ELSEVIER

Fuzzy Sets and Systems 123 (2001) 291–306

FUZZY sets and systems

www.elsevier.com/locate/fss

A new approach to fuzzy rule generation: fuzzy extension matrix☆

X.Z. Wang[a,*], Y.D. Wang[b], X.F. Xu[b], W.D. Ling[a], D.S. Yeung[c]

[a] *Department of Mathematics and Computer Science, Hebei University, Baoding, Hebei, People's Republic of China*
[b] *Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, People's Republic of China*
[c] *Department of Computing, Hong Kong Polytechnic University, HungHom, Kowloon, Hong Kong, People's Republic of China*

Received 28 December 1998; received in revised form 8 March 2000; accepted 7 July 2000

Abstract

This paper proposes a new approach to fuzzy rule generation from a set of examples with fuzzy representation. The new approach called fuzzy extension matrix incorporates the fuzzy entropy to search for paths and generalizes the concept of crisp extension matrix. By discussing paths of the fuzzy extension matrix, a new heuristic algorithm for generating fuzzy rules is introduced. Compared with the crisp extension matrix, the proposed method has the capability of handling fuzzy representation and tolerating noisy data or missing data. A case study shows that the proposed heuristic algorithm partially inherits the advantages from the crisp case such as simplicity of rules and high learning accuracy. The proposed approach offers a new, practical way to automatically acquire imprecise knowledge.

Keywords: Learning; Knowledge acquisition; Learning from fuzzy examples; Fuzzy entropy; Extension matrix; Heuristic algorithm

1. Introduction

Machine learning is the essential way to acquire intelligence for any computer systems. One of the most important branches of machine learning should be learning from examples which are usually considered to be two classes, namely, positive and negative class. The main task of learning from examples is to find a set of rules which covers positive training-examples and exclude negative training-examples, therefore, learning from examples is sometimes called concept acquisition. During the past several decades, machine learning community has developed many approaches to learning from examples. From these approaches, some

☆ This work is partially supported by Natural Science Foundation of Hebei Province and research fellowship grant (G-YY12) of Hong Kong Polytechnic University.

* Corresponding author. Department of Computing, The Hong Kong Polytechnic University, HungHom, Kowloon, Hong Kong. Tel.: +852-2766-7274; fax: +852-2766-7274.

E-mail address: csxzwang@comp.polyu.edu.hk (X.Z. Wang).

0165-0114/01/$ - see front matter
PII: S0165-0114(01)00002-1

have been selected to be intensively investigated and then successfully moved from laboratories to real-life applications.

One example is the information-entropy-based decision tree induction [11,12]. A popular and powerful heuristic algorithm for generating crisp decision trees is called ID3. The earlier version of ID3, which is based on minimum information entropy to select expanded attributes, was proposed by Quinlan in 1986 [11] and subsequently the revised version called C4.5 was given in [12]. The main advantage of decision tree induction is that the algorithm can generate a relatively smaller tree without much computational effort. In machine learning community, there has been much research on decision trees including the improvement of algorithms, the handling of different attributes and applications of decision trees in different domains (which could be found from many existing references). Another example is the extension-matrix-based rule induction [3,8]. The concept of extension matrix was first proposed by Hong [3] and then has become a powerful tool to study the computational complexity of learning from examples and to design learning algorithms. Based on the concept of extension matrix, some new learning algorithms such as AQ11 [8] and HCV [20] were developed, the optimization of learning from examples was proved to be NP-hard [3], several learning systems such as AE* [3,4] and SLFE [24] were established, and the generation of smallest fuzzy decision tree was proved to be NP-hard [17]. Additionally, the concept of extension matrix was found to be very useful to issues such as noise handling [21], feature subset selection [10,18], hyperspectral remotely sensed images classification rule acquisition [13], fuzzy production rule simplification [16], etc. Under crisp environment, a comparative study between decision trees and extension matrices shows that the decision tree induction is suitable for learning problems with large scale and the extension matrix induction is suitable for learning problems with high accuracy [5].

Traditional approaches to learning from examples, such as the above-mentioned decision tree and extension matrix, usually require attributes to take nominal-values (symbols). They obviously have the following two weaknesses. (1) If attribute-values have linguistic meaning (such as young, old, etc.) and there exist an overlapping among these attribute-values, these attribute-values are still regarded as nominal symbols and the overlapping cannot be considered in learning process. (2) If attributes take real-values, they must be transformed into nominal symbols before learning. This transformation seems to be reluctant since the numerical numbers have a linear order but nominal symbols have no order at all (i.e., the initial order information is entirely lost in the transformation). Fortunately, these two weaknesses can be overcome by incorporating fuzzy set theory into learning from examples. By using linguistic terms defined by fuzzy sets, fuzzy learning algorithms can be designed to generate fuzzy rules from examples. Compared with traditional rules, fuzzy rules represent learned knowledge more naturally to the way of human thinking and more robust in tolerating imprecise and conflict information.

Developing fuzzy rule generation approaches is very useful to the knowledge acquisition phase of artificial expert systems. One important way of developing such approaches is to generalize the existing traditional approaches. It should be noted that this generalization is not straightforward and often is difficult due to the use of membership functions. As an illustration, one can see that many different fuzzy versions of decision tree induction have been developed to handle the acquisition problem of imprecise knowledge [6,14,19,15,22]. Fuzzy decision tree algorithms not only inherit the advantages of the crisp case but also make the knowledge representation more natural.

This paper aims to generalize a traditional extension-matrix-based rule induction, i.e., to give a fuzzy version of this induction. We would like to investigate the feasibility of generalization, formalization of fuzzy extension matrices, and strengths and weaknesses of this approach to fuzzy rule generation. Moreover, we would like to check whether the fuzzy version can inherit the main advantages from the traditional version.

The organization of this paper is as follows. Section 1 is the introduction. Section 2 presents some basic concepts of extension matrix under fuzzy environment. Based on these concepts, a heuristic algorithm for generating fuzzy rules which incorporates the fuzzy entropy to search for a path is given in Section 3. The numerical experiments about the extension-matrix heuristic are conducted on a data-set called Rice Taste

Table 1
A small training set of positive examples

No.	Outlook			Temperature			Humidity		Wind	
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not_windy
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6
2	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8
3	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5
4	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6
5	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9
6	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3
7	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1
8	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8
9	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0
10	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3
11	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2

Problem and the advantages of the extension-matrix-based heuristic are verified in Section 4. Section 5 outlines our future works for investigating fuzzy extension matrices. The final section offers the conclusions of this paper.

