ISSN:0975-9646

RaajHans: A Data Mining Tool using Soft Computing Techniques

Avinash R. Pinglae[1] , Aparna A. Junnarkar[2]

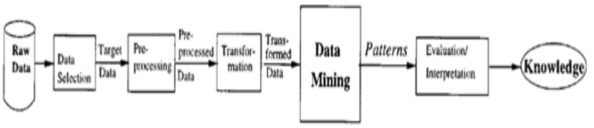
[1]Department of Computer Engineering,
University of Pune
[2]P.E.S's Modern College of Engineering,
Shivaji Nagar, Pune-5
Maharashtra, Inida

Abstract- **In today's era digital information is changing its nature and behavior on daily basis. Digital data is becoming more bulky and complex as well. Managing such data is also a challenging task to researchers. Hence they come up with data mining tools which help to manage data that is mine bulk data easily and effectively. Soft computing on other hand is a great technique to computerize the decision making system using Neural Networks and Genetic Algorithms. This information encouraged a concept of RaajHans. RaajHans will probably integrate Soft computing techniques to provide best results automatically avoiding Human manipulation. The tool will take bulk data as an input and will mine data and generate patterns using soft computing techniques in background. It will also be facilitated with a Database converter tool which will help user to input any type of data to it.**

Key terms: Data Mining, Soft Computing, Neural Network, Genetic Algorithms

1. INTRODUCTION TO DATA MINING

Data mining is a form of knowledge discovery essential for solving problems in a specific domain. Individual data sets may be gathered and studied collectively for purposes other than those for which they were originally created. New knowledge may be obtained in the process while eliminating one of the largest costs, viz., data collection [8]. Medical data, for example, often exists in vast quantities in an unstructured format. The application of data mining can facilitate systematic analysis in such cases. Medical data, however, requires a large amount of preprocessing in order to be useful. Here numeric and textual information may be interspersed, different sym- bols can be used with the same meaning, redundancy of- ten exists in data, erroneous/ misspelled medical terms are common, and the data is frequently rather sparse. A robust preprocessing system is required in order to extract any kind of knowledge from even medium-sized medical data sets[8]. The data must not only be cleaned of errors and redundancy, but organized in a fashion that makes sense to the problem.



Knowledge Discovery and Data Mining

1.1. KNOWLEDGE DISCOVERY AND DATA MINING

Data Mining [1],[2], also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. The following figure (Figure 1.1) shows data mining as a step in an iterative knowledge discovery process. In its simplest form, data mining automates the detection of relevant patterns in a database, using defined approaches and algorithms to look into current and historical data that can then be analyzed to predict future trends. Because data mining tools predict future trends and behaviors by reading through databases for hidden patterns, they allow organizations to make proactive, knowledge-driven decisions and answer questions that were previously too time-consuming to resolve[2]. Organizations that wish to use data mining tools can purchase mining programs designed for existing software and hardware platforms, which can be integrated into new products and systems as they are brought online, or they can build their own custom mining solution. For instance, feeding the output of a data mining exercise into another computer system, such as a neural network, is quite common and can give the mined data more value. This is because the data mining tool gathers the data, while the second program (e.g., the neural network) makes decisions based on the data collected. The overall KDD process is outlined in Fig. 1.1. It is interactive and iterative involving, more or less, the following steps [7].

1) Understanding the application domain: includes relevant prior knowledge and goals of the application.

2) Extracting the target data set: includes selecting a dataset or focusing on a subset of variables.
3) Data cleaning and pre-processing: includes basic operations, such as noise removal and handling of missing data. Data from real-world sources are often erroneous, incomplete, and inconsistent, perhaps due to operation error or system implementation flaws. Such low quality data needs to be cleaned prior to data mining.
4) Data integration: includes integrating multiple, heterogeneous data sources.
5) Data reduction and projection: includes finding useful features to represent the data (depending on the goal of the task) and using dimensionality reduction or transformation methods.
6) Choosing the function of data mining: includes deciding the purpose of the model derived by the data mining algorithm (e.g., summarization, classification, regression, clustering, web mining, image retrieval, discovering association rules and functional dependencies, rule extraction, or a combination of these).
7) Choosing the data mining algorithm(s): includes selecting method(s) to be used for searching patterns in data, such as deciding on which model and parameters may be appropriate.
8) Data mining: includes searching for patterns of interesting a particular representational form or a set of such representations.
9) Interpretation: includes interpreting the discovered patterns, as well as the possible visualization of the extracted patterns. One can analyse the patterns automatically or semi automatically to identify the truly interesting/ useful patterns for the user.
10) Using discovered knowledge: includes incorporating this knowledge into the performance system, taking actions based on knowledge.

