Discovery of fuzzy predicates in database

Taymi Ceruto Cordovés[1], Orenia Lapeira Mena[1], Alejandro Rosete Suárez[1], Rafael Espín Andrade[1]

[1]Higher Technical Institute José Antonio Echeverría (CUJAE), Havana, Cuba
(tceruto, olapeira,rosete)@ceis.cujae.edu.cu, espin@ind.cujae.edu.cu

Abstract

Advanced technologies have enabled us to collect large amounts of data. These data may be transformed into useful knowledge. Because of our limited ability to manually process the data, it is necessary to use automatic tools to mine useful knowledge. Many data-mining methods have been proposed which are normally restricted to a given representation, such as rules and clusters. This paper proposes FuzzyPred, a metaheuristics-based data-mining method to obtain fuzzy predicates in normal form. We believe that the patterns obtained by FuzzyPred represent an interesting new representation of knowledge that is only obtained by our proposal.

Keywords: Knowledge Discovery in Databases, Predicates, Fuzzy logic, Metaheuristics

1. Introduction

Progress in technology has made possible for organizations to collect and to store massive amounts data, referred to different topics. Successful organizations see such databases as important pieces for developing and for implementing programs and strategies to gain competitive advantage, to increase efficiency and to provide more valuable services for customers. For that reason, it is necessary to develop a new generation of computational theories and tools to assist humans in extracting useful information from real data. These theories and tools are the subject of an emerging field: knowledge discovery in databases (KDD) [1].

KDD has evolved from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, data visualization and high-performance computing. The unifying goal is to extract high-level knowledge from low-level data in the context of large data sets [1-3].

KDD refers to the overall process of discovering useful knowledge from data, and data mining (DM) refers to a particular step in this process. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining are essential to ensure that useful knowledge is derived from the data [1].

Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular set of patterns (or models) about the data. It is worth noting that the space of patterns is often infinite, and the enumeration of patterns involves some form of search in this space. Practical computational constraints place severe limits on the subspace that can be explored by a data mining method [5].

The goals of knowledge discovery are defined by the intended use of the system. We can distinguish two types of goals: (1) verification and (2) discovery. If the goal is verification, the system is limited to verifying the user's hypothesis. If the goal is discovery, then the system autonomously finds new patterns. The discovery goal may be also divided into **prediction (**where the system finds patterns for predicting the future behavior of some entities), and **description (**where the system finds patterns for presentation to a user in a human-understandable form) [1-5].

Below are explained the main tasks of mining that have been defined [1-6]:

1. **Association Rule Mining** (Apriori, Genetic Algorithms, CN2 Rules): Rules are the most ancient knowledge representations, and probably the easiest to understand. Association rules are used to represent and to identify dependencies between items in a database. Usually the condition of a rule is a predicate in certain logic, and the action is an associated class. It is often used for market basket or transactional data analysis.

2. **Classification and Prediction** (CART, CHAID, ID3, C4.5, C5.0, J48): It involves identifying data characteristics that can be used to generate a model for prediction of similar occurrences in future data. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible range of values for this attribute.

3. **Cluster Analysis** (K-Means, Neural Networks, Expectation-Maximization): It attempts to look for groups (clusters) of data items that have a strong similarity to

other objects in the same group, but are the most dissimilar to objects in other groups.

As shown below, the most conventional data mining algorithms identify the relationships among transactions using specific knowledge representation model (rules, trees, clusters). It means that any learning algorithm and knowledge representation can be only the best algorithm in a certain subset of problems.

This paper proposes a novel way of extracting interesting knowledge from transactions, called FuzzyPred (Fuzzy Predicates) [7]. The basic model of knowledge representation in FuzzyPred is logic predicates in normal forms. To do so, the proposed algorithm integrates fuzzy set concepts and metaheuristic algorithms to search for logic predicates in a given data set.

With this aim, the paper is organized as follows. Section 2 gives a brief overview of previous works. The proposed learning approach is described in Section 3. In Section 4, the behavior of the proposed approach in two different databases from an international repository (UCI Learning Maching) is analyzed. Conclusions and proposal of future work are given in Section 5.

2. Background

In order to obtain predicates, three main approaches are relevant from the literature:

Inductive Logic Programming (ILP) [8] has been defined as the intersection of inductive learning and logic programming. It is a discipline which investigates the inductive construction of first-order clausal theories from examples and background knowledge. ILP inherits its goal: to develop techniques to induce hypotheses from observations (examples) and to synthesize new knowledge from experience.

ILP has limitations, such as; it needs a set of observations (positive and negative examples), background knowledge, hypothesis language and covers relation.