2. Related concepts of extension-matrix under fuzzy environment

We use a notation F_m to denote the m-dimensional fuzzy vector space, i.e.,

$$F_m = \{(x_1, x_2, \ldots, x_m) \mid 0 \leqslant x_j \leqslant 1,\ j = 1, 2, \ldots, m\}.$$

Let $E = F_{k_1} \times F_{k_2} \times \cdots \times F_{k_n}$ be the product of n fuzzy vector spaces. An element in E, called an example, is represented in the form

$$e = (\mu_1, \mu_2, \ldots, \mu_n),$$

where $\mu_j = (a_{j1}, a_{j2}, \ldots, a_{jk_j}) \in F_{k_j}$ $(j = 1, 2, \ldots, n)$.

Consider a problem of learning from fuzzy examples where the classification is crisp. Suppose that all examples are classified into two classes, ***PE*** and ***NE***, called positive class and negative class, respectively. Let there be P positive examples, N negative examples and n attributes. The space of example labels is assumed to be $X = \{1, 2, \ldots, P + N\}$ and n attributes are $A_1, A_2, \ldots, A_n$. The value of the ith example versus the jth attribute is assumed to be a discrete fuzzy subset, denoted by μ_{ij} $(1 \leqslant i \leqslant P + N,\ 1 \leqslant j \leqslant n)$, $\mu_{ij} \in F_{k_j}$.

Each attribute corresponds to several linguistic terms which are regarded as fuzzy subsets defined on X. For arbitrarily given j, k $(1 \leqslant j \leqslant n,\ 1 \leqslant k \leqslant k_j)$, the kth linguistic term of the jth attribute is denoted by $L_j^{(k)}$ $(1 \leqslant j \leqslant n,\ 1 \leqslant k \leqslant k_j)$. When the linguistic terms of an attribute, $L_j^{(k)}$ $(1 \leqslant j \leqslant n,\ 1 \leqslant k \leqslant k_j)$, are considered as non-fuzzy labels, they will constitute the label space of the attribute. The value of an example with respect to the considered attribute is just a fuzzy set defined on the label space. In this way, the problem of learning from fuzzy examples is represented well. However, for the considered fuzzy set (the value of an example with respect to an attribute), its meaning is generally not very clear.

We illustrate the above formulation by a group of fuzzy examples shown in Tables 1 and 2 (adopted from [22] with some modification). It is easy to see from Tables 1 and 2 that $P = 11$, $N = 5$; $n = 4$; $k_1 = 3$, $k_2 = 3$, $k_3 = 2$, $k_4 = 2$, i.e., there are four attributes *Outlook*: $\{Sunny, Cloudy, Rain\}$; *Temperature*: $\{Hot, Mild, Cool\}$; *Humidity*: $\{Humid, Normal\}$; and *Wind*: $\{Windy, Not\_windy\}$. Note that the first row of the

Table 2
A small training set of negative examples

No.	Outlook			Temperature			Humidity		Wind	
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not_windy
1	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0
2	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7
3	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0
4	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7
5	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0

positive example set in Table 1, we have $\mu_{11} = (0.9, 0.1, 0.0)$; $\mu_{12} = (1.0, 0.0, 0.0)$; $\mu_{13} = (0.8, 0.2)$; $\mu_{14} = (0.4, 0.6)$. The meaning of the fourth value μ_{14} is not very clear. The positive example shown in the first row in Table 1 is represented as $e_1^+ = (\mu_{11}, \mu_{12}, \mu_{13}, \mu_{14})$.

Definition 1. Let A_j be the jth attribute and S_j be the set of all linguistic terms of A_j. Then, $[A_j \neq P_j]$ and $[A_j = P_j]$ are called propositions where $P_j \subset S_j$. The conjunctive form of several propositions, $\bigwedge_{j \in J} [A_j \circ P_j]$, is called a fuzzy rule where $J \subset \{1, 2, \ldots, n\}$, "$\circ$" denotes "$=$" or "$\neq$", and "$\wedge$" denotes "AND".

Definition 2. Consider an example $e = (\mu_1, \ldots, \mu_j, \ldots, \mu_n)$, where the jth attribute-value μ_j is a fuzzy set defined on $S_j = \{T\}$, the set of all linguistic terms of the jth attribute. The degree with which proposition $[A_j \neq P_j]$ (resp. $[A_j = P_j]$) covers the example e is defined as

$$\bigwedge_{T \in P_j} (1 - \mu_j(T)) \quad \left(\text{resp.} \bigwedge_{T \in P_j} (\mu_j(T)) \right),$$

where the crisp subset P_j is a subset of S_j and "$\wedge$" denotes the minimum operator. The degree with which a fuzzy rule covers the example e is defined as the minimum of covering degrees of its propositions.

Example 1. Consider the first positive example shown in Table 1

$$e_1^+ = (\mu_{11}, \mu_{12}, \mu_{13}, \mu_{14}) = (0.9/\mathit{sunny}, 0.1/\mathit{cloudy}, 0.0/\mathit{rain}; 1.0/\mathit{hot}, 0.0/\mathit{mild}, 0.0/\mathit{cool};$$

$$0.8/\mathit{humid}, 0.2/\mathit{normal}; 0.4/\mathit{windy}, 0.6/\mathit{Not\_windy})$$

and two fuzzy rules
R_1: $[\mathit{Outlook} \neq \{\mathit{Cloudy}, \mathit{Rain}\}]$ and
R_2: $[\mathit{Outlook} = \mathit{Sunny}] \wedge [\mathit{Humidity} \neq \mathit{Normal}]$.

The degree of R_1 covering e_1^+ is equal to $\min\{(1 - 0.1), (1 - 0.0)\} = 0.9$ and the degree of R_2 covering e_1^+ is equal to $\min\{0.9, (1 - 0.2)\} = 0.8$.

Definition 3. Let e be an example and E be a set of examples considered. $M(e)$ is defined as the vector of linguistic terms which correspond to memberships greater than or equal to 0.5; and $D(e)$ is defined as the vector of the corresponding memberships. $M(E)$ is defined by putting $M(e)$ $(e \in E)$ together. Similarly, $D(E)$ can be defined.