*1.2.*DATA MINING ALGORITHMS AND TECHNIQUES

Various algorithms and techniques like Classification, Clustering, Regression, Data Pre-processing, Prediction and Association are used for knowledge discovery from Database.

1.2.1. Classification:

Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification. In Learning the training data are analyzed by classification algorithm. In classification test data are used to estimate the accuracy of the classification rules. If the accuracy is acceptable the rules can be applied to the new data tuples. For a fraud detection application, this would include complete records of both fraudulent and valid activities determined on a record-by-record basis. The classifier-training algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. The algorithm then encodes these parameters into a model called a classifier.

Types of classification models:
- Classification by decision tree induction
- Bayesian Classification
- Neural Networks
- Support Vector Machines (SVM)
- Classification Based on Associations

1.2.2. Clustering:

Clustering can be said as identification of similar classes of objects. By using clustering techniques we can further identify dense and sparse regions in object space and can discover overall distribution pattern and correlations among data attributes. Classification approach can also be used for effective means of distinguishing groups or classes of object but it becomes costly so clustering can be used as pre-processing approach for attribute subset selection and classification. For example, to form group of customers based on purchasing patterns, to categories genes with similar functionality.

Types of clustering methods:
- Partitioning Methods
- Hierarchical Agglomerative (divisive) methods
- Density based methods
- Grid-based methods
- Model-based methods

1.2.3. Prediction:

Regression technique can be adapted for predication. Regression analysis can be used to model the relationship between one or more independent variables and dependent variables. In data mining independent variables are attributes already known and response variables are what we want to predict. Unfortunately, many real- world problems are not simply prediction. For instance, sales volumes, stock prices, and product failure rates are all very difficult to predict because they may depend on complex interactions of multiple predictor variables. Therefore, more complex techniques (e.g., logistic regression, decision trees, or neural nets) may be necessary to forecast future values. The same model types can often be used for both regression and classification. For example, the CART [8] (Classification and Regression Trees) decision tree algorithm can be used to build both classification trees (to classify categorical response variables) and regression trees (to forecast continuous response variables). Neural networks too can create both classification and regression models.

1.2.4. Association rule:

Association and correlation is usually to find frequent item set findings among large data sets. This type of finding helps businesses to make certain decisions, such as catalogue design, cross marketing and customer shop- ping behavior analysis. Association Rule algorithms need to be able to generate rules with confidence values less than one. However the number of possible Association Rules for a given dataset is generally very large and a high proportion of the rules are usually of little (if any) value.

Types of association rule:

- Multilevel association rule
- Multidimensional association rule
- Quantitative association rule

1.2.5. Regression:

Regression is a data mining function that predicts a number. Profit, sales, mortgage rates, house values, square footage, temperature, or distance could all be predicted using regression techniques. For example, a regression model could be used to predict the value of a house based on location, number of rooms, lot size, and other factors. A regression task begins with a data set in which the tar- get values are known. For example, a regression model that predicts house values could be developed based on observed data for many houses over a period of time. In addition to the value, the data might track the age of the house, square footage, number of rooms, taxes, school district, proximity to shopping centers, and so on. House value would be the target, the other attributes would be the predictors, and the data for each house would constitute a case. In the model build (training) process, a regression algorithm estimates the value of the tar- get as a function of the predictors for each case in the build data. These relationships between predictors and target are summarized in a model, which can then be applied to a different data set in which the target values are unknown. Regression models are tested by computing various statistics that measure the difference between the predicted values and the expected values. The historical data for a regression project is typically di- vided into two data sets: one for building the model, the other for testing the model. See "Testing a Regression Model". Regression modeling has many applications in trend analysis, business planning, marketing, financial forecasting, time series prediction, biomedical and drug response modeling, and environmental modeling.

