**Chapter Two**

**Data Mining Background**

**­­2.1 : Definition**

Data mining activities constitute an iterative process aimed at the analysis of large databases, with the purpose of extracting information and knowledge that may prove accurate and potentially useful for knowledge workers engaged in. [1]

A further characteristic of data mining depends on the procedure for collecting past observations and inserting them into a database. Indeed, these records are usually stored for purposes that are not primarily driven by data mining analysis. Data mining activities can be subdivided into two major investigation streams, according to the main purpose of the analysis: interpretation and prediction. [1]

**Interpretation :**

Identify regular patterns in the data and to express them through rules and criteria that can be easily understood by experts in the application domain. For example, for a company in the retail industry it might be advantageous to cluster those customers who have taken out loyalty cards according to their purchasing profile. [1]

**Prediction :**

The purpose of prediction is to anticipate the value that a random variable will assume in the future or to estimate the likelihood of future events. [1]

**2.2 : Models and Methods for Data Mining**

Classification trees or association rules, and are referred to as machine learning or knowledge discovery in databases. For example, in its simplest form, linear regression is used to relate a dependent response variable Y to an independent predictor X through a linear regression in the form Y = aX + b ,where a and b are parameters to be determined using the available past observations. For example, Y may represent the sales of a mass consumption product during a week and X the total advertisement cost during the same week. With respect to the development phases of a model, the selection of a linear function determines the type of relationship between the predictor and the response variable. A reasonable evaluation metric is the sum of the squared differences between the values of Y actually observed in the selection of a class of models to be used for learning from the past and of a specific form for representing patterns in the data. [1]

The design of a computational algorithm in order to generate the models by optimizing the evaluation metric. [1]

**2.3 : Data Mining, Classical Statistics and OLAP**

**Table (2.1): Data mining, classical statistics and OLAP [1]**

|  |  |  |
| --- | --- | --- |
| OLAP | Statistics | Data Mining |
| extraction of details and aggregate totals from data | veriﬁcation of hypotheses formulated by analysts | identiﬁcation of patterns and recurrences in data |
| Information | validation | Knowledge |
| distribution of incomes of home loan applicants | analysis of variance of incomes of home loan applicants | characterization of home loan applicants and prediction of future applicants |

**2.4 : Applications of Data Mining**

**Relational marketing :**

Identification of customer segments that are most likely to respond to targeted marketing campaigns, such as cross-selling and up-selling; identification of target customer segments for retention campaigns; prediction of the rate of positive responses to marketing campaigns; interpretation and understanding of the buying behavior of the customers; analysis of the products jointly purchased by customers, known as market basket analysis. [1]

**Fraud detection:**

Fraud may affect different industries such as telephony, insurance (false claims) and banking. [1]

**Risk evaluation :**

The purpose of risk analysis is to estimate the risk connected with future decisions, which often assume a dichotomous form. [1]

**Text mining :**

Data mining can be applied to different kinds of texts, which rep-resent unstructured data, in order to classify articles, books, documents, emails and web pages. [1]

**Image recognition :**

The treatment and classification of digital images, both static and dynamic, is an exciting subject both for its theoretical interest and the great number of applications it offers. [1]

**Web mining :**

They may prove useful for the analysis of e-commerce sites, in offering flexible and customized pages to surfers, in caching the most popular pages or in evaluating the effectiveness of an e-learning training course. [1]

**Medical diagnosis :**

Learning models are an invaluable tool within the medical field for the early detection of diseases using clinical test results. [1]

**2.5 : Representation of Input Data**

The input to a data mining analysis takes the form of a two-dimensional table, called a data set. The attributes contained in a dataset can be categorized as categorical or numerical, depending on the type of values they take on. [1]

**Categorical :**

Categorical attributes assume a finite number of distinct values, in most cases limited to less than a hundred, representing a qualitative property of an entity to which they refer. [1]

**Numerical :**

Numerical attributes assume a finite or infinite number of values and lend themselves to subtraction or division operations. Sometimes a more refined taxonomy of attributes can prove useful. [1]

**Counts :**

Counts are categorical attributes in relation to which a specific property can be true or false. These attributes can therefore be represented using Boolean variables { true, false } or binary variables { 0,1} . For example, a bank’s customers may or may not be holders of a credit card issued by the bank. [1]

**Nominal :**

Nominal attributes are categorical attributes without a natural ordering, such as the province of residence. [1]

**Ordinal :**

Ordinal attributes, such as education level, are categorical attributes that lend themselves to a natural ordering but for which it makes no sense to calculate differences or ratios between the values. [1]

**Discrete :**

Discrete attributes are numerical attributes that assume a finite number or a countable infinity of values. [1]

**Continuous :**

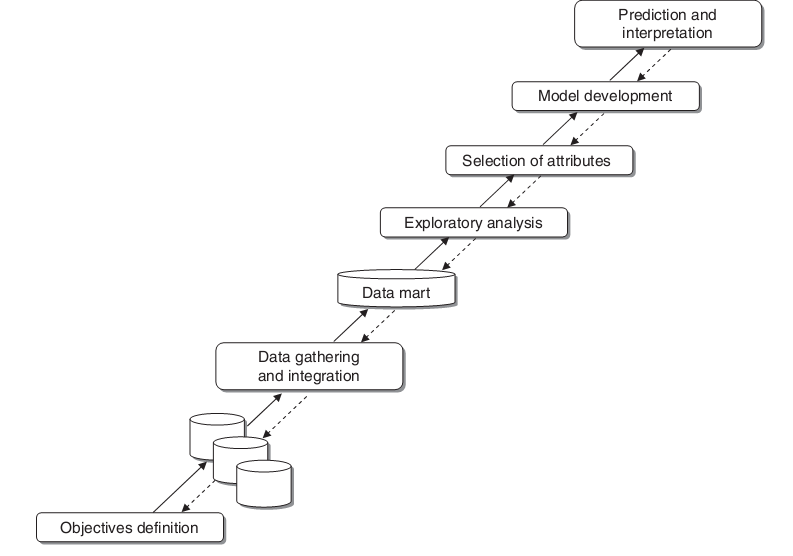
Continuous attributes are numerical attributes that assume an uncountable infinity of values. [1]

**2.6 : Data Mining Process**

Figure 2.1 shows the main phases of a generic data mining process.