For example, consider the first row e_1^+ and the third row e_3^+ shown in Table 1. It is easy to check that $M(e_1^+)=(sunny, hot, humid, not\_windy)$, $D(e_1^+)=(0.9, 1.0, 0.8, 0.6)$,

$$M(e_3^+)=\begin{pmatrix} rain & hot & humid & windy \\ rain & hot & normal & windy \\ rain & hot & humid & Not\_ \\ rain & hot & normal & Not\_ \end{pmatrix} \quad \text{and} \quad D(e_3^+)=\begin{pmatrix} 0.9 & 0.7 & 0.5 & 0.5 \\ 0.9 & 0.7 & 0.5 & 0.5 \\ 0.9 & 0.7 & 0.5 & 0.5 \\ 0.9 & 0.7 & 0.5 & 0.5 \end{pmatrix}.$$

Consider Table 2 as NE, the set of all negative examples, then $M(NE)$ and $D(NE)$ can be given as follows:

$$M(NE)=\begin{pmatrix} sunny & hot & normal & Not\_ \\ cloudy & mild & normal & Not\_ \\ cloudy & cool & normal & Not\_ \\ cloudy & mild & normal & Not\_ \\ sunny & hot & normal & Not\_ \\ sunny & mild & normal & Not\_ \end{pmatrix}, \quad D(NE)=\begin{pmatrix} 0.8 & 0.6 & 1.0 & 1.0 \\ 0.7 & 0.7 & 0.8 & 0.7 \\ 1.0 & 0.8 & 0.8 & 1.0 \\ 0.6 & 1.0 & 0.7 & 0.7 \\ 1.0 & 0.5 & 1.0 & 1.0 \\ 1.0 & 0.5 & 1.0 & 1.0 \end{pmatrix}.$$

Definition 4. Let e^+ be a positive example and NE be a set of negative examples. For each $e^- \in NE$, construct a vector $Vector(e^-)=(x_1, x_2, \ldots, x_n)$ as follows:

$$x_j=\begin{cases} |(D(e^+))_j-(D(e^-))_j| & \text{if } (M(e^+))_j=(M(e^-))_j, \\ (D(e^-))_j & \text{if } (M(e^+))_j\neq(M(e^-))_j, \end{cases} \quad j=1,2,\ldots,n,$$

where, for arbitrary vector V, $(V)_j$ denotes the jth component of V. The extension matrix of e^+ under background $M(NE)$ is defined as $\{Vector(e^-)\,|\,e^- \in NE\}$, denoted by $EM(e^+)|_{NE}$, in short, $EM(e^+)$. The positive example e^+ is called a seed for generating the extension matrix.

Example 2. Consider the first row in Table 1 as the positive example e^+ and Table 2 as the negative example set NE. Then, the extension matrix of e^+ under background $M(NE)$ is

$$D(NE)=\begin{pmatrix} 0.1 & 0.4 & 1.0 & 0.4 \\ 0.7 & 0.7 & 0.8 & 0.1 \\ 1.0 & 0.8 & 0.8 & 0.4 \\ 0.6 & 1.0 & 0.7 & 0.1 \\ 0.1 & 0.5 & 1.0 & 0.4 \\ 0.1 & 0.5 & 1.0 & 0.4 \end{pmatrix} \quad \text{where } M(NE)=\begin{pmatrix} sunny & hot & normal & Not\_ \\ cloudy & mild & normal & Not\_ \\ cloudy & cool & normal & Not\_ \\ cloudy & mild & normal & Not\_ \\ sunny & hot & normal & Not\_ \\ sunny & mild & normal & Not\_ \end{pmatrix}.$$

Definition 5. Consider an extension matrix $EM=(L_{ij})$. An element of EM, L_{ij}, is called a dead-element if $L_{ij}\leqslant\alpha$ where α is a threshold called dead-element standard. A path refers to a connection of non-dead-elements of the extension matrix.

Example 3. Consider the extension matrix $DN(E)$ in Example 2. For the dead-element standard $\alpha=0.4$, there will be ten dead-elements which are $L_{11}, L_{51}, L_{61}, L_{12}$ and L_{i4} $(i=1,2,\ldots,6)$. The connection $L_{13}\to L_{21}\to L_{31}\to L_{41}\to L_{53}\to L_{63}$ constitutes a complete path of the extension matrix while $L_{13}\to L_{21}$ an incomplete path (see Fig. 1 where $*$ represents dead-element).

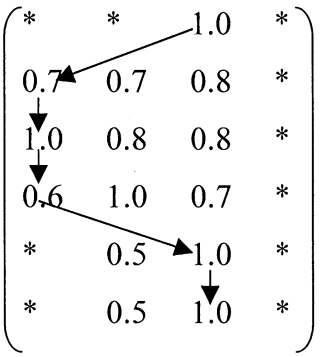

Fig. 1. An extension matrix and its one path.

The main purpose of introducing extension matrix under fuzzy environment is to establish the relation between a path of extension matrix and a fuzzy rule. We have the following proposition.

Proposition 1. *Each path of an extension matrix* (*defined in Definition* 5) *corresponds to a fuzzy rule* (*denoted as* $\bigwedge_{j\in J}[A_j \neq P_j]$). *Each proposition* $[A_j \neq P_j]$ *in the fuzzy rule* $\bigwedge_{j\in J}[A_j \neq P_j]$ *corresponds to one or several elements of the path.*

Example 4. Consider the path $L_{13} \to L_{21} \to L_{31} \to L_{41} \to L_{53} \to L_{63}$ in Example 3 (see Fig. 1). Noting the background $M(NE)$ in Example 2, one can verify that the fuzzy rule corresponding to this path is $[Outlook \neq Cloudy] \wedge [Humidity \neq Normal]$.

In the following we do not differentiate a fuzzy rule from a path corresponding the fuzzy rule.

