

A fuzzy inductive learning strategy for modular rules

Ching-Hung Wang^{a,c}, Jau-Fu Liu^a, Tzung-Pei Hong^{b,*}, Shian-Shyong Tseng^a

^a *Institute of Computer and Information Science, National Chiao-Tung University, Hsin-Chu, 30050, Taiwan, ROC*

^b *Department of Information Management, Kaohsiung Polytechnic Institute, Kaohsiung, 84008, Taiwan, ROC*

^c *Chunghwa Telecommunication Laboratories, Ministry of Transportation and Communication, Chung-Li, 32617, Taiwan, ROC*

Received December 1996; received in revised form April 1997

Abstract

In real applications, data provided to a learning system usually contain linguistic information which greatly influences concept descriptions derived by conventional inductive learning methods. The design of learning methods to learn concept descriptions in working with vague data is thus very important. In this paper, we apply fuzzy set concepts to machine learning to solve this problem. A fuzzy learning algorithm based on the *maximum information gain* is proposed to manage linguistic information. The proposed learning algorithm generates fuzzy rules from “soft” instances, which differ from conventional instances in that they have class membership values. Experiments on the Sports and the Iris Flower classification problems are presented to compare the accuracy of the proposed algorithm with those of some other learning algorithms. Experimental results show that the rules derived from our approach are simpler and yield higher accuracy than those from some other learning algorithms. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Expert systems; Fuzzy machine learning; Fuzzy sets; Knowledge acquisition; Measure of fuzziness; Membership functions

1. Introduction

Various learning methods have been developed for inducing rules from collections of examples [6, 8, 17–21]. Among these learning approaches, *inductive learning* may be the most commonly used in real-world application domains. Inductive learning is basically a process of inferring concept descriptions that include positive instances and exclude negative instances. Traditional inductive learning methods are however inapplicable to some application domains, since data in the real world usually contain vagueness and ambiguity.

Vagueness and ambiguity most commonly result from attributes insufficient to appropriately describe objects, or when experts, teachers, or users are not quite sure what classes given objects belong to. The boundaries of a piece of information used may not be clear-cut, and each object may be expressed as

* Corresponding author. E-mail: tphong@nas05.kpi.edu.tw.

a linguistic “input–output” relation. Each attribute that describes an object could thus be defined as a fuzzy set. As an example, the object *dangerous dogs* may be expressed as “*Dog A* has a **large body** and **long hairs**, and it is **dangerous** with 0.8 degree of certainty”. “**Large**”, “**long**”, and “**dangerous**” are fuzzy linguistic terms. Since attributes and classifications used to express objects represent human perceptions and desires, they are vague by nature. A **crisp classification** that distinguishes between positive and negative instances is often artificial; instead, fuzzy or ambiguous classifications of instances are commonly seen in the real world.

Vagueness in general greatly influences concept descriptions derived by conventional inductive learning methods [27]. Some kinds of learning problems arising in working with vague data were discussed in [2, 3, 5, 10, 12, 23, 24]. The design of learning methods to work well with vague data is thus very important. Several successful learning strategies based on ID3 have been proposed [7, 15, 20–22, 25, 27]; most of these use tree-pruning and fuzzy logic techniques. As for version-space-based learning strategies, Wang et al. proposed a fuzzy version space learning strategy for managing vague information [23]. In this paper, we propose a fuzzy inductive learning algorithm (FIL) based on *maximum fuzzy information gain* to induce a set of fuzzy modular rules from “soft” training instances, which differ from conventional instances in that they have class membership values [15, 24, 25]. This learning approach can solve some problems of inductive learning in vague learning environments.

The remainder of this paper is organized as follows. Some related concepts and terms are reviewed in Section 2. A generalized inductive learning is introduced in Section 3. A fuzzy inductive learning algorithm (FIL) is proposed in Section 4. An example illustrating the learning process of the proposed algorithm is described in Section 5. The time complexity of the proposed learning algorithm is analyzed in Section 6. Experimental results from the Sports and the IRIS flower classification problems are reported in Section 7. Finally, conclusions are given in Section 8.

2. Review of related concepts and terms

In this section, we review concepts and terms important to this paper.

2.1. Fuzzy set

A fuzzy set is an extension of a crisp set. Crisp sets allow only full membership or no membership at all, whereas fuzzy sets allow partial membership. In other words, an element may belong to more than one set. In a crisp set, the membership or non-membership of an element x in set A is described by a characteristic function $u_A(x)$, where

$$u_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

Fuzzy set theory extends this concept by defining partial membership, which can take values ranging from 0 to 1:

$$u_A: X \rightarrow [0, 1],$$

where X refers to the universal set defined for a specific problem.

Assuming that A and B are two fuzzy sets with membership functions of $u_A(x)$ and $u_B(x)$, then the following fuzzy operators can be defined.

(1) The *intersection* operator:

$$u_{A \cap B}(x) = u_A(x) \tau u_B(x),$$

where $\tau: [0, 1] * [0, 1] \rightarrow [0, 1]$ is a *t-norm* operator satisfying the following conditions [26]:
for each $a, b, c \in [0, 1]$:

- (i) $a \tau 1 = a$;
- (ii) $a \tau b = b \tau a$;
- (iii) $a \tau b \geq c \tau d$ if $a \geq c, b \geq d$;
- (iv) $a \tau b \tau c = a \tau (b \tau c) = (a \tau b) \tau c$.