**Definition of objectives :**

Data mining analyses are carried out in specific application domains and are intended to provide decision makers with useful knowledge. [1]



**Figure (2.1): Data mining process [1]**

**Data gathering and integration :**

Once the objectives of the investigation have been identified, the gathering of data begins. Data may come from different sources and therefore may require integration. Data sources may be internal, external or a combination of the two. In some instances, data sources are already structured in data warehouses and data marts for OLAP analyses and more generally for decision support activities. These are favorable situations where it is sufficient to select the attributes deemed relevant for the purpose of a data mining analysis. There is a risk, however, that, in order to limit memory uptake, the information stored in a data warehouse has been aggregated and consolidated to such an extent as to render useless any subsequent analysis.[1]

**Exploratory analysis :**

The data is carried out with the purpose of getting acquainted with the available information and carrying out data cleansing, data cleansing occurs at a semantic level. First of all, the distribution of the values for each attribute is studied, using histograms for categorical attributes and basic summary statistics for numerical variables. [1]

**Attribute Selection :**

Attributes is evaluated in relation to the goals of the analysis. Attributes that prove to be of little use are removed, in order to cleanse irrelevant information from the dataset. [1]

**Model development and validation :**

Once a high quality dataset has been assembled and possibly enriched with newly defined attributes, pattern recognition and predictive models can be developed. Usually the *training* of the models is carried out using a sample of records extracted from the original dataset. Then, the predictive accuracy of each model generated can be assessed using the rest of the data. [1]

**Prediction and interpretation :**

Knowledge workers may be able to use it to draw predictions and acquire a more in-depth knowledge of the phenomenon of interest. [1]

**2.7 : Analysis Methodologies**

One can draw a first fundamental distinction between supervised and unsupervised learning processes. [1]

**Supervised learning :**

In a supervised (or direct) learning analysis, a target attribute either represents the class to which each record belongs, Supervised learning processes are therefore oriented toward prediction and interpretation with respect to a target attribute. [1]

**Unsupervised learning :**

Unsupervised (or indirect) learning analyses are not guided by a target attribute. Taking the distinction even further, seven basic data mining tasks can be identified:

* characterization and discrimination
* classification
* regression
* time series analysis
* association rules
* clustering
* description and visualization

**Characterization and discrimination :**

Where a categorical target attribute exists, before starting to develop a classification model, it is often useful to carry out an exploratory analysis whose purpose is twofold. On the one hand, the aim is to achieve a characterization by comparing the distribution of the values of the attributes for the records belonging to the same class. [1]

**Classification :**

A set of observations is available, usually represented by the records of a dataset, whose target class is known. Observations may correspond, for instance, to mobile phone customers anther binary class may indicate whether a given customer is still active or has churned. Each observation is described by a given number of attributes whose value is known; in the previous categorical in classification problems and therefore takes on a finite and usually rather small number of values. In most applications the target is even represented by a binary variable. The categorical nature of the target determines the distinction between classification and regression. [1]

**Regression :**

Used when the target variable takes on continuous values egression. Based on the available explanatory attributes, the goal is to predict the value of the target variable for each observation. If one wishes to predict the sales of a product based on the promotional campaigns mounted and the sale price, the target variable may take on a very high number of discrete values and can be treated as a continuous variable. [1]

**Time series :**

Time and is therefore associated with adjacent periods on the time axis. In this case, the sequence of values of the target variable is said to represent a time series. [1]

**Association rules :**

Association rules, also known as affinity groupings ,aroused to identify interesting and recurring associations between groups of records of a dataset. For example, it is possible to determine which products are purchased together in a single transaction and how frequently. [1]

**Clustering :**

The term cluster refers to a homogeneous subgroup existing within a population. Clustering techniques are therefore aimed at segmenting a heterogeneous population into a given number of subgroups composed of observations that share similar characteristics; observations included in different clusters have distinctive features. [1]

**Description and visualization :**

In contrast to clustering and association rules, descriptive analysis does not pursue any particular grouping or partition of the records in the dataset, an effective and concise description of information is very helpful, since it may suggest possible explanations of hidden patterns in the data and lead to a better understanding the phenomena to which the data refer. [1]

**2.8 : Classification**

Classification models are supervised learning methods for predicting the value of a categorical target attribute, unlike regression models which deal with numerical attributes. Starting from a set of past observations whose target class is known, classification models are used to generate a set of rules that allow the target class of future examples to be predicted.