Rule extraction is the most important purpose of learning from examples. Suppose that a set of examples with crisp representation is classified into two classes (positive and negative class) and this set of examples is non-noisy, one may learn a set of rules which covers the positive examples and exclude the negative examples exactly. The approaches to learning are various, such as, decision trees, genetic algorithms, neural networks, especially extension matrices discussed here. It is useful to extend some existing traditional learning techniques to their fuzzy versions. So far, many crisp approaches to extracting rules have had their fuzzy versions such as fuzzy ID3 and fuzzy neural networks, but the fuzzy version of extension matrix approach has not been found. In the following, we will develop the fuzzy rule generation algorithm based on extension matrices. Proposition 1 has established the relation between a fuzzy rule and a path of extension matrix, that is, a fuzzy rule represented as $\bigwedge_{j\in J}[A_j \neq P_j]$ corresponds to a path of extension matrix. Therefore, the problem of fuzzy rule extraction can be converted into a search for paths of an extension matrix. That is just one of the most significant purposes of introducing extension matrix under fuzzy environment.

3. Heuristic algorithm of extension matrix for generating fuzzy rules

3.1. The fuzzy entropy of a path

Definition 6. Let PE be a set of positive examples, NE be a set of negative examples, $e^+ \in PE$, $EM(e^+)$ be the extension matrix of e^+ under background $M(NE)$, $PATH$ be a path of the extension matrix $EM(e^+)$ and the fuzzy rule corresponding to $PATH$ be R. The degrees of $PATH$ covering PE and NE are defined as

$$p = \sum\{\text{degree of } R \text{ covering } e \mid e \in PE\}$$

and

$$n = \sum \{\text{degree of } R \text{ covering } e \mid e \in NE\},$$

respectively. The fuzzy entropy of the path is defined as

$$Entropy(PATH) = -\frac{p}{p+n}\log\frac{p}{p+n} - \frac{n}{p+n}\log\frac{n}{p+n}.$$

The fuzzy entropy of a path of an extension matrix denotes the fuzziness of the path. That is the contrast degree of covering positive examples and negative examples. Consider the path $L_{13} \to L_{21} \to L_{31} \to L_{41} \to L_{53} \to L_{63}$ in Example 4, this path corresponds to the fuzzy rule $[Outlook \neq Cloudy] \wedge [Humidity \neq Normal]$. Noting that the degrees of the path covering PE and NE are

$$p = 0.8 + 0.1 + 0.5 + 0.3 + 0.0 + 0.6 + 0.0 + 0.7 + 0.1 + 0.1 + 1.0 = 4.2$$

and

$$n = 0.0 + 0.2 + 0.0 + 0.3 + 0.0 = 0.5,$$

respectively, one can compute that

$$E(PATH) = -\frac{4.2}{4.7}\log\frac{4.2}{4.7} - \frac{0.5}{4.7}\log\frac{0.5}{4.7} = 0.46.$$

That indicates that the considered path, to a great degree, covers the set of positive examples (the degree of covering each of Examples 1, 3, 6, 8 and 11 is not less than 0.5) and excludes the set of negative examples.

Information-entropy based on probabilistic models (i.e., Shannon entropy) is a well-known concept to describe probabilistic uncertainty (randomness). Subsequently, this concept was extended to describe the possibility of distribution's uncertainty (fuzziness). A typical extension was given in [1]. Here, fuzzy entropy used is based on this extension. That is, the degree with which a rule ($PATH$) covers positive examples (respective negative examples) is regarded as the corresponding possibility. The fuzzy entropy of the possibility distribution $(p/(p+n), n/(p+n))$ represents the fuzziness of the rule ($PATH$), i.e., represents to what degree the considered rule ($PATH$) covers both positive examples and excludes negative examples. It is clear that a fuzzy rule ($PATH$) with small fuzzy entropy is desirable. If the fuzzy entropy is equal to 0, then the rule corresponding to the path will cover a class and excludes the other class exactly, degenerating to the case of learning from crisp examples. In the fuzzy case, it is impossible that the fuzzy entropy decreases to 0. Hence, we desire that the fuzzy entropy is as small as possible. The following heuristic algorithm for generating fuzzy rules is developed just according to the minimum fuzzy entropy criterion.

3.2. Heuristic algorithm of fuzzy extension matrix

Initial state: PE – a set of positive examples, NE – a set of negative examples, $PATH$ – a path of the extension matrix, *Rule_set* – a rule set which initially is set to be empty. Let α, β, γ be three thresholds with the property: $0.5 \leqslant \alpha \leqslant 1$, $0 < \beta < 0.5$, $0 < \gamma < 0.5$. Use CPE to denote a subset of PE with the property that, for each $e \in CPE$, the degree of $PATH$ covering e is not less than α, use $DN(PATH)$ to denote the averaged degree of $PATH$ covering NE, and use γ to denote the dead-element standard.

Step 1: Randomly select a positive example e^+ from PN, generate the extension matrix $EM(e^+)$, and put $PATH = empty$, $NewPATH = empty$, $PE1 = PE$.

Step 2: **IF** all of non-dead elements of $EM(e^+)$ had appeared in $PATH$, then put $PE = PE - CPE$ and goto Step 4 **ELSE FOR** (c_i, one non-dead-element of $EM(e^+)$ which is not used in $PATH$), put $NewPATH = PATH \cup \{\text{non-dead-elements } c_i\}$, compute $Entropy(NewPATH)$

ENDFOR. Select the non-dead-element c_k, which corresponds to minimum $Entropy(NewPATH)$.

Step 3: Put $NewPATH = PATH \cup \{\text{non-dead-elements } c_k\}$, compute $Entropy(PATH)$, $Entropy(NewPATH)$ and $DN(NewPATH)$. **IF** $Entropy(NewPATH) < Entropy(PATH)$ **THEN** replace $PATH$ with $NewPATH$, make a sign (already used) to c_k and other elements being equal to c_k in the corresponding column, goto Step 2; **ELSE** $PE = PE - CPE$.

Step 4: **IF** $DN(NewPATH) < \beta$, then output the fuzzy rule, L, corresponding to $PATH$ and put $Rule\_set = Rule\_set \cup \{L\}$, **ELSE IF** $PE1 \neq PE$ **THEN** go to Step 1.

Step 5: Output *Rule_set* and *PE* (the positive examples which cannot be covered) [**END**].

The key points included within the above heuristic algorithm are as follows. In the process of seeking one $PATH$ of a extension matrix, the fuzzy entropy of the $PATH$ is regarded as a standard for extending the $PATH$ under the condition that the averaged degree with which the final path covers *NE* does not exceed the threshold β. If the fuzzy entropy is decreasing, then select the element with minimum entropy and add it in the $PATH$.

We illustrate that the heuristic of extension matrix with Tables 1 and 2. To facilitate extension matrix generation, the membership degree 0.5 in Tables 1 and 2 is specially treated as follows.

