Cybrarians Journal
journal.cybrarians.org
E-ISSN 1687-2215

Foundation of data mining: understanding the infrastructure of the knowledge factory

Ahmed Ramadan Ahmed

Business intelligent Developer, Data Analyst, Software Developer,

Raya Corporation, Egypt

mr.ahmed.ramadan@live.com

Abstract

This article is mainly speak to whom interested in knowledge discovery or data analysis and it`s techniques, also this article discuses the data mining (DM) and show what is this thing.

Anyone want to build a techniques for develop a data mining project or how to build it this article will help him in that issue this article focus on basics about data mining what is that-in conceptual and analysis skill- and who can use it where we can use it and how we apply this techniques in the business reality you will also know how data mining works and the algorithms that associated with data mining you will also knows what is the data mining project lifecycle and the knowledge creation process in a simple words you will be able to design a conceptual and logic design for a data mining project after reading this article.

Keywords: Data Mining, Data mining Algorisms, Data Models

How to cite this research

Ahmed Ramadan. Foundation of data mining: understanding the infrastructure of the knowledge factory .- cybrarians journal .- no 19 (June 2009) .- Accessed .- Available at:

DM introduction

This part focus on building a knowledge background of what is DM and setup the basic definition for it

What is exactly DM?

DM in a simple word is objective analysis of information you already collect this mean DM is NOT a program it`s a process and series of tools that help to analysis data and information that you are already collect to create knowledge to support decision making . For example you may collect demographical data through membership application or information about products or who attend seminars through registration database, if you simply query this database to know who attend last meeting or what the most product sold so you perform a DM. of course you can use a sophisticated techniques to ask some questions and get different answers but it all in the same cycle data analysis , and more time you spent to collect the more data you get the more quality you get .

But the misconception is if we have some inconsistence data entry problems this problems can`t be solved except with analysis tools that allows you to fix those problems and generate a cleaned and approved data to help you to discover the users or customer behaviors, or predict what they will do…

In previous we always ask what happen but with DM we will know what will happen and why.

Questions about DM?

What is the limitation of DM?

MIT – 1st ranked researching university in over all the world – consider DM on of top ten technologies that will change the world but mean the wrong collected data by users or customers on purpose , duplicate data or records , lack of data standards, human errors affect the result

So collected data must apply uniform ways of inputting data, so we have to take in mind this process of data or person-related rather than technology related.

How we can apply DM techniques in a organization?

First of all there is a series of question need to be answered by the organization First of all what data they are already collect it may be registration Database (DB) or purchasing DB or it membership DB I mean what information they currently have Second of all what decision did they want to take with this information like , did they want to make decisions to how to get more members or how to sell additional products or how to get

additional attendees to seminars or what is the business objectives they want to be solved

Where the organization should start to apply DM?

This diagram shows start point and the no end point

Skills you should have to deal with DM

If we look at the roots of data mining Core, we find that they are mainly derived from three fields: statistics, Machine learning and database. And Of course there is a lot of skills that you will need to deal with DM but

There are three Main required skills

Analytical skill this skill let you design a logic DM models and how to translate organizations problems to understandable problems that can solve by DM models (which this article is focus on).

Technical skill and I mean by that how to use software to build efficient DM models, how to train, update or manage the models and how to query models.

Statistical skill how choose the appropriate algorism to solve a certain problem and how we should set the models parameters.

DM benefits.

The benefits of data mining are that it allows you to explore your data and look for relationships, patterns and trends. Provided these relationships are statistically sound, it enables you to investigate further and then act on the patterns and trends for the benefit of the business

So DM is your weapon to dominate the world but I'll sort a few reasons that lead you to use DM

Discover your data:

Organizations have gathered huge amounts of data through many applications. With all of this data to explore, Organizations want to be able to find hidden patterns to help guide their business strategies.

Increasing competition:

Competition is high as a result of modern marketing and distribution channels such as the Internet and telecommunications. Organizations are facing worldwide competition, and the key to business success is the ability to retain existing customers and acquire new ones. Data mining contains technologies that allow Organizations to analyze factors that affect these issues.

Major products and vendors for DM

There are hundreds of data mining product and consulting companies. KDNuggets(kdnuggets.com) has an extended list of most of these companies and their products in the data mining field. Here we list a few the major data mining product companies.

SAS: SAS is probably the largest data mining product vendor in terms of the market share. SAS has been in the statistics field for decades. SAS Base contains a very rich set of statistical functions that can be used for all sorts of data analysis. It also has a powerful script language called SAS Script. SAS Enterprise Miner was introduced in 1997. It provides the user with a graphical flow environment for model building, and it has a set of popular data mining algorithms, including decision trees, neural network, regression, association, and so on. It also supports text mining.

SPSS: SPSS is another major statistics company. It has a number of data mining products including SPSS base and Answer Tree (decision trees). SPSS acquired a British company ISL in late 1998 and inherited the

Clementine data mining package. Clementine was one of the first companies to introduce the data mining flow concept, allowing users to clean data, transform data, and train models in the same workflow environment. Clementine also has tools to manage data mining project cycle.