The opportunities afforded by classification extend into several different application domains: selection of the target customers for a marketing campaign, fraud detection, image recognition, early diagnosis of diseases, text cataloguing and spam email recognition are just a few examples of real problems that can be framed within the classification paradigm. [1]

2.8.1 : Classification Problems

In a classification problem, we have a dataset *D* containing *m* observations described in terms of *n explanatory* attributes and a categorical *target* attribute. [1]

The explanatory attributes, also called *predictive variables*, may be partly categorical and partly numerical. The target attribute is also called a *class* or *label* , while the observations are also termed *examples* or *instances*. Unlike regression, for classification models the target variable takes a finite number of values. In particular, we have a *binary* classification problem if the instances belong to two classes only, and a *multiclass* or *multicategory* classification if there are more than two classes. The purpose of a classification model is to identify recurring relationships among the explanatory variables which describe the examples belonging to the same class. Such relationships are then translated into *classification rules* which are used to predict the class of examples for which only the values of the explanatory attributes are known. The rules may take different forms depending on the type of model used. [1]

2.8.2 : Taxonomy of Classification Models

We can distinguish four main categories of classification models: Heuristic models, Separation models, Regression models, and Probabilistic models.

**Heuristic models :**

Heuristic methods make use of classification procedures based on simple and intuitive algorithms. This category includes *nearest neighbor* methods, based on the concept of distance between observations, and *classification* *trees*, which use *divide-and-conquer* schemes to derive groups of observations that are as homogeneous as possible with respect to the target class. [1]

**Separation models :**

Separation models divide the attribute space *H* into disjoint regions {S1, S2, . . . , SH}, separating the observations based on the target class. [1]

**Regression models :**

Regression models, These for prediction of continuous target variables, make an explicit assumption concerning the functional form of the conditional probabilities, which correspond to the assignment of the target class by the supervisor. [1]

**Probabilistic models :**

In probabilistic models, a hypothesis is formulated regarding the functional form of the conditional probabilities of the observations given the target class, known as class-conditional probabilities. Naive Bayes classifiers and Bayesian networks are well-known families of probabilistic methods. [1]

**2.9 : Evaluation of Classification Models**

Classification methods can be evaluated based on several criteria, as follows :

**Accuracy :**

Evaluating the accuracy of a classification model is crucial for two main reasons. First, the accuracy of a model is an indicator of its ability to predict the target class for future observations. Based on their accuracy values, it is also possible to compare different models in order to select the classifier associated with the best performance. [1]

**Speed :**

Classification methods characterized by longer computation times may be applied to a small-size training set obtained from a large number of observations by means of random sampling schemes. [1]

**Robustness :**

A classification method is *robust* if the classification rules generated, as well as the corresponding accuracy, do not vary significantly as the choice of the training set and the test set varies, and if it is able to handle missing data and outliers. [1]

**Scalability :**

The scalability of a classifier refers to its ability to learn from large datasets, and it is inevitably related to its computation speed. [1]

**Interpretability :**

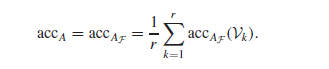
If the aim of a classification analysis is to interpret as well as predict, then the rules generated should be simple and easily understood by knowledge workers and experts in the application domain. [1]

2.9.1 : Holdout Method

The *holdout* estimation method involves subdividing the *m* observations available into two disjoint subsets *T* and *V*, for training and testing purposes respectively, and then evaluating the accuracy of the model through the accuracy acc*A(V)* on the test set. [1]

2.9.2 : Repeated Random Sampling

The *repeated random sampling* method involves replicating the holdout method a number *r* of times. For each repetition a random independent sample *Tk* is extracted, which includes *t* observations, and the corresponding accuracy acc*A(Vk)* is evaluated, where *Vk* = *D* − *Tk*. At the end of the procedure, the accuracy of the classifier *AF* is estimated using the sample mean: [1]



The total number *r* of repetitions may be evaluated a priori using techniques from statistical inference for determining the appropriate sample size. [1]

2.9.3 : Cross-Validation

The method of *cross-validation* offers an alternative to repeated random sampling techniques and guarantees that each observation of the dataset *D* appears the same number of times in the training sets and exactly once in the test sets.[1]

2.9.4 : Confusion Matrices

The accuracy measurement methods described above are not always adequate for discriminating among models, and in some instances they may even yield paradoxical results. [1]

2.9.5 : ROC Curve Charts

*Receiver operating characteristic* (ROC) curve charts allow the user to visually evaluate the accuracy of a classifier and to compare different classification models. [1]

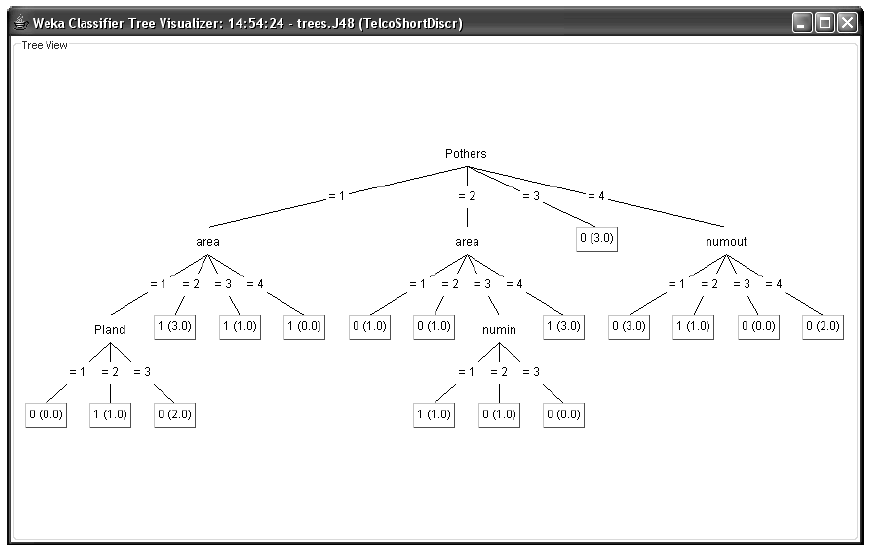
An ROC chart is a two-dimensional plot with the proportion of false positives fp on the horizontal axis and the proportion of true positives tp on the vertical axis. [1]