PE:

1	0.9, 0.1, 0.0,	1.0, 0.0, 0.0,	0.8, 0.2,	0.4, 0.6
2	0.0, 0.7, 0.3,	0.8, 0.2, 0.0,	0.1, 0.9,	0.2, 0.8
3	0.0, 0.1, 0.9,	0.7, 0.3, 0.0,	0.5, 0.0,	0.5, 0.0
4	0.0, 0.1, 0.9,	0.7, 0.3, 0.0,	0.0, 0.5,	0.5, 0.0
5	0.0, 0.1, 0.9,	0.7, 0.3, 0.0,	0.5, 0.0,	0.0, 0.5
6	0.0, 0.1, 0.9,	0.7, 0.3, 0.0,	0.0, 0.5,	0.0, 0.5
7	0.0, 0.7, 0.3,	0.0, 0.3, 0.7,	0.7, 0.3,	0.4, 0.6
8	0.0, 0.3, 0.7,	0.0, 0.0, 1.0,	0.0, 1.0,	0.1, 0.9
9	1.0, 0.0, 0.0,	1.0, 0.0, 0.0,	0.6, 0.4,	0.7, 0.3
10	0.9, 0.1, 0.0,	0.0, 0.3, 0.7,	0.0, 1.0,	0.9, 0.1
11	0.7, 0.3, 0.0,	1.0, 0.0, 0.0,	1.0, 0.0,	0.2, 0.8
12	0.9, 0.1, 0.0,	0.2, 0.8, 0.0,	0.1, 0.9,	1.0, 0.0
13	0.0, 0.9, 0.1,	0.0, 0.9, 0.1,	0.1, 0.9,	0.7, 0.3
14	0.0, 0.0, 1.0,	0.0, 0.0, 1.0,	1.0, 0.0,	0.8, 0.2

NE:

1	0.8, 0.2, 0.0,	0.6, 0.4, 0.0,	0.0, 1.0,	0.0, 1.0
2	0.2, 0.7, 0.1,	0.3, 0.7, 0.0,	0.2, 0.8,	0.3, 0.7
3	0.0, 1.0, 0.0,	0.0, 0.2, 0.8,	0.2, 0.8,	0.0, 1.0
4	0.2, 0.6, 0.2,	0.0, 1.0, 0.0,	0.3, 0.7,	0.3, 0.7
5	1.0, 0.0, 0.0,	0.0, 0.5, 0.0,	0.0, 1.0,	0.0, 1.0
6	1.0, 0.0, 0.0,	0.5, 0.0, 0.0,	0.0, 1.0,	0.0, 1.0

Take $\alpha = 0.7$ and $\beta = 0.2$, use FE to denote the fuzzy entropy of the extended path and DN to denote the averaged degree of the path covering the negative example set NE, the computed result (five fuzzy rules) is listed as follows.

```
(1) Seed: the 11th positive example
    PATH: [Humidity≠Normal]  FE=0.294 DN=0.117
    PATH: [Humidity≠Normal]∧[Temperature≠Mild] FE=0.215 DN=0.067
    Final PATH (Final rule): FE=0.141 DN=0.033
      [Humidity≠Normal]∧[Temperature≠Mild]∧[Outlook≠Cloudy]
    Degrees of covering positive examples:
    1 0.80; 3 0.70; 5 0.70; 9 0.60; 11 0.70; 14 1.00;
(2) Seed: the eighth positive example
    PATH: [Temperature≠Hot] FE=0.688 DN=0.767
    PATH: [Temperature≠Hot]∧[Outlook≠Sunny] FE=0.674 DN=0.450
    PATH: [Temperature≠Hot,Mild]∧[Outlook≠Sunny] FE=0.625 DN=0.217
    Final PATH (Final rule): FE=0.490 DN=0.083
      [Temperature≠Hot,Mild]∧[Outlook≠Sunny,Cloudy]
    Degrees of covering positive examples: 8 0.70;
(3) Seed: the fourth positive example
    Final PATH: (Final rule): [Wind≠Not_windy] FE=0.353 DN=0.100
    Degrees of covering positive examples:
    4 1.00; 10 0.90; 12 1.00; 13 0.70;
(4) Seed: the second positive example
    PATH: [Outlook≠Sunny] FE=0.693 DN=0.467
    PATH: [Outlook≠Sunny]∧[Temperature≠Mild] FE=0.660 DN=0.217
    Final PATH (Final rule): FE=0.593 DN=0.117
      [Outlook≠Sunny]∧[Temperature≠Mild,Cool]
    Degrees of covering positive examples: 2 0.80; 6 0.70;
(5) Seed: the seventh positive example
    Final PATH: (Final rule): [Humidity≠Normal] FE=0.693 DN=0.117
    Degrees of covering positive examples: 7 0.70;
```

The main purpose of learning from examples is to extract a set of rules from a set of examples described as positive and negative. This set of rules, in the crisp case, covers all positive examples and excludes all negative examples exactly. In the fuzzy case, both the training examples and the extracted rules are fuzzy, and the extracted rules only cover positive example set and exclude negative example set to some extent. From a set of examples with fuzzy representation such as Tables 1 and 2, a set of fuzzy rules taking the form $\wedge[ATTRIBUTE \neq linguistic\ term(s)]$ can be extracted by using the proposed extension matrix heuristic algorithm. Let the set of extracted fuzzy rules be $\{R_1, R_2, \ldots, R_n\}$ and e be an example to be classified. Define $P(e) = \max_{1 \leqslant i \leqslant n} \{d_i \mid d_i \text{ is the degree of } R_i \text{ covering } e\}$. $P(e)$ is regarded as the degree of e belonging to the category described by positive examples. When crisp decision is needed, a threshold ε should be given. Using the given threshold, the process of making crisp decision can be described as: if $P(e)$ is not less than ε then the novel example e is classified to the positive class, else e is regarded as a negative example.

Using the learned five fuzzy rules, one can test the Positive example set (Table 1) and the Negative example set (Table 2). The matching results are shown in Tables 3 and 4.

It is easy to see from Tables 3 and 4 that the classification accuracy for the two training sets is 100% if the threshold ε is taken within the interval $(0.3, 0.7]$. Generally speaking, the learning accuracy from a set of examples with fuzzy representation does not reach 100%. The learning accuracy depends on the representation of training examples and the selection of learning parameters such as several thresholds. That is verified by the case study in Section 4.

