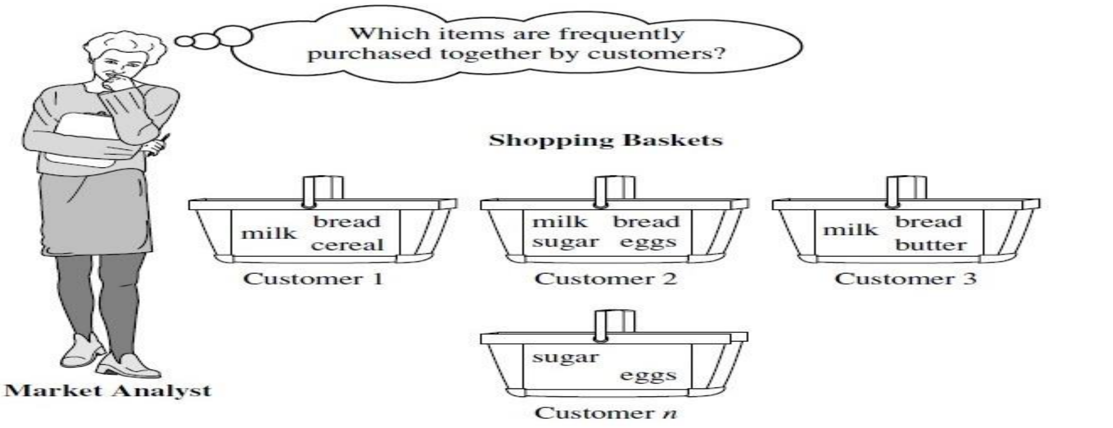
DATA MINING

Frequent Patters Mining

Unit 3

Mining Frequent Patterns, Associations, and Correlations

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including frequent itemsets, frequent subsequences (also known as sequential patterns), and frequent substructures. A *frequent itemset* typically refers to a set of items that often appear together in a transactional data set—for example, milk and bread, which are frequently bought together in grocery stores by many customers. A frequently occurring subsequence, such as the pattern that customers, tend to purchase first a laptop, followed by a digital camera, and then a memory card, is a (*frequent*) *sequential pattern*. A substructure can refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences.



Market basket analysis.

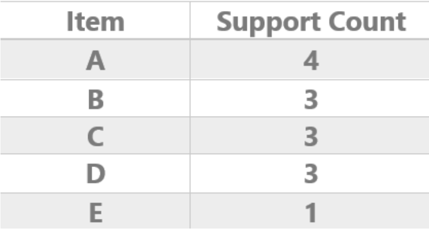
Association

It is used to find a correlation between two or more items by identifying the hidden pattern in the data set and hence also called relation analysis*.* This method is used in market basket analysis to predict the behavior of the customer.

Example

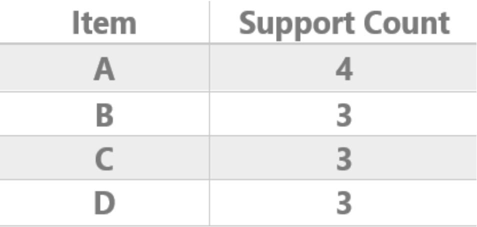
Association analysis. Suppose that, as a marketing manager at *AllElectronics*, you want to know which items are frequently purchased together (i.e., within the same transaction). An example of such a rule, mined from the *AllElectronics* transactional database, is

Buys(*X*, “*computer*”)=>*buys*(*X*, “*software*”) [*support* = 1*%*, *confidence* = 50*%*],



Item	Support Count
A	4
B	3
C	3
D	3
E	1

Customer set1



Item	Support Count
A	4
B	3
C	3
D	3

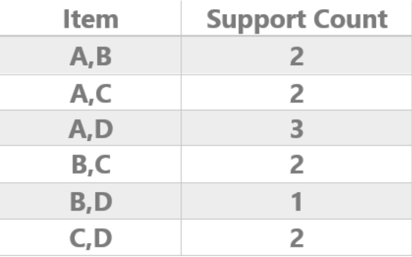
Frequent itemset from 1st scan

where *X* is a variable representing a customer. A **confidence**, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well. A 1% **support** means that 1% of all the transactions under analysis show that computer and software are purchased together. This association rule involves a single attribute or predicate (i.e., *buys*) that repeats. Association rules that contain a single predicate are referred to as **single-dimensional association rules**. Dropping the predicate notation, the rule can be written simply as “*computer* =>*software* [1%, 50%].” Suppose, instead, that we are given the *AllElectronics* relational database related to purchases. A data mining system may find association rules like

age(X, “20..29”)^*income(X*, “40K..49K”)=>*buys*(*X*, “laptop”)

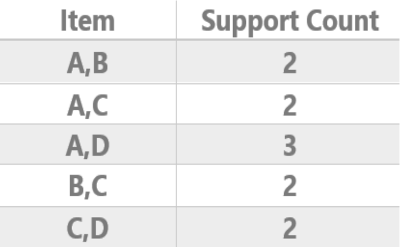
[*support* = 2*%*, *confidence* = 60*%*].

