Applied Intelligence 18, 179–193, 2003

Fuzzy Inductive Learning Strategies

CHING-HUNG WANG
Chunghwa Telecommunication Laboratories, Ministry of Transportation and Communications, Chung-Li 32617, Taiwan, R.O.C.
ching@ms.tl.gov.tw

CHANG-JIUN TSAI
Institute of Computer and Information Science, National Chiao-Tung University, Hsin-Chu 30050, Taiwan, R.O.C.
sstseng@cis.nctu.edu.tw

TZUNG-PEI HONG
Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, R.O.C.
tphong@nuk.edu.tw

SHIAN-SHYONG TSENG
Institute of Computer and Information Science, National Chiao-Tung University, Hsin-Chu 30050, Taiwan, R.O.C.

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. Design of learning methods for 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 AQR learning strategy is proposed to manage linguistic information. The proposed learning algorithm generates fuzzy linguistic rules from "soft" instances. 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.

Keywords: AQR algorithm, fuzzy classification, fuzzy inductive learning, machine learning, soft instances

1. Introduction

Among machine learning approaches [1–5], inductive learning from instances may be commonly used in real-world application domains. Famous examples are decision-tree approaches [6–8] or AQR-based approaches [9, 10]. Inductive learning is basically a process of inferring concept descriptions that include positive instances and exclude negative instances. Traditional inductive learning procedures are however inapplicable to some real domains, since data in the real world usually contain vagueness and ambiguity. Fuzzy techniques can then be adopted to manage this kind of domains [11, 12].

Vagueness and ambiguity most commonly result from inappropriate or inadequate attributes being used to describe objects, or when experts, teachers, or users are not quite sure what classes given objects belong to.

The boundaries of pieces of information used may not be clear-cut, and each object may be expressed as a linguistic "input-output" relationship. 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 describe 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 formation by conventional inductive learning methods [13]. It may make the learning process fail or derive null concept description. The design of learning methods to work well with vague data is thus very important. Some kinds of inductive learning problems arising from working with vague data are discussed in [11, 12, 14–18]. Several successful learning strategies based on ID3 have been proposed [19–21]; most of these use tree-pruning and fuzzy logic techniques. As for Version-Space-based learning strategies [22], Wang et al. proposed a fuzzy version-space learning algorithm to manage linguistic information [23]. Besides, Sudkamp and Hammell proposed the methods of interpolation, completion, and learning fuzzy rules for fuzzy inference systems [12]. In this paper, we propose a fuzzy learning algorithm based on the AQR learning strategy [9, 10] to induce a fuzzy rule set from "soft" training instances. This learning approach can overcome some inductive learning problems in vague learning environments.

The remainder of this paper is organized as follows. Some related concepts and terms are reviewed in Section 2. The AQR learning strategy is reviewed in Section 3. The concepts of fuzzy inductive learning are introduced in Section 4. A fuzzy inductive learning algorithm (FAQR) based on AQR is proposed in Section 5. Experimental results on the Sports and on the IRIS flower classification problems are reported in Section 6. Finally, discussion and future work are given in Section 7.

2. Review of Related Concepts and Terms

In this section, we briefly review concepts and terms used in this paper.

2.1. *Fuzzy Set Concepts*

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:

$$u_A\colon X \to [0, 1],$$

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

Assuming that A and B are two fuzzy sets with respective 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 $\tau\colon [0, 1] * [0, 1] \to [0, 1]$ is a *t-norm* operator satisfying the following conditions [24] 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 $\rho\colon [0, 1] * [0, 1] \to [0, 1]$ is an *s-norm* operator satisfying the following conditions [25]: 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 the 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

An *instance space* is a set of instances that can be legally described by a given instance language. Instance spaces can be divided into two classes: *attribute-based* instance spaces and *structured* instance spaces [26]. In an attribute-based instance space, each instance can be represented by *one* or *several* attributes. Attribute-based instance spaces are of primary concern here.

The entire instance space is partitioned into several *classes*, each with its own class name. Instances belonging to the same class possess certain common properties. A concept is then a classification rule used to describe a certain class. For example, the concept of prime number is , "x is prime if x is an integer, and x is divisible only by x and 1."

A *hypothesis space* is a set of hypotheses that can be legally described by a concept description language (generalization language). Five kinds of expressions [25] are often used in representing hypotheses: *pure conjunctive form*, *pure disjunctive form*, *internal disjunctive form*, *DNF* and *CNF*. In this paper, the proposed learning strategy is mainly concerned with DNF (disjunctive normal form) expressed as follows:

$$C_1 \text{ or } C_2 \text{ or } \ldots \text{ or } C_x,$$

where C_i is a pure conjunctive expression.

Formally the concepts derived by our method could be represented in the following grammar:

If *<cover>* then predict *<class>*, where
<cover> = *<complex*$_1$*>* or ... or *<complex*$_x$*>*,
<complex>=*<selector*$_1$*>* and ... and *<selector*$_y$*>*,
<selector>=*<attribute relationship value>*.

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.

Conventional inductive learning is aimed at finding a concept description R that correctly describes all instances in the training set. 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^+ represents a positive instance and e^- represents a negative one, $\subset$ and $\not\subset$ are relationship descriptors that mean "*covered by*" and "*not covered by*", respectively.

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

3. Review of the AQR Learning Strategy

AQR is an inductive learning system [9] that uses the basic AQ algorithm [10] to generate a set of classification rules. When building classification rules, AQR performs a heuristic search through hypothesis space to determine the descriptions that account for all positive instances and no negative instances. AQR processes the training instances in stages; each stage generating a single rule, and then removing the instances it covers from the training set. This step is repeated until enough rules have been found to cover all instances in the chosen class. The AQR algorithm is described below:

AQR algorithm:

Let POS be a set of positive instances.
Let NEG be a set of negative instances.

STEP 1. Let COVER be the empty cover.
STEP 2. While COVER does not cover all instances in POS, process the following steps. Otherwise, stop the procedure and return COVER.

STEP 3. Select a SEED, i.e., a positive instance not covered by COVER.
STEP 4. Call procedure GENSTAR to generate the set STAR, which is a set of complexes that covers SEED but that covers no instances in ENG.
STEP 5. Let BEST be the best complex in STAR according to user-defined criteria.
STEP 6. Add BEST as an extra disjunct to COVER.