Table 3
The matching result of positive example set

No.	1	2	3	4	5	6	7	8	9	10	11
Matching degree	0.8	0.8	0.7	0.7	0.7	0.7	0.9	1.0	1.0	0.7	1.0

Table 4
The matching result of negative example set

No.	1	2	3	4	5
Matching degree	0.2	0.3	0.2	0.3	0.0

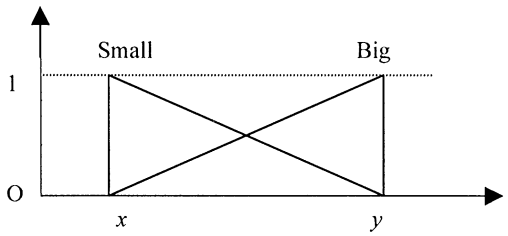

Fig. 2. Two membership functions.

3.3. The capability of tolerating noisy data and missing data

The crisp extension matrix algorithm for generating production rules requires the positive example set to be consistent with the negative example set. It does not permit noisy data appearing in the training set. Our proposed approach incorporates the fuzzy entropy such that the fuzzy extension matrix algorithm, to some extent, has the capability of tolerating noisy or missing data.

We illustrate the capability of tolerating noisy or missing data. Consider a toy problem of learning from examples with two attributes A1 and A2. Assume that the training data contains one positive example $e^+ = (x, y)$ and a set NE with 99 negative examples $e_j^- = (y - \varepsilon_1, y - \varepsilon_2)$ $(j = 1, 2, \ldots, 98)$, and $e_{99}^- = (x + \varepsilon_1, y - \varepsilon_2)$ where the 99th negative example is likely to be a noisy-data, x is supposed to be less than y, and ε_1 or ε_2 is a very small positive perturbation of the value y. These two numerical attributes are fuzzified in terms of two linguistic terms, namely "Small" and "Big" (corresponding to values x and y, respectively), with the membership functions shown in Fig. 2. After fuzzification, the learning problem can be represented as Table 11 (one positive example) and Table 12 (99 negative examples).

The traditional extension matrix algorithm cannot deal with this case. We try to handle the problem using the proposed method. Noting that the perturbations are supposed to be positive and very small, the extension matrix $EM(e+)$ and the negative background $M(NE)$ can be represented as follows:

$$M(NE) = \begin{pmatrix} Big & Big \\ \vdots & \vdots \\ Big & Big \\ Small & Big \end{pmatrix}_{99\times 2} \quad \text{and} \quad EM(e^+) = \begin{pmatrix} 1-\varepsilon_1 & * \\ \vdots & \vdots \\ 1-\varepsilon_1 & * \\ * & * \end{pmatrix}_{99\times 2},$$

Table 5
Rough sketch of rice taste data

No.	Favor	Appearance	Taste	Stickiness	Toughness	Overall evaluation
1	0.532	0.931	1.571	1.600	0.00	1.874
2	0.699	1.534	1.760	1.944	−0.875	1.706
⋮	⋮	⋮	⋮	⋮	⋮	⋮
105	0.163	1.088	1.074	0.683	−0.135	0.913

where $*$ represents dead-element. It is obvious that the searched path is $L_{1,1} \to L_{2,1} \to \cdots \to L_{98,1}$ which corresponds to the production rule $[A1 \neq Big]$. To a great extent, this fuzzy rule covers the positive example and excludes the negative example set (the fuzzy entropy of the path is approximately $\log 2$ and the averaged degree with which the rule covers the set of negative examples is about 0.01).

Worth mentioning is that the missing data can be dealt with in a similar way. Continue to discuss the above toy-example. Assume that the 99th negative example is $(?, y-\varepsilon)$ in which the first value is lost and ε is a positive small perturbation. One way of dealing with the missing value is to consider the missing value to be a "don't care"-value [2,23], which means that the missing value belongs to each linguistic term with membership degree 1. In this way, the 99th negative example is represented as a fuzzy set

$$e_{99}^{-} = 1.0/Small + 1.0/Big + \varepsilon/Small + (1.0-\varepsilon)/Big.$$

Consequently, the extension matrix $EM(e^+)$ and the negative background $M(NE)$ become the following forms, respectively:

$$M(NE) = \begin{bmatrix} Big & Big \\ \vdots & \vdots \\ Big & Big \\ Small & Big \\ Big & Big \end{bmatrix}_{100\times 2} \quad \text{and} \quad EM(e^+) = \begin{bmatrix} 1-\varepsilon_1 & * \\ \vdots & \vdots \\ 1-\varepsilon_1 & * \\ * & * \\ 1-\varepsilon & * \end{bmatrix}_{100\times 2},$$

in which the symbol $*$ represents dead-element. Similar to the case of noisy data, the searched path is $L_{1,1} \to L_{2,1} \to \cdots \to L_{98,1} \to L_{100,1}$ corresponding to the fuzzy rule $[A1 \neq Big]$. The fuzzy entropy of this path is slightly different from that of the path in handling noisy case.

4. Experimental analysis

We select a data-set (called Rice Taste Problem, adopted from [9]) to verify our extension matrix heuristic algorithm. The Rice Taste data set consists of 105 examples with five numerical attributes which are Favor, Appearance, Taste, Stickiness and Toughness, respectively. The classification takes a form of continuous value representing the overall evaluation for the rice. The rough sketch of the data set is illustrated in Table 5. In our study, examples with negative overall evaluation are regarded as positive examples and the others are regarded as negative examples.

4.1. Fuzzifying attribute values

This is a process of fuzzifying numerical numbers into linguistic terms. From many alternatives, we select the fuzzy clustering algorithm. It is based on self-organized learning and can generate some type of membership functions [7].