Genetic Programming (GP) [9] is based on Genetic Algorithms. The main idea is to obtain a mathematic expression that relates some variables to a given target variable. The mathematic expression is often expressed as a tree, where the internal nodes are operators (such as addition, multiplication, etc), and the terminal nodes are variables or constants. Each tree is evaluated in each example according to the error between the result of its application and the target variable.

GP has as limitations the exclusive use of genetic algorithms, the learning is supervised, and the trees that are obtained can vary of size (it implies that it is necessary to implement limits in the growth). Our proposal is different in two senses: learning can be unsupervised and other metaheuristic algorithms (not genetic algorithms) may be used.

Discovering Fuzzy Association Rules [10-12]. The definition of linguistic terms is based on the fuzzy set theory. Hence, the rules having these terms are called fuzzy association rules. In fact, the use of fuzzy techniques has been considered as one of the key components of DM systems because of the affinity with the human knowledge representation. This approach is based mainly in Genetic Algorithms or Ant Colony Optimization.

The structure of the knowledge is predefined (if/then rules) and it offers a convenient format for expressing pieces of knowledge, but it is just a format which can cover specific semantic. The antecedent is composed by atomic expressions connected with logic operator and the variables that appear cannot repeat.

The rules are only conditional relation among conditions and conclusions. This restricts the possibility to obtain a new knowledge, where the main logical relation is the conjunction or disjunction or equivalence or any free combination of logical operators.

In general, several models of knowledge are impossible to be obtained by the previous methods. For instance, in a fuzzy database with variables A, B, C, and D, the following knowledge models may not be obtained:

- (A and B) or (not B and C)
- (A and B and not D)
- (B) or (not B and C) or (D)

It is worth noting that some of these predicates may be part of the antecedent of a rule. However, they alone are not obtained as knowledge, and truth value of this knowledge is never calculated. It is important to note that if some of these predicates have a high truth value they represent useful and valuable knowledge that describe the data, but they are never obtained by the previous methods. The reason behind this is that the models of knowledge representation in the previous methods are limited to rules, trees and clusters. The three predicates presented before are examples of normal forms that were not obtained by them.

As our proposal is to obtain any sort of predicate which allows the presence of different connectives among the values of any variable, none of these notable approaches are directly usable. The aim is to obtain fuzzy predicates with great truth values in the given examples. To the best of our knowledge, there has not been another attempt to obtain fuzzy predicates before.

3. The fuzzy data mining algorithm

FuzzyPred is the name of the algorithm proposed in this paper. It is a hybrid KDD method. It constitutes an interesting variant because it is a generalization regarding the methods that today exists for knowledge discovery (fuzzy or not) [7].

The KDD process is interactive and iterative, involving numerous stages with many decisions made by the user. The same as in FuzzyPred, because it involves the following stages: data selection, special pre-processing using fuzzy transformation, application of metaheuristics to help data mining process, and storage and visualization of results.

3.1. Data selection

Data selection is aimed at choosing the dataset to be analyzed. Some of the data was not pertinent to the data mining and was ignored. Normally only a part of the real-world database is usually selected [1, 3].

Once the data resources available are identified and selected, they need to be cleaned, built into the desired form, and formatted.

3.2. Data pre-processing

This consists of all the actions taken before the actual data analysis process starts. Data pre-processing includes data cleaning, data transformation, and other activities improving the quality of the dataset [2].

Data cleaning is an intensive procedure that is absolutely necessary for successful data mining. Data cleaning involved the identification of missing, inconsistent or mistaken values. Some algorithms can be applied at this stage of discovering and removing (or correcting) “outliers” in data [4]. Tools used in this step provide a picture of distributions, and statistics such as maxima, minima, mean values, etc.

On the other hand, some selected data may have different formats because they are chosen from different data sources. There is no unique procedure and the only criterion is to transform the data for convenience of use during the data mining stage [1-5].

Transformation of attributes is absolutely essential if the chosen learning method can only handle specific types of attributes. In this case we propose that the attributes are expressed in linguistic terms, which are more natural and understandable for human beings [5].

Fuzzy set theory was first proposed by Zadeh in 1965 [13]. This theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [6]. The use of fuzzy sets to describe association between data extends the types of relationships that may be represented. It facilitates the interpretation of pattern in linguistic terms, and it avoids unnatural boundaries in the partitioning of the attribute domains [6, 13].

Attribute ranges may be better described by fuzzy partitions (classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt). The proposed fuzzy mining algorithm uses membership functions to transform each numerical and nominal value into a fuzzy set in linguistic terms.