Some instances of a *t-norm* operator $a \tau b$ are $\min(a, b)$ and $a * b$.

- (2) The *union* operator:

$$u_{A \cup B}(x) = u_A(x) \rho u_B(x),$$

where $\rho: [0, 1] * [0, 1] \rightarrow [0, 1]$ is an *s-norm* operator satisfying the following conditions [26]:
for each $a, b, c \in [0, 1]$:

- (i) $a \rho 0 = a$;
- (ii) $a \rho b = b \rho a$;
- (iii) $a \rho b \geq c \rho d$ if $a \geq c, b \geq d$;
- (iv) $a \rho b \rho c = a \rho (b \rho c) = (a \rho b) \rho c$.

Some instances of an *s-norm* operator $a \rho b$ are $\max(a, b)$ and $a + b - a * b$.

- (3) The α -*cut* operator:

$$A_\alpha(x) = \{x \in X \mid u_A(x) \geq \alpha\},$$

where A_α is an α -*cut* of a fuzzy set A . A_α contains all elements in the universal set X that have a membership grade in A greater than or equal to the specified value of α .

These fuzzy operators will be used in our learning algorithm to derive fuzzy *if-then* rules.

2.2. Inductive learning

Conventional inductive learning is aimed at finding a concept description R that correctly describes all instances in the training set [6, 8, 17–21]. If E is a training set divided into two subsets: P (the set of positive instances) and N (the set of negative instances), then conventional inductive learning attempts to find a concept description R such that the following conditions are met:

$$\forall e^+ \in P \Rightarrow e^+ \subset R \quad \text{and} \quad \forall e^- \in N \Rightarrow e^- \not\subset R,$$

where e^+ is a positive instance and e^- is a negative one, \subset and $\not\subset$ are relationship descriptors that mean “covered by” and “not covered by”, respectively. The concepts derived from traditional inductive learning methods are usually represented in the following grammar:

If $\langle \text{cover} \rangle$ then predict $\langle \text{class} \rangle$, where
 $\langle \text{cover} \rangle = \langle \text{complex}_1 \rangle$ or ... or $\langle \text{complex}_x \rangle$,
 $\langle \text{complex} \rangle = \langle \text{selector}_1 \rangle$ and ... and $\langle \text{selector}_y \rangle$,
 $\langle \text{selector} \rangle = \langle \text{attribute relationship value} \rangle$.

A *selector* relates a variable to a value. For example, “color = red”, “height = tall”, and “weight > 60 kg” are all *selectors*. A conjunction of *selectors* forms a *complex*. A *cover* is a disjunction of *complexes* describing all positive instances and no negative instances of the concept.

Generally, conventional inductive learning methods only work well in ideal domains that contain no vague data. In order to handle linguistic information, the conventional inductive learning must be generalized.

3. Generalized inductive learning

Since data in real-world applications usually contain linguistic information, conventional inductive learning procedures may be inapplicable to some application domains. Fuzzy concepts can then be applied to such conventional inductive learning approaches. The generalized inductive learning task is one of finding a concept description \tilde{R} such that the following conditions are met:

$$\forall \tilde{e} \in_{\alpha} \tilde{P} \Rightarrow \tilde{e} \tilde{\subset}_{\alpha} \tilde{R} \quad \text{and} \quad \forall \tilde{e} \in_{\alpha} \tilde{N} \Rightarrow \tilde{e} \tilde{\not\subset}_{\alpha} \tilde{R},$$

where \tilde{R} is a fuzzy concept description, \forall is a linguistic quantifier of type “almost all”, “most”, etc. [16], \tilde{P} denotes a set of “soft” positive instances and \tilde{N} denotes a set of “soft” negative instances, $\tilde{\subset}_{\alpha}$ and $\tilde{\not\subset}_{\alpha}$ are fuzzy relationship descriptors that mean “ α -covered by” and “ α -not covered by”, respectively. Each instance \tilde{e} can be considered a *soft instance*. *Soft instances* differ from conventional instances in that they have class membership values. The membership value $u_{\tilde{P}}(\tilde{e})$ specifies the degree to which instance \tilde{e} belongs to the positive class \tilde{P} , and the membership value $u_{\tilde{N}}(\tilde{e})$ specifies the degree to which instance \tilde{e} belongs to the negative class \tilde{N} . When the value of $u_{\tilde{P}}(\tilde{e})$ is greater than or equal to a predefined significant level α , instance \tilde{e} is then said to α -belong to the class \tilde{P} ($\tilde{e} \in_{\alpha} \tilde{P}$). The inductive learning is thus generalized to find a concept description, \tilde{R} , that includes almost all “soft” positive instances and excludes almost all “soft” negative instances.

A “soft” training instance is represented here by selectors with a class membership value. Each selector is represented as $[A r 'v']$, where A is an attribute, r is a crisp or fuzzy relationship, and v is a crisp or fuzzy value. An example of a “soft” training instance is shown below:

\tilde{e} : [height = 190 cm] and [weight = 80 kg], he is a basketball player,

with class membership value $u_{\text{basketball_player}}(\tilde{e}) = 0.8$,

where both [height = 190 cm] and [weight = 80 kg] are *crisp* selectors, and $u_{\text{basketball_player}}(\tilde{e})$ is a class membership value that specifies the degree to which \tilde{e} belongs to the class *basketball_player*.

