

DiagData: A Tool for Generation of Fuzzy Inference System

Silvia Maria Fonseca Silveira Massruhá, Raphael Fuini Riccieti, Helano Povoas Lima and Carlos Alberto Alves Meira

Laboratory of Computational Intelligence, Embrapa Agricultural Informatics, São Paulo State 13083-886, Brazil

Received: May 6, 2011 / Accepted: September 15, 2011 / Published: March 20, 2012.

Abstract: In this paper, it described the architecture of a tool called DiagData. This tool aims to use a large amount of data and information in the field of plant disease diagnostic to generate a disease predictive system. In this approach, techniques of data mining are used to extract knowledge from existing data. The data is extracted in the form of rules that are used in the development of a predictive intelligent system. Currently, the specification of these rules is built by an expert or data mining. When data mining on a large database is used, the number of generated rules is very complex too. The main goal of this work is minimize the rule generation time. The proposed tool, called DiagData, extracts knowledge automatically or semi-automatically from a database and uses it to build an intelligent system for disease prediction. In this work, the decision tree learning algorithm was used to generate the rules. A toolbox called Fuzzygen was used to generate a prediction system from rules generated by decision tree algorithm. The language used to implement this software was Java. The DiagData has been used in diseases prediction and diagnosis systems and in the validation of economic and environmental indicators in agricultural production systems. The validation process involved measurements and comparisons of the time spent to enter the rules by an expert with the time used to insert the same rules with the proposed tool. Thus, the tool was successfully validated, providing a reduction of time.

Key words: Prediction modelling, data mining, decision tree, machine learning, fuzzy inference system, fuzzygen.

1. Introduction

Nowadays, the phytopathologies have generated a large amount of data and information in the field of the plant disease diagnosis and control resulting from their experiments and publications. The challenge in this work is to use these data and information to extract knowledge so that it can predict the onset of a disease and prevent its spread. The traditional methods of data analysis usually perform queries using SQL (structured query language), OLAP (on-line analytical processing) tools and data visualization tools. However, the tools often fail to answer more complex questions involving all possible relationships and associations that exist in a large amount of data.

Therefore, it is necessary to use techniques of data

analysis supported by computer, allowing the self-extracting (or semi-automatic) of new knowledge from a large data repository. This field of research is called KDD (knowledge discovery from database) and DM (data mining). This process of extracting knowledge of the database is aimed at finding knowledge from one set of data to be used in decision-making. This area of research is multidisciplinary and incorporates techniques used in various areas such as database, artificial intelligence and statistics. The techniques used in MD should not be viewed as a substitute for other forms of data analysis, but rather as complementary tools to improve the results of the explorations made [1].

Embrapa has generated a large amount of data about plant diseases. For instance, there are many books, papers and reports published for diagnosis and control of plant diseases. These texts were written by experts in phytopathology to be used by farmers or extension

Corresponding author: Silvia Maria Fonseca Silveira Massruhá, scientific researcher, Ph.D., main research fields: artificial intelligence, fuzzy logic, machine-learning, expert systems, diagnostic systems. E-mail: silvia@cnpia.embrapa.br.

technicians for diagnosis of plant diseases. However, if the information is not available at hand farmers can use wrong dosages or chemical products to fight a disease and that may put in risk consumers' health and cause damages to the ecosystem. In this case, diagnostic expert systems can be an alternative tool to help the experts in decision-making concerning the identification of diseases and control methods [2].

The reliability of a diagnostic expert system depends on the quantity and quality of knowledge that it handles, i.e., the number of diseases it can diagnose and the appropriate knowledge representation constructed by the domain expert. This can be achieved by the knowledge engineer with a knowledge acquisition procedure. However, the knowledge acquisition process is the bottleneck in the development of any expert system [2].

In the diagnosis domain, the task performed by the expert can be thought of as a classification process, in which diseases are assigned to classes or categories determined by their properties. In a classification model, the connection between classes and properties can either be defined by something simple, such as a flowchart, or complex such as the executable models represented by computer programs. In the latter, the classification models can be built in two ways: (a) by interviewing the relevant experts of the domain; and (b) by constructing inductively, through the generalization from specific examples contained in numerous recorded classifications.