2.9.6 : Cumulative Gain and Lift Charts

The *lift* measure corresponds to the intuitive idea of evaluating the accuracy of a classifier based on the density of positive observations inside the set that has been identified based on model predictions. [1]

**2.10 : Classification Trees**

*Classification trees* are perhaps the best-known and most widely used learning methods in data mining applications. The reasons for their popularity lie in their conceptual simplicity, ease of usage, computational speed, robustness with respect to missing data and outliers and, most of all, the interpretability of the rules they generate. To separate the observations belonging to different classes, methods based on trees obtain simple and explanatory rules for the relationship existing between the target variable and predictive variables. [1]



**Figure (2.2): Example of a classification tree [1]**

**Top-down induction of decision trees :**

**1.** In the initialization phase, each observation is placed in the root node of the tree. The root is included in the list L of active nodes.

**2.** If the list L is empty the procedure is stopped, otherwise a node J belonging to the list L is selected, is removed from the list and is used as the node for analysis.

**3.** The optimal rule to split the observations contained in J is then determined, based on an appropriate preset criterion. The splitting rule generated in this way is then applied, and descendant nodes are constructed by subdividing the observations contained in J. For each descendant node the conditions for stopping the subdivision are verified. If these are met, node J becomes a leaf, to which the target class is assigned according to the majority of the observations contained in J. Otherwise, the descendant nodes are added to the list L. Finally, step 2 is repeated.

**Splitting rules :**

For each node of the tree it is necessary to specify the criteria used to identify the optimal rule for splitting the observations and for creating the descendant nodes. As shown in the next section, there are several alternative criteria, which differ in the number of descendants, the number of attributes and the evaluation metrics. [1]

**Stopping criteria :**

At each node of the tree different *stopping* criteria are applied to establish whether the development should be continued recursively or the node should be considered as a leaf. In this case too, various criteria have been proposed, which result in quite different topologies of the generated trees, all other elements being equal. [1]

**Pruning criteria :**

Finally, it is appropriate to apply a few *pruning* criteria, first to avoid excessive growth of the tree during the development phase (*pre-pruning*), and then to reduce the number of nodes after the tree has been generated (*post-pruning*). [1]

2.10.1 : Splitting Rules

Classification trees can be divided into *binary* and *general* trees based on the maximum number of descendants that each node is allowed to generate. [1]

**Binary trees :**

A tree is said to be *binary* if each node has at most two branches. Binary trees represent in a natural way the subdivision of the observations contained at a node based on the value of a binary explanatory attribute. When dealing with categorical attributes with more than two classes, binary trees should necessarily form two groups of categories in order to perform a split. Numerical attributes can be separated based on a threshold value. Finally, binary trees may also be used to develop multicategory classification. [1]

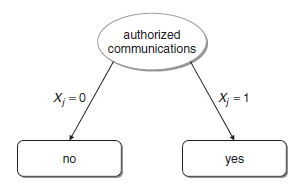
**Multi-split classification trees :**

A tree is said to be *multi-split* if each node has an arbitrary number of branches. This allows multi-valued categorical explanatory attributes to be handled more easily. [1]

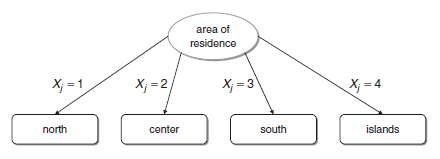
**Univariate trees :**

For *univariate* trees the splitting rule is based on the value assumed by a single explanatory attribute. If the selected attribute is categorical, the observations at a given node are divided by the mean. [1]

For example, for a binary attribute taking the values {0*,*1}, the two subsets can correspond to the values {0} and {1}. For a non-binary categorical attribute. If a numerical attribute, like the age of a customer, may divided to a scope as <= 45 and > 45.



**Figure (2.3): Univariate split for a binary attribute [1]**



**Figure (2.4): Univariate split for a nominal attribute [1]**

****

**Figure (2.5): Univariate split for a numerical attribute [1]**

**Multivariate trees :**

For multivariate trees, the partition of the observations at a given node is based on the value assumed by a function *ϕ(x*1*, x*2*, . . . , xn)* of the attributes and leads to a rule of the form *ϕ(***x***)* ≤ *b* or *ϕ(***x***)>b*. [1]

2.10.2 : Univariate Splitting Criteria

Although they are usually characterized by a lower accuracy, the algorithms that develop classification trees based on univariate rules are more popular than their multivariate counterpart, partly because of the simplicity and interpretability of the rules generated and partly because they were proposed first. [1]

2.10.3 : Development of a Classification Tree

The main point of the development of two classification trees obtained by using respectively the entropy and the Gini index is that the last one are easy to use because its dose not depend on LOGs operations, unlike the entropy index. [1]

2.10.4 : Stopping Criteria and Pruning Rules

Stopping criteria are a set of rules used at each node during the development of a tree in order to determine whether it is appropriate to create more branches and generate descendant nodes or whether the current node should become a leaf. [1]

A node becomes a leaf of the tree when at least one of the following conditions occurs:

**Node size :**

The node contains a number of observations that is below a preset minimum threshold value. [1]

**Purity :**

The proportion of observations at the node and belonging to the same class is above a preset maximum threshold value that corresponds to the accuracy that one wishes to achieve. [1]

**Improvement :**

The possible subdivision of the node would generate a gain that is below a preset minimum threshold value. [1]