The membership functions may be obtained from the expert information (if it is available) or by a normalization process. If the latter is the case, it is necessary to perform a fuzzy partition of the input variable spaces dividing each universe of discourse into a number of equal or unequal partitions, to select a kind of membership function and to assign one fuzzy set to each subspace.

In our case, the definition of membership functions is based on Xfuzzy 3.0 [14] (in particular we used Xfedit). It provides a graphic interface to facilitate the description of fuzzy sets as Figure 1 shows. The tool is formed by a group of windows that allow the user to create and to publish the membership functions for each attribute.

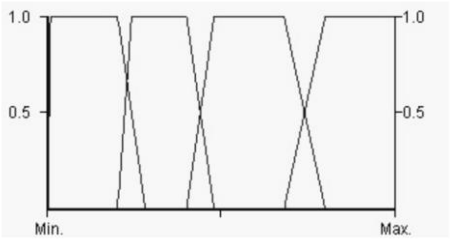


Fig. 1: Xfedit in Xfuzzy 3.0

For the experiments one linguistic term (selected by data miner) is used for each attribute. It implies that the number of fuzzy variables for the data-mining process will be the same as the original number of attributes.

The application of the proposal involves applying specific algorithms for extracting patterns from data sets in each particular representation. Our data mining method can be viewed as three primary algorithmic components: (1) model representation, (2) model evaluation, and (3) search.

3.3. Model representation

Model representation is the language used to describe discoverable patterns. If the representation is too limited, then some models of knowledge may not be obtained from the examples. It is important that a data analyst fully comprehend the representational assumptions that might be inherent in a particular method [1].

So, the aim of this work is to obtain a set of linguistic expression like predicates. Predicates may interpreted as hierarchy trees where their basic predicates measure the convenience of attributes for each alternative (a combination of values of the variables), and conjunction and disjunction are aggregation operators of preferences. The truth value of a fuzzy predicate is a mapping from the universe set X to the continuous interval [0, 1], instead of the classical set {0, 1}.

In general, a predicate may be a tree where each internal node may be a fuzzy operator (such as conjunction, disjunction, and negation) and each leaf is a fuzzy variable from the database.

In this paper we restrict this hierarchical representation (tree) to a normal form (such as conjunctive or disjunctive normal forms). In classical logic, for each predicate there is a logically equivalent formula in conjunctive/disjunctive normal form [15, 16]. This implies that the normal forms in classical logic can be seen as general models to represent logic predicates. This is not exactly true in fuzzy logic, because the truth value of a formula depends on the type of fuzzy operator used. Despite of this fact, we believe that the normal forms have a similar form of generality for fuzzy predicates.

In the following, we represent the predicates in a compact description of relations based on Normal Forms. It is worth clarifying that the equivalences that are obtained from this alternative are more true than false in Multivalued logic, but not absolutely true.

In particular, we use the Conjunctive and Disjunctive Normal Forms (CNF and DNF) with connectives. The propositional connectives ($\wedge$,$\vee$,$\neg$) using in normal forms symbolize operations on sentences.

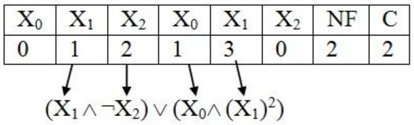
- p $\wedge$ q is true if and only if both p and q are true. It is called conjunction, and symbolizes the inclusive use of “and” in natural language.
- p $\vee$ q is false if and only if both p and q are false. It is called disjunction, and symbolizes the use of “or” in natural language.
- $\neg$p is true when p is false, and conversely. It is called the negation of p.

We use a fixed-length chromosome and Integer Coding to represent an individual, but in reality the predicate has variable size because we add a special value that indicates the absence of that variable in the predicate. Each variable involved in the predicate can take different values according to the following scale. In general, each predicate is represented by a vector, where each component represents a variable or term. Each term may have a value in the following scale.

- 0: it is not in the predicate
- 1: it’s in the predicate
- 2: it appears denied
- 3: it appears with the modifier “very”
- 4: it appears with the modifier “hyper”
- 5: it appears with the modifier “something”

In the case of the modifiers (values of 3, 4 and 5), the truth value is calculated as the power of the original truth. The power differs depending on the modifier. The value that is used for the modifier “very” is 2, for “hyper” is 3, and for “something” is 0.5. These powers are used for intensifying or moderating the value of truth of the variable [15-16].