The selectors used to describe derived concepts may, however, be different from those used to describe training instances since some derived concept selectors may be expressed in fuzzy terms. For example, a fuzzy concept may be represented as

IF [height = “tall”] and [weight = “heavy”] THEN he is a basketball_player,

with membership value $u = 0.8$,

where [height = “tall”] and [weight = “heavy”] are *fuzzy* selectors, and u represents the strength of the rule.

Selectors used in the *instance space* must therefore be transformed into representations in the *hypothesis space* for fuzzy matching. Let $u_{s_i}(\tilde{e})$ represent the degree of matching between selector s_i in the hypothesis space and the corresponding selector in the instance \tilde{e} . The value of $u_{s_i}(\tilde{e})$ ranges between 0 and 1, and is used to represent the degree to which instance \tilde{e} is covered by s_i ; 0 indicates complete exclusion and 1 indicates complete inclusion. When the value of $u_{s_i}(\tilde{e})$ is greater than or equal to a predefined significant level α , selector s_i is said to α -cover instance \tilde{e} .

Assume that we have an instance \tilde{e} and a complex $C_j = s_{j_1} \wedge s_{j_2} \wedge \dots \wedge s_{j_m}$. The degree of instance \tilde{e} covered by complex C_j is evaluated as

$$u_{C_j}(\tilde{e}) = u_{s_{j_1}}(\tilde{e}) \wedge u_{s_{j_2}}(\tilde{e}) \wedge \dots \wedge u_{s_{j_m}}(\tilde{e})$$

or, more generally,

$$u_{C_j}(\tilde{e}) = u_{s_{j_1}}(\tilde{e}) \tau u_{s_{j_2}}(\tilde{e}) \tau \dots \tau u_{s_{j_m}}(\tilde{e}),$$

where τ is a *t-norm* operator.

The value of $u_{C_j}(\tilde{e})$ is thus used to represent the fuzzy degree of instance \tilde{e} covered by complex C_j . When the value of $u_{C_j}(\tilde{e})$ is greater than or equal to a predefined significant level α , complex C_j is then said to α -cover instance \tilde{e} .

The concept description \tilde{R} indicates the disjunction of complexes, say, C_1, C_2, \dots, C_x , and is denoted as $\tilde{R} = C_1 \vee C_2 \cdots \vee C_x$. The degree of instance \tilde{e} covered by the concept description \tilde{R} is thus evaluated as

$$u_{\tilde{R}}(\tilde{e}) = u_{C_1}(\tilde{e}) \vee u_{C_2}(\tilde{e}) \vee \cdots \vee u_{C_x}(\tilde{e})$$

or, more generally,

$$u_{\tilde{R}}(\tilde{e}) = u_{C_1}(\tilde{e}) \rho u_{C_2}(\tilde{e}) \rho \cdots \rho u_{C_x}(\tilde{e}),$$

where ρ is an s -norm operator.

The concept of fuzzy matching is used in the following proposed learning algorithm to handle vagueness.

4. A fuzzy inductive strategy for learning modular rules

Various fuzzy inductive learning strategies have been developed for handling noise, uncertainty and vagueness [16, 22, 23, 24, 27]. Among these learning methods, fuzzy decision trees [22, 25, 27] and their various learning approaches are commonly used in real-world applications. They are concerned with finding the most relevant overall attributes. The heuristics of minimizing “entropy” is used to determine which attribute should be selected next in the decision tree. However, this may cause derived rules to be too specific, entailing irrelevant tests in the conditions of rules.

Here, a fuzzy inductive learning algorithm based on the PRISM learning strategy [6] is proposed to handle vagueness and eliminate irrelevant tests occurring in the rule. The proposed learning algorithm maximizes fuzzy information gain instead of minimizing entropy in inducing modular rules [15, 24, 25]. It concentrates on finding relevant attribute–value pairs, rather than just attributes. During induction, the actual amount of information contributed by each attribute–value pair (selector) is evaluated for a specific classification, and the one with the maximum fuzzy information gain is then selected and added to the induced rule. Each attribute–value pair (selector) can be thought of as a message, and the classification δ_k can be similarly thought of as an initial event. Given a message s_i , the amount of fuzzy information gain about an event δ_k is defined as

$$I(\delta_k|s_i) = \log_2 \left(\frac{H(\delta_k|s_i)}{H(\delta_k)} \right) = \log_2 (H(\delta_k|s_i)) - \log_2 (H(\delta_k)),$$

where $H(\delta_k|s_i)$ and $H(\delta_k)$ are, respectively, the subsequent and antecedent fuzzy information, and are defined as follows:

$$H(\delta_k|s_i) = \frac{\sum_{j=1}^n u_{\delta_k}(e_j) \tau u_{s_i}(e_j)}{\sum_{j=1}^n u_{s_i}(e_j)}, \quad H(\delta_k) = \frac{1}{n} \sum_{j=1}^n u_{\delta_k}(e_j),$$

where n is the size of the training set, e_j is the j th instance in the training set, and $u_{\delta_k}(e_j)$ is a class membership value specifying the degree to which instance e_j belongs to event δ_k .