**2.11 : Bayesian Methods**

Bayesian methods belong to the family of probabilistic classification models. They explicitly calculate the *posterior* probability *P(y*|**x***)* that a given observation belongs to a specific target class by means of Bayes’ theorem, once the *prior* probability *P(y)* and the class conditional probabilities *P(***x**|*y)* are known. [1]

Unlike other methods described in this chapter, which are not based on probabilistic assumptions, Bayesian classifiers require the user to estimate the probability *P(***x**|*y)* that a given observation may occur, provided it belongs to a specific class. [1]

2.11.1 : Naive Bayesian Classifiers

In spite of the simplifying assumption of conditional independence of the attributes, which makes it easy to compute the conditional probabilities, the empirical evidence shows that Bayesian classifiers are often able to achieve accuracy levels which are not lower than those provided by classification trees or even by more complex classification methods. [1]

2.11.2 : Bayesian Networks

*Bayesian networks*, also called *belief networks*, allow the hypothesis of conditional independence of the attributes to be relaxed, by introducing some reticular hierarchical links through which it is possible to assign selected stochastic dependencies that experts of the application domain deem relevant. [1]

**2.12 : Logistic Regression**

*Logistic regression* is a technique for converting binary classification problems into linear regression ones. Logistic regression models present the same difficulties described in connection with regression models, from which they derive. [1]

**2.13 : Neural Networks**

A neural network is an oriented graph consisting of nodes, which in the biological analogy represent neurons, connected by arcs, which correspond to dendrites and synapses. [1]

2.13.1 : The Rosenblatt Perceptron

The Rosenblatt perceptron corresponds to a linear separation of the observations based on the target class. The aim of the iterative procedure is therefore to determine the coefficients of the separating hyperplane. [1]

2.13.2 : Multi-level Feed-forward Networks

It is includes the following components :

**Input nodes :**

The purpose of the input nodes is to receive as input the values of the explanatory attributes for each observation. Usually, the number of input nodes equals the number of explanatory variables. [1]

**Hidden nodes :**

Hidden nodes apply given transformations to the input values inside the network. Each node is connected to incoming arcs that go from other hidden nodes or from input nodes, and it is connected with outgoing arcs to output nodes or to other hidden nodes.[1]

**Output nodes :**

Output nodes receive connections from hidden nodes or from input nodes and return an output value that corresponds to the prediction of the response variable. In classification problems, there is usually only one output node.[1]



**Figure (2.6): Example of neural networks [1]**

**2.14 : Support Vector Machines**

Support vector machines identify a set of examples, called *support vectors*, which appear to be the most representative observations for each target class. [1]

2.14.1 : Structural Risk Minimization

As already observed, a classification algorithm *AF* defines an appropriate hypothesis space *F* and a function *f*\* ∈ *F* which optimally describes the relationship between the class value *y* and the vector of explanatory variables **x**.[1]

In order to describe the criteria for selecting the function *f*\*, let *V (y,f(***x***))* denote a loss function which measures the discrepancy between the valuesreturned by the predictive function *f (***x***)* and the actual values of the class *y*.[1]

To select an optimal hypothesis *f*\* ∈ *F*, decision theory suggests minimizingthe *expected risk* functional, defined as: [1]

C:\Users\MHT~1.M\AppData\Local\Temp\msohtmlclip1\01\clip_image001.png

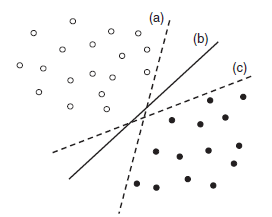
where *P(***x***, y)* = *P***x***,y (***x***, y)* denotes the joint probability distribution over R*n* × *H* of the examples *(***x***, y)* from which the instances in the dataset *D* are assumed to be independently drawn. [1]

Since the distribution *P(***x***, y)* is generally unknown, in place of the expected risk one is naturally led to minimize the *empirical risk* over the training set *T*, defined as: [1]

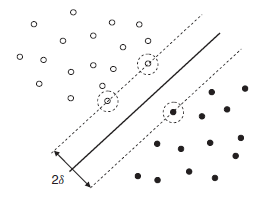


2.14.2 : Maximal Margin Hyperplane for Linear Separation

For linear separating functions represented by the hyperplanes in the space R*n*, the minimization of the right-hand side can be traced back to the maximization of the *margin of separation*, which will be described in the following figures. [1]



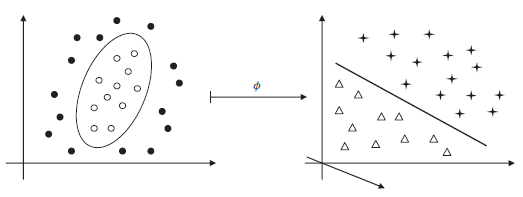
**Figure (2.7): The lines represent possible separating hyperplanes in the tow-dimensional space for the points in the diagram [1]**



**Figure (2.8): Maximal margin separating hyperplanes and canonical hyper-planes for a linearly separable dataset [1]**

2.14.3 : Nonlinear Separation

Linear separating functions are not able to perform accurate classifications when the set of examples is intrinsically characterized by a nonlinear pattern, like that shown in figure. [1]



**Figure (2.9): Nonlinear separation achieved by means of transformations in the feature space [1]**

**2.15 : Association Rules**

This sections (until 2.17) talking generally about association rules, chapter five describe this concept by case study example and detailed information through Apriori algorithm.