The rule indicates that of the *AllElectronics* customers under study, 2% are 20 to 29 years old with an income of $40,000 to $49,000 and have purchased a laptop (computer) at *AllElectronics*. There is a 60% probability that a customer in this age and income group will purchase a laptop. Note that this is an association involving more than one attribute or predicate (i.e., *age, income*, and *buys*). Adopting the terminology used in multidimensional databases, where each attribute is referred to as a dimension, the above rule can be referred to as a **multidimensional association rule**.



Item	Support Count
A,B	2
A,C	2
A,D	3
B,C	2
B,D	1
C,D	2

Customers set 2



Item	Support Count
A,B	2
A,C	2
A,D	3
B,C	2
C,D	2

Frequent itemset from 2nd scan

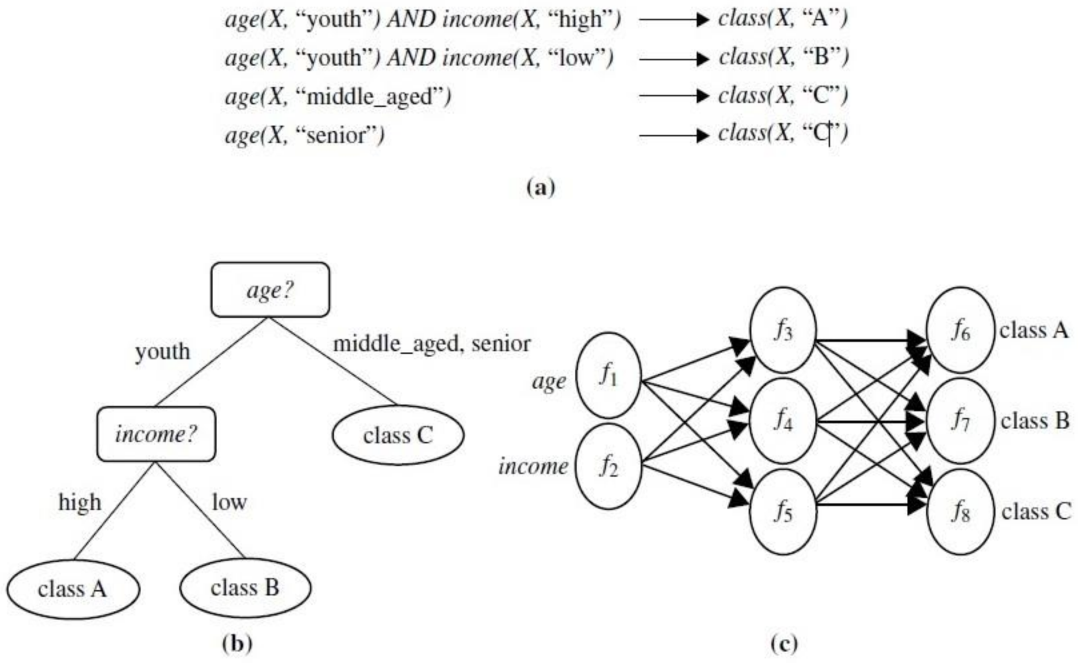
Classification and Regression for Predictive Analysis:

Classification is the process of finding a **model** (or function) that describes and distinguishes data classes or concepts. The model are derived based on the analysis of a set of **training data** (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the class label is unknown. *"How is the derived model presented?"* The derived model may be represented in various forms, such as *classification rules* (i.e., *IF-THEN rules*), *decision trees*, *mathematical formulae*, or *neural networks.*

Decision Tree: A **decision tree** is a flowchart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions.

Neural Network:

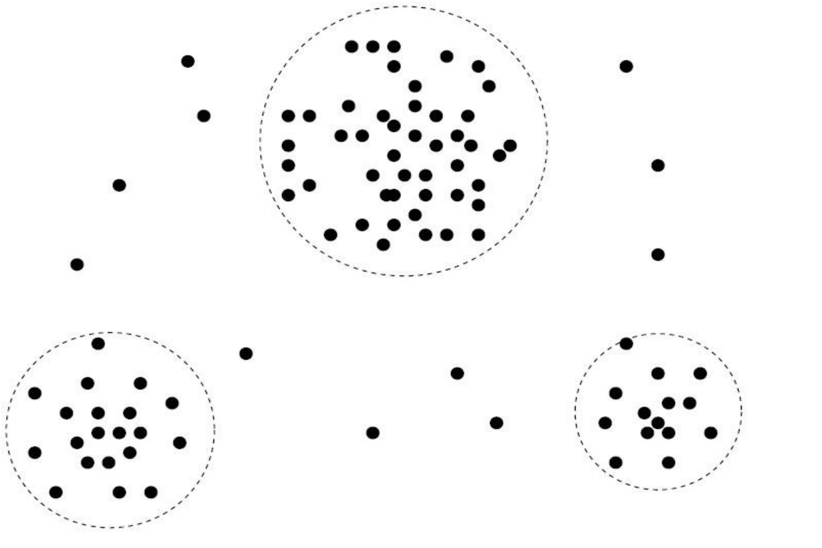
This method or model is based on biological neural networks. It is a collection of neurons like processing units with weighted connections between them. They are used to model the relationship between inputs and outputs. It is used for classification, regression analysis, data processing etc