The proposed fuzzy learning algorithm then uses the fuzzy information gain function to determine which selector is chosen and when the specialization process is performed. The specialization operation decides whether a chosen selector should be added to the condition part of an induced rule. Since fuzzy information exists in the training set, the induced rule will not be absolutely true, but will be only partially true. The fuzzy strength of an induced rule “IF \tilde{C} THEN δ_k ” can thus be determined using the fuzzy Bayes measurement

function, $B(\delta_k|\tilde{C})$, which is defined as follows [27]:

$$B(\delta_k|\tilde{C}) = \frac{\sum_{j=1}^n u_{\tilde{C}}(e_j) \tau u_{\delta_k}(e_j)}{\sum_{j=1}^n u_{\tilde{C}}(e_j)},$$

where \tilde{C} and δ_k are, respectively, the condition and conclusion of the induced rule.

When the accuracy of the induced rule is above a predefined truth level β , i.e. $B(\delta_k|\tilde{C}) \geq \beta$, then the specialization process is terminated indicating that an induced rule has been obtained. After a rule has been induced, instances α -covered by this rule are removed from the original training set. The above operations are then repeated until all instances α -belonging to class δ_k have been removed. The fuzzy learning algorithm is stated below.

Fuzzy inductive algorithm for learning modular rules

IF the training set contains instances of more than one classification, then for each classification, δ_k , in turn:

STEP 1. Initiate a null complex \tilde{C} .

STEP 2. Measure the fuzzy information gain, $I(\delta_k|s_i)$, of the classification δ_k for each possible selector s_i from the training set.

STEP 3. Choose a selector s_i for which $I(\delta_k|s_i)$ is maximum.

STEP 4. Add selector s_i to \tilde{C} , i.e., $\tilde{C} = \tilde{C} \wedge s_i$, and calculate $B(\delta_k|\tilde{C})$.

STEP 5. If $B(\delta_k|\tilde{C})$ is above a predefined truth level β , then execute STEP 6; otherwise, create a new training set in which each instance is α -covered by the selector s_i , and go to STEP 2.

STEP 6. Form the rule "IF \tilde{C} THEN δ_k ".

STEP 7. Remove all instances α -covered by the rule "IF \tilde{C} THEN δ_k " from the original training set.

STEP 8. Repeat STEP 1 to STEP 7 until all instances α -belonging to class δ_k in the original training set have been removed.

When the rules for one classification have been induced, the training set is restored to its initial state and the algorithm is applied again to induce a set of rules covering the next classification. Below, an example is used to illustrate this learning process.

5. An example

This is a simple domain for deciding what sport to play according to Sunday's weather. A small training set with fuzzy membership values is shown in Table 1 [27]. Each instance is described by four fuzzy attributes (*Outlook*, *Temperature*, *Humidity*, *Wind*) and one fuzzy classification (*Sport*). Each attribute has the values shown below.

$$\begin{aligned} \text{Outlook} &= \{\text{Sunny}, \text{Cloudy}, \text{Rain}\}, & \text{Humidity} &= \{\text{Humid}, \text{Normal}\}, \\ \text{Temperature} &= \{\text{Cool}, \text{Mild}, \text{Hot}\}, & \text{Wind} &= \{\text{Windy}, \text{Not_windy}\}. \end{aligned}$$

Classifications include the following sports:

$$\text{Sports} = \{\text{Swimming}, \text{Volleyball}, \text{Weight_lifting}\}.$$

The proposed learning procedure for inducing rules of the classification *Swimming* is demonstrated first. Let α be 0.5 and β be 0.7. The information gain of each selector for classification *Swimming* is calculated from Table 1 and is shown in Table 2.

Table 1
Set of training instances for sports to play depending on Sunday's weather

Case	Outlook			Temperature			Humidity		Wind		Sports		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not-windy	Volleyball	Swimming	W-lifting
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
3	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

Table 2
The information gain of each selector for classification *Swimming*

Outlook			Humidity		Temperature			Wind	
Sunny	Cloudy	Rain	Humid	Normal	Hot	Mild	Cool	Windy	Not-windy
1.093	-0.0995	-0.819	0.58	0	1.1593	0	-2.09	0.093	0.45

Among these selectors, selector “*Temperature is Hot*” has the maximum fuzzy information gain evaluated as follows:

$$\begin{aligned}
 &I(\text{Swimming}|\text{Hot}) \\
 &= \log_2 \left(\frac{(1.0+0.6+0.8+0.3+0.7+1.0+1.0+0.2+0.5)/(0.8+0.6+0.6+0.1+0.8+0.7+0.5)}{(0.8+0.7+0.6+0.1+0.8+0.3+0.7+0.2+0.6)/16} \right) \\
 &= \log_2 \left(\frac{0.67}{0.3} \right) = 1.1593.
 \end{aligned}$$

The selector “*Temperature is Hot*” is added to the induced rule, and the rule “IF *Temperature is Hot* THEN *Swimming*” is thus generated. The fuzzy strength of this rule is then calculated as follows:

$$B(\text{Swimming}|\text{Hot}) = \frac{0.8+0.6+0.6+0.1+0.8+0.7+0.5}{1.0+0.6+0.8+0.3+0.7+1.0+1.0+0.2+0.5} = \frac{4.1}{6.1} = 0.67.$$

Since the rule strength, $B(\text{Swimming}|\text{Hot}) = 0.67$, is lower than the predefined truth level β (0.7), the specialization process by repeating STEP 2–4 is then performed. Also, a new training set which is α -covered by the selector “Temperature is Hot” is generated and shown in Table 3.