IBM: IBM has a data mining product called Intelligent Miner, developed by an IBM German subsidiary. Intelligent Miner contains a set of algorithms and visualization tools. Intelligent Miner exports mining models in Predictive Modeling Markup Language (PMML), which was defined by the Data Mining Group (DMG), an industry organization. PMML documents are Extensible Markup Language (XML) files containing the descriptions of model patterns and statistics of training dataset. These files can be loaded by DB2 database for prediction purpose.

Microsoft Corporation: Microsoft was the first major database vendor to include data mining features in a relational database. SQL Server 2000, released in September 2000, contains two patented data mining algorithms: Microsoft Decision Trees and Microsoft Clustering. Apart from these algorithms, the most important data mining feature is the implementation of OLE DB for Data Mining. OLE DB for Data Mining is an industry standard that defines a SQL-style data mining language and a set of schema row sets targeted at database developers. This API makes it very easy to embed data mining components, especially prediction features, into user applications. Now MS release the Business intelligent

Studio 2005 and 2008 (By the way I prefer MS Business intelligent Studio 2005 or later) which a very powerful tool to implement DM models

Oracle: Oracle 9i shipped in 2000, containing a couple of data mining algorithms based on association and Naïve Bayes. Oracle 10g includes many more data mining tools and algorithms. Oracle also incorporated the Java Data Mining API, which is a Java package for data mining tasks.

Angoss: Angoss' Knowledge STUDIO is a data mining tool that includes the power to build decision trees, cluster analysis, and several predictive models, allowing users to mine and understand their data from many different perspectives. It includes powerful data visualization tools to support and explain the discoveries. Angoss also has a set of content viewer controls, which work with data mining algorithms in SQL Server 2000. Its algorithms can also be plugged into the SQL Server platform.

KXEN: KXEN is a data mining software provider based in France. It has a number of data mining algorithms, including SVM, regression, time series, segmentation, and so forth. It also provides data mining solutions for OLAP cubes. It developed an Excel add-in that allows users to do data mining in a familiar Excel environment.

How DM works

Business problems for DM

We can say that the core of DM is the Algorisms that use to solve problems but what are Business problems for DM listed below some of it (not all)

Cross-selling: What products are customers likely to purchase? Cross selling is an important business challenge for retailers. Many retailers, especially online retailers, use this feature to increase their sales. For example, if you go to online bookstores such as Amazon.com to purchase a book, you may notice that the Web site gives you a set of recommendations about related books. These recommendations can be derived from data mining analysis

Fraud detection: Is this insurance claim fraudulent? Insurance companies process thousands of claims a day. It is impossible for them to investigate each case. Data mining can help to identify those claims that are more likely to be false.

Risk management: Should the loan be approved for this customer? This is the most common question in the banking scenario. Data mining techniques can be

used to score the customer's risk level, helping the manager make an appropriate decision for each application.

Customer segmentation: Who are my customers? Customer segmentation helps marketing managers understand the different profiles of customers and take appropriate marketing actions based on the segments.

Sales forecast: How many cases of Products will I sell next week in this store? What will the inventory level be in one month? Data mining forecasting techniques can be used to answer these types of time-related questions.

Churn analysis (best customers): Every business would like to retain as many customers as possible. Churn analysis can help marketing managers understand the reason for customer churn, improve customer relations, and eventually increase customer loyalty

Data mining can be used to solve hundreds of business problems. Based on the nature of these problemsthis is a simple diagram that show the DM Architecture

DM Algorisms

We can divide the algorisms into these categories

1. **Classification**

Classification is one of the most popular data mining tasks. Business problems like churn analysis and risk management usually involve classification.

Classification refers to assigning cases into categories based on a predictable attribute. Each case contains a set of attributes, one of which is the class attribute (predictable attribute). The task requires finding a model that describes the class attribute as a function of input attributes. In the College Plans dataset previously described, the class is the College Plans attribute with two states: Yes and No. To train a classification model, you need to know the class value of input cases in the training dataset, which are usually the historical data. Data mining algorithms that require a target to learn against are considered supervised algorithms. Typical classification algorithms include decision trees, neural network, and Naïve Bayes.

Decision Trees Algorithm

The Decision Trees algorithm is a classification and regression algorithm provided by SQL Server Analysis Services for use in predictive modeling of both discrete and continuous attributes.

For discrete attributes, the algorithm makes predictions based on the relationships between input columns in a dataset. It uses the values, known as states, of those columns to predict the states of a column that you designate as predictable. Specifically, the algorithm identifies the input columns that are correlated with the predictable column. For example, in a scenario to predict which customers are likely to purchase a bicycle, if nine out of ten younger customers buy a bicycle, but only two out of ten older customers do so, the algorithm infers that age is a good predictor of bicycle purchase. The decision tree makes predictions based on this tendency toward a particular outcome.

For continuous attributes, the algorithm uses linear regression to determine where a decision tree splits.

If more than one column is set to predictable, or if the input data contains a nested table that is set to predictable, the algorithm builds a separate decision tree for each predictable column

How the Algorithm Works

The Decision Trees algorithm builds a data mining model by creating a series of splits in the tree. These splits are represented as nodes. The algorithm adds a node to the model every time that an input column is found to be significantly correlated with the predictable column. The way that the algorithm determines a split is different depending on whether it is predicting a continuous column or a discrete column.