A code example with its corresponding predicate is shown in the figure 2:



X_0	X_1	X_2	X_0	X_1	X_2	NF	C
0	1	2	1	3	0	2	2

$$(X_1 \wedge \neg X_2) \vee (X_0 \wedge (X_1)^2)$$

Fig. 2 Encoding of a predicate

In the chromosome the variables can appear twice or more times, i.e. they are included in two or more clauses. The two last positions in the chromosome represent the type of normal form (1 means CNF, 2 means DFN), and the number of clauses (C).

3.4. Model evaluation

Model evaluation criteria are quantitative statements (or fit functions) of how good is a particular pattern. This is a key factor for the extraction of knowledge because the search will be guided by this value. Furthermore, quality measures express the importance and relevance of the results obtained [2, 3].

For each solution the unique aspect that was considered in this paper is the truth value, which depends on the number of clauses, variables and rows of the data set. The truth value is calculated using fuzzy logic operators. Among operations on fuzzy sets are the basic and commonly used complementation, union and intersection, as proposed by Zadeh [13].

The main feature of the fuzzy logic is that it does not give a unique definition of the classic operations as the union or the intersection. We have studied the characteristics of several fuzzy operators for decision-making. For example, Zadeh operators (min-max) are insensitive. In this case, the change of one variable may not change the value of the result (0.5 $\wedge$ 0.5 = 0.5; 0.5 $\wedge$ 0.8 = 0.5). The probability operators are not idempotent; the conjunction of two variables, with the same values, does not result in the same number ($0.5 \wedge 0.5 = 0.25$; $0.5 \wedge 0.8 = 0.4$), in fact the results with a lower value [17, 18].

We also studied other operators such as Hamacher, Einstein, Lukasiewicz, Drastic and it was determined that they have the same problem, because they are associative [17, 18]. On the contrary, the compensatory fuzzy logic is sensitive and idempotent [19, 20]. This aspect is very important for the correct interpretation of the results. We have also studied other type of operators, such as Harmonic, Geometric and Arithmetic Mean, and their dual [19]. All these have a value between the maximum and minimum.

The fitness assignment for the predicates is performed as Figure 3 shows.

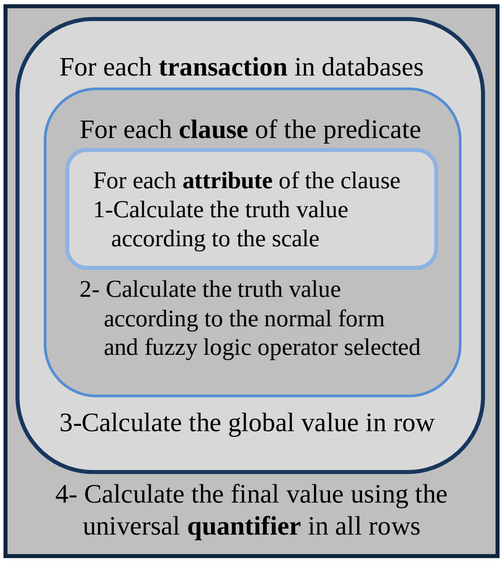

For each **transaction** in databases
For each **clause** of the predicate
For each **attribute** of the clause
1-Calculate the truth value according to the scale
2- Calculate the truth value according to the normal form and fuzzy logic operator selected
3-Calculate the global value in row
4- Calculate the final value using the universal **quantifier** in all rows

Fig. 3: Fitness assignment

This is a simple example to show how to evaluate a determined predicate using the process described in the Figure 3:

$(X_1 \wedge \neg X_2) \vee (X_0 \wedge (X_1)^2)$

In Table I, the first three columns represent the values of the attributes in the original database and the last four columns represent the result of the first step that calculate the truth value of the attribute according to the scale previously explained.

Table I First step of the evaluation

X_0	X_1	X_2	X_1	$\neg X_2$	X_0	$(X_1)^2$
0.1	0.2	0.5	0.2	0.8	0.1	0.04
0.8	0.3	0.9	0.3	0.7	0.8	0.09
0.4	0.5	0.5	0.5	0.5	0.4	0.25
0.5	0.7	0.2	0.7	0.3	0.5	0.49
0.1	0.8	0.3	0.8	0.8	0.1	0.64
1	0.4	0.9	0.4	0.4	1	0.16

In Table II, the first two columns represent the computation of the value for each clause according to the normal form (DNF) and the fuzzy operator selected (in this case the minimum of Zadeh is used). The last column represents the third step, the calculation of the value of the all predicate in the row (in this case the maximum of Zadeh is used to calculate the disjunctions of clauses).