Information gains for other selectors except the selector “Temperature is Hot” are then re-calculated from Table 3. The selector “Outlook is Cloudy” then has the maximum information gain evaluated as follows:

$$I(\text{Swimming}_{\text{Hot}}|\text{Cloudy}) = \log_2 \left(\frac{(0.1+0.2+0.6+0.3)/(0.1+0.2+0.7+0.1+0.3)}{(0.8+0.7+0.6+0.8+0.7+0.6)/7} \right) = \log_2 \left(\frac{0.86}{0.6} \right) = 0.519.$$

The selector “Outlook is Cloudy” is added to the induced rule, and the rule “IF Temperature is Hot and Outlook is Cloudy, THEN Swimming” is thus generated. The fuzzy strength of this rule is calculated as follows:

$$B(\text{Swimming}|\text{Hot} \wedge \text{Cloudy}) = \frac{0.1+0.2+0.6+0.1+0.3}{0.1+0.2+0.7+0.3+0.1+0.3+0.1} = \frac{1.3}{1.8} = 0.72.$$

Table 3
New training set α -covered by selector “Temperature is Hot”

Case	Outlook			Temperature			Humidity		Wind		Sports		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not-windy	Volleyball	Swimming	W-lifting
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
3	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

Table 4
New set of instances

Case	Outlook			Temperature			Humidity		Wind		Sports		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not-windy	Volleyball	Swimming	W-lifting
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

Table 5
New set of instances

Case	Outlook			Temperature			Humidity		Wind		Sports		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not-windy	Volleyball	Swimming	W-lifting
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0

- Rule 1: IF *Temperature* is *Hot*, *Outlook* is *Cloudy*, THEN *Swimming* ($B=0.72$).
 Rule 2: IF *Temperature* is *Hot*, *Outlook* is *Sunny*, THEN *Swimming* ($B=0.85$).
 Rule 3: IF *Temperature* is *Mild*, *Wind* is *Not-windy*, THEN *Volleyball* ($B=0.78$).
 Rule 4: IF *Outlook* is *Rain*, THEN *Weight-lifting* ($B=0.89$).
 Rule 5: IF *Temperature* is *Cool*, THEN *Weight-lifting* ($B=0.88$).
 Rule 6: IF *Wind* is *Windy*, THEN *Weight-lifting* ($B=0.71$).

Fig. 1. Six rules induced by the proposed fuzzy learning algorithm for the sport domain.

Since the rule strength (0.72) is greater than the predefined truth level β (0.7), the specialization process is terminated and the modular rule “IF *Temperature* is *Hot* and *Outlook* is *Cloudy*, THEN *Swimming*” is output. All instances α -covered by this rule are then removed from Table 1 to form the new set shown in Table 4.

Since cases 1, 2, 9, 11, 16 in Table 4 still α -belong to the classification *Swimming*, the learning procedure must generate a new rule to cover these instances. The information gain of each selector for classification *Swimming* is calculated from Table 4. Among these selectors, selector “*Temperature* is *Hot*” has the maximum information gain (1.237), and is added to the new induced rule. The rule “IF *Temperature* is *Hot* THEN *Swimming*” is then generated. Since the fuzzy rule strength, $B(\text{Swimming}|\text{Hot}) = 0.67$, is lower than the predefined truth level β (0.7), the specialization process is then performed. Finally, a new modular rule “IF *Temperature* is *Hot* and *Outlook* is *Sunny*, THEN *Swimming*” is obtained. All instances α -covered by this rule are then removed from Table 4 to form the new training set shown in Table 5. In Table 5, there are no instances α -belonging to the classification “*Swimming*”. The learning process for classification “*Swimming*” is thus finished. The same procedure is then performed to induce rules for the other classes.

The six rules induced by the proposed fuzzy learning algorithm are listed in Fig. 1.

6. Time complexity analysis

The time complexity of the proposed learning algorithm is analyzed in this section using the following notation.

1. s = the number of selectors;
2. c = the number of classes;

Table 6
Time complexity of the proposed learning algorithm

Step no.	Time complexity
Step 1	$O(1)$
Step 2	$O(s)*O(n)$
Step 3	$O(s)$
Step 4	$(O(1) + O(n))$
Step 5	$t*(O(n) + (\text{STEP 2-4}))$
Step 6	$O(1)$
Step 7	$O(n)$
Step 8	$r*(\text{STEP 1-7})$

3. n = the size of training set;
4. t = the maximum number of selectors in an induced rule;
5. r = the maximum number of induced rules for one classification.

The processing time of the proposed algorithm includes evaluating the information gain and the specialization process. Let $T(n)$ denote the time complexity of the proposed fuzzy learning algorithm in dealing with n training instances. The time complexity of each step is listed in Table 6.