The first approach was adopted in the development of a preliminary version of an expert system for diagnosis of corn diseases on the web [3]. In that system, decision trees were generated from the interviews with domain experts and resources from the literature in the corn diseases area. After doing so, an expert system was built whose inference flows from the consequences (symptoms) to the causes (diseases) [2].

In this paper, it shows how the second approach (data mining techniques) can be used during the acquisition process. To do so, it developed a tool to

extract knowledge from structured data. This tool, called DiagData, aims to help the process of extraction of information from database by identifying groups of similar data in such a way that rules can be inducted and an inference system generated.

The paper is organized as follows. Section 2 describes some concepts of the data mining and uncertain reasoning in an integrated approach. Section 3 presents the DiagData architecture. Finally, Section 4 brings the results obtained so far as well as future work in our research project.

2. Methods and Data

The data mining can be summarized as the nontrivial extraction of the implicit, previously unknown, interesting, and potentially useful information (usually in the form of knowledge patterns or models) from data. The extracted knowledge is used to describe the hidden regularity of data, to make prediction, or to aid human users in other ways. The popularity of data mining is due to demands from various real-world applications in decision-making. An important aspect for scalable data mining is through efficient algorithms. The machine learning refers artificial intelligence tasks with improved performance and these techniques can be used in data mining tasks.

Machine learning algorithms [4] have proved to be of great practical value in a variety of application domains. There are especially useful in data mining problems where large databases may contain valuable implicit regularities that can be discovered automatically (e.g., to analyse outcomes of medical treatments from patient databases or to learn general rules for credit worthiness from financial databases).

A well-known tree induction algorithm adopted from machine learning is ID3 or C4.5, proposed by Quinlan [5, 6], which employs a process of constructing a decision tree in a top-down approach.

According to Chen [7], a decision tree is a hierarchical representation that can be used to determine the classification of an object by testing its

values for certain properties. In a decision tree, a leaf node denotes a decision (or classification) while a non-leaf node denotes a property used for decision (color, size, etc.). It is preferred the shortest path to reach leaf, because it implies the fewest possible number of questions are needed. The examples are used to guide the construction of a decision tree. The main algorithm is a recursive process. At each stage of this process, it selected a property based on the information gain calculated from the training examples. In addition, rule induction can be used in conjunction with tree induction. The rule induction can be served as a post processing of tree induction using ID3 or C4.5. In general, a rule can be constructed by following a particular path from root to a leaf in the decision tree, with the variables and their values involved in all the non-leaf nodes as the condition, and the classification variable as the consequence.

Decision trees have been widely used in data mining tasks. They also have been presented as a good tool to study the epidemiology of plant diseases such as described in Ref. [8].

The data mining tasks can be completed by uncertain reasoning techniques, such as fuzzy logic, bayesian networks, neural networks. Whereas probability theory is aimed at coping with randomness in reasoning, fuzzy

logic deals with a different kind of uncertainty, namely, vagueness. Fuzzy logic, first developed by Zadeh [9], provides an approximate but effective means of describing the behavior of systems that are too complex, ill-defined, or not easily analyzed mathematically. Fuzzy logic is an extension of the boolean logic for handling uncertain and imprecise knowledge. Fuzzy logic uses fuzzy set theory in which a variable is a member of one or more sets, with a specified degree of membership in a range, expressed mathematically as the interval [10]. Fuzzy variables are processed using a system called fuzzy inference system. It involves fuzzification, fuzzy inference and defuzzification. The fuzzification process converts a crisp input value to a fuzzy value. The fuzzy inference is responsible for drawing conclusions from the knowledge base. The defuzzification process converts the fuzzy control actions into a crisp control action. Then, fuzzy systems can provide crisp, exact control actions. A technique used in defuzzification is the centroid method.