A common problem in data mining models is that the model becomes too sensitive to small differences in the training data, in which case it said to be over-fitted or over-trained. An overfitted model cannot be generalized to other data sets. To avoid overfitting on any particular set of data

Predicting Discrete Columns

The way that the Decision Trees algorithm builds a tree for a discrete predictable column can be demonstrated by using a histogram. The following diagram shows a histogram that plots a predictable column, Car Buyers, against an input column, Age. The histogram shows that the age of a person helps distinguish whether that person will purchase a car.

The correlation that is shown in the diagram would cause the Decision Trees algorithm to create a new node in the model.

As the algorithm adds new nodes to a model, a tree structure is formed. The top node of the tree describes the breakdown of the predictable column for the overall population of customers. As the model continues to grow, the algorithm considers all columns.

Predicting Continuous Columns

When the Decision Trees algorithm builds a tree based on a continuous predictable column, each node contains a regression formula. A split occurs at a point of non-linearity in the regression formula. For example, consider the following diagram.

The diagram contains data that can be modeled either by using a single line or by using two connected lines. However, a single line would do a poor job of representing the data. Instead, if you use two lines, the model will do a much better job of approximating the data. The point where the two lines come together is the point of non-linearity, and is the point where a node in a decision tree model would split. For example, the node that corresponds to the point of non-linearity in the previous graph could be represented by the following diagram. The two equations represent the regression equations for the two lines.

Data Required for Decision Tree Models

When you prepare data for use in a decision trees model, you should understand the requirements for the particular algorithm, including how much data is needed, and how the data is used.

The requirements for a decision trees model are as follows:

- **A single key column** Each model must contain one numeric or text column that uniquely identifies each record. Compound keys are not permitted.
- **A predictable column** Requires at least one predictable column. You can include multiple predictable attributes in a model, and the predictable attributes can be of different types, either numeric or discrete. However,

increasing the number of predictable attributes can increaseprocessing time.

- **Input columns** Requires input columns, which can be discrete or continuous. Increasing the number of input attributes affects processing time.

Creating Predictions

After the model has been processed, the results are stored as a set of patterns and statistics, which you can use to explore relationships or make predictions.

Naïve Bayes Algorithm

The Naive Bayes algorithm is a classification algorithm provided by SQL Server Analysis Services for use in predictive modeling. The name Naive Bayes derives from the fact that the algorithm uses Bayes theorem but does not take into account dependencies that may exist, and therefore its assumptions are said to be naive.

This algorithm is less computationally intense than other algorithms, and therefore is useful for quickly generating mining models to discover relationships between input columns and predictable columns. You can use this algorithm to do initial explorations of data, and then later you can apply the results to create additional mining models with other algorithms that are more computationally intense and more accurate.

Example

As an ongoing promotional strategy, the marketing department for the Adventure Works Cycle company has decided to target potential customers by mailing out fliers. To reduce costs, they want to send fliers only to those customers who are likely to respond. The company stores information in a database about demographics and response to a previous mailing. They want to use this data to see how demographics such as age and location can help predict response to a promotion, by comparing potential customers to customers who have similar characteristics and who have purchased from the company in the past. Specifically, they want to see the differences between those customers who bought a bicycle and those customers who did not.

By using the Naive Bayes algorithm, the marketing department can quickly predict an outcome for a particular customer profile, and can therefore determine which customers are most likely to respond to the fliers. they can also visually investigate specifically which input columns contribute to positive responses to fliers.

How the Algorithm Works

The Naive Bayes algorithm calculates the probability of every state of each input column, given each possible state of the predictable column. as shown in the following graphic.

The Naive Bayes Viewer lists each input column in the dataset, and shows how the states of each column are distributed, given each state of the predictable column. You can use this view to identify the input columns that are important for differentiating between states of the predictable column. For example, in the Commute Distance column shown here, if the customer commutes from one to two miles to work, the probability that the customer will buy a bike is 0.387, and the probability that the customer will not buy a bike is 0.287. In this example, the algorithm uses the numeric information, derived from customer characteristics such as commute distance, to predict whether a customer will buy a bike..

Data Required for Naive Bayes Models

When you prepare data for use in training a Naive Bayes model, you should understand the requirements for the algorithm, including how much data is needed, and how the data is used.

The requirements for a Naive Bayes model are as follows:

- **A single key column** Each model must contain one numeric or text column that uniquely identifies each record. Compound keys are not allowed.
- **Input columns** In a Naive Bayes model, all columns must be either discrete or discretized columns. For a Naive Bayes model, it is important to ensure that the input attributes are independent of each other.
- **At least one predictable column** the predictable attribute must contain discrete or discredited values. The values of the predictable column can be treated as input and frequently are, to find relationships among the columns.

Making Predictions

After the model has been trained, the results are stored as a set of patterns, which you can explore or use to make predictions.

You can create queries to return predictions about how new data relates to the predictable attribute, or you can retrieve statistics that describe the correlations found by the model.

Neural Network Algorithm

the Neural Network algorithm combines each possible state of the input attribute with each possible state of the predictable attribute, and uses the training data to calculate probabilities. You can later use these probabilities for classification or regression, and to predict an outcome of the predicted attribute, based on the input attributes.