We have

$$\begin{aligned}
 T(n) &= c*[O(1) + O(s)*O(n) + O(s) + (O(1) + O(n)) + t*(O(n) + (\text{Step 2-Step 4})) \\
 &\quad + O(1) + O(n) + r*(\text{STEP 1-7})] \\
 &= c*[O(1) + O(s)*O(n) + O(s) + (O(1) + O(n)) + t*(O(n) + O(1) + O(s*n) + O(s) + O(n)) \\
 &\quad + O(1) + O(n) + r*(\text{STEP 1-7})] \\
 &= c*[O(1) + O(s*n) + O(s) + O(n) + O(t) + O(t*s*n) + O(t*s) + O(t*n) + r*(\text{STEP 1-7})] \\
 &= c*[O(1) + O(s*n) + O(s) + O(n) + O(t) + O(t*s*n) + O(t*s) + O(t*n) + O(r) + O(r*s*n) \\
 &\quad + O(r*s) + O(r*n) + O(r*t) + O(r*t*s*n) + O(r*t*s) + O(r*t*n)] \\
 &= O(c*r*t*s*n).
 \end{aligned}$$

7. Experiments

To demonstrate the effectiveness of the proposed fuzzy inductive learning algorithm, we applied it to two application domains. One decides what sport to play according to the Sunday's weather, using the instances described in Table 1 [27]. The other one classifies Fisher's Iris data, which contain 150 training instances [11]. The fuzzy learning algorithm was implemented in C language on a SUN SPARC/20 workstation and run 100 times. The accuracy of the proposed method was compared with those of other learning algorithms on the same application domains to demonstrate performance. The experiments are described below.

7.1. The sport domain

Due to its simplicity, the sport classification problem is easily used to test and interpret the performance of the proposed approach. In this experiment, two induction methods were run on this problem: our proposed approach, and the Yuan and Shaw approach [27]. The Yuan and Shaw approach constructs a decision tree

based on fuzzy entropy. The decision tree generated by the Yuan and Shaw approach for this problem domain is shown in Fig. 2 [27].

The corresponding rules were:

- Rule a: IF *Temperature* is *Mild*, *Wind* is *Not-windy*, THEN *Volleyball* ($B = 0.78$).
- Rule b: IF *Temperature* is *Hot*, *Outlook* is *Cloudy*, THEN *Swimming* ($B = 0.72$).
- Rule c: IF *Temperature* is *Hot*, *Outlook* is *Sunny*, THEN *Swimming* ($B = 0.85$).
- Rule d: IF *Temperature* is *Hot*, *Outlook* is *Rain*, THEN *Weight-lifting* ($B = 0.73$).
- Rule e: IF *Temperature* is *Cool*, THEN *Weight-lifting* ($B = 0.88$).
- Rule f: IF *Temperature* is *Mild*, *Wind* is *Windy*, THEN *Weight-lifting* ($B = 0.81$).

With the derived six classification rules, the classification accuracy for the training data is shown in Table 7. The classification for a given object is obtained using the following steps:

1. For each rule, calculate the membership of the condition for the object based on its attributes. The conclusion membership (the classification to a class) is set equal to the condition membership.

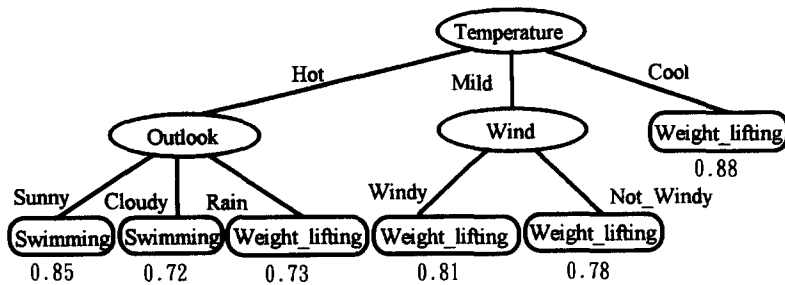


Fig. 2. Decision tree derived by the Yuan and Shaw approach on the sport classification domain.

Table 7
Accuracy of the Yuan and Shaw approach on the sport classification domain

Case	Classification known in training data			Classification derived from learned rules			
	Volleyball	Swimming	W-lifting	Volleyball	Swimming	W-lifting	Results
1	0.0	0.8	0.2	0.0	0.9	0.0	r
2	1.0	0.7	0.0	0.4	0.6	0.0	w
3	0.3	0.6	0.1	0.2	0.7	0.3	r
4	0.9	0.1	0.0	0.7	0.3	0.3	r
5	0.0	0.0	1.0	0.3	0.1	0.9	r
6	0.2	0.0	0.8	0.3	0.0	0.7	r
7	0.0	0.0	1.0	0.0	0.0	1.0	r
8	0.7	0.0	0.3	0.2	0.0	0.8	w
9	0.2	0.8	0.0	0.0	1.0	0.0	r
10	0.0	0.3	0.7	0.1	0.0	0.7	r
11	0.4	0.7	0.0	0.0	0.7	0.0	r
12	0.7	0.2	0.1	0.7	0.0	0.3	r
13	0.0	0.0	1.0	0.0	0.2	0.8	r
14	0.0	0.0	1.0	0.3	0.0	0.7	r
15	0.0	0.0	1.0	0.0	0.0	1.0	r
16	0.8	0.6	0.0	0.5	0.5	0.0	a

Note: w = wrong classification; a = ambiguity; r = right classification.

2. When two or more rules classify an object into the same class with different degrees of membership, take the maximum as the class membership value.
3. An object may be classified into several classes with different degrees of membership. When classification to only one class is required, select the class with the highest membership.

Among the 16 training cases, 13 cases (except cases 2, 8, 16) were correctly classified. The classification accuracy was 81%.

Next, the sport classification problem was run using our proposed fuzzy inductive learning algorithm with the two parameters α and β being respectively, 0.5 and 0.7. The set of fuzzy rules induced by our proposed approach is shown in Fig. 1.

According to the derived six classification rules, the classification accuracy for the training data is shown in Table 8. The classification accuracy was 81%.