In the proposed approach, it developed a tool called DiagData where the inputs are databases. Techniques of data mining such as decision tree are used to extract rules from databases. Then the rules are used to generate a fuzzy inference system in the web (Fig. 1).

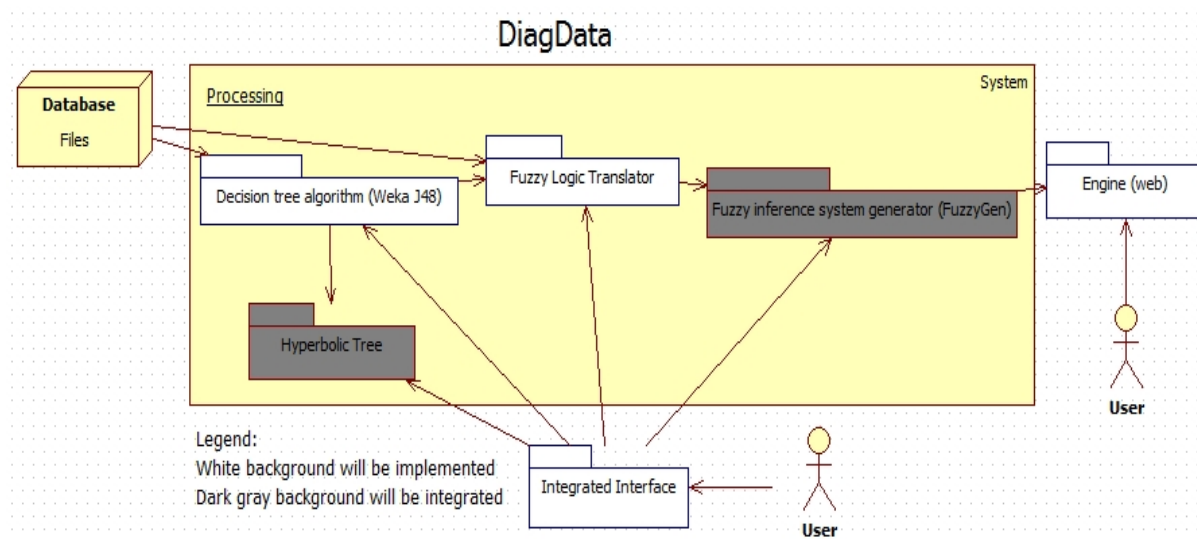


Fig. 1 The DiagData architecture.

The DiagData tool was developed in Java Standard Edition (JAVA SE). DiagData has 3 main modules: the decision tree builder, the fuzzy translator and the fuzzy inference system generator.

The inputs of this tool are two files, a training file and other testing file and the output is a fuzzy inference system. The training and testing files are inputs of the decision tree builder. In the implementation of this module, it was used the J48 algorithm of the software WEKA. The J48 implements the C4.5 algorithm proposed by Quinlan [6]. The rules generated by this module can be completed to generate the fuzzy system. The user can visualize the rules, confusion matrix, the training and testing files in the format .TXT. The rules can be visualized in the graphic format as hiperbolic tree. Afterwards, a FCL(fuzzy control language) file can be used for inference. The user can access the fuzzy variables and the results of the fuzzy inference engine. In the implementation of the fuzzy inference system generator used the FuzzyGen tool [10]. In the development of this tool, it used the API jFuzzyLogic [11] and in the Java SE platform for desktop. The

FuzzyGen generates a FIS (fuzzy inference system) in the FCL format (fuzzy control language) aiming the data interoperability [12].

3. Results

As Fig. 2 shows, the DiagData can be divided into four main stages: knowledge extracting, fuzzy modeling, inference engine and graphic results.

In the first phase, the training and testing files are inputs of this tool. The rules are generated from these files. It is not necessary the testing file, the rules can be generated from training file only.