age(X, "youth") AND income(X, "high") → class(X, "A")
age(X, "youth") AND income(X, "low") → class(X, "B")
age(X, "middle_aged") → class(X, "C")
age(X, "senior") → class(X, "C")
(a)
age?
youth
middle_aged, senior
income?
class C
high
low
class A
class B
(b)
age
income
f1
f2
f3
f4
f5
f6
f7
f8
class A
class B
class C
(c)

A classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

Cluster Analysis

Unlike classification and regression, which analyze class-labelled (training) data sets, **clustering** analyzes data objects without consulting class labels. In many cases, class labelled data may simply not exist at the beginning. Clustering can be used to generate class labels for a group of data.



A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

Example

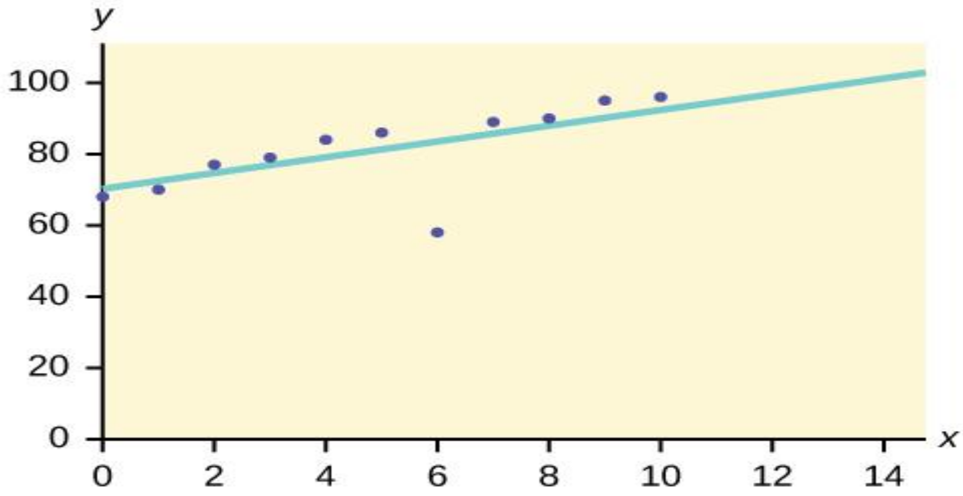
Cluster analysis. Cluster analysis can be performed on *AllElectronics* customer data to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing. Figure shows a 2-D plot of customers with respect to customer locations in a city. Three clusters of data points are evident.

Outlier Analysis

A data set may contain objects that do not comply with the general behavior or model of the data. These data objects are **outliers**. Many data mining methods discard outliers as noise or exceptions. However, in some applications (e.g., fraud detection) the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as **outlier analysis** or **anomaly mining**.

Example:

Outlier analysis. Outlier analysis may uncover fraudulent usage of credit cards by detecting purchases of unusually large amounts for a given account number in comparison to regular charges incurred by the same account. Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.



Pattern Mining:

pattern mining can be classified using the following criteria:

Basic patterns: A frequent pattern may have several alternative forms, including a simple frequent pattern, a closed pattern, or a max-pattern. To review, a **frequent pattern** is a pattern (or itemset) that satisfies a minimum support threshold. A pattern *p* is a **closed pattern** if there is no superpattern *p*0 with the same support as *p*. Pattern *p* is a **max-pattern** if there exists no frequent superpattern of *p*. Frequent patterns can also be mapped into **association rules**, or other kinds of rules based on interestingness measures. Sometimes we may also be interested in **infrequent** or **rare patterns** (i.e., patterns that occur rarely but are of critical importance, or **negative patterns** (i.e., patterns that reveal a negative correlation between items).