Although the accuracy and the number of rules derived using our approach are the same as those derived using the Yuan and Shaw fuzzy decision tree approach [27], the rules derived using our approach are, however, simpler than Yuan and Shaw's. Some irrelevant tests of the rules derived using our approach have been removed. For example, the selector "Temperature is Hot" in Rule *d* does not appear in Rule 4, and the selector "Temperature is Mild" in Rule *f* does not appear in Rule 6.

7.2. The IRIS domain

The Iris problem is stated as follows. There are three species of Iris flowers to be distinguished: *Setosa*, *Versicolor*, and *Virginica*. There are 50 training instances for each class. Each training instance is described by four attributes: *Sepal Length (SL)*, *Sepal Width (SW)*, *Petal Length (PL)*, and *Petal Width (PW)*. All four of the attributes are numerical domains. The membership functions of each attribute used in this experiment are defined in Fig. 3.

Table 8
Accuracy of our approach on the sport classification domain

Case	Classification known in training data			Classification derived from learned rules			
	Volleyball	Swimming	W-lifting	Volleyball	Swimming	W-lifting	Results
1	0.0	0.8	0.2	0.0	0.9	0.4	r
2	1.0	0.7	0.0	0.4	0.6	0.0	w
3	0.3	0.6	0.1	0.2	0.7	0.3	r
4	0.9	0.1	0.0	0.7	0.3	0.3	r
5	0.0	0.0	1.0	0.3	0.1	0.9	r
6	0.2	0.0	0.8	0.3	0.0	0.7	r
7	0.0	0.0	1.0	0.0	0.3	1.0	r
8	0.7	0.0	0.3	0.2	0.0	0.8	w
9	0.2	0.8	0.0	0.0	1.0	0.7	r
10	0.0	0.3	0.7	0.1	0.0	0.9	r
11	0.4	0.7	0.0	0.0	0.7	0.2	r
12	0.7	0.2	0.1	0.7	0.0	0.3	r
13	0.0	0.0	1.0	0.0	0.2	1.0	r
14	0.0	0.0	1.0	0.3	0.0	0.7	r
15	0.0	0.0	1.0	0.0	0.0	1.0	r
16	0.8	0.6	0.0	0.5	0.5	0.0	a

Note: w = wrong classification; a = ambiguity; r = right classification.

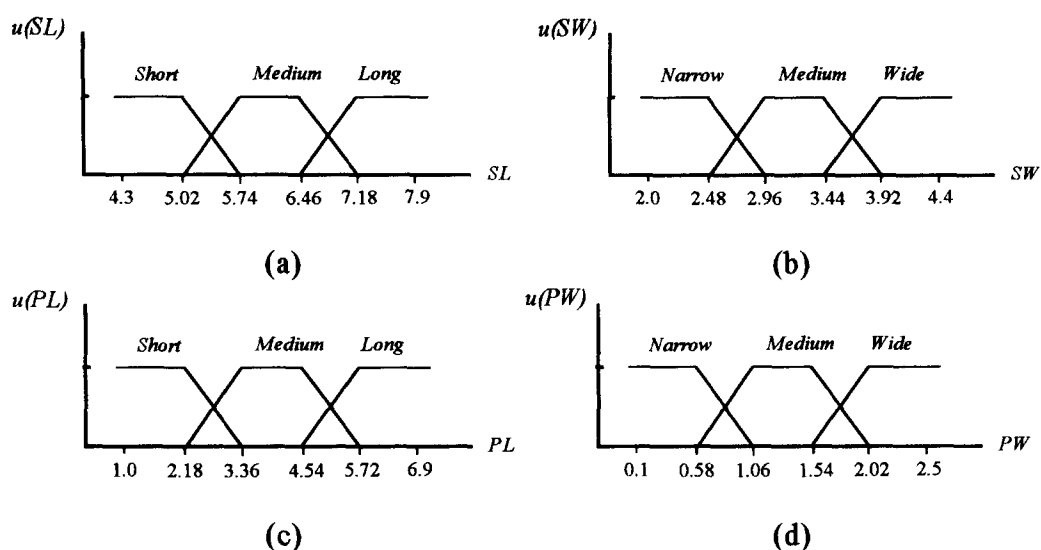


Fig. 3. The membership functions of the attributes for the Iris flower classification problem.

Table 9
Accuracy of six learning algorithms on the Iris Flower Problem

Algorithm \ Class	Setosa	Virginica	Versicolor	Average
FIL	100	98	94	97.33
GVS	100	94	94	96.00
IVSM	100	93.33	94.00	95.78
NT growth	100	93.50	91.13	94.87
Dasarathy	100	98	86	94.67
C4	100	91.07	90.61	93.89

Since the training set includes only 150 instances, a method called *N-fold cross validation* [4] was adopted for this small set of examples. All instances were randomly divided into N subsets of as nearly equal size as possible. For each i , $i = 1, \dots, N$, the i th subset was used as a test set, and the other subsets were combined into a training set. In the experiments, the data were partitioned into 10 subsets, each with 15 instances composed of five positive training instances and 10 negative training instances. The fuzzy learning algorithm was then run on the training instances to derive promising rules. Finally, the rules derived were then tested on the remaining test data. Classification rates were then averaged across all possible groups. The set of rules derived using our approach was:

- Rule 1: IF PL is *Short*, THEN Iris Setosa ($B = 0.99$).
- Rule 2: IF PL is *Medium*, THEN Iris Versicolor ($B = 0.89$).
- Rule 3: IF PL is *Long*, THEN Iris Virginica ($B = 0.97$).
- Rule 4: IF PW is *Wide*, THEN Iris Virginica ($B = 0.93$).