The training and testing files in the .arff format are uploaded when the “Open” button is clicked in training and testing area, respectively. When the “rules extracting” button is clicked, the system runs the J48 using the two files, training and testing. This algorithm generates the rules and confusion matrix that are saved in the two files called “regras.txt” and “confusion_matrix.txt”, respectively. In this phase, the training and testing files are saved in the “train. txt” and “test.txt”. The user can visualize all files in the

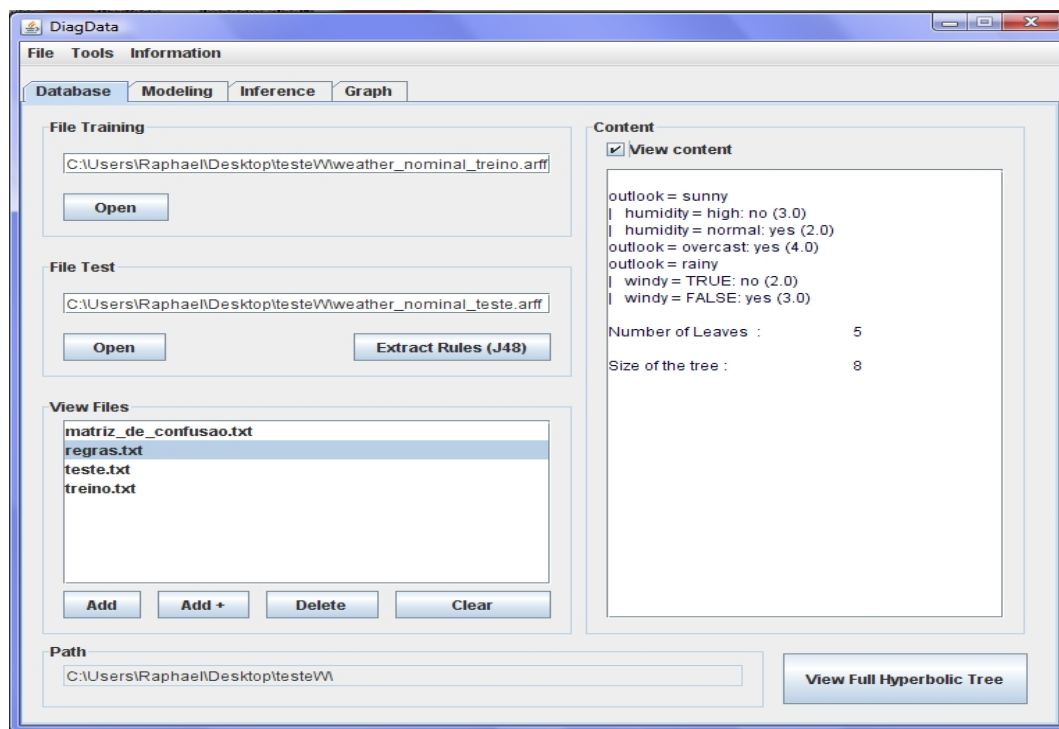


Fig. 2 The first form of DiagData.

format.TXT (Fig. 2). The rules can also be visualized in the graphic format as hiperbolic tree (Fig. 3). In the second phase, the fuzzy modelling is created from rules generated when pressing the button “Loading variables and rules”. The name of the model, the input and output variables and the rules are generated from the model as shown in Fig. 4.

In the “inference engine” phase, the system will try to infer the result. The user will have to choose a model generated in the previous step, by clicking the button “...”. This file is in the FCL format, as shown in Fig. 5. By clicking the “Upload” button, the user upload the entire contents of the file in the system and in the table “Upload the values of input variables ...”, will appear the input variables listed in the file that is uploaded.

The user must enter the values of variables in the “Value” of the table “Upload the values of input variables ...”, as shown in Fig. 6.

Finished entering the values for the variables and clicking on “Inferring results” will be presented in the form at the “Results ...”, as shown in Fig. 7, the contents of each variable, their fuzzy values, the rules and their weights. In this phase, the system infers the outcome based on the rules generated by the step to knowledge extracting. To assign values to which they want to input variables, using fuzzy logic system will infer a result. These results—the fuzzy ranges of the input and output variables and the generated results, can be viewed in Fig. 7, which presents in graphical format.