GENSTAR procedure:

STEP 1. Let STAR be the set containing the empty complex.
STEP 2. While any complex in STAR covers some negative instances in NEG, process the following steps; otherwise, stop the procedure and return STAR.
STEP 3. Select a negative instance E_{neg} covered by a complex in STAR.
STEP 4. Specialize complexes in STAR to exclude E_{neg} by the following substeps:

(a) Let EXTENSION be all selectors that cover SEED, but not E_{neg};
(b) Let STAR be the set $\{x \cap y \mid x \in \text{STAR}, y \in \text{EXTENSION}\}$;
(c) Remove all complexes in STAR subsumed by other complexes.

STEP 5. Remove the worst complexes from STAR until the size of STAR $\leq$ maxstar (a user-defined maximum).

Unfortunately, the AQR learning strategy only works well in ideal domains where no vague data are present. When such data are present, AQR cannot work well. However, the effective use of learning systems in real-world applications depends substantially upon their ability to handle linguistic information. In this paper, we thus apply the concept of fuzzy sets to the AQR learning strategy to solve this problem.

4. Fuzzy Inductive Learning

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

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

where $\tilde{R}$ is a fuzzy concept description, $\tilde{\forall}$ is a linguistic quantifier of type "almost all", "most", etc. [13], $\tilde{P}$ denotes a fuzzy positive class and $\tilde{N}$ denotes a fuzzy negative class, $\tilde{\subset}_\alpha$ and $\tilde{\not\subset}_\alpha$ are fuzzy relationship descriptors that mean "*α-covered by*" and "*α-not covered by*" respectively, and $\tilde{e} \in_\beta \tilde{P}$ and $\tilde{e} \in_\beta \tilde{N}$ represent that $\tilde{e}$ *β-belongs* to $\tilde{P}$ and $\tilde{N}$ respectively. When the degree of instance $\tilde{e}$ covered by $\tilde{R}$ is greater than or equal to a predefined significance level α, $\tilde{R}$ is then said to *α-cover* instance $\tilde{e}$. Each instance $\tilde{e}$ can then 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 significance level β, instance $\tilde{e}$ is then said to *β-belong* to the class $\tilde{P}$ (i.e., $u_{\tilde{P}}(\tilde{e}) \geq \beta$, represented as $\tilde{e} \in_\beta \tilde{P}$). The set of "soft" positive instances *β-belonging* to the class $\tilde{P}$ is thus denoted as $\tilde{P}_\beta$. Similarly, the set of "soft" negative instances *β-belonging* to the class $\tilde{N}$ is denoted as $\tilde{N}_\beta$. Fuzzy inductive learning thus attempts to find a concept description, $\tilde{R}$, that *α-covers* almost all "soft" positive instances in $\tilde{P}_\beta$ and almost no "soft" negative instances in $\tilde{N}_\beta$.

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 $u_{basketball\_player}(\tilde{e}) = 0.8$,

where both [height = 190 cm] and [weight = 80 kg] are *crisp* selectors, and $u_{basketball\_player}(\tilde{e})$ 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 baseketball_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 *instance space* must therefore be transformed into representations in *hypothesis space* for fuzzy matching. Let $u_{\tilde{s}_i}(\tilde{e})$ represent the degree of matching between selector $\tilde{s}_i$ in the hypothesis space and the corresponding selector in instance $\tilde{e}$. The value of $u_{\tilde{s}_i}(\tilde{e})$ ranges between 0 and 1, and is used to represent the degree to which instance $\tilde{e}$ is covered by $\tilde{s}_i$; 0 indicates complete exclusion and 1 indicates complete inclusion. When the value of $u_{\tilde{s}_i}(\tilde{e})$ is greater than or equal to a predefined significant level α, selector $\tilde{s}_i$ is said to *α-cover* instance $\tilde{e}$.

Assume that we have an instance $\tilde{e}$ and a complex $\tilde{C}_j = \tilde{s}_{j_1} \wedge \tilde{s}_{j_2} \wedge \cdots \wedge \tilde{s}_{j_m}$. The degree of instance $\tilde{e}$ covered by complex $\tilde{C}_j$ is evaluated as:

$$u_{\tilde{c}_j}(\tilde{e}) = u_{\tilde{s}_{j_1}}(\tilde{e}) \wedge u_{\tilde{s}_{j_2}}(\tilde{e}) \wedge \cdots \wedge u_{\tilde{s}_{j_m}}(\tilde{e}),$$

or more generally,

$$u_{\tilde{c}_j}(\tilde{e}) = u_{\tilde{s}_{j_1}}(\tilde{e}) \ \tau \ u_{\tilde{s}_{j_2}}(\tilde{e}) \ \tau \cdots \tau \ u_{\tilde{s}_{j_m}}(\tilde{e}),$$

where τ is a *t-norm* operator.

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

The concept description $\tilde{R}$ indicates the disjunction of complexes, say, $\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_x$, and is denoted as $\tilde{R} = \tilde{C}_1 \vee \tilde{C}_2 \cdots \vee \tilde{C}_x$. The degree of instance $\tilde{e}$ covered by the concept description $\tilde{R}$ is thus evaluated as:

$$u_{\tilde{R}}(\tilde{e}) = u_{\tilde{c}_1}(\tilde{e}) \vee u_{\tilde{c}_2}(\tilde{e}) \vee \cdots \vee u_{\tilde{c}_x}(\tilde{e}),$$

or more generally,

$$u_{\tilde{R}}(\tilde{e}) = u_{\tilde{c}_1}(\tilde{e}) \, \rho \, u_{\tilde{c}_2}(\tilde{e}) \, \rho \ \cdots \rho u_{\tilde{c}_x}(\tilde{e}),$$

where ρ is an *s-norm* operator.

When the value of $u_{\tilde{R}}(\tilde{e})$ is greater than or equal to a predefined significance level α, $\tilde{R}$ is then said to *α-cover* instance $\tilde{e}$. The concept of fuzzy matching is used in our proposed learning algorithm to handle vagueness.

5. The Fuzzy AQR Learning Strategy

In this section, we propose a fuzzy AQR learning algorithm that can induce linguistic concept descriptions from a set of "soft" training instances. The membership values of classes and attributes for soft training instances are assumed known in advance. Several approaches for getting appropriate membership functions were proposed in the past, and can be adopted here. Membership functions may be subjectively assigned by domain experts or derived through a delphi negotiation process [27]. They may also be formed by equally dividing the value domains into several fuzzy regions. Fuzzy clustering techniques may also be used to get the membership functions from the example distribution [28–31]. Since this paper focuses on learning fuzzy rules from soft training examples, the acquisition of membership values will not further be discussed here.