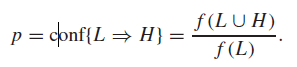
**Table (2.2): Example of a dataset consisting of transactions defined over the set of objects[1]**

|  |  |  |
| --- | --- | --- |
| Identifier *ti* |  | Transaction *Ti* |
| 001 |  | {*a, c*} |
| 002 |  | {*a, b, d*} |
| 003 |  | {*b, d*} |
| 004 |  | {*b, d*} |
| 005 |  | {*a, b, c*} |
| 006 |  | {*b, c*} |
| 007 |  | {*a, c*} |
| 008 |  | {*a, b, e*} |
| 009 |  | {*a, b, c, e*} |
| 010 |  | {*a, e*} |

*O*={*a, b, c, d, e*} = {bread*,* milk*,* cereals*,* coffee*,* tea}. [1]

**Single-dimension association rules :**

Given two itemsets L ⊂ O and H ⊂ O such that L ∩ H =∅and a transaction T, an association rule is a probabilistic implication denoted by L ⇒H with the following meaning: if L is contained in T, then H is also contained in T with a given probability p, termed the confidence of the rule in D and defined as: [1]



The set L is called the antecedent or body of the rule, and H is the consequentor head: [1]

C:\Users\MHT~1.M\AppData\Local\Temp\msohtmlclip1\01\clip_image001.png

The rule L ⇒H is said to have a support s in D if the proportion of Ns actions containing both L and H is equal to s, that is, if: [1]

C:\Users\MHT~1.M\AppData\Local\Temp\msohtmlclip1\01\clip_image001.png

The number NT of possible association rules increases exponentially as the number n of objects increases, according to the formula: [1]

C:\Users\MHT~1.M\AppData\Local\Temp\msohtmlclip1\01\clip_image001.png

The problem of generating strong association rules may be divided into two successive phases: first, the generation of *frequent itemsets*; and second, the generation of *strong rules*. [1]

**Generation of frequent itemsets :**

The aim of generating *frequent itemsets* is to extract all sets of objects whose relative frequency is greater than the assigned minimum support. This phase is more burdensome from a computational viewpoint than the subsequent phase of rule generation. Therefore, several algorithms have been proposed for obtaining the frequent itemsets in an efficient manner. In the following section we will describe the most popular one, known as the *Apriori* algorithm. [1]

**Generation of strong rules :**

After all frequent itemsets have been generated, we can proceed to the next phase of identifying *strong rules*. It is necessary to separate the objects contained in each frequent itemset according to all possible combinations of head and body of the rule and verify if the confidence of the rule in turn exceeds the minimum threshold. [1]

**Lift of a rule :**

Lift values greater than 1 indicate that the rule being considered is more effective than the relative frequency of the head in predicting the probability that the head is contained in some transaction of the dataset. In this case body and head of the rule are positively associated. Conversely, if the lift is less than 1 the rule is less effective than the estimate obtained through the relative frequency of the head. [1]



2.16 : **Apriori Algorithm**

The *Apriori* algorithm is a more efficient method of extracting strong rules contained in a set of transactions. During the first phase the algorithm generates the frequent itemsets in a systematic way, without exploring the space of all candidates, while in the second phase it extracts the strong rules.[1]

The theoretical assumption on which the Apriori algorithm is based consists of a property called the *Apriori principle*. [1]

**Apriori algorithm: generation of frequent itemsets :**

**1.** The transactions in the dataset are scanned to compute the relative frequency of each object. Objects with a frequency lower than the support threshold *s*min are discarded. At the end of this step, the collection of all frequent 1-itemsets has been generated, and the iteration counter is set to *k*=2.

**2.** The candidate *k*-itemsets are iteratively generated starting from the (*k* − 1)-itemsets, determined during the previous iteration.

**3.** The support of each candidate *k*-itemset is computed by scanning through all transactions included in the dataset.

**4.** Candidates with a support lower than the threshold *s*min are discarded.

**5.** The algorithm stops if no *k*-itemset has been generated. Otherwise, set *k* = *k* + 1 and return to step 2.

2.16.1 : Generation of Frequent Itemsets

The Apriori algorithm generates frequent itemsets iteratively, starting with the frequent 1-itemsets and then determining the frequent *k*-itemsets based upon the frequent (*k* − 1)-itemsets generated during the previous step. The total number of iterations is equal to *k*max + 1, where *k*max denotes the maximum cardinality of a frequent itemset in the dataset *D*. [1]

2.16.2 : Generation of Strong Rules

The second phase, generation of the strong rules, receives as input a list of all frequent itemsets, generated during the first phase, each associated with a relative frequency greater than the minimum threshold *s*min. [1]

**Apriori algorithm: generation of strong rules :**

1. The list of frequent itemsets generated during the first phase is scanned. If the list is empty, the procedure stops. Otherwise, let *B* be the next itemset to be considered, which is then removed from the list.
2. The set *B* of objects is subdivided into two non-empty disjoint subsets *L* and *H* = *B* − *L*, according to all possible combinations.
3. For each candidate rule *L* ⇒ *H*, the confidence is computed as: 
4. If *p* ≥ *p*min the rule is included into the list of strong rules, otherwise it is discarded.