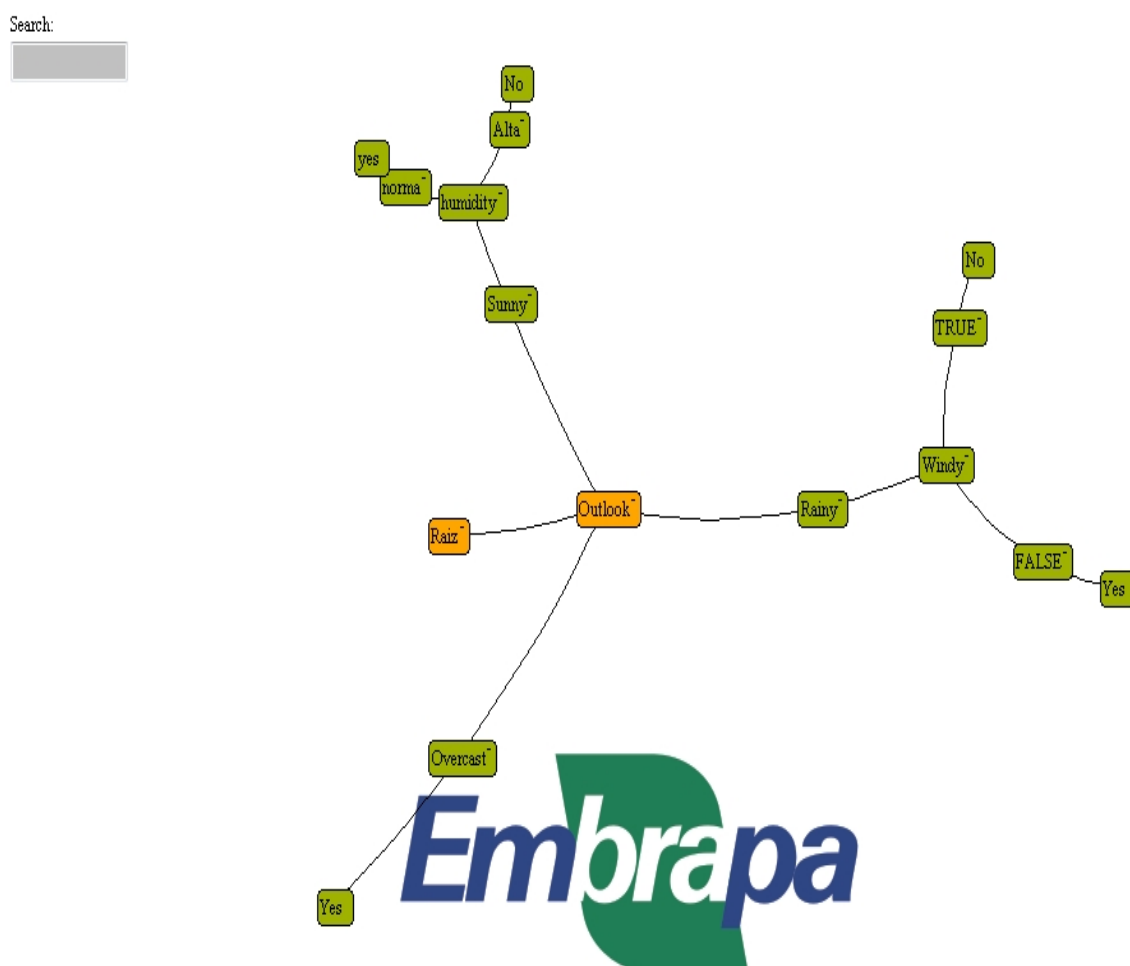


Fig. 3 The tree generated from rules.