In the proposed method, the concept descriptions no longer necessarily *include/exclude* all *positive/negative* instances presented, since linguistic information exists in the "soft" training set $\tilde{E}$. Two fuzzy measurement functions, $u_{include}(\tilde{R})$ and $u_{exclude}(\tilde{R})$, are used to evaluate the "goodness" of a derived concept $\tilde{R}$. The fuzzy measurement function, $u_{include}(\tilde{R})$, used to evaluate the degree of including "soft" positive instances by the concept description $\tilde{R}$ is defined as follows:

$$u_{include}(\tilde{R}) = \frac{\sum_{\tilde{e} \in_\beta \tilde{P}} (u_{\tilde{P}}(\tilde{e}) \ \tau \ u_{\tilde{R}}(\tilde{e}))}{\sum_{\tilde{e} \in_\beta \tilde{P}} u_{\tilde{P}}(\tilde{e})}.$$

Similarly, $u_{include}(\tilde{C})$, used to evaluate the degree of including "soft" positive instances by complex $\tilde{C}$, is defined as follows:

$$u_{include}(\tilde{C}) = \frac{\sum_{\tilde{e} \in_\beta \tilde{P}} (u_{\tilde{P}}(\tilde{e}) \ \tau \ u_{\tilde{C}}(\tilde{e}))}{\sum_{\tilde{e} \in_\beta \tilde{P}} u_{\tilde{P}}(\tilde{e})}.$$

The fuzzy measurement function, $u_{exclude}(\tilde{R})$, used to evaluate the degree of excluding "soft" negative instances by the concept description $\tilde{R}$, is defined as follows:

$$u_{exclude}(\tilde{R}) = \frac{\sum_{\tilde{e} \in_\beta \tilde{N}} (u_{\tilde{N}}(\tilde{e}) \ \tau \ (1 - u_{\tilde{R}}(\tilde{e})))}{\sum_{\tilde{e} \in_\beta \tilde{N}} \ u_{\tilde{N}}(\tilde{e})}.$$

Correspondingly, $u_{exclude}(\tilde{C})$, used to evaluate the degree of excluding "soft" negative instances by complex $\tilde{C}$, is defined as follows:

$$u_{exclude}(\tilde{C}) = \frac{\sum_{\tilde{e}\in_\beta \tilde{N}} (u_{\tilde{N}}(\tilde{e})\ \tau\ (1 - u_{\tilde{C}}(\tilde{e})))}{\sum_{\tilde{e}\in_\beta \tilde{N}}\ u_{\tilde{N}}(\tilde{e})}.$$

A complex $\tilde{C}$ with a higher $u_{include}(\tilde{C})$ membership value possesses more truthful inclusion of "*soft*" positive training instances, and a complex $\tilde{C}$ with a higher $u_{exclude}(\tilde{C})$ possesses more truthful exclusion of "*soft*" negative training instances. A complex that includes much positive information may also possibly include much negative information. Correspondingly, a complex that excludes much negative information may also possibly exclude much positive information. Clearly these kinds of complexes are not sure to be better than complexes that include both a little fuzzy positive and a little fuzzy negative information. Which complex $\tilde{C}$ is suitable thus depends on both $u_{include}(\tilde{C})$ and $u_{exclude}(\tilde{C})$. $u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{C})$ is then used to make this determination, which is defined as follows:

$$u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{C}) = u_{include}(\tilde{C})\ \rho\ u_{exclude}(\tilde{C}),$$

where ρ is a *union* operator or an *addition* operator. Note that if the maximum operator is used for the *union* operator, the cost for using the *union* operator is constant. It only needs the time of finding the maximum of two real numbers. Similarly, if the minimum operator is used for the τ operator, the cost for using the *intersection* operator is constant. The costs for $u_{include}(\tilde{C})$ and $u_{exclude}(\tilde{C})$are then proportional to the numbers of positive and negative soft examples.

Similarly, $u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{R})$ is used to evaluate the performance of the derived concept description $\tilde{R}$, which is defined as follows:

$$u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{R}) = u_{include}(\tilde{R})\ \rho\ u_{exclude}(\tilde{R}).$$

The fuzzy AQR learning strategy consists of two main phases: generation and testing. The generation phase generates and collects possible fuzzy complexes into a large set; the testing phase then evaluates each element in this set according to the value of $u_{\tilde{\forall}+\tilde{\forall}^-}$. The best fuzzy complex as an extra disjunct is then added to the set of concept descriptions. This procedure is repeated until all "soft" positive instances in $\tilde{P}_\beta$ have been α-*covered* by the set of concept descriptions. The fuzzy AQR learning algorithm is stated below.

INPUT:

A set of "soft" positive and negative training instances.

OUTPUT:

A fuzzy concept description $\tilde{R}$ that α-*covers* almost all "soft" positive instances in $\tilde{P}_\beta$ and almost no "soft" negative instances in $\tilde{N}_\beta$.

Fuzzy AQR Learning Algorithm:

STEP 1. Let $\tilde{R}$ be an empty set.
STEP 2. While $\tilde{R}$ does not α-*cover* all "soft" positive instances in $\tilde{P}_\beta$ (i.e., $\exists\ \tilde{e} \in \tilde{P}_\beta$, $\tilde{e}\tilde{\not\subset}_\alpha \tilde{R}$), process the following steps. Otherwise, stop the procedure and return $\tilde{R}$.
STEP 3. Select a *SEED* that is a "soft" positive instance not α-*covered* by $\tilde{R}$ and having the highest $u_{\tilde{P}}(\tilde{e})$ among all soft positive instances.
STEP 4. Call procedure *GenComplex* to generate $\widehat{C}_{set}$, which is a set of complexes that α-*cover SEED* and α-*cover* no "soft" negative instances in $\tilde{N}_\beta$ (i.e., $\forall\ \tilde{C}_i \in \widehat{C}_{set}$, $\forall \tilde{e} \in \tilde{N}_\beta$, $SEED\ \tilde{\subset}_\alpha \tilde{C}_i$ & $\tilde{e}\ \tilde{\not\subset}_\alpha \tilde{C}_i$).
STEP 5. Select the *complex* $\tilde{C}_{best}$ that has the highest $u_{\tilde{\forall}+\tilde{\forall}^-}$ value in $\widehat{C}_{set}$.
STEP 6. Add $\tilde{C}_{best}$as an extra disjunct to $\tilde{R}$ (i.e., $\tilde{R} = \tilde{R} \vee \tilde{C}_{best}$), and then GO TO STEP 2.