**2.17 : General Association Rules**

The association rules described in the previous sections are *binary*, *asymmetric* and *one-dimensional*. The rules are *one-dimensional* becausethey involve only one logical dimension of data from multi-dimensional datawarehouses and data marts. However, it is possible to identify more general association rules that maybe useful across a range of applications. [1]

**Association rules for symmetric binary attributes :**

It is possible to extend the analysis with symmetric attributes to the case of asymmetric binary attributes, in order to apply the algorithms for the generation of association rules described in previous sections. In fact, a simple transformation can be used: for each symmetric binary variable, two asymmetric binary variables are introduced which correspond respectively to the presence of a given value and to the presence of its opposite. If, for example, the symmetric variable *communications* represents the answer provided in a registration form by customers regarding the consent to receive communications from the company, two asymmetric binary variables *consent* and *do not consent*, taking on the values {0,1} in relation to the value assumed by the attribute *communications*, can be introduced for each record. [1]

**Association rules for categorical attributes :**

In other situations the attributes may be categorical but not binary. In the previous registration form examples there may be attributes such as the province of residence or the level of education. In this case too it is possible to introduce, for each categorical attribute, a set of asymmetric binary variables, equal in number to the levels of the categorical attribute. Each binary variable takes the value 1 if in the corresponding record the categorical attribute assumes the level associated with the binary variable, otherwise it takes the value 0. [1]

**Association rules for continuous attributes :**

When dealing with continuous attributes, such as the age or income of those who have filled out a registration form, it is appropriate to proceed in two sequential phases. First, the continuous attribute is transformed into a categorical variable through one of the discretization techniques reviewed in Chapter 6. Then the discretized attribute is also transformed into asymmetric binary variables, as previously described, in order to proceed with the extraction of strong rules. [1]

**Multidimensional association rules :**

A rule in the context of market basket analysis may appear in the form ‘if a customer buys a digital camera, is 30–40 years old, and has an annual average expenditure of $300–500 on electronic equipment, then she will also buy a color printer with probability 0.78’. This is a three-dimensional rule, since the body consists of three dimensions (purchased items, age and expenditure amount), while the head refers to a purchase. Observe that the numerical variables that represent the age and the expenditure amount have previously been discretized, as described in the previous paragraph. [1]

**Multi-level association rules** :

The design of a data warehouse. Items are usually grouped by type and in turn by sales department. A possible way to remedy the rarefaction of objects in the transactions is to transfer the analysis to a higher level in the hierarchy of concepts. In this way the number of objects decreases and consequently the number of transactions containing the same object increases. Also the computational complexity significantly decreases, due to the reduced number of objects, so that the algorithms for rule extraction become much more efficient. [1]

**Sequential association rules :**

Often the transactions are recorded according to a specific temporal sequence. For example, the transactions for a loyalty card holder correspond to the sequence of sale receipts. By the same token, the transactions that gather the navigation paths followed by a given web user are associated with the temporal sequence of the sessions. In situations like these, analysts are often interested in extracting association rules that take into account temporal dependencies. [1]

**2.18 : Clustering**

The purpose of clustering methods is the identification of homogeneous groups of records called *clusters*. [1]

2.18.1 : Clustering Methods

Clustering models have long been used in various disciplines, such as social sciences, biology, astronomy, statistics, image recognition, processing of digital information, marketing and data mining. [1]

Clustering methods must fulfill a few general requirements, as indicated below :

**Flexibility :**

A flexible clustering algorithm should also be able to analyze datasets containing categorical attributes. [1]

**Robustness :**

The robustness of an algorithm manifests itself through the stability of the clusters generated with respect to small changes in the values of the attributes of each observation. [1]

**Efficiency :**

Efficiently in order to guarantee reasonable computing times for large problems. In the case of massive datasets, one may also resort to the extraction of samples of reduced size in order to generate clusters more efficiently. [1]

2.18.2 : Taxonomy of Clustering Methods

There are four types of taxonomy :

**Partition methods :**

Develop a subdivision of the given dataset into a predetermined number *K* of non-empty subsets. [1]

**Hierarchical methods :**

Carry out multiple subdivisions into subsets, based on a tree structure and characterized by different homogeneity thresholds within each cluster and inhomogeneity thresholds between distinct clusters. [1]

**Density-based methods :**

Density-based methods can identify clusters of non-convex shape and effectively isolate any possible outliers. [1]

**Grid methods :**

Obtaining a grid structure consisting of cells. Subsequent clustering operations are developed with respect to the grid structure and generally achieve reduced computing times. [1]

2.18.3 : Affinity Measures

In many instances this can be obtained by defining an appropriate notion of distance between each pair of observations. [1]

The definition of an appropriate notion of distance depends on the nature of the attributes that make up the dataset *D*. [1]

**Numerical attributes :**

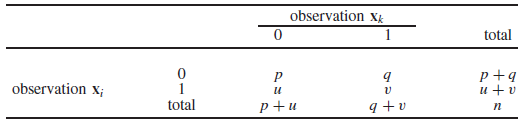
If all *n* attributes in a dataset are numerical, we may turn to the *Euclidean distance* between the vectors associated with the pair of observations which is so called because in order to reach one point from another we have to travel along two sides of a rectangle having the two points as its opposite vertices. [1]



**Figure (2.10): Euclidean distance (full line) and Manhattan distance (dotted line) between two points in the plane [1]**

**Binary attributes :**

Assumes only one of the two values 0 or 1. Assume for the moment that all *n* attributes in the dataset *D* are binary. In order to define a metric, reference should be made to the contingency table associated with the *n* attributes at two distinct observations **x***i* and **x***k*, as indicated in Figure 2.11 below. [1]



**Figure (2.11): Contingency table for a binary attribute [1]**

The value *p* is the number of binary attributes, *q* is the number of attributes for which observation **x***i* takes the value 0 and observation **x***k* takes the value 1. The value *u* is the number of attributes for which **x***i* assumes the value 1 and **x***k* assumes the value 0. [1]

**Nominal categorical attributes :**

Interpreted as a generalization of a symmetric binary attribute, for which the number of distinct values is greater than two. [1]

Nominal categorical variable by means of a number of asymmetric binary attributes equal to the number of distinct values assumed by the nominal variable. Once the nominal variables in the dataset have been replaced by the categorical attributes. [1]