A mining model that is constructed with the Neural Network algorithm can contain multiple networks, depending on the number of columns that are used for both input and prediction, or that are used only for prediction. The number of networks that a single mining model contains depends on the number of states that are contained by the input columns and predictable columns that the mining model uses.

Example

The Neural Network algorithm is useful for analyzing complex input data, such as from a manufacturing or commercial process, or business problems for which a significant quantity of training data is available but for which rules cannot be easily derived by using other algorithms.

Suggested scenarios for using the Neural Network algorithm include the following:

- Marketing and promotion analysis, such as measuring the success of a direct mail promotion or a radio advertising campaign.
- Predicting stock movement, currency fluctuation, or other highly fluid financial information from historical data.
- Analyzing manufacturing and industrial processes.
- Text mining.
- Any prediction model that analyzes complex relationships between many inputs and relatively fewer outputs.

How the Algorithm Works

The Neural Network algorithm creates a network that is composed of up to three layers of neurons. These layers are an input layer, an optional hidden layer, and an output layer.

Input layer: Input neurons define all the input attribute values for the data mining model, and their probabilities.

Hidden layer: Hidden neurons receive inputs from input neurons and provide outputs to output neurons. The hidden layer is where the various probabilities of the inputs are assigned weights. A weight describes the relevance or importance of a particular input to the hidden neuron. The greater the weight that is assigned to an input, the more important the value of that input is. Weights can be negative, which means that the input can inhibit, rather than favor, a specific result.

Output layer: Output neurons represent predictable attribute values for the data mining model.

Data Required for Neural Network Models

A neural network model must contain a key column, one or more input columns, and one or more predictable columns.

Data mining models that use the Neural Network algorithm are heavily influenced by the values that you specify for the parameters that are available to the algorithm. The parameters define how data is sampled, how data is distributed or expected to be distributed in each column, and when feature selection is invoked to limit the values that are used in the final model.

Creating Predictions

After the model has been processed, you can use the network and the weights stored within each node to make predictions. A neural network model supports regression, association, and classification analysis, Therefore, the meaning of each prediction might be different.

1. **Clustering**

Clustering is also called segmentation. It is used to identify natural groupings of cases based on a set of attributes. Cases within the same group have more or less similar attribute values.

Figure below displays a simple customer dataset containing two attributes: age and income. The clustering algorithm groups the dataset into three segments based on these two attributes. Cluster 1 contains the younger population with a low income. Cluster 2 contains middle-aged customers with higher incomes. Cluster 3 is a group of senior individuals with a relatively low income.

Clustering is an unsupervised data mining task. No single attribute is used to guide the training process. All input attributes are treated equally. Most clustering algorithms build the model through a number of iterations and stop when the model converges, that is, when the boundaries of these segments are stabilized.

The algorithm uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and creating predictions.

Clustering models identify relationships in a dataset that you might not logically derive through casual observation. For example, you can logically discern that people who commute to their jobs by bicycle do not typically live a long distance from where they work. The algorithm, however, can find other characteristics about bicycle commuters that are not as obvious. In the following diagram, cluster A represents data about people who tend to drive to work, while cluster B represents data about people who tend to ride bicycles to work.

The clustering algorithm differs from other data mining algorithms, such as the Decision Trees algorithm, in that you do not have to designate a predictable column to be able to build a clustering model. The clustering algorithm trains the model strictly from the relationships that exist in the data and from the clusters that the algorithm identifies.

Example

Consider a group of people who share similar demographic information and who buy similar products from Amazon web site. This group of people represents a cluster of data. Several such clusters may exist in a database. By observing the columns that make up a cluster, you can more clearly see how records in a dataset are related to one another.

How the Algorithm Works

The Clustering algorithm first identifies relationships in a dataset and generates a series of clusters based on those relationships. A scatter plot is a useful way to

visually represent how the algorithm groups data, as shown in the following diagram. The scatter plot represents all the cases in the dataset, and each case is a point on the graph. The clusters group points on the graph and illustrate the relationships that the algorithm identifies.

After first defining the clusters, the algorithm calculates how well the clusters represent groupings of the points, and then tries to redefine the groupings to create clusters that better represent the data. The algorithm iterates through this process until it cannot improve the results more by redefining the clusters.

You can customize the way the algorithm works by selecting a specifying a clustering technique, limiting the maximum number of clusters, or changing the amount of support required to create a cluster.

Data Required for Clustering Models

When you prepare data for use in training a clustering model, you should understand the requirements for the particular algorithm, including how much data is needed, and how the data is used.

The requirements for a clustering model are as follows:

- **A single key column** Each model must contain one numeric or text column that uniquely identifies each record. Compound keys are not allowed.
- **Input columns** Each model must contain at least one input column that contains the values that are used to build the clusters. You can have as many input columns as you want, but depending on the number of values in each column, the addition of extra columns can increase the time it takes to train the model.
- **Optional predictable column** The algorithm does not need a predictable column to build the model, but you can add a predictable column of almost any data type. The values of the predictable column can be treated as input to the clustering model, or you can specify that it be used for prediction only. For example, if you want to predict customer income by clustering ondemographics such as region or age, you would specify income as**Predict Only** and add all the other columns, such as region or age, as inputs.