GenComplex Procedure:

STEP 1. Let$\widehat{C}_{set}$ be a set of single-selector complexes that α-*cover SEED*.
STEP 2. While at least one *complex* in$\widehat{C}_{set}$ α-*covers* a "soft" negative instance in $\tilde{N}_\beta$ (i.e., $\exists\ \tilde{C}_j \in \widehat{C}_{set}$, $\exists\ \tilde{e} \in \tilde{N}_\beta$, $\tilde{e}\tilde{\subset}_\alpha \tilde{C}_j$), process the following steps; otherwise, stop the procedure and return$\widehat{C}_{set}$.
STEP 3. Select a $\tilde{C}_j$ with the smallest value $u_{exclude}(\tilde{C}_j)$ in$\widehat{C}_{set}$.
STEP 4. Select a soft negative instance $\tilde{e}$ with the highest $u_{\tilde{N}}(\tilde{e})$ among those α-*covered* by $\tilde{C}_j$.
STEP 5. Specialize all *complexes* in $\widehat{C}_{set}$ to α-*not cover* negative instance $\tilde{e}$ using the following substeps:

(a) Let$\widehat{S}$ be the set of selectors that α-*cover SEED*, but not $\tilde{e}$.
(b) Let $\widehat{C}_{set}$ be the set $\{\tilde{C}_j \wedge \tilde{S}_k \mid \tilde{C}_j \in \text{old}\ \widehat{C}_{set}, \tilde{S}_k \in \widehat{S}\}$.
(c) Remove all *complexes* in $\widehat{C}_{set}$ subsumed by other *complexes* (i.e., if $\tilde{C}_i$ subsumes $\tilde{C}_j$ and $u_{\tilde{\forall}+\forall^-(\tilde{C}_i)} \geq u_{\tilde{\forall}+\forall^-(\tilde{C}_j)}$, then drop $\tilde{C}_j$ from$\widehat{C}_{set}$).

STEP 6. Remove the worst complexes from $\widehat{C}_{set}$ until the size of $\widehat{C}_{set} \leq \theta$ (a user-defined threshold).

The fuzzy AQR learning algorithm performs a heuristic search of hypothesis space to determine the fuzzy concept descriptions that *α-cover* all "soft" positive instances and no "soft" negative instances. It induces rules in stages; each stage generates a *complex*. When the learning process terminates, the *complexes* are output to form a set of rules. If the concept description $\tilde{R}$ is a disjunction of complexes $\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_k$, then $\tilde{R}$ is represented in the form of rules as follows:

Rule 1: IF $\tilde{C}_1$, then the positive class with membership value $u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{C}_1)$;
Rule 2: IF $\tilde{C}_2$, then the positive class with membership value $u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{C}_2)$;

⋮

Rule k: IF $\tilde{C}_k$, then the positive class with the membership value $u_{\tilde{\forall}+\tilde{\forall}^-}(\tilde{C}_k)$.

Below, an example is used to clarify the learning process.

Example 6. This is a simple domain in which the fuzzy concept of dangerous dogs is to be derived. Each instance is described by three fuzzy attributes with membership values for two classes. Each dog is described by its ***color*** (e.g., *black*, *brown*), its ***size*** (e.g., *small*, *medium*, *large*), its ***coat*** (e.g., *short*, *long*), and its ***class*** (e.g., *dangerous*, *non-dangerous*).

Assume the four linguistic training instances in Table 1 are given for training.

Assume that the size of complexes maintained in $\widehat{C}_{set}$ is set at 5 ($\theta_{=5}$), and parameters α and β are respectively set at 0.5 and 0.5. The concept for dangerous dogs is learned by the proposed algorithm as follows:

Step 1: Initialize $\tilde{R}$ as an empty set.
Step 2: Since $\tilde{R}$ does not *α-cover* all "soft" positive instances in $\tilde{P}_\beta$, the following steps are executed.
Step 3: Since $\tilde{e}_1$ is not covered by $\tilde{R}$ and has the highest $u_{dangerous}$ value ($=0.80$) among the soft positive examples, $\tilde{e}_1$ is thus selected as the *SEED*.
Step 4: Call procedure *GenComplex* to generate complexes $\widehat{C}_{set}$ that *α-cover* $\tilde{e}_1$ and *α-cover* no "soft" negative instances as follows:

a. Find the single-selector complexes that *α-cover* $\tilde{e}_1$. Since the selectors "***size*** = large", "***color*** = black", and "***coat*** = long" *α-cover* $\tilde{e}_1$, $\widehat{C}_{set} =$ {"***size*** = large", "***color*** = black", "***coat*** = long"}.
b. Since *complex* "***coat*** = long" in $\widehat{C}_{set}$ *α-covers* "soft" negative instances $\tilde{e}_4$ and $\tilde{e}_3$, and *complex* "***color*** = black" *α-covers* $\tilde{e}_4$, the following steps are executed.
c. Assume the minimum operator is used as the τ operator. The $u_{exclude}$ value for the complex "***coat*** = long" is calculates as:

$$u_{exclude}(\text{"}coat = long\text{"}) = \frac{\text{Min}(0.6, 1-0.8) + \text{Min}(0.9, 1-1.0)}{0.6+0.9} = 0.133.$$

The $u_{exclude}$ value for the complex "***color*** = black" is calculates as:

$$u_{exclude}(\text{"}color = black\text{"}) = \frac{\text{Min}(0.6, 1-0.1) + \text{Min}(0.9, 1-0.9)}{0.6+0.9} = 0.467.$$

Since complex "***coat*** = long" has a smaller $u_{exclude}$ value than "***color*** = black" has, "***coat*** = long" is then chosen for later processing.
d. Among the soft negative instances *α-covered* by "***coat*** = long", instance $\tilde{e}_4$ has the highest $u_{\tilde{N}}$ value (=0.9). $\tilde{e}_4$ is then chosen.

Table 1. Four training instances in the dangerous dog domain.