- **Based on the *abstraction* levels involved in a pattern:** Patterns or association rules may have items or concepts residing at high, low, or multiple abstraction levels. For example, suppose that a set of association rules mined includes the following rules where *X* is a variable representing a customer:
  buys(*X*, “*computer*”)=>*buys*(*X*, “*printer* ”)
  buys(*X*, “*laptop computer*”)=>*buys*(*X*, “*color laser printer* ”)
  We refer to the rule set mined as consisting of **multilevel association rules**. If, instead, the rules within a given set do not reference items or attributes at different abstraction levels, then the set contains **single-level association rules**.
- **Based on the *number of dimensions* involved in the rule or pattern:** If the items or attributes in an association rule or pattern reference only one

dimension, it is a **single-dimensional association rule/pattern**. If a rule/pattern references two or more dimensions, such as *age, income*, and *buys*, then it is a **multidimensional association rule/pattern**. The following is an example of a multidimensional rule: *age*(*X*, “20: : :29”)^*income*(*X*, “52*K* : : :58*K*”)=>*buys*(*X*, “*iPad* ”).

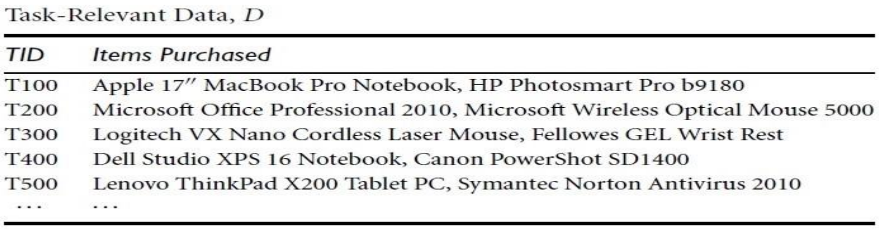
- **Based on the *types of values* handled in the rule or pattern:** If a rule involves associations between the presence or absence of items, it is a **Boolean association rule**. If a rule describes associations between quantitative items or attributes, then it is a **quantitative association rule**.
- **Based on the *constraints* or *criteria* used to mine *selective patterns*:** The patterns or rules to be discovered can be **constraint-based** (i.e., satisfying a set of userdefined constraints), **approximate**, **compressed**, **near-match** (i.e., those that tally the support count of the near or almost matching itemsets), **top-*k*** (i.e., the *k* most frequent itemsets for a user-specified value, *k*), **redundancy-aware top-*k*** (i.e., the top-*k* patterns with similar or redundant patterns excluded), and so on.
- **Based on *kinds of data and features* to be mined:** Given relational and data warehouse data, most people are interested in itemsets. Thus, frequent pattern mining in this context is essentially **frequent itemset mining**, that is, to mine frequent *sets of items*. However, in many other applications, patterns may involve sequences and structures.
- **Based on *application domain-specific semantics*:** Both data and applications can be very diverse, and therefore the patterns to be mined can differ largely based on their domain-specific semantics. Various kinds of application data include spatial data, temporal data, spatiotemporal data, multimedia data (e.g., image, audio, and video data), text data, time-series data, DNA and biological sequences, software programs, chemical compound structures, web structures, sensor networks, social and information networks, biological networks, data streams, and so on.
- **Based on *data analysis usages*:** Frequent pattern mining often serves as an intermediate step for improved data understanding and more powerful data analysis. For example, it can be used as a feature extraction step for classification, which is often referred to as **pattern-based classification**.

Pattern Mining in Multilevel, Multidimensional Space

Multidimensional associations involve more than one dimension or predicate (e.g., rules that relate what a customer *buys* to his or her *age*). *Quantitative association rules* involve numeric attributes that have an implicit ordering among values (e.g., *age*). *Rare patterns* are patterns that suggest interesting although rare item combinations. *Negative patterns* show negative correlations between items.

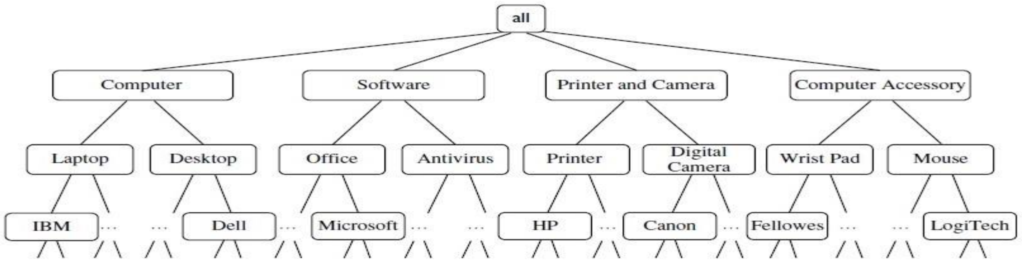
MULTILEVEL ASSOCIATION RULES:

- Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules.
- Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework.
- Rules at high concept level may add to common sense while rules at low concept level may not be useful always.
    - Using uniform minimum support for all levels:
- When a uniform minimum support threshold is used, the search procedure is simplified.
- The method is also simple, in that users are required to specify only one minimum support threshold.
- The same minimum support threshold is used when mining at each level of abstraction.
- For example, in Figure, a minimum support threshold of 5% is used throughout.
- (e.g. for mining from “computer” down to “laptop computer”).
- Both “computer” and “laptop computer” are found to be frequent, while “desktop computer” is not.
- Using reduced minimum support at lower levels:
    - Each level of abstraction has its own minimum support threshold.
    - The deeper the level of abstraction, the smaller the corresponding threshold is.
    - For example, in Figure, the minimum support thresholds for levels 1 and 2 are 5% and 3%, respectively.
    - In this way, “computer,” “laptop computer,” and “desktop computer” are all considered frequent.