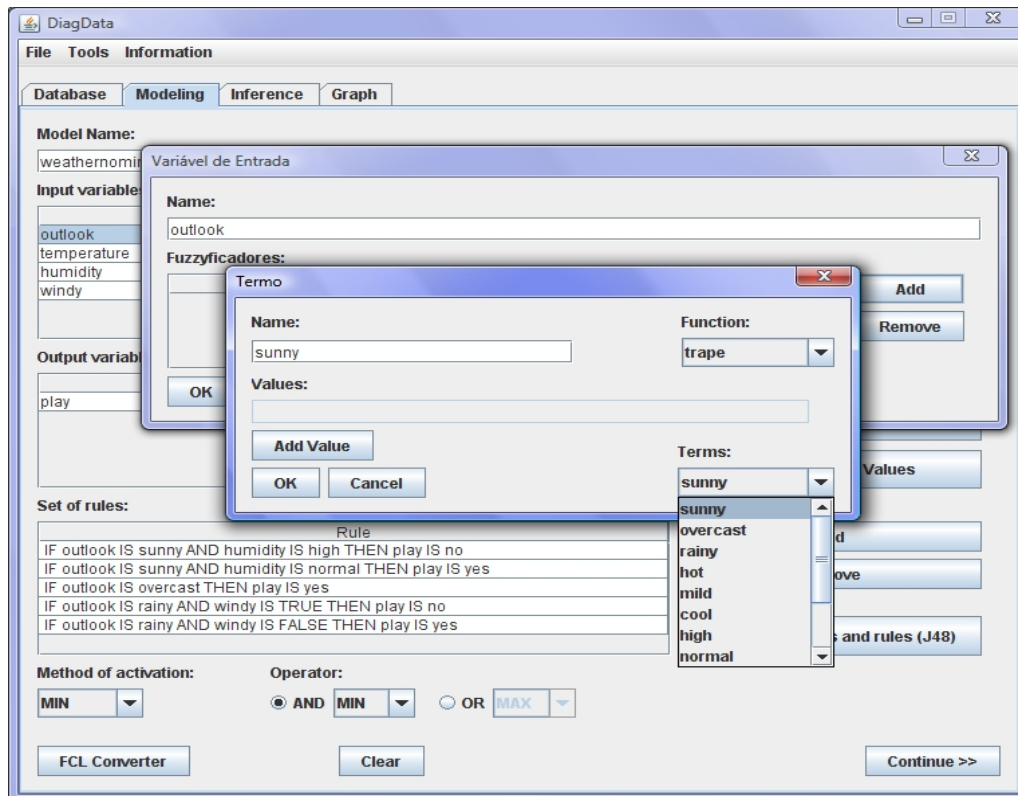


Fig. 4 Modelling phase.

```

FUNCTION_BLOCK jogar_tenis
-----
VAR_INPUT
    humidity : REAL;
    vento : REAL;
    outlook : REAL;
END_VAR

VAR_OUTPUT
    jogar : REAL;
END_VAR

FUZZIFY humidity
    TERM alta := trape 70 75 75 100;
    TERM normal := trape 0 0 70 75;
END_FUZZIFY

FUZZIFY vento
    TERM true := trian 0 1 0;
    TERM false := trian 0 0 0;
END_FUZZIFY

FUZZIFY outlook
    TERM sunny := trape 0 0 10 20;
    TERM overcast := trape 10 20 40 50;
    TERM rainy := trape 40 50 50 100;
END_FUZZIFY

DEFUZZIFY jogar
    TERM sim := trian 0 1 0;
    TERM nao := trian 0 0 0;
    ACCU : MAX;
    METHOD : COG;
END_DEFUZZIFY

RULEBLOCK No1
    ACT : MIN;
    AND : MIN;
    RULE 1 : IF outlook IS sunny AND humidity IS normal THEN jogar
IS sim;
    RULE 2 : IF outlook IS rainy AND vento IS true THEN jogar IS
sim;
    RULE 3 : IF outlook IS overcast THEN jogar IS sim;
    RULE 4 : IF outlook IS rainy AND vento IS false THEN jogar IS
sim;
    RULE 5 : IF outlook IS sunny AND humidity IS alta THEN jogar IS
nao;
END_RULEBLOCK

END_FUNCTION_BLOCK
    
```

Fig. 5 FCL file.