**Ordinal categorical attributes :**

Assume that the values of each ordinal categorical attribute are represented through the corresponding position in the natural order. After we have carried out the transformation indicated for all the ordinal variables, we can use the measures of distance previously introduced for numerical attributes.[1]

**Mixed composition attributes :**

Mixed composition, in the sense that some of them are numerical, while others are binary symmetric or binary asymmetric or nominal categorical or ordinal categorical. [1]

**2.19 : Partition Methods**

Partition methods start with an initial assignment of the *m* available observations to the *K* clusters. Then, they iteratively apply a reallocation technique whose purpose is to place some observations in a different cluster.[1]

Partition methods are therefore of a heuristic nature, in the sense that they are based on a myopic logic typical of the class of so-called greedy methods, and at each step they make the choice that locally appears the most advantageous. [1]

The *K-means* method and the *K-medoids* method, which will be described next, are two of the best-known partition algorithms. [1]

2.19.1 : *K*-means Algorithm

The *K-means* algorithm receives as input a dataset *D*, a number *K* of clusters to be generated and a function dist*(***x***i ,* **x***k)* that expresses the inhomogeneity between each pair of observations, or equivalently the matrix **D** of distances between observations. [1]

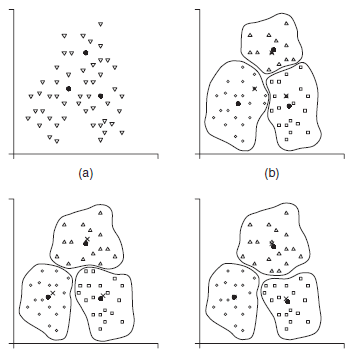
**Procedure - *K*-means algorithm :**

**1.** During the initialization phase, *K* observations are arbitrarily chosen in *D* as the centroids of the clusters.

**2.** Each observation is iteratively assigned to the cluster whose centroid is the most similar to the observation, in the sense that it minimizes the distance from the record.

**3.** If no observation is assigned to a different cluster with respect to the previous iteration, the algorithm stops.

**4.** For each cluster, the new centroid is computed as the mean of the values of the observations belonging to the cluster, and then the algorithm returns to step 2.



**Figure (2.12): An example of application of the K-means algorithm [1]**

2.19.2 : *K*-medoids Algorithm

The *K*-medoids algorithm proceeds like the *K*-means method, using medoids instead of centroids. The *K* medoids algorithm requires a large number of iterations and is not suited to deriving clusters for large datasets. A few variants have therefore been proposed, based on the extraction of samples of observations to which the *K*-medoids method can be applied. [1]

**2.20 : Hierarchical Methods**

Hierarchical clustering methods are based on a tree structure. Unlike partition methods, they do not require the number of clusters to be determined in advance. Hence, they receive as input a dataset *D* containing *m* observations and a matrix of distances dist*(***x***i ,* **x***k)* between all pairs of observations. [1]

Most hierarchical algorithms resort to one of five alternative measures: minimum distance, maximum distance, mean distance, distance between centroids, and Ward distance. [1]

**Minimum distance :**

*Minimum distance* is dissimilarity between two clusters is given by the minimum distance among all pairs of observations such that one belongs to the first cluster and the other to the second cluster. [1]

**Maximum distance :**

*Maximum distance* is dissimilarity between two clusters is given by the maximum distance among all pairs of observations such that one belongs to the first cluster and the other to the second cluster. [1]

**Mean distance :**

Is dissimilarity between two clusters via the mean of the distances between all pairs of observations belonging to the two clusters. [1]

**Distance between centroids :**

Is dissimilarity between two clusters through the distance between the centroids representing the two clusters. [1]

**Ward distance :**

The criterion of *Ward distance*, based on the analysis of the variance of the Euclidean distances between the observations, is slightly more complex than the criteria described above. Indeed, it requires the algorithm to first calculate the sum of squared distances between all pairs of observations belonging to a cluster. Afterwards, all pairs of clusters that could be merged at the current iteration are considered, and for each pair the total variance is computed as the sum of the two variances between the distances in each cluster. [1]

2.20.1 : Agglomerative Hierarchical Methods

Agglomerative methods are *bottom-up* techniques in which each single observation initially represents a distinct cluster. [1]

**Procedure - Agglomerative algorithm :**

**1.** In the initialization phase, each observation constitutes a cluster. The distance between clusters therefore corresponds to the matrix Dof the distances between all pairs of observations.

**2.** The minimum distance between the clusters is then computed, and the two clusters *Ch* and *Cf* with the minimum distance are merged, thus deriving a new cluster *Ce*. The corresponding minimum distance dist*(Ch,Cf )* originating the merger is recorded.

**3.** The distance between the new cluster *Ce*, resulting from the merger between *Ch* and *Cf* , and the preexisting clusters is computed.

**4.** If all the observations are included into a single cluster, the procedure stops. Otherwise it is repeated from step 2.

2.20.2 : Divisive Hierarchical Methods

Divisive algorithms are the opposite of agglomerative methods, in that they are based on a *top-down* technique, which initially places all the observations in a single cluster. This is then subdivided into clusters of smaller size, so that the distances between the generated subgroups are minimized. [1]

**2.21 : Evaluation of Clustering Models**

The evaluation of the predictive accuracy is part of the development process of a model and is based on specific numerical indicators. The same does not apply to clustering methods and, more generally, to unsupervised learning models. Even though the absence of a target attribute makes the evaluation of an unsupervised model less direct and intuitive, it is possible to define reasonable measures of quality and significance for clustering methods. [1]