	Size			*Color*		*Coat*		*Class*	
Case	*Small*	*Medium*	*Large*	*Black*	*Brown*	*Short*	*Long*	*Dangerous*	*Non-dangerous*
$\tilde{e}_1$	0.1	0.2	0.9	0.8	0.2	0.1	0.9	0.8	0.2
$\tilde{e}_2$	0.0	0.0	1.0	0.0	1.0	0.9	0.2	0.7	0.3
$\tilde{e}_3$	0.2	0.8	0.3	0.1	0.9	0.2	0.8	0.4	0.6
$\tilde{e}_4$	0.9	0.1	0.0	0.9	0.1	0.0	1.0	0.1	0.9

e. Specialize all *complexes* in $\widehat{C}_{set}$ to *α-not cover* the negative instance $\tilde{e}_4$ using the following steps:
   e.1. Among the three selectors "***size*** = large", "***color*** = black", and "***coat*** = long" which *α-cover* $\tilde{e}_1$, only "***size*** = large" does not *α-cover* $\tilde{e}_4$. $\tilde{S}$ is then {"***size*** = large"}.
   e.2. New $\widehat{C}_{set} = \{\tilde{C}_j \wedge \tilde{S}_k \mid \tilde{C}_j \in \text{old}\,\widehat{C}_{set},\ \tilde{S}_k \in \tilde{S}\}$ = {"***size*** = large"}.
   e.3. Since there is only one complex in $\widehat{C}_{set}$, no subsumption relationships exist. No complex is thus removed.

f. Since only one complex exists in $\widehat{C}_{set}$ and θ is set at 5, no complex is removed from $\widehat{C}_{set}$.

Step 5: *Complex* "***size*** = large" is the best one in $\widehat{C}_{set}$ since only one complex exist in $\widehat{C}_{set}$. The $u_{include}$ value for complex "***size*** = large" is calculates as:

$$u_{include}(\text{"}size = large\text{"}) = \frac{\text{Min}(0.8, 0.9) + \text{Min}(0.7, 1.0)}{0.8 + 0.7} = 1.0.$$

The $u_{exclude}$ value for complex "***size*** = large" is calculates as:

$$u_{exclude}(\text{"}size = large\text{"}) = \frac{\text{Min}(0.6, 1 - 0.3) + \text{Min}(0.9, 1 - 0.0)}{0.6 + 0.9} = 1.0.$$

Assume the maximum operator is used for the ρ operation. $u_{\tilde{\forall}^+\tilde{\forall}^-}$ ("***size*** = large") is then calculated as :

$$u_{\tilde{\forall}^+\tilde{\forall}^-}(\text{"}\boldsymbol{size} = large\text{"}) = \text{Max}(1, 1) = 1.$$

Note that in addition to the maximum operator, the addition operator can also be used for the ρ operation.

Step 6: Add *complex* "***size*** = large" as an extra disjunct to $\tilde{R}$. Since $\tilde{R}$ is originally empty, the new $\tilde{R}$ is {"***size*** = large"}.

Since $\tilde{R}$ = {"***size*** = large"} has *α-covered* all "soft" positive instances, the procedure is thus terminated. The concept $\tilde{R}$ for dangerous dogs can be expressed in the form of rules as:

IF *the size of dogs is* ***large***, THEN *dogs are* ***dangerous***, $u_{\tilde{\forall}^+\tilde{\forall}^-} = 1$.

6. Experiments

Two application domains were used to demonstrate the effectiveness of the proposed fuzzy AQR learning algorithm (FAQR). One decided what sport to play according to Sunday's weather, using the instances described in [21]. The other one classified Fisher's Iris data, which contain 150 instances. The fuzzy AQR learning algorithm was implemented in C language on a SUN SPARC/20 workstation and run 100 times on average since a fixed number of training examples are randomly chosen from the whole set of instances. The accuracy of the proposed method was compared with those of other learning algorithms on the same application domains. These experiments are described below.

6.1. The Sport Domain

This is a simple domain for deciding what sport to play according to Sunday's weather. A small set of training instances, each with fuzzy membership values, is shown in Table 2 [21]. Each instance is described by four fuzzy attributes (*Outlook*, *Temperature*, *Humidity*, *Wind*) and one fuzzy classification (*Sport*). Each attribute has the values shown below.

$$\begin{aligned} \text{Outlook} &= \{\text{Sunny, Cloudy, Rain}\}, \\ \text{Humidity} &= \{\text{Humid, Normal}\}, \\ \text{Temperature} &= \{\text{Cool, Mild, Hot}\}, \\ \text{Wind} &= \{\text{Windy, Not\_Windy}\}. \end{aligned}$$

Classifications include the following sports:

$$Sports = \{Swimming, Volleyball, Weight\_Lifting\}.$$

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 Yuan and Shaw's approach [21]. Yuan and Shaw's approach constructs a fuzzy decision tree based on a measurement of ambiguity. The rules generated by Yuan and Shaw's approach for this problem domain are shown below:

Rule *a*: IF *Temperature* is *Mild*, *Wind* is *Not-windy*, THEN *Volleyball* ($u = 0.78$);
Rule *b*: IF *Temperature* is *Hot*, *Outlook* is *Cloudy*, THEN *Swimming* ($u = 0.72$);

Table 2. A set of training instances in the sport domain.

	Outlook			Temperature			Humidity		Wind		Sports		
Case	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

Rule c: IF *Temperature* is *Hot*, *Outlook* is *Sunny*, THEN *Swimming* ($u = 0.85$);
Rule d: IF *Temperature* is *Hot*, *Outlook is Rain*, THEN *Weight-lifting* ($u = 0.73$);
Rule e: IF *Temperature* is *Cool*, THEN *Weight-lifting* ($u = 0.88$);
Rule f: IF *Temperature* is *Mild*, *Wind* is *Windy*, THEN *Weight-lifting* ($u = 0.81$).

Each derived classification rule was associated with a class membership value u. The classification for a given object was obtained using the following steps:

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

The classification accuracy of the training data is shown in Table 3 [21]. 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 three parameters α, β and θ being respectively 0.5, 0.6 and 100. The set of fuzzy rules induced by our proposed approach is shown below.

Rule 1: IF *Temperature* is *Hot*, THEN *Swimming* ($u = 0.95$).
Rule 2: IF *Wind* is *Not-windy*, THEN *Volleyball* ($u = 0.95$).
Rule 3: IF *Outlook* is *Rain*, THEN *Weight-lifting* ($u = 0.95$).
Rule 4: IF *Temperature* is *Cool*, THEN *Weight-lifting* ($u = 0.93$).
Rule 5: IF *Wind* is *Windy*, THEN *Weight-lifting* ($u = 0.68$).

The classification accuracy for the training data according to the five classification rules derived is shown in Table 4. The classification accuracy was 94%.