Table II Second and third step of the evaluation

$X_1 \wedge \neg X_2$	$X_0 \wedge (X_1)^2$	$(X_1 \wedge \neg X_2) \vee (X_0 \wedge (X_1)^2)$
0.2	0.04	0.2
0.3	0.09	0.3
0.5	0.25	0.5
0.3	0.49	0.49
0.8	0.64	0.8
0.4	0.16	0.4

The fourth step consists of using the universal quantifier (conjunction of the values obtained in the last column of the Table II). The truth value of this predicate $(X_1 \wedge \neg X_2) \vee (X_0 \wedge (X_1)^2)$ is 0.8.

3.5. Search strategy

Search strategy is very important and before defining it, the first thing that it is necessary to keep in mind is the dimension of the space of the search. A search method consists of two components: (1) parameter search and (2) model search [21]

In many cases the data mining problem has been reduced to purely an optimization task: find the patterns that optimize the evaluation criteria [1].

In the last years, a new kind of approximate algorithm has emerged which basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space. These methods are nowadays commonly called metaheuristics [21, 22].

Recently some researchers in the field tried to propose a definition of metaheuristics. Summarizing, we outline fundamental properties which characterize metaheuristics [22]:

- Metaheuristics are strategies that “guide” the search process.
- The goal is to efficiently explore the search space in order to find (near) optimal solutions.
- Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
- Metaheuristic algorithms are approximate and usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- Metaheuristics are not problem-specific.

Metaheuristics represent an important class of techniques to solve, approximately, hard combinatorial optimization problems for which the use of exact methods is impractical. Metaheuristics such as Tabu Search (TS), Hill Climbing (HC), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolutionary Computation (EC), Iterated Local Search (ILS), and Simulated Annealing (SA) have been used each of them in isolation, or in combination [21, 22].

Each metaheuristic is different according to some aspects. Some of them are population based (GA, EC), and others are trajectory methods (TS, HC). Although, they are based on different philosophies, the mechanisms to efficiently explore a search space are all based on intensification and diversification [21, 22].

Many successful applications have been reported for all of them. According to the No Free Lunch Theorem [23] it is impossible to say which the best of all metaheuristics is. It depends on the encoding of the problem, the correct selection of the objective function as well as the operators. The only possibility is to make experiments with different parameters.

In particular, we use an open source library called BICIAM [24].

In FuzzyPred the learning task will be formulated as an optimization problem. This class of the problem can be defined by:

- a set of variables (depends on data selection)
- variable domains (depends on the scale)
- an objective function

The global process in FuzzyPred tries to get predicates with high truth value. For that reason the algorithm tries to maximize it as it is shown next:

```
BEGIN
   Predicate Set = Ø
   Initialize parameters
   $P_I$ = Generate an initial solution
   Predicate Set = Predicate Set  + $P_I$
   REPEAT
     Pc = Generate new solution according to the metaheuristic selected
     If Pc is accepted
        $P_I$ =   Pc
        Predicate Set=Predicate Set + $P_c$
   While stop condition is not verified
   Return Predicate Set
 END
```

The final result is the concatenation of the predicates obtained by running the algorithm several times.

3.6. Post-processing and interpretation

Post-processing translates discovered patterns into forms acceptable for human beings. Interpretation of results is an important aspect of the KDD process. The discovered patterns should be presented to a user in an understandable manner. Presenting numerous patterns, without any scoring, is very inefficient. For that reason some measures of importance, relevance, and interestingness are required [1-5].

Four functions were implemented: DecreaseVariables, PredicateWithEqualsClauses, EvidentPredicates and DeleteRepeatedSolutions. In each one of them, the objective is to improve the legibility.

The behavior of each one of these methods is detailed next:

- DecreaseVariables: It refines the predicate by reducing unnecessary variables. For this, it removes a random variable from the predicate in order to obtain a reduced version of the predicate. Then, it evaluates the reduced predicate. If the truth value of the reduced predicate is greater than the original predicate, then the reduced predicate is conserved as a better solution. This process is repeated until no improvement is obtained. The effect of this function is to improve the outcomes and to reduce the number of variables in the predicates to facilitate their interpretation.

- PredicateWithEqualsClauses: It excludes the repeated clauses from a predicate.
  Example: $(X_0 \vee X_3) \wedge (X_0 \vee X_3)$

- EvidentPredicates: It deletes the predicates that have the property of containing an atom and it’s denied, because these predicates are obvious in spite of its high truth value.
  Example: $(\neg X_0)^{0.5} \vee (X_0)^{0.5}$

- DeleteRepeatedSolutions: It eliminates from the list of predicates, those repeated predicates. The function considers that all the possible permutations inside each clause are equal.
  Example: $\neg X_1 \vee X_{2,}\ \ X_2 \vee \neg X_1$

These functions should be executed in a predetermined order, i.e. in the order that they were presented. Otherwise, the result of the application of a function may deteriorate the results obtained in a previous function.