Creating Predictions

After the model has been trained, the results are stored as a set of patterns, which you can explore or use to make predictions.

You can create queries to return predictions about whether new data fits into the clusters that were discovered, or to obtain descriptive statistics about the clusters.

1. **Association**

Association is another popular data mining task. Association is also called market basket analysis. A typical association business problem is to analyze a sales transaction table and identify those products often sold in the same shopping basket. The common usage of association is to identify common sets of items (frequent item sets) and rules for the purpose of cross-selling. In terms of association, each product, or more generally, each attribute/value pair is considered an item. The association task has two goals: to find frequent

item sets and to find association rules. Most association type algorithms find frequent item sets by scanning the

dataset multiple times. The frequency threshold (support) is defined by the user before processing the model. For example, support = 2% means that the model analyzes only items that appear in at least 2% of shopping carts. A frequent item set may look like {Product = “Pepsi”, Product = “Chips”, Product = “Juice”}.

Each item set has a size, which is the number of items that it contains. The size of this particular item set is 3.

Apart from identifying frequent item sets based on support, most association type algorithms also find rules. An association rule has the form A, B => C with a probability, where A, B, C are all frequent item sets. The probability is also referred to as the confidence in data mining literature. The probability is a threshold value that the user needs to specify before training an association model. For example, the following is a typical rule: Product = “Pepsi”, Product = “Chips” => Product = “Juice” with an 80% probability. The interpretation of

this rule is straightforward. If a customer buys Pepsi and chips, there is an 80% chance that he or she may also buy juice. Figure below displays the product association patterns. Each node in the figure represents a product, each edge represents the relationship. The direction of the edge represents the direction of the prediction. For example, the edge from Milk to Cheese indicates that those who purchase milk might also purchase cheese.

The Association algorithm is an association algorithm is useful for recommendation engines. A recommendation engine recommends products to customers based on items they have already bought, or in which they have indicated an interest..

Example

The E-pay web site company is redesigning the functionality of its Web site. The goal of the redesign is to increase sell-through of products. Because the company records each sale in a transactional database, they can use the Association algorithm to identify sets of products that tend to be purchased together. They can then predict additional items that a customer might be interested in, based on items that are already in the customer's shopping basket.

Data Required for Association Models

When you prepare data for use in an association rules model, you should understand the requirements for the particular algorithm, including how much data is needed, and how the data is used.

The requirements for an association rules model are as follows:

- **A single key** column Each model must contain one numeric or text column that uniquely identifies each record. compound keys not permitted.
- **A single predictable column** An association model can have only one predictable column. Typically it is the key column of the nested table, such as the field that lists the products that were purchased. The values must be discrete
- **Input columns** . The input columns must be discrete. The input data for an association model often is contained in two tables. For example, one table might contain customer information while another table contains customer purchases. You can input this data into the model by using a nested table.

Creating Predictions

After the model has been processed, you can use the rules and item sets to make predictions or to perform a promotions. In an association model, a prediction tells you what item is likely to occur given the presence of the specified item, and the prediction can include such information as the probability, the support, or the importance.

Performance

The process of creating item sets and counting correlations can be time-consuming. Although the Association Rules algorithm uses optimization techniques to save space and make processing faster, you should know that that performance issues might occur under conditions such as the following:

- Data set is large with many individual items.
- Minimum item set size is set too low.

To minimize processing time and reduce the complexity of the item sets, you might try grouping related items by categories before you analyze the data.

- **Regression**

The regression task is similar to classification. The main difference is that the predictable attribute is a continuous number. Regression techniques have been widely studied for centuries in the field of statistics. Linear regression and logistic regression are the most popular regression methods. Other regression techniques include regression trees and neural networks. Regression tasks can solve many business problems. For example, they can be used to predict coupon redemption rates based on the face value, distribution method, and distribution volume, or to predict wind velocities based on temperature, air pressure, and humidity.

Linear Regression Algorithm

The Linear Regression algorithm is a variation of the Decision Trees algorithm that helps you calculate a linear relationship between a dependent and independent variable, and then use that relationship for prediction.

The relationship takes the form of an equation for a line that best represents a series of data. For example, the line in the following diagram is the best possible linear representation of the data.

Each data point in the diagram has an error associated with its distance from the regression line. The coefficients a and b in the regression equation adjust the angle and location of the regression line. You can obtain the regression equation by adjusting a and b until the sum of the errors that are associated with all the points reaches its minimum.

There are other kinds of regression that use multiple variables, and also nonlinear methods of regression. However, linear regression is a useful and well-known method for modeling a response to a change in some underlying factor.

Example

You can use linear regression to determine a relationship between two continuous columns. For example, you can use linear regression to compute a trend line from manufacturing or sales data. You could also use the linear regression as a precursor to development of more complex data mining models, to assess the relationships among data columns.

Although there are many ways to compute linear regression that do not require data mining tools, the advantage of using the Linear Regression algorithm for this task is that all the possible relationships among the variables are automatically computed and tested. You do not have to select a computation method, such as solving for least squares. However, linear regression might oversimplify the relationships in scenarios where multiple factors affect the outcome.