The accuracy of rules derived from FAQR is higher than that derived using Yuan and Shaw's fuzzy decision

Table 3. Accuracy of Yuan and Shaw's approach on the sport classification domain.

	Classification known in training data			Classification derived from learned rules			
Case	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	0.7	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

w: Wrong classification; a: Ambiguity; r: Right classification.

Table 4. Accuracy of our approach on the sport classification domain.

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

w: Wrong classification; a: Ambiguity; r: Right classification.

tree approach for the sport domain. Fewer and simpler rules were derived using FAQR than were derived using Yuan and Shaw's approach since the latter is based on decision trees. In decision-tree-based approaches, attributes, instead of attribute values, are chosen to grow. An attribute with many promising branches could then be chosen even though it has some undesired branches. Overspecialization is then usually seen in decision-tree approaches. For example, the selector "*Temperature* is *Mild*" in Rule a does not appear in Rule 2, the selector "*Temperature* is *Hot*" in Rule d does not appear in Rule 3, and the selector "*Temperature* is *Mild*" in Rule f does not appear in Rule 5.

Note that when an unknown example is classified by more than two rules, its class is determined by the one with the highest membership value. Thus, some general rules, such as Rule 1 in the sport domain, are not further specialized.

6.2. The IRIS Domain

The Iris problem is as follows: There are three species of Iris flowers to be distinguished: *Setosa*, *Versicolor*, and *Verginica*. There are 50 training instances for each class. Each training instance is described by four attributes: *Sepal Length* ($S.L.$), *Sepal Width* ($S.W.$), *Petal Length* ($P.L.$), and *Petal Width* ($P.W.$). All four of the attributes are numerical domains. The membership functions for each attribute used in this experiment are defined in Fig. 1.

Since the training set included only 150 instances, a method called *N-fold cross validation* [32] was adopted for this small set of samples. All instances were randomly divided into N subsets of as nearly equal size as possible. For each n, $n = 1, \ldots, N$, the n-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 ten subsets, each with fifteen instances composed of five positive training instances and ten negative training instances. The fuzzy learning algorithm then ran on training instances to derive promising rules. Finally, the most promising rules derived were then tested on the remaining data subset. Classification rates were then averaged across all possible groups. The set of rules derived using our approach was:

Rule 1: IF $P.L.$ is *Short*, THEN Iris Setosa ($u = 0.99$).
Rule 2: IF $P.L.$ is *Medium* and, THEN Iris Versicolor ($u = 0.89$).
Rule 3: IF $P.L.$ is *Long*, THEN Iris Virginica ($u = 0.97$).
Rule 4: IF $P.W.$ is *Wide*, THEN Iris Virginica ($u = 0.93$).

The average classification accuracy was 100% for *Setosa*, 98% for *Versicolor*, and 94% for *Virginia*. The accuracy of some other learning algorithms on the Iris Flower Classification Problem was examined in [33] by Hirsh for reference. The methods studied were Hirsh's Incremental Version Space Merging [33], Aha and Kibler's noise-tolerant NT-growth [34], Dasarathy's pattern-recognition approach [35], and Quinlan's C4 [8]. The accuracy of the generalized version-space learning algorithm (GVS) on the Iris Flower Problem was examined in [36] by Hong and Tseng. Table 5

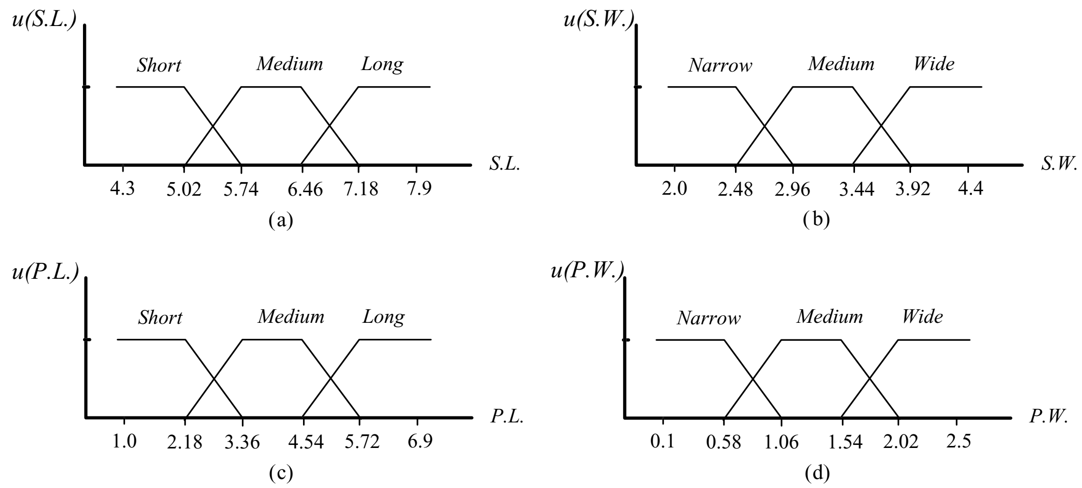

Figure 1. The given membership functions for each attribute in the IRIS domain.

Table 5. Accuracy of six learning algorithms on the iris flower problem.

Class	Setosa	Viginica	Versicolor	Average
Algorithm				
FAQR	100	96	94	96.67
GVS	100	94	94	96.00
IVSM	100	93.33	94.00	95.78
NTgrowth	100	93.50	91.13	94.87
Dasarathy	100	98	86	95.67
C4	100	91.07	90.61	93.89

compares the accuracy of our learning algorithm with those of the others. It can easily be seen that the accuracy of our method is as high as or higher than any of the other learning methods. Our approach cannot, however, be concluded to have the best performance among these approaches since the experimental conditions may be different. For example, our approach adopted the N-fold cross validation method, but the others could adopt different validation methods. More tests have to be done in the future.

6.3. *The CAI Domain*

In the CAI domain, there exist many learning records of students. These records can be used to assist teachers in analyzing the learning performance of students and refining the construction of teaching materials. In the experiment, 500 learning records are used as the training instances, each of which contains five quiz grades and a total grade, represented in numerical domains. The first 10 learning records are shown in Table 6.

The numerical grades of each quiz are then mapped to three performance levels, *Good*, *Average* and *Not Good*. The range between the upper bound and the lower bound of the grades of each quiz are equally divided to form the membership functions as shown in Fig. 2.

The set of fuzzy rules induced from the learning records by our proposed approach is shown below.