DiagData

File Tools Information

Database Modeling **Inference** Graph

Select the input file ...

C:\Users\Raphael\Desktop\testeVWeather_teste.fcl

Charge

Provide the values of input variables ...

Variable	value
windy	1
humidity	75
outlook	20

Result ...

play :

Defuzzifier : CenterOfGravity
 Latest defuzzified value: 0.5000000000000003
 Aggregation method: max
 Term: yes NaN Trapezoidal : 0.0 , 0.0 , 1.0 , 1.0
 Term: no NaN Trapezoidal : 0.0 , 0.0 , 0.0 , 0.0

windy :

Value: 1.0
 Aggregation method: sum
 Term: false 0.0 Trapezoidal : 0.0 , 0.0 , 0.0 , 0.0
 Term: true 0.0 Trapezoidal : 0.0 , 0.0 , 1.0 , 1.0

1 (0.0) if (outlook is rainy) and (windy is true) then play is no [weight: 1.0]
 2 (0.0) if (outlook is sunny) and (humidity is high) then play is no [weight: 1.0]
 3 (1.0) if outlook is overcast then play is yes [weight: 1.0]
 4 (0.0) if (outlook is rainy) and (windy is false) then play is yes [weight: 1.0]
 5 (0.0) if (outlook is sunny) and (humidity is normal) then play is yes [weight: 1.0]

Inferred result

Exit

Fig. 6 The inference engine form.

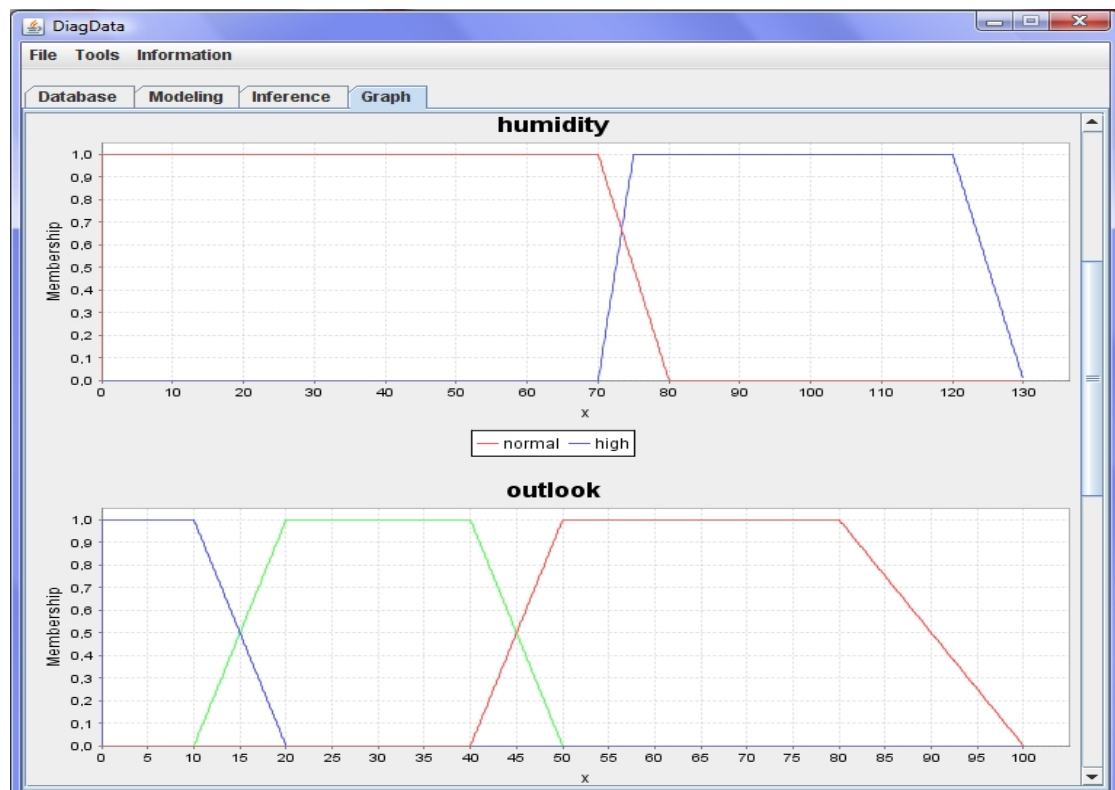


Fig. 7 The results showed in the graphic format.