Types of Regression Methods:

- Linear Regression
- Multivariate Linear Regression
- Nonlinear Regression
- Multivariate Nonlinear Regression

1.2.6. Data Pre-processing and visualization:

Data pre-processing is the process of manipulating data prior to actual mining steps. It aims at improving the quality or the performance of data mining algorithms. Data pre-processing can be divided into several categories: Data cleaning, integration, transformation, reduction and compression. I will introduce several techniques which aid this process under these different goals. Data visualization on the other hand provides methods to efficiently display data to the human mind. Visualization is mainly used in the context of data exploration, which is the combined human and computer analysis of data. I will introduce plotting techniques for different kinds of data, like single- and multidimensional data.

2. Architecture

The architecture of RaajHans is desktop based. It can be installed and run on desktops having minimum requirements of P4 system. As the tool is going to implement using java platform, it will be compatible with any kind of systems environment. In short we can say that the RaajHans would be a platform independent data mining tool. This feature of RaajHans can make it more popular than any other data mining tool.

After installation of the RaajHans, it can be used to input large amount of data in form of SQL queries or Oracle database. The database converter tool will help RaajHans to convert the database inputted in form of SQL queries or oracle database into the .arff file i.e. at- tribute reference file.

Mathematical Model:

Let S be the system which will perform data mining operations β on input set X. We can define system S as

S={s, e, X, Y, F_{me}, F_{friend}, DD, NDD}

Where,

s=Distinct start of system

e=Distinct end of system

X=set of input

Y=set of output

F_{me}=Central Function. It defines a important part of program.

Φ=Constant vectors

All vectors will be column vectors.

As the tool is going to perform database operations on given data set

Let X is a data set on which data mining operations are to be performed.

Hence

X=$A_{m\times n}$ {m=No. of Rows and n=No. of Columns}

f(x) will denote set of minimizers of x on set X where x∈X

now,

f(x)=β{ $A_{m\times n}$}

Here β defines set of data mining operations to be performed on data set A.

$R_{m\times n}$ = β{ $A_{m\times n}$}

$$g(x) = \begin{cases} 0 \; if \; x \in \beta \\ 1 \; if \; x \notin \beta \end{cases}$$

Here g(x) represents a finite set of operations carried out on data element x.

Now we can write probability of carrying out a data mining operation of data set X.

$P(\beta\{A_{m\times n}\}) = \{P(g(x))$

But as we know g(x) is non-deterministic

Hence we can write

$$\min_{x,y}\{\frac{e^A x}{m} + \frac{e^A y}{n}\}$$

Where y is output element and $y \in Y$

Hence we can write

$$R_{m\times n} = \min_{x,y}\{\frac{e^A x}{m} + \frac{e^A y}{n}\} \qquad \text{……..(A)}$$

From equation (A) we can write system S as

$$S = \{X\,|X = (1,2,3 - -n), Y|Y = R_{m\times n,}\ \frac{e^A x}{m} + \frac{e^A y}{n}\}$$

2.1. DATA:

2.1.1. Attribute Reference file Format:

A file format specifies the organization of information, at some level of abstraction, contained in one or more byte streams that can be exchanged between systems. An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the Header information, which is followed the Data information. The Header of the ARFF file con- tains the name of the relation, a list of the attributes (the columns in the data), and their types. The Data of the ARFF file looks like the following:

```
@DATA
5.1,3.5,1.4,0.2,heart-disease
4.9,3.0,1.4,0.2,heart-diease
```

Lines that begin with a % are comments. The @RELATION, @ATTRIBUTE and @DATA declarations are case insensitive.

The ARFF Header Section:

The ARFF Header section of the file contains relation declaration and attributes declarations.

The @relation Declaration:

The relation name is defined as the first line in the ARFF file. The format is: @relation hrelation − nameiwherehrelation− nameiisastring.

The string must be quoted if the name includes spaces.