Let X be the considered data set. We intend to cluster X into k linguistic terms T_j, $j=1,2,\ldots,k$. For simplicity, we assume the shape of membership function to be triangular. These linguistic terms, T_j $(j=1,2,\ldots,k)$, will have triangular membership functions as follows:

$$T_1(x)=\begin{cases}1, & x\leqslant a_1,\\ (a_2-x)/(a_2-a_1), & a_1<x<a_2,\\ 0, & x\geqslant a_2,\end{cases}$$

$$T_k(x)=\begin{cases}1, & x\geqslant a_k,\\ (x-a_{k-1})/(a_k-a_{k-1}), & a_{k-1}<x<a_k,\\ 0, & x\leqslant a_{k-1},\end{cases}$$

$$T_j(x)=\begin{cases}0, & x\geqslant a_{j+1},\\ (a_{j+1}-x)/(a_{j+1}-a_j), & a_j<x<a_{j+1},\\ (x-a_{j-1})/(a_j-a_{j-1}), & a_{j-1}<x<a_j,\\ 0, & x\leqslant a_{j-1},\end{cases}\qquad 1<j<k.$$

Each pair of adjacent membership functions crosses at the membership value 0.5. Only parameters needed to be determined are k centers $\{a_1,a_2,\ldots,a_k\}$. An effective method to determine these centers is Kohonen-feature-maps algorithm [7]. At initial time, k centers are set to be distributed evenly on the range of X. Let

$$A=\{a_1,a_2,\ldots,a_k\},\quad d(X,A)=\sum_{x\in X}\min_i|x-a_i|.$$

The centers will be adjusted iteratively. Each iteration consists of three steps:
(1) randomly take a value x from X, denoted by $x[n]$;
(2) search for an integer m such that $|x[n]-a_m[n]|=\min_j|x[n]-a_j[n]|$;
(3) put $a_m[n+1]=a_m[n]+\alpha(x[n]-a_m[n])$ and keep other centers unchanged,
where n is the iteration time and α is the learning rate.

The iteration ends when $d(X,A)$ converges.

It is easy to see from this clustering algorithm that the sum of memberships is equal to 1 for any value. In fact, it is unnecessary for the set of linguistic terms to form such a partition. If the cross-point between adjacent linguistic terms is not 0.5 or the membership function changes to another form from the triangular shape, the memberships sum fails to be 1 generally.

4.2. Effect of different linguistic terms and different thresholds

For the positive example set and negative example set, around 70% examples are randomly selected as the training sets. Two groups of different linguistic terms shown in Tables 6 and 7 are obtained by using the iterative algorithm described in Section 4.1. The experiment is repeated 12 times and then the averaged values of test targets are shown in Table 8.

Table 6
The first set of linguistic terms

Attribute	No. of terms	Centers of term
Favor	2	−0.91, 0.16
Appearance	2	−0.56, 0.79
Taste	4	−3.07, −1.23, −0.23, 0.60
Stickiness	2	−0.53, 0.86
Toughness	2	−0.43, 0.44

Table 7
The second set of linguistic terms

Attribute	No. of terms	Centers of term
Favor	3	−1.97, −0.70, 0.22
Appearance	2	−0.56, 0.79
Taste	2	−0.80, 0.50
Stickiness	3	−1.15, 0.02, 1.08
Toughness	3	−0.60, 0.03, 1.21

Table 8
Test results of the two set of linguistic terms

	$(\alpha, \beta, \varepsilon)$	No. of rules	Correct rate of *PE* (%)	Error rate of *NE* (%)	Accuracy of the test set (%)
First set	(0.6, 0.2, 0.8)	5.4	95.2	4.5	90.8
Second set	(0.7, 0.2, 0.6)	6.5	97.8	3.4	94.4

Table 9
The effect of thresholds

$(\alpha, \beta, \varepsilon)$	No. of rules	Correct rate of *PE* (%)	Error rate of *NE* (%)	Accuracy of the test set (%)
(0.75, 0.2, 0.5)	6	92.2	2.6	89.2
(0.70, 0.2, 0.6)	7	96.5	3.2	93.1
(0.65, 0.3, 0.7)	6	88.3	2.8	85.5

Table 8 shows us that the learning result including the number of fuzzy rules, the accuracy of testing training sets, the error rate of covering negative examples and the accuracy of novel examples classification depends on the selection of linguistic terms, where α is the threshold of covering positive examples, β is the maximum averaged degree of covering negative example set and ε is the threshold for crisp classification. It is very important but very difficult to select appropriate number of linguistic terms and their centers such that the learning result attains the optimization for a given heuristic.

Different thresholds will result in different learning consequences. Using the second set of terms shown in Table 7, different learning consequences can be obtained by selecting different thresholds α, β and ε. The learning consequences are shown in Table 9.

One problem is how to select parameters $\alpha, \beta, \varepsilon$ such that the performance of our method attains an optimum. That is a difficult problem since the performance of $\alpha, \beta, \varepsilon$ depends strongly on the representation of data sets used in real applications. Usually, the assignment of these parameters is given by domain users according to their requirements. Of course, that does not guarantee an optimum. It is possible that a combination of these parameter values results in a poor performance. In other words, we cannot guarantee that the performance of our method is better than fuzzy ID3 for any values of $\alpha, \beta, \varepsilon$. To obtain an optimum for these parameters, one promising method is to refine them in a connection structure, but the current paper does not discuss this kind of refinement.

Table 10
A comparison with fuzzy-ID3

Algorithms	No. of rules	Correct rate of *PE* (%)	Correct rate of all training examples (%)
Fuzzy extension matrix	6.5	97.8	94.4
Fuzzy-ID3	13.8	91.0	86.2

Table 11
One positive example

No.	A1		A2	
	Small	Big	Small	Big
1	1.0	0.0	1.0	0.0

Table 12
99 negative examples

No.	A1		A2	
	Small	Big	Small	Big
1	ε_1	$1-\varepsilon_1$	ε_2	$1-\varepsilon_2$
2	ε_1	$1-\varepsilon_1$	ε_2	$1-\varepsilon_2$
⋮		⋮		⋮
98	ε_1	$1-\varepsilon_1$	ε_2	$1-\varepsilon_2$
99	$1-\varepsilon_1$	ε_1	$1-\varepsilon_2$	ε_2

4.3. Comparison to fuzzy-ID3

One powerful heuristic for generating crisp decision trees is ID3. The earlier version of ID3, which is based on minimum information entropy to select expanded attributes, was proposed by Quinlan in 1986 [11] and subsequently the revised version called C4.5 was given in [12]. As the increasing uncertainty is incorporated into the knowledge-based system, it is found that using crisp ID3 to represent imprecise knowledge is not enough. The fuzzy version of ID3, based on minimum fuzzy entropy, has been suggested by several authors [6,14,19,15,22]. The learning result of fuzzy-ID3 heuristic is a fuzzy decision tree which can be converted into a set of fuzzy rules. For completing this brief comparison, we do not directly use C4.5 programs in the simple experiment given in Section 4, but use fuzzy ID3 algorithm, since C4.5 can generate only crisp rules.