The average classification accuracy was 100% for *Setosa*, 98% for *Versicolor*, and 94% for *Virginia*. The accuracies of some other learning algorithms on the Iris Flower Classification Problem are also shown in Table 9 for comparison. The methods studied were Hirsh's Incremental Version Space Merging [13], Aha

and Kibler's noise-tolerant NT-growth [1], Dasarathy's pattern-recognition approach [9], Quinlan's C4 [21], and Hong and Tseng's generalized version space learning algorithm (GVS) [14]. It can easily be seen that the accuracy of our method is the highest among all the listed learning methods.

8. Conclusion

In this paper, we have proposed a fuzzy inductive learning algorithm based on fuzzy information gain to generate fuzzy *if-then* rules. This approach can solve problems conventional inductive learning methods have with fuzzy set, and find promising inference rules. Also, the proposed approach eliminates irrelevant tests in the rules. Experimental results show that our method yields high accuracy, and the induced rules are concise. The proposed method is thus a flexible and efficient fuzzy inductive learning method for modular rules.

Acknowledgments

The authors would like to thank the anonymous referees for their very constructive comments.

References

- [1] D.W. Aha, D. Kibler, Noise-tolerant instance-based learning algorithms, Proc. 11th Internat. Joint Conf. on Artificial Intelligence, Detroit, MI, 1989, pp. 794–799.
- [2] A.F. Blishun, Fuzzy learning models in expert systems, Fuzzy Sets and Systems 22 (1987) 57–70.
- [3] M. Botta, A. Giordana, L. Saitta, Learning in uncertain environments, in: R.R. Yager, L.A. Zadeh (Eds.), An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, Dordrecht, 1992, pp. 281–296.
- [4] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Tree, Wadsworth International Group, Belmont, CA, 1984.
- [5] L.M. de Campos, S. Moral, Learning rules for a fuzzy inference model, Fuzzy Sets and Systems 59 (1993) 247–257.
- [6] J. Cendrowska, PRISM: an algorithm for inducing modular rules, Internat. J. Man-Machine Stud. 27 (1987) 349–370.
- [7] R.L.P. Chang, T. Pavlidis, Fuzzy decision tree algorithms, IEEE Trans. Systems Man Cybernet. 7 (1977) 28–35.
- [8] P. Clark, T. Niblett, The CN2 induction algorithm, Machine Learning 3 (1989) 261–283.
- [9] B.V. Dasarathy, Noising around the neighborhood: a new system structure and classification rule for recognition in partially exposed environments, IEEE Trans. Pattern Analysis Machine Intelligence 2 (1980) 67–71.
- [10] M. Delgado, A. Gonzalez, An inductive learning procedure to identify fuzzy systems, Fuzzy Sets and Systems 55 (1993) 121–132.
- [11] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.
- [12] A. Gonzalez, A learning methodology in uncertain and imprecise environments, Internat. J. Intelligent Systems 10 (1995) 357–371.
- [13] H. Hirsh, Generalizing version spaces, Machine Learning 17 (1994) 5–46.
- [14] T.P. Hong, S.S. Tseng, A generalized version space learning algorithm for noisy and uncertain data, IEEE Trans. Knowledge Data Eng. 9 (1997) 336–340.
- [15] H. Ichihashi, T. Shirai, K. Nagasaka, T. Miyoshi, Neuro-fuzzy ID3 – a method of inducing decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, Proc. IEEE WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms, Nagoya, Japan, 1994, pp. 23–33.
- [16] J. Kacprzyk, C. Iwanski, Fuzzy logic with linguistic quantifiers in inductive learning, in: L.A. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, Wiley, New York, 1992, pp. 465–478.
- [17] R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Machine Learning: An Artificial Intelligence Approach, vol. 1, Morgan Kaufmann, Palo Alto, CA, 1983.
- [18] R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Machine Learning: An Artificial Intelligence Approach, vol. 2, Morgan Kaufmann, Palo Alto, CA, 1986.
- [19] T.M. Mitchell, Generalization as search, Artificial Intelligence 18 (1982) 203–226.
- [20] J.R. Quinlan, Decision tree as probabilistic classifier, Proc. 4th Internat. Machine Learning Workshop, Morgan Kaufmann, San Mateo, CA, 1987, pp. 31–37.

- [21] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
- [22] J. Rives, FID3: fuzzy induction decision tree, Proc. 1st Internat. Symp. on Uncertainty Modeling and Analysis, 1990, pp. 457–462.
- [23] C.H. Wang, T.P. Hong, S.S. Tseng, Inductive learning from fuzzy examples, Proc. 5th IEEE Internat. Conf. on Fuzzy Systems, New Orleans, 1996, pp. 13–18.
- [24] C.H. Wang, J.F. Liu, T.P. Hong, S.S. Tseng, FILSMR: a fuzzy inductive learning strategy for modular rules, Proc. 6th IEEE Internat. Conference on Fuzzy Systems, Barcelona, Spain, June 25–29, 1997, pp. 1289–1294.
- [25] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1992, pp. 265–268.
- [26] S. Weber, A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms, Fuzzy Sets and Systems 11 (1983) 115–134.
- [27] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.