The @attribute Declarations:

Attribute declarations take the form of an ordered sequence of @attribute statements. Each attribute in the data set has its own @attribute statement which uniquely defines the name of that attribute and its data type. The order the attributes are declared indicates the column position in the data section of the file. For example, if an attribute is the third one declared then RaajHans expects that all that attributes values will be found in the third comma delimited column. The format for the @attribute statement is:

@attribute<attribute-name><datatype>where the <attribute-name>must start with an alphabetic character. If spaces are to be included in the name then the entire name must be quoted. The keywords numeric, string and date are case insensitive.

Numerical Attributes

Numeric attributes can be real or integer numbers.

Nominal attributes

Nominal values are defined by providing an <nominal-specification >listing the possible values:

{<nominalname1>, <nominal-name2 >, <nominal-name3 >,}

For example, the class value of the Iris dataset can be defined as follows:

@ATTRIBUTE class {Heart-diease,Iris-versicolor,Iris-virginica}

Values that contain spaces must be quoted.

String attributes

String attributes allow us to create attributes containing arbitrary textual values. This is very useful in text-mining applications, as we can create datasets with string attributes, then write RaajHans Filters to manipulate strings (like StringToWordVectorFilter). String attributes are declared as follows:

@ATTRIBUTE LCC string

Date attributes

Date attribute declarations take the form:

@attribute<name >date [<date-format >]

where ¡name¿ is the name for the attribute and -format is an optional string specifying how date values should be parsed and printed. The default format string accepts the ISO-8601 combined date and time format: ”yyyy-MM-dd’T’HH:mm:ss”.

Dates must be specified in the data section as the corresponding string representations of the date/time

ARFF Data Section

The ARFF Data section of the file contains the data declaration line and the actual instance lines.

The @data Declaration The @data declaration is a single line denoting the start of the data segment in the file. The format is:

@data

The instance data

Each instance is represented on a single line, with carriage returns denoting the end of the instance.

Attribute values for each instance are delimited by commas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute). Missing values are represented by a single question mark, as in:

```
@data
4.4,?,1.5,?,Heart-diease
```

Values of string and nominal attributes are case sensitive, and any that contain space must be quoted, as follows:

```
@relation LCCvsLCSH
@attribute LCC string
@attribute LCSH string
@data
```

AG5, 'Encyclopedias and dictionaries.;Twentieth century.'
AS262, 'Science – Soviet Union – History.'

Dates must be specified in the data section using the string representation specified in the attribute declaration. For example:

```
@RELATION Timestamps
@ATTRIBUTE timestamp DATE ”yyyy-MM-dd HH:mm:ss”
@DATA”2001-04-03 12:12:12”
```

Sparse ARFF files

Sparse ARFF files are very similar to ARFF files, but data with value 0 are not be explicitly represented. Sparse
ARFF files have the same header (i.e @relation and @attribute tags) but the data section is different. Instead of representing each value in order, like this:

```
@data
0, X, 0, Y, ”class A”
0, 0, W, 0, ”class B”
```

the non-zero attributes are explicitly identified by attribute number and their value stated, like this:

```
@data
1 X, 3 Y, 4 ”class A”
2 W, 4 ”class B”
```

Each instance is surrounded by curly braces, and the format for each entry is: index ><space><value >where index is the attribute index (starting from 0). Note that the omitted values in a sparse instance are 0; they are not ”missing” values! If a value is unknown, you must explicitly represent it with a question mark (?).

2.2. CODE:

RaajHans is a comprehensive data mining tool which will include basic data mining techniques. Hence the basic module structure of RaajHans would be including all these Data mining techniques. Along with that it will be having extra modules for GA and Fuzzy C-means algorithm. Let us discuss in brief.

2.2.1. Module A: Classification

Clicking on the classifier tab after loading a dataset into RaajHans and selecting the choose tab will bring up a menu with a number of choices for the classifier that is to be applied to the dataset. Note that you have 4 options on how to test the model you're building: Using the test set, a training set (you will need to specify the location of the training set in this case), cross validation and a percentage. The achieved accuracy of your model will vary, depending on the option you select. One pitfall to avoid is to select the training set as a test set, as that will result in an underestimate of the error rate. The resulting model, with a lot of additional information will be displayed after you click on start. What exactly is contained in the output can be determined under options. One of the things to watch out for is that the confusion matrix is displayed, as this gives a lot more information than just the prediction accuracy. Other useful things are the options showing up when right clicking the results list on the bottom right. For example, this is where you can load and save the models you built, as well as save the results page. Another fact to keep in mind is that RaajHans gives hints on how to achieve the same result from the command line: look at what is displayed next to the Choose button and how it changes with the options that you select. This information can also be found towards the top of your results page.