How the Algorithm Works

The Linear Regression algorithm is a variation of the Decision Trees algorithm. When you select the Linear Regression algorithm, a special case of the Decision Trees algorithm is invoked, with parameters that constrain the behavior of the algorithm and require certain input data types. Moreover, in a linear regression model, the whole data set is used for computing relationships in the initial pass, whereas a standard decision trees model splits the data repeatedly into smaller subsets or trees.

Data Required for Linear Regression Models

When you prepare data for use in a linear regression model, you should understand the requirements for the particular algorithm. This includes how much data is needed, and how the data is used. The requirements for this model type are as follows:

- **A single key** column Each model must contain one numeric or text column that uniquely identifies each record. Compound keys are not permitted.
- **A predictable** column Requires at least one predictable column. You can include multiple predictable attributes in a model, but the predictable attributes must be continuous numeric data types. You cannot use a datetime data type as a predictable attribute even if the native storage for the data is numeric.

- **Input columns** Input columns must contain continuous numeric data and be assigned the appropriate data type.

Creating Predictions

After the model has been processed, the results are stored as a set of statistics together with the linear regression formula, which you can use to compute future trends. For general information about how to

In addition to creating a linear regression model by selecting the Linear Regression algorithm, if the predictable attribute is a continuous numeric data type, you can create a decision tree model that contains regressions. In this case, the algorithm will split the data when it finds appropriate separation points, but for some regions of data, will create a regression formula instead.

Logistic Regression Algorithm

The Logistic Regression algorithm is a variation of the Neural Network algorithm. Logistic regression is a well-known statistical technique that is used for modeling binary outcomes, such as a yes-No outcome.

Logistic regression is highly flexible, taking any kind of input, and supports several different analytical tasks:

- Use demographics to make predictions about outcomes, such as risk for a certain disease.
- Explore and weight the factors that contribute to a result. For example, find the factors that influence customers to make a repeat visit to a store.
- Classify documents, e-mail, or other objects that have many attributes.

Example

Consider a group of people who share similar demographic information and who buy products from a company. By modeling the data to relate to a specific outcome, such as purchase of a target product, you can see how the demographic information contributes to someone's likelihood of buying the target product.

How the Algorithm Works

Logistic regression is a well-known statistical method for determining the contribution of multiple factors to a pair of outcomes. The implementation uses a modified neural network to model the relationships between inputs and outputs. The effect of each input on the output is measured, and the various

inputs are weighted in the finished model. The name logistic regression comes from the fact that the data curve is compressed by using a logistic transformation, to minimize the effect of extreme values.

Data Required for Logistic Regression Models

When you prepare data for use in training a logistic regression model, you should understand the requirements for the particular algorithm, including how much data is needed, and how the data is used.

The requirements for a logistic regression model are as follows:

A single key column Each model must contain one numeric or text column that uniquely identifies each record. Compound keys are not allowed.

Input columns Each model must contain at least one input column that contains the values that are used as factors in analysis. You can have as many input columns as you want, but depending on the number of values in each column, the addition of extra columns can increase the time it takes to train the model.

At least one predictable column the model must contain at least one predictable column of any data type, including continuous numeric data. The values of the predictable column can also be treated as inputs to the model, or you can specify that it be used for prediction only. Nested tables are not allowed for predictable columns, but can be used as inputs.

Creating Predictions

After the model has been trained, you can create queries against the model content to get the regression coefficients and other details, or you can use the model to make predictions.

1. **Forecasting**

Forecasting is yet another important data mining task. What will the stock value of CASE 30 be tomorrow? What will the sales amount of Pepsi be next month? Forecasting can help to answer these questions. It usually takes as an input time series dataset, for example a sequence of numbers with an attribute representing time. The time series data typically contains adjacent observations, which are order-dependant. Forecasting techniques deal with general trends and periodicity. The most popular time series technique is ARIMA, which stands for Auto Regressive Integrated Moving Average model.

Figure below contains two curves. The solid line curve is the actual time series data on Microsoft stock value, while the dotted curve is a time series model based on the moving average forecasting technique.

Time Series Algorithm

The Time Series algorithm provides regression algorithms that are optimized for the forecasting of continuous values, such as product sales, over time. Whereas other algorithms, such as decision trees, require additional columns of new information as input to predict a trend, a time series model does not. A time series model can predict trends based only on the original dataset that is used to create the model.You can also add new data to the model when you make a prediction and automatically incorporate the new data in the trend analysis.

The following diagram shows a typical model for forecasting sales of a product in four different sales regions over time. The model that is shown in the diagram shows sales for each region plotted as red, yellow, purple, and blue lines. The line for each region has two parts:

- Historical information appears to the left of the vertical line and represents the data that the algorithm uses to create the model.
- Predicted information appears to the right of the vertical line and represents the forecast that the model makes.

The combination of the source data and the prediction data is called a series.

An important feature of the Time Series algorithm is that it can perform cross prediction. If you train the algorithm with two separate, but related, series, you can use the resulting model to predict the outcome of one series based on the behavior of the other series. For example, the observed sales of one product can influence the forecasted sales of another product. Cross prediction is also useful for creating a general model that can be applied to multiple series. For example, the predictions for a particular region are unstable because the series lacks good quality data. You could train a general model on an average of all four regions, and then apply the model to the individual series to create more stable predictions for each region.