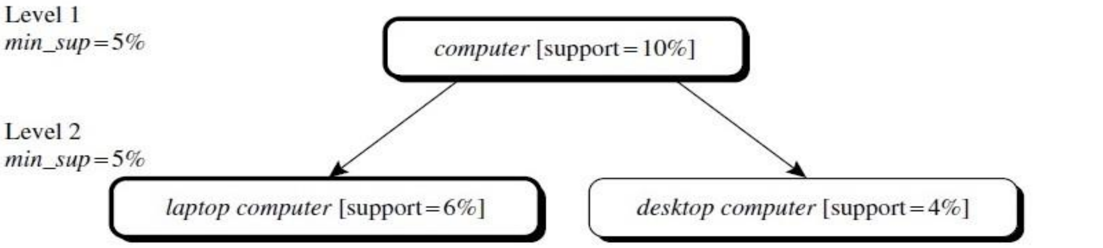


Task-Relevant Data, *D*

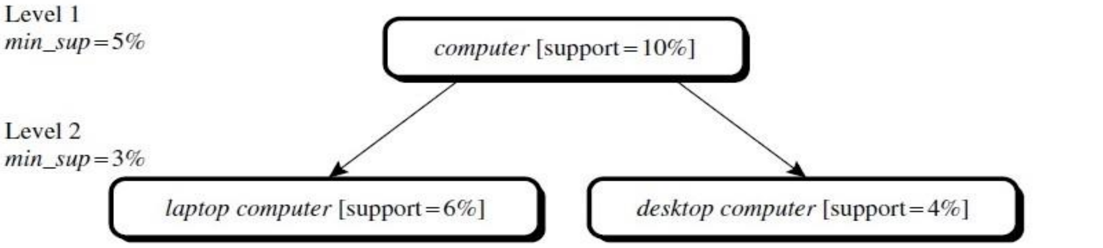
TID	*Items Purchased*
T100	Apple 17″ MacBook Pro Notebook, HP Photosmart Pro b9180
T200	Microsoft Office Professional 2010, Microsoft Wireless Optical Mouse 5000
T300	Logitech VX Nano Cordless Laser Mouse, Fellowes GEL Wrist Rest
T400	Dell Studio XPS 16 Notebook, Canon PowerShot SD1400
T500	Lenovo ThinkPad X200 Tablet PC, Symantec Norton Antivirus 2010
...	...



Concept hierarchy for *AllElectronics* computer items.



Multilevel mining with uniform support.



Multilevel mining with reduced support.

MULTIDIMENSIONAL ASSOCIATION RULES:

1.In Multi dimensional association:

- Attributes can be categorical or quantitative.
- Quantitative attributes are numeric and incorporates hierarchy.
- Numeric attributes must be discretized.
- Multi dimensional association rule consists of more than one dimension:

Eg: buys(X,”IBM Laptop computer”)=>buys(X,”HP Inkjet Printer”)

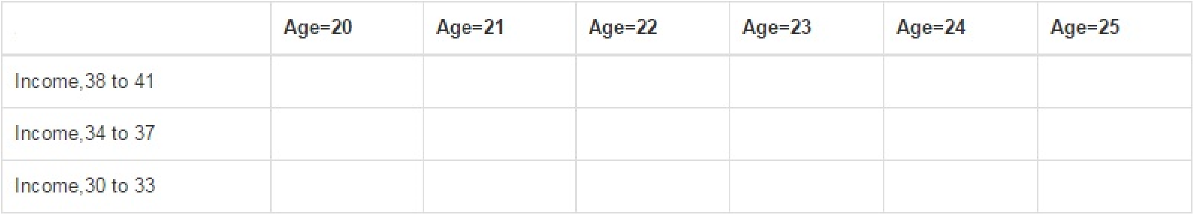
2.Three approaches in mining multi dimensional association rules:

1.Using static discritization of quantitative attributes.

- Discretization is static and occurs prior to mining.
- Discretised attributes are treated as categorical.
- Use a priori algorithm to find all k-frequent predicate sets(this requires k or k+1 table scans).
- Every subset of frequent predicate set must be frequent.
- Eg: If in a data cube the 3D cuboid (age, income, buys) is frequent implies (age, income), (age, buys), (income, buys) are also frequent.
- Data cubes are well suited for mining since they make mining faster.
- The cells of an n-dimensional data cuboid correspond to the predicate cells.