2.2.2. Module B: Clustering

The clustering option is very similar to the classification described above, with a few differences regarding the options you select. . Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification. In Learning the training data are analyzed by classification algorithm. In classification test data are used to estimate the accuracy of the classification rules. If the accuracy is acceptable the rules can be applied to the new data tuples. For a fraud detection application, this would include complete records of both fraudulent and valid activities determined on a record-by-record basis. The classifier-training algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. The algorithm then encodes these parameters into a model called a classifier.

2.2.3. Module C: Prediction

Regression technique can be adapted for predication. Regression analysis can be used to model the relationship between one or more independent variables and dependent variables. In data mining independent variables are attributes already known and response variables are what we want to predict. Unfortunately, many real- world problems are not simply prediction. For instance, sales volumes, stock prices, and product failure rates are all very difficult to predict because they may depend on complex interactions of multiple predictor variables. Therefore, more complex techniques (e.g., logistic regression, decision trees, or neural nets) may be necessary to forecast future values. The same model types can of- ten be used for both regression and classification. For example, the CART [8] (Classification and Regression Trees) decision tree algorithm can be used to build both classification trees (to classify categorical response variables) and regression trees (to forecast continuous response variables). Neural networks too can create both classification and regression models.

2.2.4. Module D: Association rule

RaajHans will also provide three algorithms to extract association rules from non-numerical data as shown in the picture below.

- Apriori Algorithm
- FP-Growth Algorithm.
- Fuzzy C-means

2.2.5. Module E: Experimenter

The experimenter, which can be run from GUI, is a tool that allows you to perform more than one experiment at a time, maybe applying different techniques to a datasets, or the same technique repeatedly with differ- ent parameters. After selecting new, which initializes a new experiment with default parameters, you can select where you want to store the results of your experiment by using browse (there are a number of choices avail- able for the format of your results file). You can then change the default parameters if desired (watch out for the option of selecting classification or regression). For example, you can add more datasets, delete the ones you already selected as well as add and delete algorithms applied to your selected datasets. You can also the type of experiment (cross validation or a percentage split for the training and test set). After running your experiment by selecting Start from the Run tab, your results will be stored in the specified Results file if the run was successful. You then need to load this file into RaajHans from the Analysis pane to see your results.

2.2.6. Module F: Knowledge Flow

The knowledge flow is an alternative interface to the functionality provided by the RaajHans data mining package. RaaJHans components can be selected from a tool bar, positioned a layout canvas, and connected into a directed graph to model a complete system that Processes and analyzes data.

Data Sources:

- used to indicate where data is coming from
- supports various file types and sources
- Configurable for

- file name of data source.

-Data set or instance (incremental) loading.

2.2.7. Module G: Visualization

Used to visually display outputs

- supports performance and summaries

-comparable to options from Explorer interface

- Data Visualizer - component that can pop up a panel for visualizing data in a single large 2D scatter plot.
- Scatter Plot Matrix - component that can pop up a panel containing a matrix of small scatter plots (clicking on a small plot pops up a large scatter plot).
- Attribute Summarizer - component that can pop up a panel containing a matrix of histogram plots - one for each of the attributes in the input data.
- Model Performance Chart - component that can pop up a panel for visualizing threshold curves.
- Text Viewer - component for showing textual data. Can show data sets, classification performance statistics
- Graph Viewer - component that can pop up a panel for visualizing tree based models.
- Strip Chart - component that can pop up a panel that displays a scrolling plot of data (used for viewing the online performance of incremental classifiers).
- Example1: Heart Disease Diagnosis

1. Specify a Data Source
2. Specify which attribute is the class
3. Specify cross validation
4. Specify evaluation
5. Specify evaluation output
6. To allow viewing of decision trees per fold
7. Run experiments.