For the set of positive examples and the set of negative examples, around 70% examples are randomly selected as the training sets. The set of linguistic terms shown in Table 7 is used. Using extension matrix heuristic and fuzzy-ID3 heuristic, respectively, the experiment is repeated 12 times and then the averaged values of test targets are shown in Table 10.

It can be seen from Table 10 that, with regard to the simplicity of fuzzy rules, the training accuracy and the test accuracy, the extension matrix heuristic is superior to the fuzzy-ID3. The training time of extension matrix heuristic is slightly longer than that of fuzzy-ID3, but the training-time difference between the two algorithms is not significant when the training set is not very big. To some extent, this experiment gives

the comparative strengths and weaknesses between the popular fuzzy ID3 and the proposed fuzzy extension matrix heuristic. Incompletely it indicates such a guideline that the proposed heuristic is more suitable for learning problems with high accuracy and without large scale.

5. Future work

An efficient way to develop fuzzy rule generation techniques is to extend existing traditional methodologies of generating crisp rules to their fuzzy version. Although this paper has made an initial attempt to extend an existing technique for rule generation (called extension matrix) to its fuzzy version, the following problems related to fuzzy extension matrices need to be further investigated in the future.

(1) Exploring effectiveness of the current method to solve learning from examples with large scale.
(2) Addressing fuzziness of the output variable (classification attribute), i.e., addressing whether a further generalization of matrix is made to get rid of the restriction of crisp concepts positive and negative classes.
(3) Making more experiments to further check to what degree the current method is robust against noisy data and missing data, and to further compare this method with some existing fuzzy generation methods to give their comparative strengths and weaknesses.
(4) Investigating how to select some parameters (such as $\alpha, \beta, \varepsilon$ used in the paper) such that the performance of our method attains an optimum. An attempt to refine these parameters in a connection structure is being made.

6. Conclusions

This paper deals with the fuzzy rule generation from a set of examples with fuzzy representation by using extension matrix. A new heuristic algorithm for generating fuzzy rules, which is based on extension matrix and incorporates the concept of fuzzy entropy, is developed. Compared with the crisp extension matrix, the proposed method has the capability of handling fuzzy representation and tolerating noisy or missing data. The proposed heuristic algorithm partially inherits the advantages from the crisp case such as simplicity of rules and high learning accuracy.

References

[1] A. De Luca, S. Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy set theory, Inform. Control 20 (1972) 301–312.
[2] R. Gopalan, M.G. Nair, Discovering fuzzy association rules in large databases, Proc. 9th Australian Database Conference, Perth, 2–3 February 1998, pp. 165–176.
[3] J.R. Hong, AE1: extension matrix approximate method for general covering problem, International J. Comput. Inform. Sci. 14 (6) (1985) 421–437.
[4] J.R. Hong, AE5: multi-purpose system of learning from examples, J. Comput. 12 (2) (1989) 98–105 (in Chinese).
[5] J.R. Hong, Extension matrix theory of learning from examples, J. Comput. 14 (2) (1991) 401–410 (in Chinese).
[6] H. Ichihashi, T. Shirai, K. Nagasaka, T. Miyoshi, Neuro-fuzzy ID3, Fuzzy Sets and Systems 81 (1996) 157–167.
[7] T. Kohonen, Self-Organization and Associate Memory, Springer, Berlin, 1988.
[8] R.S. Michalski et al., The multi-purpose incremental learning systems AQ15 and its application to three medical domain, Proc. 5th Natl. Conf. on Artificial Intelligence, Morgan Kaufman, Philadelphia, PA, 1986, pp. 1041–1045.
[9] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets and Systems 86 (1997) 251–270.
[10] G.L. Qian, W.H. Shu, B. Chen, G.R. Quan, Research on a heuristic algorithm of feature subset selection based on entropy, J. Software 9 (1998) 911–916 (in Chinese).

[11] J.R. Quinlan, Induction of decision trees, Mach. Learning 1 (1986) 81–106.
[12] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman, Mateo, CA, 1993.
[13] L.X. Sun, Y.M. Zhang, Acquiring hyperspectral remotely sensed images classification rules using inductive learning, Proc. SPIE – The International Society for Optical Engineering, SPIE, Bellingham, WA, USA, 1998, pp. 164–168.
[14] T. Tani, M. Sakoda, Fuzzy modeling by ID3 algorithm and its application to prediction of outlet temperature, Proc. IEEE Internat. Conf. on Fuzzy Systems, San Diego, CA, 8–12 March 1992, pp. 923–930.
[15] X.Z. Wang, J.R. Hong, On the handling of fuzziness for continuous-valued attributes in decision tree generation, Fuzzy Sets and Systems 99 (1998) 283–290.
[16] X.Z. Wang, J.R. Hong, Learning optimization in simplifying fuzzy rules, Fuzzy Sets and Systems 106 (1999) 349–356.
[17] X.Z. Wang et al., On the optimization of fuzzy decision trees, Fuzzy Sets and Systems 112 (2000) 117–125.
[18] X.Z. Wang, E.C.C. Tsang, D.S. Yeung, A problem of selecting optimal subset of fuzzy-valued features, Proc. IEEE Internat. Conf. on System, Man, and Cybernetics, October 12–15, 1999, pp. 361–366.
[19] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan, 17–22 July 1992, pp. 265–268.
[20] X.D. Wu, HCV induction algorithm, Proc. – ACM Comput. Sci. Conference ACM, New York, NY, USA, 1993, pp. 168–175.
[21] X.D. Wu, J. Krisar, P. Mahlen, Noise handling with extension matrices, Proc. Internat. Conf. on Tools with Artificial Intelligence, IEEE, Piscataway, NJ, USA, 1995, pp. 190–197.
[22] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.
[23] Y. Yuan, H. Zhuang, A genetic algorithm for generating fuzzy classification rules, Fuzzy Sets and Systems 84 (1996) 1–19.
[24] C.H. Zhu, F.L. Xiong, System of learning from examples for knowledge acquisition, Proc. IEEE Internat. Conf. on Expert Systems for Development, IEEE Computer Society Press, Los Alamitos, CA, USA, 1994, pp. 273–276.