2.Using dynamic discretization of quantitative attributes:

- Known as mining Quantitative Association Rules.
- Numeric attributes are dynamically discretized.
- Eg: age(X,”20..25”) Λ income(X,”30K..41K”)=>buys (X,”Laptop Computer”)



	Age=20	Age=21	Age=22	Age=23	Age=24	Age=25
Income,38 to 41						
Income,34 to 37						
Income,30 to 33						

GRID FOR TUPLES

3.Using distance based discritization with clustering.

This id dynamic discretization process that considers the distance between data points.

- It involves a two step mining process:
  - Perform clustering to find the interval of attributes involved.
  - Obtain association rules by searching for groups of clusters that occur together.
- The resultant rules may satisfy:
  - Clusters in the rule antecedent are strongly associated with clusters of rules in the consequent.
  - Clusters in the antecedent occur together.
  - Clusters in the consequent occur together.

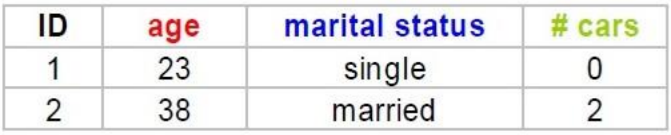
Mining Quantitative Association Rules

we introduce three methods that can help overcome
this difficulty to discover novel association relationships:
(1) a data cube method,
(2) a clustering-based method, and
(3) a statistical analysis method

Quantitative Association Rules

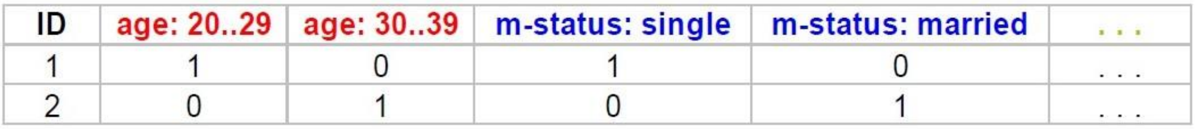
- Up to now: associations of boolean attributes only
- Now: numerical attributes, too
- Example:

Original database



ID	age	marital status	# cars
1	23	single	0
2	38	married	2

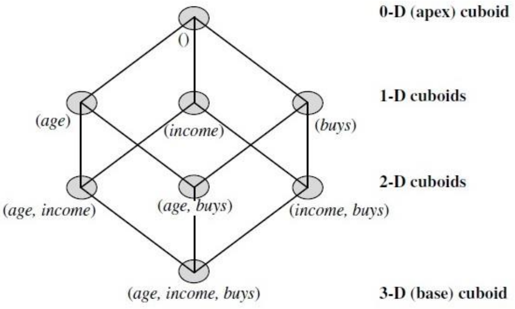
Boolean database



ID	age: 20..29	age: 30..39	m-status: single	m-status: married	. . .
1	1	0	1	0	. . .
2	0	1	0	1	. . .

Static Discretization of Quantitative Attributes

- Discretized prior to mining using concept hierarchy.
- Numeric values are replaced by ranges.
- In relational database, finding all frequent k-predicate sets will require k or k+1 table scans.
- Data cube is well suited for mining.
- The cells of an n-dimensional cuboid correspond to the predicate sets.
- Mining from data cubes can be much faster.



Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains the three predicates *age*, *income*, and *buys*.

Constraint-Based Frequent Pattern Mining

Constraint-based pattern mining is **a generalization of frequent itemset mining**. The constraints can include the following:

- **Knowledge type constraints:** These specify the type of knowledge to be mined, such as association, correlation, classification, or clustering.
- **Data constraints:** These specify the set of task-relevant data.
- **Dimension/level constraints:** These specify the desired dimensions (or attributes) of the data, the abstraction levels, or the level of the concept hierarchies to be used in mining.
- **Interestingness constraints:** These specify thresholds on statistical measures of rule interestingness such as support, confidence, and correlation.
- **Rule constraints:** These specify the form of, or conditions on, the rules to be mined. Such constraints may be expressed as metarules (rule templates), as the maximum or minimum number of predicates that can occur in the rule antecedent or consequent, or as relationships among attributes, attribute values, and/or aggregates.

Metarule-Guided Mining of Association Rules

Metarules may be based on the analyst's experience, expectations, or intuition regarding the data or may be automatically generated based on the database schema.

Example:

Metarule-guided mining. Suppose that as a market analyst for *AllElectronics* you have access to the data describing customers (e.g., customer age, address, and credit rating) as well as the list of customer transactions. You are interested in finding associations between customer traits and the items that customers buy.