Data Required for Time Series Models

When you prepare data for use in training any data mining model, make sure that you understand the requirements for the particular model and how the data is used.

Each forecasting model must contain a case series, which is the column that specifies the time slices or other series over which change occurs. For example, the data in the previous diagram shows the series for historical and forecasted sales over a period of several months. For this model, each region is a series, and the date column contains the time series, which is also the case series. In other models, the case series can be a text field or some identifier such as a customer ID or transaction ID. However, a time series model must always use a date, time, or some other unique numeric value for its case series.

The requirements for a time series model are as follows:

- **A single key time column** Each model must contain one numeric or date column that is used as the case series, which defines the time slices that the model will use. The data type for the key time column can be either a datetime data type or a numeric data type. However, the column must contain continuous values, and the values must be unique for each series. The case series for a time series model cannot be stored in two columns, such as a Year column and a Month column.
- **A predictable column** Each model must contain at least one predictable column around which the algorithm will build the time series model. The data type of the predictable column must have continuous values. For example, you can predict how numeric attributes, such as income, sales, or temperature, change over time. However, you cannot use a column that contains discrete values, such as purchasing status or level of education, as the predictable column.
- **An optional series key column** Each model can have an additional key column that contains unique values that identify a series. The optional series key column must contain unique values. For example, a single model can contain sales for many product models, as long as there is only one record for each product name for every time slice.

Creating Time Series Predictions

By default, when you view a time series model, Analysis Services shows you five predictions for the series. However, you can create queries to return a variable number of predictions, and you can extra columns to the predictions to return descriptive statistics.

- **Sequence Analysis**

Sequence analysis is used to find patterns in a discrete series. A sequence is composed of a series of discrete values (or states). For example, a DNA sequence is a long series composed of four different states: A, G, C, and T. A Web click sequence contains a series of URLs. Customer purchases can also be modeled as sequence data. For example, a customer first buys a car, then sound system, and finally a NOS. Both sequence and time series data contain adjacent observations that are dependant. The difference is that the sequence series contains

discrete states, while the time series contains continuous numbers.

Sequence and association data are similar in the sense that each individual case contains a set of items or states. The difference between sequence and association models is that sequence models analyze the state transitions, while the association model considers each item in a shopping cart to be equal and independent. With the sequence model, buying a car before buying sound system is a different sequence than buying sound system before a car. With an association algorithm, these are considered to be the same item set.

Figure below displays Web click sequences. Each node is a URL category. Each line has a direction, representing a transition between two URLs. Each transition is associated with a weight, representing the probability of the transition between one URL and the other.

Sequence analysis is a relatively new data mining task. It is becoming more important mainly due to two types of applications: Web log analysis and DNA analysis. There are several different sequence techniques available today such as Markov chains.

1. **Deviation Analysis**

Deviation analysis is for finding those rare cases that behave very differently from others. It is also called outlier detection, which refers to the detection of significant changes from previously observed behavior. Deviation analysis can be used in many applications. The most common one is credit card fraud detection. To identify abnormal cases from millions of transactions is a very challenging task. Other applications include network intrusion detection, manufacture error analysis, and so on.

There is no standard technique for deviation analysis. It is still an actively researched topic. Usually analysts employ some modified versions of decision

trees, clustering, or neural network algorithms for this task. In order to generate significant rules.

.

When to Use What

Analytical Problem	Examples	Algorithms
Classification: Assign cases to predefined classes	Credit risk analysis Churn analysis Customer retention	Decision Trees NaiveBayes Neural Nets
Segmentation: Taxonomy for grouping similar cases	Customer profile analysis Mailing campaign	Clustering Sequence Clustering
Association: Advanced counting for correlations	Market basket analysis Advanced data exploration	Decision Trees Association
Time Series Forecasting: Predict the future	Forecast sales Predict stock prices	Time Series
Prediction: Predict a value for a new case based on values for similar cases	Quote insurance rates Predict customer income	All
Deviation analysis: Discover how a case or segment differs from others	Credit card fraud detection Network infusion analysis	All

DM project life cycle (knowledge Creation process)

We can consider DM models as any system so as we know any system consist of three main parts inputs, process and outputs

Know let`s go to apply this concept on DM

We will find that shape

Translating the Data Mining Process into Steps

As you've just learned, data mining is a process. Though the end step is clearly building a mining model, the steps leading up to the creation of the model determine the success of your solution. While there are a multitude of approaches to the data mining process, all of them roughly translate into the distinct steps

Step 1—Problem Definition

Before you build a mining model, you need to understand the data you will work with and clearly define the business problem you are trying to solve. This includes analyzing the business requirements, defining the scope of the problem, defining the metrics by which the model will be evaluated, and defining the final objective for the data mining project. These tasks translate into questions like:

What is your organization is looking for?

Which attribute of the dataset do you want to try to predict?

What types of relationships are you trying to find?

Do you want to make predictions from the data mining model or just look for interesting patterns and associations?

How is the data distributed?

How are the columns related, or if there are multiple tables, how are the tables related?