Post-processing makes also possible to visualize and to store the extracted patterns. A standard data mining language or other standardization efforts will facilitate the systematic development of datamining solutions, to improve interoperability among multiple data mining systems and functions, and to use of data mining systems in industry and society.

Recent efforts in this direction include Predictive Model Markup Language (PMML) created by Data Mining Group. PMML is an XML-based language that enables the definition and sharing of predictive models between applications [25]. It is the de facto standard to represent predictive models and for this reason it was necessary to include it in FuzzyPred.

PMML follows an intuitive structure to illustrate a data mining model and it can be described by the following components [25]:

- Header: It contains general information about the PMML document, such as copyright information for the model, its description, and information about the application used to generate the model such as name and version.
- Data Dictionary: It contains definitions for all the possible fields used by the model. It is here that a field is defined as continuous, categorical, or ordinal. Depending on this definition, the appropriate value ranges are then defined as well as the data type (such as, string or double).
- Model: It contains the definition of the data mining model.
- Mining Schema: It lists all fields used in the model. This can be a subset of the fields as defined in the data dictionary.

In the version 4.1 of PMML several specific elements are included for the following techniques of modeling:

- AssociationModel.
- BaselineModel.
- ClusteringModel.
- NaiveBayesModel.
- MiningModel.
- NearestNeighborModel.
- NeuralNetwork.
- RegressionModel y GeneralRegressionModel.
- RuleSetModel.
- SequenceModel.
- Scorecard.
- SupportVectorMachineModel.
- TextModel.
- TimeSeriesModel.
- TreeModel.

FuzzyPred is a new way of obtaining knowledge that uses a different model and therefore it was necessary to adapt the original RuleSetModel (the nearest model) defined in PMML in order to create a new model called FuzzyPredicateModel. The labels "Header" and "DataDictionary" are maintained. In addition, FuzzyPredicateModel includes two fundamental labels: "MiningSchema" and "PredicateSet".

Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is interactive [1-5].

This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, reports, matrices, curves or cubes. The variety, quality, and flexibility of visualization tools may strongly influence the usability, interpretability and attractiveness of a data mining system [1-5].

FuzzyPred visualizes the predicates obtained in a tree supported in the tool SpaceTree [26] as Figure 4 shows.

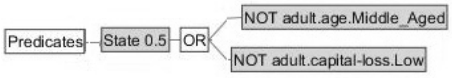


Fig. 4: Example of visualization

4. Experiments

To illustrate the performance of the proposed method, two databases with different characteristics have been chosen: Adult and Forest Fire [27]. In Adult we are going to develop two studies using different variables. Afterwards we shall carry out results analysis where we look at the behavior of the method.

The following values have been considered in each experiment:

- Metaheuristics used for mining fuzzy predicates: Random Search, Hill Climbing, Genetic Algorithm

- Genetic process: 20 individuals, 0.9 as crossover probability, 0.5 as mutation probability
- 30 repetitions were executed, each one with a maximum number of 500 iterations.
- Compensatory Fuzzy Operator (Geometric mean and dual) were used to evaluate the predicates.

All of the experiments were performed using a Intel(R) Core (TM) Duo, 2.66 GHz CPU with 2Gb of memory and running Windows 7 Ultimate de 64 bits.

The first study was with the real-world database **Adult**. It contains data of people that were interviewed in 1994, conducted by the Bureau of the Census for statistics. It was originally used to predict if the monetary entrance exceeded the 50 thousand american dollars a year. Ronny Kohaviy and Barry Becker transcribed the data from a public use microdata tape supplied by the Bureau of the Census [27]. This database consists of 32561 records with 15 attributes each one.

To develop the different experiments, we extracted the 3 quantitative attributes from them: “age”, “hours-per-week”, “capital-gain”.

The fuzzy sets corresponding to the linguistic labels for a linguistic variable are specified by means of the corresponding membership functions. They can be defined by the user or defined by means of a uniform partition if the expert knowledge is not available.

In this experiment, uniform partitions with trapezoidal membership functions are used:

- for the variable <u>age</u> we define 4 linguistic labels: young, middle-aged, senior, old
- for the variable <u>hours-per-week</u>: Part-time, Full-time, Over-time, Too-Much
- for the variable <u>capital-gain</u>: none, low, high

In the first test the selected labels were the following ones:

- Senior Age
- High capital gain
- Over_time hours-per-week

An example of interesting fuzzy predicate obtained with the proposed approach using Hill Climbing is the follow:

High capital gain ∨ Over_time hours-per-week
Truth Value: 0

This predicate shows a singular knowledge. It means that the people with high capital gain (> 55555) are not predominant in the "Adult" database, although their hours per week are over time (40-70).