Rule 1: IF Grade of Quiz 4 is *Good*, THEN Learning Performance is *Good*.
Rule 2: IF Grade of Quiz 1 is *Average* and Grade of Quiz 2 is *Good* and Grade of Quiz 3 is *Good*, THEN Learning Performance is *Average*.
Rule 3: IF Grade of Quiz 1 is *Not Good*, THEN Learning Performance is *Not Good*.

Table 6. A partial set of training instances in the CAI domain.

Student ID	Q1	Q2	Q3	Q4	Q5	Total
1	12	14	18	3	9	56/100
2	10	6	12	6	7	41/100
3	3	6	6	1	5	21/100
4	8	10	8	2	8	36/100
5	16	18	20	20	20	94/100
6	0	3	3	1	4	11/100
7	1	8	6	4	10	29/100
8	2	3	3	0	3	11/100
9	12	16	14	4	14	60/100
10	6	8	12	2	10	38/100
:	:	:	:	:	:	:
:	:	:	:	:	:	:
Upper bound	16	18	20	20	20	94
Lower bound	0	3	3	1	3	11

Rule 4: IF Grade of Quiz 5 is *Good*, THEN Learning Performance is *Good*.
Rule 5: IF Grade of Quiz 2 is *Not Good* and Grade of Quiz 3 is *Not Good*, THEN Learning Performance is *Not Good*.
Rule 6: IF Grade of Quiz 4 is*Not Good*, THEN Learning Performance is *Not Good*.

The above fuzzy rules can be shown to teachers for analyzing the discriminating power of each quiz. For example, Rules 1 and 6 may show that Quiz 4 has a high discriminating power for test, because if students get good grades on Quiz 4, their learning performance is good, and if their grades on Quiz 4 is not good, then their learning performance is not good.

7. Discussion and Future Work

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

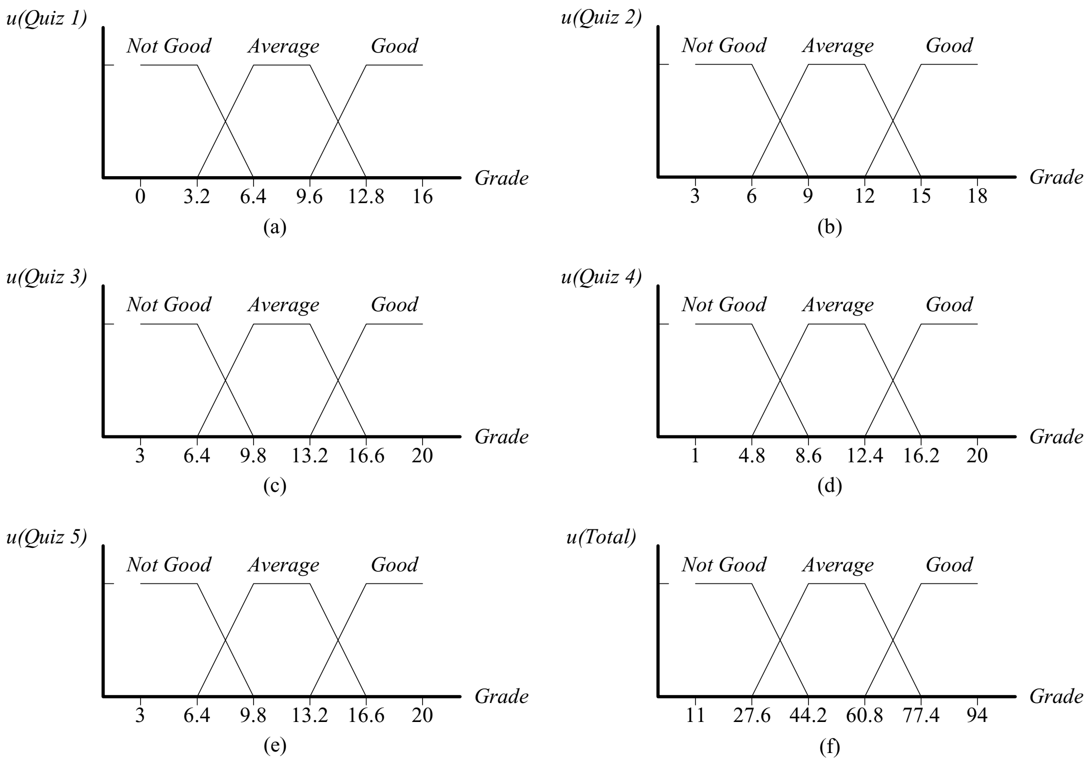

Figure 2. The membership functions for each quiz in the CAI domain.

Three parameters α, β and θ are used in the proposed algorithm. Their values will affect the coverage degree of examples, the execution time and the complexity of rules. A trade-off exists among these goals since they cannot be achieved at the same time. As a guidance, larger values of α and β will cause more certain fuzzy rules but less coverage of soft training examples. Larger θ values will increase the search breadth, and thus derive better rules with a higher possibility. The computational time will, however, accordingly increase along with larger θ values. Users can thus tune these parameters according to their learning goals.

Our method assumes that membership values of classes and attributes for training examples are known in advance. In [28–30], we proposed some fuzzy learning methods to automatically derive membership functions. Designing an integration approach to simultaneously learn rules and membership functions based on the proposed algorithm will then be focused in the future. We hope the bottleneck of membership function acquisition can thus be avoided.

Acknowledgment

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

References

1. Y. Kodratoff and R.S. Michalski, *Machine Learning: An Artificial Intelligence Approach*, vol. 3, Toiga: Palo Alto, CA, 1990.
2. R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, vol. 1, Toiga: Palo Alto, CA, 1983.
3. R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, vol. 2, Toiga: Palo Alto, CA, 1984.
4. R.S. Michalski and G. Tecuci, *Machine Learning: A Multistrategy Approach*, vol. 4, Morgan Kaufmann Publishers: San Francisco, CA, 1994.
5. R.S. Michalski, I. Bratka, and M. Kubat, *Machine Learning and Data Mining: Methods and Applications*, Wiley: Chichester, England, 1998.
6. J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *Int. J. of Man-Machine Studies*, vol. 27, pp. 349–370, 1987.