$P1(X, Y) \wedge P2(X,W) \Rightarrow buys(X, \text{“}office\ software\text{”}),$

where *P*1 and *P*2 are **predicate variables** that are instantiated to attributes from the given database during the mining process, *X* is a variable representing a customer, and *Y* and *W* take on values of the attributes assigned to *P*1 and *P*2, respectively.

$Age(X, \text{“30..39”}) \wedge income(X, \text{“41}K\text{..60}K\text{”}) \Rightarrow buys(X, \text{“}office\ software\text{”}).$

$$(P1 \wedge P2 \wedge \_\ \_\ \_ \wedge Pl\ ) \Rightarrow (Q1 \wedge Q2 \wedge \_\ \_\ \_ \wedge Qr\ )$$

where *Pi* (*i* = 1, 2,… , *l*) and *Qj* (*j* = 1, 2,…., *r*) are either instantiated predicates or predicate variables. Let the number of predicates in the metarule be *p* = l+*r*. To find interdimensional association rules satisfying the template,

- We need to find all frequent *p*-predicate sets, *Lp*.
- We must also have the support or count of the *l*-predicate subsets of *Lp* to compute the confidence of rules derived from *Lp*.

Classification using frequent patterns

There are two types of classification using frequent patterns:

- Associative classification model as well as association rules, which are generated from frequent patterns and used for classifications
- Discriminative frequent pattern-based classification

Associative Classification

you will learn about associative classification. The methods discussed are
CBA, CMAR, and CPAR.
Association rules are mined in a two-step process consisting of *frequent itemset mining* followed by *rule generation*. The second step analyzes the frequent itemsets to generate association rules. All association rules must satisfy certain criteria regarding their "accuracy" (or *confidence*) and the proportion of the data set that they actually represent (referred to as *support*).

age =youth^credit = OK=>buys computer = yes [*support* = 20%, *confidence* = 93%],

where ^ represents a logical "AND." We will say more about confidence and support later. More formally, let *D* be a data set of tuples. Each tuple in *D* is described by *n* attributes, *A*1, *A*2, …… , *An*, and a class label attribute, *Aclass* . All continuous attributes are discretized and treated as categorical (or nominal) attributes. An **item**, *p*, is an attribute– value pair of the form (*Ai* , *v*), where *Ai* is an attribute taking a value, *v*. A data tuple ***X*** D .*x*1, *x*2, …. , *xnl* satisfies an item, *p* = (*Ai* , *v*), if and only if *xi* = *v*, where *xi* is the value of the *i*th attribute of ***X***. Association rules can have any number of items in the rule antecedent (left side) and any number of items in the rule consequent (right side). However, when mining association rules for use in classification, we are only interested in association rules of the form *p*1 ^*p*2 ^….*pl* =>*Aclass* = *C*, where the rule antecedent is a conjunction of items, *p*1, *p*2, …. , *pl* (*l* <=*n*), associated with a class label, *C*. For example, a confidence of 93% for Rule (9.21) means that 93% of the customers in *D* who are young and have an OK credit rating belong to the class *buys computer* = *yes*.

In general, associative classification consists of the following steps:

1. Mine the data for frequent itemsets, that is, find commonly occurring attribute–value pairs in the data.

2. Analyze the frequent itemsets to generate association rules per class, which satisfy confidence and support criteria.

3. Organize the rules to form a rule-based classifier.

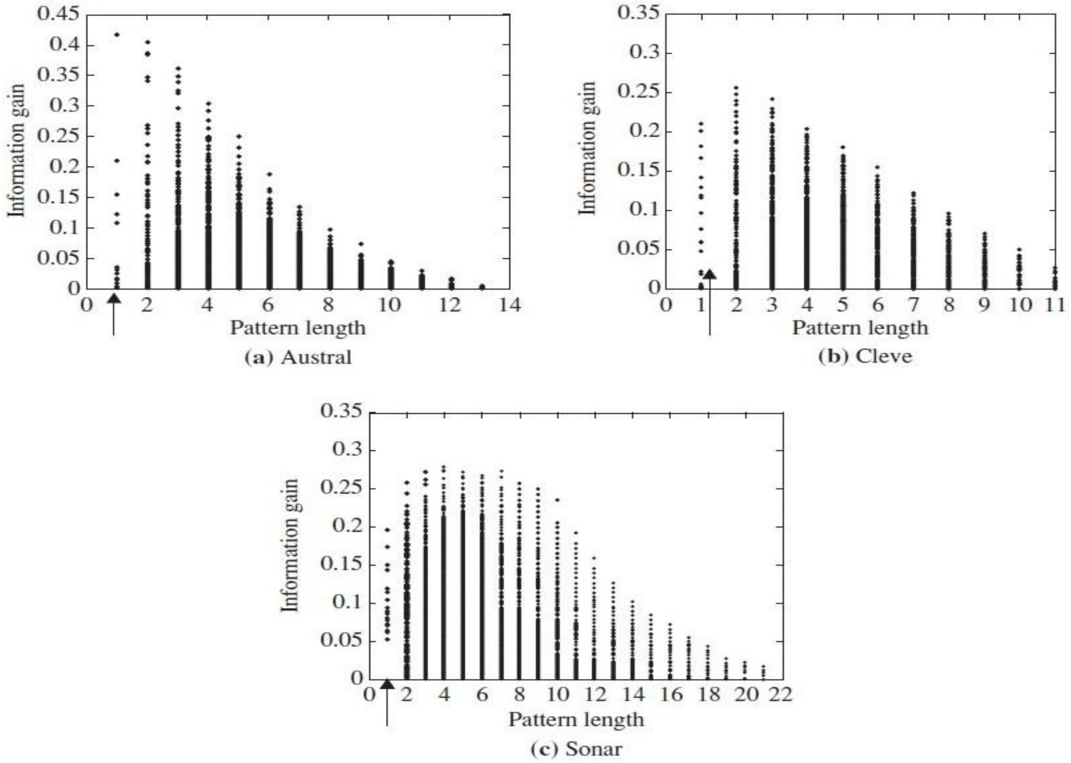
One of the earliest and simplest algorithms for associative classification is **CBA** (Classification Based on Associations). CBA uses an iterative approach to frequent itemset mining, where multiple passes are made over the data and the derived frequent itemsets are used to generate and test longer itemsets. The complete set of rules satisfying minimum confidence and minimum support thresholds are found and then analyzed for inclusion in the classifier. CBA uses a heuristic method to construct the classifier, where the rules are ordered according to decreasing precedence based on their confidence and support. If a set of rules has the same antecedent, then the rule with the highest confidence is selected to represent the set. When classifying a new tuple, the first rule satisfying the tuple is used to classify it. The classifier also contains a default rule, having lowest precedence, which specifies a default class for any new tuple that is not satisfied by any other rule in the classifier. In this way, the set of rules making up the classifier forma *decision list*.

CMAR (Classification based on Multiple Association Rules) differs from CBA in its strategy for frequent itemset mining and its construction of the classifier. It also employs several rule pruning strategies with the help of a tree structure for efficient storage and retrieval of rules. CMAR adopts a variant of the *FP-growth* algorithm to find the complete set of rules satisfying the minimum confidence and minimum support thresholds. FP-growth uses a tree structure, called an *FP-tree*, to register all the frequent itemset information contained in the given data set, *D*. This requires only two scans of *D*. The frequent itemsets are then mined from the FP-tree. CMAR uses an enhanced FP-tree that maintains the distribution of class labels among tuples satisfying each frequent itemset.

CPAR (Classification based on Predictive Association Rules) takes a different approach to rule generation, based on a rule generation algorithm for classification known as FOIL (Section 8.4.3). FOIL builds rules to distinguish positive tuples (e.g., *buys computer =yes*) from negative tuples (e.g., *buys computer =no*). For multiclass problems, FOIL is applied to each class.

Discriminative Frequent Pattern–Based Classification

"*Why not consider frequent patterns as combined features, in addition to single features when building a classification model?*" This notion is the basis of **frequent pattern– based classification**—the learning of a classification model in the feature space of single attributes *as well as* frequent patterns. Many of the frequent patterns generated in frequent itemset mining are indiscriminative because they are based solely on support, without considering predictive power. That is, by definition, a pattern must satisfy a user-specified minimum support threshold, *min sup*, to be considered frequent. For example, if *min sup*, is, say, 5%, a pattern is frequent if it occurs in 5% of the data tuples.



(a) Austral

(b) Cleve

(c) Sonar

Single feature versus frequent pattern: Information gain is plotted for single features (patterns of length 1, indicated by arrows) and frequent patterns (combined features) for three UCI data sets. *Source:* Adapted from Cheng, Yan, Han, and Hsu [CYHH07].

The *general framework for discriminative frequent pattern–based classification* is as follows.

1. Feature generation: The data, *D*, are partitioned according to class label. Use frequent itemset mining to discover frequent patterns in each partition, satisfying minimum support. The collection of frequent patterns, F, makes up the feature candidates.

2. Feature selection: Apply feature selection to F, resulting in FS, the set of selected (more discriminating) frequent patterns. Information gain, Fisher score, or other evaluation measures can be used for this step. Relevancy checking can also be incorporated into this step to weed out redundant patterns. The data set *D* is transformed to *D*`, where the feature space now includes the single features as well as the selected frequent patterns, F_S.

3. Learning of classification model: A classifier is built on the data set *D*0. Any learning algorithm can be used as the classification model.