The second test was with the same databases, but the linguistic terms selected were different:

- Middle Age
- Low capital-gain
- PartTime hours per week

An example of fuzzy predicate obtained with the proposed approach using Genetic Algorithm is the follow:

(¬Middle Age ∧ ¬Low capital gain ∧ ¬PartTime hours per week)
Truth Value: 0.99

This example shows that in the database exist many people that were not in the middle age (20-50), that they have not low capital gain (< 55555) and they do not work on part-time style.

The last part of our study used the **Forest Fire** database [27]. It was created in 2007 by Paulo Cortez y Anibal Morais in the Minho University, Portugal. This database consists of 517 records with 13 attributes each one. To develop a basic experiment, we extracted the following four quantitative attributes:

- Temp: Temperature in Celsius grades.
- RH: Relative humidity in %.
- Wind: Speed of the Wind in Km/h.
- Area: burnt area of the forest in hectares.

The initial linguistic partitions are composed of three or four linguistic terms with uniformly distributed trapezoidal:

- <u>Temp</u>: cold, cool, hot
- <u>RH</u>: little, medium, high
- <u>Wind</u>: slow, medium, fast
- Area: little, medium, big

The linguistic terms selected are:

- Cool Temp
- Medium RH
- Medium Wind
- Little Area

Using Random Search the result was:

(Cool Temp.^{2} ∨ Little Area$^{0.5}$) ∧ (Medium Wind^{2})
Truth Value: 0.82

This predicate means that in the database is very common that the temperature is very pleasant or that the burnt area is small, and the speed of the wind is medium. This knowledge is also reaffirmed by the following obtained predicate:

(¬MediumWind^{2} ∧ ¬Cool Temp$^{0.5}$) ∨ ¬Little Area
Truth Value: 0.

This singular predicate with truth value zero indicates that this never happens in the database. It coincides exactly with the previous one, because the change is that the variables appear denied (an important capacity of the proposed system).

This kind of fuzzy predicates lets us represent knowledge about patterns of interest in an explanatory and understandable form which can be used by the experts.

It is worth noting that Adult and Forest Fire have been extensively used. However, the predicates presented in this section have been never obtained before. This is caused by the singular nature of the representation based on normal form that is presented in this paper.

5. Conclusion

In this paper, a novel and interesting data mining algorithm, called FuzzyPred has been proposed. In general, it can be said that the obtained predicates represent regularity in the databases and they can be used to provide some valuable knowledge for the experts.

The original contributions of the proposed approach include:

- The learning process is not supervised.
- The structure of the knowledge is not totally restricted, but it focuses only on fuzzy predicates. It represents a more flexible structure to allow each variable to take more than one value, and to facilitate the extraction of more general knowledge.
- Fuzzy logic contributes to the interpretability of the extracted predicates due to the use of a knowledge representation nearest to the expert.
- It is possible to use different fuzzy operators to calculate the truth value of the predicate (although compensatory is privileged because it has demonstrated to be highly efficient in the context of decision making).
- There is more than one search method (metaheuristics) available.

There is still much work to be done in this field. Some guidelines for future research in this direction include:

- The inclusion in the algorithm of interestingness quality measures to guide the discovery process (and combinations of them) as objective functions in order to obtain fuzzy predicates with better properties.
- Our method assumes that the membership functions are known in advance. We will propose some fuzzy learning methods to automatically derive the membership functions.
- Test of our proposal in databases with a great number of selected attributes.

6. References

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD Process for Extracting Useful Knowledge", *Communications of the Acm*, Vol. 39, pp. 27-34, 1996.

[2] J. Han and M. Kamber, "Data Mining: Concepts and Techniques (2nd edition), *The Morgan Kaufmann Series in DataManagement Systems*, ISBN: 978-1-55860-901-3,pp. 1-14, 2006.

[3] M. Berry, M. Linoff and S. Gordon, "Data Mining Techniques", *John Wiley & Sons*, ISBN: 0-47L-47b4-3, pp. 11-40, 2004.

[4] J. Hernández, M. Ramírez and C. Ferri "Introducción a la minería de datos", *Ed. Pearson Education of Madrid*, ISBN 84-205-4091-9, pp. 3-125, 2004.