3. Operation

3.1. Role of Soft Computing:

Soft computing methodologies (involving fuzzy sets, neural networks, genetic algorithms, and rough sets) are most widely applied in the data mining step of the overall KDD process. Fuzzy sets provide a natural framework for the process in dealing with uncertainty. Neural networks and rough sets are widely used for classification and rule generation. Genetic algorithms (GAs) are involved in various optimization and search processes, like query optimization and template selection. Other approaches like case based reasoning [5] and decision trees [3] are also widely used to solve data mining problems.

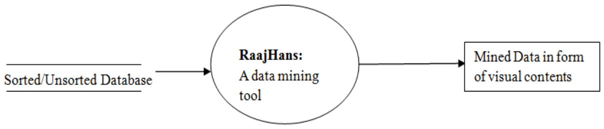
Recently various soft computing methodologies have been applied to handle the different challenges posed by data mining. The main constituents of soft computing, at this juncture, include fuzzy logic, neural networks, genetic algorithms, and rough sets. Each of them contributes a distinct methodology for addressing problems in its domain. This is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human-interpretable, low cost, approximate solution, as compared to traditional techniques. Let us first describe the roles and significance of the individual soft computing tools and their hybridizations, followed by the various systems developed for handling the different functional aspects of data mining. A suitable preference criterion is often optimized during mining. It may be mentioned that there is no universally best data mining method; choosing particular soft computing tool(s) or some combination with traditional methods is entirely dependent on the particular application and requires human interaction to decide on the suitability of an approach.

3.1.1. Fuzzy Sets:

The modeling of imprecise and qualitative knowledge, as well as the transmission and handling of un- certainty at various stages are possible through the use of fuzzy sets. Fuzzy logic is capable of supporting, to a reasonable extent, human type reasoning in natural form. It is the earliest and most widely reported constituent of soft computing. The development of fuzzy logic has led to the emergence of soft computing. In this section we provide a glimpse of the available literature pertaining to the use of fuzzy sets in data mining.

The notion of interestingness, which encompasses several features such as validity, novelty, usefulness, and simplicity, can be quantified through fuzzy sets. Fuzzy dissimilarity of a discovered pattern with a user-defined vocabulary has been used as a mea- sure of this interestingness. There is a growing in- disputable role of fuzzy set technology in the realm of data mining [7]. Various data browsers have been implemented using fuzzy set theory [6]. Analysis of real-world data in data mining often necessitates simultaneous dealing with different types of variables, viz., categorical/symbolic data and numerical data. Nauck [5] has developed a learning algorithm that creates mixed fuzzy rules involving both categorical and numeric attributes. Pedrycz [5] discusses some constructive and fuzzy set-driven computational vehicles of knowledge discovery, and establishes the relationship between data mining and fuzzy modeling. The role of fuzzy sets is categorized below based on the different functions of data mining that are modeled.

3.1.2. Genetic Algorithms: GA is adaptive, robust, efficient, and global search method, suitable in situations where the search space is large. They optimize a fitness function, corresponding to the preference criterion of data mining, to arrive at an optimal solution using certain genetic operators. Knowledge discovery systems have been developed using genetic programming concepts. The MASSON system [4], where intentional information is extracted for a given set of objects, is popular. The problem addressed is to find common characteristics of a set of objects in an



