster all points of a cluster. The distances are computed according to a given distance measure, e.g. Euclidean distance.

Example



In this example the k-means algorithm terminates after two iterations (the colors indicate the current clusters, the red points are the current centroids).

Properties of k-Means

- Strengths
  - Relatively efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Often terminates at a local optimum
  - The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms
- Weaknesses
  - Applicable only when mean is defined, therefore not applicable to 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

This assessment follows the list of criteria for evaluating clustering methods that we have introduced earlier.

References

- Textbook
  - Jiawei Han, Data Mining: concepts and techniques, Morgan Kaufman, 2000, ISBN 1-55860-489-8
- Some relevant research literatue
  - R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.