[5] B. Sierra, "Aprendizaje Automático: Conceptos básicos y avanzados", *Ed. Pearson Prentice Hall*, ISBN 10: 84-8322-318-5, pp. 15-82, 2006.

[6] K. Venugopal, K. Srinivasa and L. Patnaik, "Soft Computing for Data Mining Applications". *Studies in Computational Intelligence, Springer-Verlag, Berlin Heidelberg*, Vol. 190, ISBN 978-3-642-00192-5, pp. 1-15, 2009.

[7] T. Ceruto, A. Suarez and R. Espin, "Método para obtener Predicados Difusos a partir de datos utilizando metaheurísticas", *Revista Internacional de Investigación de Operaciones (RIIO)*, Colombia, ISSN 2145 - 9517, Vol. 1, pp. 29-37, 2010.

[8] S. Muggleton and L. De Raedt, "Inductive Logic Programming: Theory and Methods." *The Journal of Logic Programming*, Vol. 19-20, pp. 629-679, 1994.

[9] D. Goldberg and J. Koza, "Genetic Programming Theory and Practice V", *Springer Science+Business Media*, ISBN-13: 978-0-387-76307-1, pp. 1-13 , 2008.

[10] T. Hong and Y. Lee, "An Overview of Mining Fuzzy Association Rules". *Fuzzy Sets and Their Extensions: Representation, Aggregation and Mode*ls, Springer Berlin / Heidelberg, pp. 397-410, 2008.

[11] M. Delgado, N. Manín, M. Martín-Bautista, et al. "Mining Fuzzy Association Rules: An Overview", *Soft Computing for Information Processing and Analysis, Springer Berlin / Heidelberg*. Vol. 164, pp. 351-373, 2005.

[12] M. De Cock, C. Cornelis and E. Kerre, “Fuzzy Association Rules: a Two-Sided Approach”, *FIP2003 (International Conference on Fuzzy Information processing: Theories and Applications*), pp. 385-390, 2003.

[13] L. Zadeh, “Fuzzy Sets”, *Information Contro*l, Vol. 8, pp-338-353, 1965.

[14] Xfuzzy Home Page, “Fuzzy logic design tools”, Available: www.imse-cnm.csic.es/Xfuzzy/

[15] E. Trillas, “On a model for the meaning of predicates. A naive approach to the genesis of fuzzy sets”, *Studies in Fuzziness and Soft Computing*, Vol. 243 (9), pp. 175-205, 2009.

[16] A.D. Bruno, “Normal forms”, *Mathematics and Computers in Simulation*, Vol. 45, pp. 413-427, 1998.

[17] M. Mizumoto, "Fuzzy Sets and their Operations I", *Information and Control,* Vol. 48(1), pp. 31-48, 1981.

[18] M Mizumoto. "Fuzzy Sets and their Operations II", *Information and Control,* Vol. 50(2), pp. 160-174, 1981.

[19] M. Mizumoto, "Pictorial Representactions of fuzzy conectives, Part II:cases of Compensatory operators and Self-dual operators", *Fuzzy Sets and Systems*, Vol. 32, pp. 45-79, 1989.

[20] R. Espin, E. Fernandez, G. Mazcorro, J. Marx-Gómez and M. Lecich, “Compensatory Logic: A fuzzy normative model for decision making”, *Investigación Operacional*, Vol. 27 (2), pp. 178-193, 2006.

[21] E. Talbi, “Metaheuristics: From Design to Implementation”, *Ed. John Wiley & Sons*, ISBN 978-0-470-27858-1, pp. 18-29, 2009.

[22] C. Blum and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”, *ACM Computing Surveys*, Vol. 35(3), pp. 268–308, 2003.

[23] D. Wolpert, W. Macready, “No Free Lunch Theorems for Optimization”, *IEEE Transactions on Evolutionary Computation*, Vol. 1, pp .67-82, 1997.

[24] J. Fajardo and A. Suarez, " Algoritmo Multigenerador de Soluciones para la competencia y colaboración de generadores metaheurísticos”, *Revista Internacional de Investigación de Operaciones (RIIO)*, Colombia, ISSN 2145 - 9517, Vol. 1 (0), pp-57-62, 2010.

[25] Data Mining Group, “Welcome to DMG”, Available: www.dmg.org, 4/6/2012

[26] SpaceTree, “SpaceTree”, Available: www.cs.umd.edu/hcil/spacetree/, 12/6/2012

[27] C. Blake and C. Merz, "UCI Repository of machine learning databases" *University of California, Irvine, Dept. of Information and Computer Sciences*, Available: http://archive.ics.uci.edu/ml/, 1988.