7. J.R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.
8. J.R. Quinlan, "C4.5: Programs for machine learning," 1993.
9. P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, vol. 3, pp. 261–283, 1989.
10. R.S. Michalski, "On the quasi-minimal solution of the general covering problem," in *Proceedings of the Fifth International Symposium on Information Processing*, 1969, pp. 25–128.
11. A. Giordana and L. Saitta, "Machine learning," in *Handbook of Fuzzy Computation*, edited by E. Ruspini, P. Bonissone, and W. Pedrycz, IOP Publishing Ltd., 1998.
12. T. Sudkamp and R.J. Hammell, "Interpolation, completion, and learning fuzzy rules," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 2, pp. 332–342, 1994.
13. Y. Kodratoff and M. Manago, "Generalization and noise," *Int. J. of Man-Machine Studies*, vol. 27, pp. 181–204, 1987.
14. A.F. Blishun, "Fuzzy Learning models in expert systems," *Fuzzy Sets and Systems*, vol. 22, pp. 57–70, 1987.
15. M. Botta, A. Giordana and L. Saitta, "Learning in uncertain environments," *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, pp. 281–296, 1992.
16. L.M. de Campos and S. Moral, "Learning rules for a fuzzy inference model," *Fuzzy Sets and Systems*, vol. 59, pp. 247–257, 1993.
17. M. Delgado and A. Gonzalez, "An inductive learning procedure to identify fuzzy systems," *Fuzzy Sets and Systems*, vol. 55, pp. 121–132, 1993.
18. A. Gonzalez, "A learning methodology in uncertain and imprecise environments," *Int. J. of Intelligent Systems*, vol. 10, pp. 357–371, 1995.
19. R.L.P. Chang and T. Pavliddis, "Fuzzy decision tree algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 28–35, 1977.
20. J. Rives, "FID3: Fuzzy induction decision tree," *The First International Symposium on Uncertainty Modeling and Analysis*, 1990, pp. 457–462.
21. Y. Yuan and M.J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 69, pp. 125–139, 1995.
22. T.M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, pp. 203–226, 1982.
23. C.H. Wang, T.P. Hong, and S.S. Tseng, "Inductive learning from fuzzy examples," *The Fifteenth IEEE International Conference on Fuzzy Systems*, pp. 13–18, 1996.
24. J. Kacprzyk and C. Iwanski, "Fuzzy logic with linguistic quantifiers in inductive learning," *Fuzzy Logic for the Management of Uncertainty*, pp. 465–478, 1992.
25. D. Haussler, "Quantifying inductive bias: A.I. learning algorithms and Valiant's learning framework," *Artificial Intelligence*, vol. 36, pp. 177–221, 1998.
26. P.H. Winston, *Artificial Intelligence*, Third Edition, Addison-Wesley Publishing, 1992.
27. C. Feng, "Fuzzy multicriteria decision-making in distribution of factories: An application of approximate reasoning," *Fuzzy Sets and Systems*, vol. 71, pp. 197–205, 1995.
28. T.P. Hong and J.B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems*, vol. 103, no. 3, pp. 389–404, 1999.
29. T.P. Hong and J.B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 127–140, 2000.
30. T.P. Hong and C.Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, vol. 84, pp. 33–47, 1996.
31. F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*, John Wiley, 1999.
32. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. stone, Classification and Regression Tree, 1984.
33. H. Hirsh, "Generalizing version spaces," *Machine Learning*, vol. 17, pp. 5–46, 1994.
34. D.W. Aha and D. Kibler, "Detecting and removing noisy instances from concept descriptions," Technical Report University of California, 1989, pp. 12–88.
35. B.V. Dasarathy, "Noising around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 67–71, 1980.
36. T.P. Hong and S.S. Tseng, "A generalized version space learning algorithm for noisy and uncertain data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 9, no. 2, pp. 336–340, 1997.

Ching-Hung Wang is currently working with Telecomm Laboratories of Chunghwa Telecomm Co., Ltd. He received the B.S. degree in computer and information science from National Chiao Tung University in 1990, and the M.S. degree in computer and information science from National Chiao Tung University in 1997. For his contribution of the Master Thesis, he was awarded the Long-Terng Award of Hong-Chi Computer Co., Ltd in 1990. He passed the "National Senior Officialdom Examination" in 1992 and became a licensed computer engineer ever since. In 1997, he received the Ph.D. degree in computer and information science from National Chiao Tung University.

His current research interests are intelligent transportation systems (ITS), electronic toll collection (ETC), machine learning, data mining, genetic algorithms, neural networks, and fuzzy logic. He is a member of Taiwanese Association for Artificial Intelligence and Taiwanese Association for Intelligent Transportation Systems.

Chang-Jiun Tsai is currently a Ph.D. candidate at the Institute of Computer and Information Science in the National Chiao Tung

University of Hsinchu, Taiwan, R.O.C. His research interests include artificial intelligence, expert system, machine learning, data mining, and computer-assisted learning. He received his B.S. in applied mathematics from National Chung Hsing University and his M.S. in computer and information science from National Chiao Tung University.

Tzung-Pei Hong received his B.S. degree in chemical engineering from National Taiwan University in 1985, and his Ph.D. degree in computer science and information engineering from National Chiao-Tung University in 1992. From 1992 to 1994, he was an associate professor at the Department of Computer Science in Chung-Hua Polytechnic Institute. He was an associate professor from 1994 to 1999 and a professor from 1999 to 2001 at the Department of Information Management in I-Shou University. He is currently a professor at the Department of Electrical Engineering in National University of Kaohsiung. He was also the director of the library and information center in National University of Kaohsiung, and was in charge of the whole computerization and library planning from the start. His current research interests include parallel processing, machine learning, data mining, soft computing, management information systems, and www applications. Dr. Hong is a member of the Association for Computing Machinery, the IEEE, the Chinese Fuzzy Systems Association, the Taiwanese Association for Artificial Intelligence, and the Institute of Information and Computing Machinery.

Shian-Shyong Tseng received his Ph.D. degree in Computer Engineering from the National Chiao Tung University in 1984. Since August 1983, he has been on the faculty of the Department of Computer and Information Science at National Chiao Tung University, and is currently a Professor there. From 1988 to 1991, he was the Director of the Computer Center at National Chiao Tung University. From 1991 to 1992 and 1996 to 1998, he acted as the Chairman of Department of Computer and Information Science. Form 1992 to 1996, he was the Director of the Computer Center at Ministry of Education and the Chairman of Taiwan Academic Network (TANet) management committee. In December 1999, he founded Taiwan Network Information Center (TWNIC) and is now the Chairman of the board of directors of TWNIC. His current research interests include parallel processing, expert systems, computer algorithm, and internet-based application.