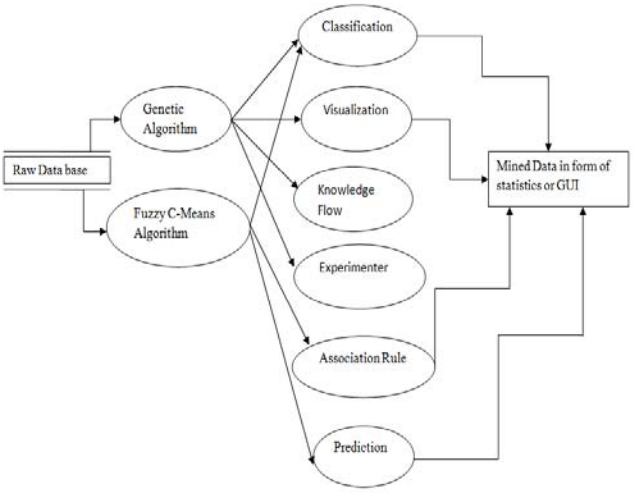
Zero Level DFD of RaajHans

The Data Flow Diagram shown above is the Zero Level DFD of RaajHans. The figure concludes that the tool is to be considered as a zero bubble and Database is the raw data which can be initially sorted or in unsorted form. Data also can be in form of SQL database or Oracle database. RaajHans will be having a converter tool integrated in it which will help to convert the SQL database into Attribute Reference file format. After converting it to .arff file which is attribute reference file, the actual data mining task can be perform on the file. The result of data mining task can be in form of statistics or us- ing some Visualization techniques. LetâĂŹs take deeper look to the system using One-Level DFD. Object oriented database. Genetic programming is used to automatically generate, evaluate, and select object-oriented queries. GA is also used for several other purposes like fusion of multiple data types in multimedia databases, and automated program generation for mining multimedia data. However, the literature in the domain of GA-based data mining is not as rich as that of fuzzy sets [6]. We pro- vide below a categorization of few such interesting systems based on the functions modeled.

4. Development

The development of Tool will include the designing of modules depicted in chapter 3. Development of GUI of the RaajHans can be done using a programming language Java as it is platform independent. Algorithms also have to be done using the Java and SQL server. To understand the exact flow of the system, let us discuss the architectural diagrams of the RaajHans.

4.1. One-Level DFD:



Above figure shows the schematic representation of RaajHans. The One-Level DFD shows the exact flow of data from a raw data file through various data mining operations. Here we can see that Association rule classification and other techniques are producing output in form of statistics and the statistics are then can be converted to graphical representation.

5. Summary And Conclusion

Current research focuses on emerging technique of soft computing in the domain of Data Mining. Effective utilization of Soft Computing techniques for mining bulk amount of data can make Data mining task powerful. As there are several tools available for data mining in market along with various techniques, one can effectively recognize a good result using such tools. RaajHans on other side is a tool which emphasizes the performance of data mining task using soft computing techniques. Genetic algorithms, Fuzzy sets and Neural Network embedded in the tool will help improve the data mining task. The RaajHans will guide the user to choose a most suit- able pattern among generated ones. Neural network will help to preserve the knowledge gained by past data mining tasks. Neural network will also help the tool to improve its performance after day by day use. Genetic Algorithms are most recommended to generate patterns from the data. RaajHans will also try to overcome the problems associated with RaajHans like specific file format as in- put [2], visualization limitations, and inconsistency of generated patterns.

REFERENCES

[1] Weka User Manual

[2] Ian H. Witten, EiebFrank,Len Trigg Practical Machine Learning tools and Techniques with Java implementation, Mark Hall, Aug 1999

[3] KanhaiyaLal, N.C.Mahanti Role of soft computing as a tool in data miningâĂŹ. Depart- ment of Computer Sc. Engg., Birla Institute of Technology Patna, Bihar, India, Department of Applied Mathematics, Birla Institute of Tech- nologyMesra, Ranchi,India.

[4] Sankar K. Pal,Soft data mining, computa- tional theory of perceptions, and rough-fuzzy approach, Machine Intelligence Unit, Indian Statistical Institute, Kolkata,India.17 March 2003.

[5] R.S.Michalski, M.Kubat, I. Bratko, âĂŸMa- chine Learning and Data Mining: Methods and ApplicationsâĂŹ, 1998.

[6] Akira Imada, âĂŸAn Introduction to Soft Com- putingâĂŹ, December 17, 2003

[7] SushmitaMitra, Sankar K. Pal, and PabitraMi- tra, âĂŸData Mining in Soft Computing Frame- work: A SurveyâĂŹIEEE transactions on neural networks, vol. 13, no. 1, january 2002.

[8] Mrs. Bharati M. Ramageri, âĂŸData mining techniques and applicationsâĂŹ Indian Journal of Computer Science and Engineering, Vol. 1 No. 4 301-305