These are the questions that you need to be able to answer before you can begin to work with the data. To find the answers, you may need to conduct a data availability study, investigating the needs of the business users with respect to the data available. If the data won't support what the users need to find out, you may need to redefine the project.

Step 2—Data Preparation

You've defined the problem that you are going to try to solve—now what? Well, first you need to find the inputs related to this business problem. Collecting the data can be a cumbersome task. This data is usually scattered across a

company and stored in different formats. But do not narrow your focus! Find all data that is related to the business problem.

Often, the original data is collected through an OLTP – On Line Transactional Process which is a normalized database model - system and contains inconsistencies. Entries are missing or flawed; forexample, the data might show that a customer bought a product before she was born or shops regularly at a store 2,000 miles from her home. Before you begin to build the models, you need to fix these problems. In other words, you must "clean" the data. The problem is that cleaning the data is not a straightforward process. Maybe the person shopping 2,000 miles from her home has two residences and lives an equal amount of time at both. Usually, you are working with a very large dataset and can't look through every transaction personally. Therefore, you need to use some form of automation to explore the data and find the inconsistencies. Exploration techniques can include calculating the minimum and maximum values, calculating the mean and standard deviations, and looking at the distribution of the data. In the end, you need to decide which data seems flawed and devise a strategy for fixing the problem.

In preparing the data, you often have to transform columns of the dataset before building a mining model. For example, to determine whether your company's compensation strategy is equitable, you may try to predict salaries based on age, experience, length of time with the company, and other factors. The data you use to create your model contains a large number of possible values for the salary of an employee—in essence; it is a continuous attribute, a column with a large number of states. To make your final model more focused, you need to discretize the data. This simply means creating a limited number of buckets (salary ranges) such as low, medium, and high, and replacing the values in the column with the appropriate bucket name. You may also want to define a new column based on existing columns. For example, you may not have a column that details the total cost of retaining an employee, including such things as health insurance and other perks, but you could easily make one by adding up each cost and displaying it in a new column.

Step 3—Model Building

The most important concept in data mining knows your data. If you don't understand the structure of your dataset, how can you know what to ask, or which columns to include in your data mining model? Imagine that you are at an important business meeting but did not prepare. If you ask questions during the meeting, they will probably not make sense, reducing your effectiveness. The

same holds true for data mining. If you build models without knowing your data, you will ask the wrong questions, reducing the model's effectiveness.

Before building the model, you need to randomly separate the original dataset into separate training (model-building) and testing (validation) datasets. You use the training data to build the model. Then you test the accuracy of the model by creating prediction queries against the testing dataset. Because you know the outcome of the predictions (the data comes from the same set used to train the model), you can calculate the accuracy of the model's performance.

Sometimes the attribute that you are trying to predict has a very high distribution of one state, and a very low distribution of another state. For example, in our dataset, the number of positive responses in the predictable column is about 5 percent, while the number of negative responses is about 95 percent. There is a chance that there are not enough occurrences of the positive response to generate the strong relationships that will allow us to create predictions. One way to solve this problem is to over-sample the data, which means that we artificially boost the number of positive responses but randomly remove a number of the records that correspond to negative responses.

After you explore the data and select columns to include in the model, you can build your models using the training dataset. This process happens exactly the way it sounds—you pass the data through the algorithm to train the model. Each algorithm also contains adjustable parameters that can affect the outcome of the model. The result of the training process is a mathematical model you can either use to explore the data (as in the case of a clustering algorithm) or to create predictions (as in the case of a decision tree algorithm). How well you choose the columns to include in the model and how you alter parameters of the model ultimately determine the performance of the models. With that said, here are the steps for building the model:

Select columns.

Select a model.

Adjust parameters.

Train the model.

Step 4—Model Validation

After you build a model, you need to know how well it performs. You do not want to move the model into a production environment until you know how well it predicts. Often, you build several models and then compare how they perform

against each other. This is where you use the testing dataset that you previously set aside.

Step 5—Deployment of the Model into Production

This is where all of your hard work begins to show results. After you build the models and measure their effectiveness, you can deploy them in a production environment, the place where the models will be used in the business decision-making process. Updating the model is part of the deployment strategy. As more data comes into the organization, you need to develop a process for rebuilding the models, thus improving their effectiveness.

Step 6—Meta Data Management

The information that is associated with exploring the data and building the models is useful for you to save. This includes columns that were removed, models that were previously built, and the effectiveness of those models. Managing this data can become a project in itself, but it is a very important step. Typically, you store this information in a database, where it is available through queries, like any other data.

References

- www.data-mining-guide.net
- Microsoft TechNet Library
- Data Mining: Foundations and Practice- Tsau Young Lin, Ying Xie, Anita Wasilewska and Churn-Jung Liau (Eds.)
- MSDN Microsoft Developer Network
- SQL Server Books online
- SQL Server 2000 Resource Kit and Appendix
- THE DIMENSIONAL FACT MODEL: A CONCEPTUAL MODEL FOR DATA WAREHOUSES1(MATTEO GOLFARELLI, DARIO MAIO and STEFANO RIZZI

  DEIS - Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy)
- Data Mining: Algorithms and Applications Matrix Math Review
- SQL Server 2005 Resource Kit and Appendix
- Association forum of chicagoland video from YouTube