4. Conclusion

This paper presented an evaluation of the use of the DiagData to extract information from structured information using data mining techniques. Tests carried out with corn diseases showed very good results comparing the decision tree constructed by the expert, which is based on grouping of symptoms, with the similarity among tree nodes calculated from the DiagData tool. However, the DiagData tool generates a large amount of rules from model. Then, it is necessary for the expert to select the main rules. This process of selection is semi-automatic because the expert has to validate these rules. After, the user has to enter the fuzzy values of the variables of selected rules. In real applications, there is very often no sharp boundary between variable ranges so that fuzzy variables is often better suited for the data. Membership degrees between zero and one are used in fuzzy variables instead of crisp assignments of the data. The DiagData allows user validate the selected rules because it generates automatically the fuzzy inference system. Thus, the user can refine the rules and generate the system again and quickly to correct it. Note that the DiagData is a tool to help the experts build the expert system but it doesn't eliminate them. The DiagData has also been used in the validation of economic and environmental indicators in agricultural production systems. In future work, it is intended to validate the DiagData with large databases to verify whether the results can be improved.

Although the initial validation is on agriculture, DiagData can be used in several domains, since it was developed to be independent of language and subject.

References

- [1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Academic Press, USA, 2006, pp. 5-10.
- [2] S.M.F.S. Massruhá, J.P. Dutra, S.A. Cruz, S. Sandri, J. Wainer, M.A.B. Morandi, An object-oriented framework for virtual diagnosis, in: Biennial Conference of European Federation of IT in Agriculture, Glasgow Caledonian University, Glasgow, 2007, p. 6.
- [3] S.M.F.S. Massruhá, E. Souza, L.A.S. Romani, S.A.B. Cruz, Virtual services for agricultural technology transfer, in: European Federation of IT in Agriculture, Food and The Environment-Efita 99, Bonn, Germany, 1999, pp. 53-62.
- [4] T. Mitchell, Machine Learning, MacGraw Hill, USA, 1997, p. 413.
- [5] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81-106.
- [6] J.R. Quinlan, C4.5: Programs for Machine Learning, The Morgan Kaufmann series in machine learning, Morgan Kaufmann, Amsterdam, 1993, p. 302.
- [7] Z. Chen, Data Mining and Uncertain Reasoning: An Integrated Approach, John Wiley, USA, 2001, p. 370.
- [8] C.A.A. Meira, L.H.A. Rodrigues, S.A. Moraes, Analysis of coffee rust epidemics with decision tree, Tropical Plant Pathology 33 (2008) 114-124.
- [9] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.
- [10] H.P. Lima, S.M.F.S. Massruhá, Sistema Fuzzy Gen: Manual do usuário, Série Documentos 96, Embrapa Informática Agropecuária, Campinas, 2009, p. 24 (in Portuguese).
- [11] FuzzyLogic-Open Source Fuzzy Logic Library, 2011 [Online], <http://jfuzzylogic.sourceforge.net> (accessed May, 2, 2011).
- [12] International Electrotechnical Commission (IEC), Technical Committee no. 65: Industrial Process Measurement and Control, Sub-committee 65 B: Devices IEC 1131-Programmable Controllers, Part 7-Fuzzy Control Programming, Committee Draft CD 1.0 (Rel. 19 Jan 97) [Online], 1997, <http://www.fuzzytech.com/binaries/iecccd1.pdf>. (accessed May, 2, 2011)