中華民國八十四年全國計算機會議論文集
Proceedings of National Computer Symposium 1995

應用歸納學習策略由數值資料產生模糊規則
Generating Fuzzy Rules from Numerical Data by Inductive Learning Strategy

王景弘　　洪宗貝　　曾憲雄
Ching-Hung Wang[†‡], Tzung-Pei Hong[‡], and Shian-Shyong Tseng[†]

國立交通大學資訊科學研究所
Institute of Computer and Information Science National Chiao-Tung University[†]
私立高雄工學院資訊管理系
Department of Information Management Kaohsiung Polytechnic Institute[‡]
交通部電信研究所
Directorate General of Telecommunication Laboratories[‡]

摘要

本論文，我們企圖將模糊集合概念應用於機器學習領域。在此我們有幾點假設，（一）我們假設所有訓練例子的類別是具有正、負程度的隸屬性；（二）測試例子及概念之間可以做不精確性的推論；（三）訓練資料可以具有誤錯、不確定、及模糊訊息。基於上述的假設，一個新的歸納學習問題可以視為學習找出一個概念描述來含蓋大部份的正例及排除大部份的負例。所以，在此我們提出一個模糊學習法來處理例子有中具有誤錯、不確定、及模糊訊息的問題，並可結合不同的專家知識，產生最一致的概念空間。

關鍵詞：樣本空間，模糊樣本空間，歸納學習，分類

Abstract

In this paper, we have attempted to apply the concept of fuzzy sets to machine learning. We assume, first, that the classification into positive and negative examples in the training set is a degree of positiveness and negativeness between 0 and 1; second, that an inexact matching between a concept description and an example is allowed; and third, that data may contain wrong, uncertain, and linguistic information. A new inductive learning problem is then formulated so as to induce a concept description that covers almost all of positive examples and almost none of negative ones. Therefore, a fuzzy learning algorithm by the Version-Space strategy is proposed to manage wrong, uncertain, and linguistic information under imprecision and noise environments.

Keywords: version space, fuzzy version space, inductive learning, classification.

1. Introduction

Among the machine learning approaches [9], *inductive learning* from examples may be the most commonly used in many real world application domains. Inductive learning is a process of inferring a concept description that describes all positive examples and excludes all negative examples. These traditional inductive learning procedures are however inapplicable to some application domains, since data in the real world usually contain wrong, uncertain, or linguistic information. A ***crisp classification*** to distinguish positive and negative examples is often artificial; instead, fuzzy or ambiguous classification of examples is commonly seen in the real world.

In real applications, data provided to learning systems by experts, teachers, or users usually contain ***wrong***, ***uncertain***, and ***linguistic*** information. Wrong, uncertain, and linguistic information will in general greatly influence the information and use of the concepts derived. Modifying traditional inductive learning methods to work well in noisy and vague environments is then very important. Several successful learning strategies based on ID3 have

been proposed [11][12][13]; most of these used tree-pruning and fuzzy logic techniques. As to Version-Space based learning strategies, Mitchell proposed a multiple version space learning strategy for managing wrong information [9]. Hirsh handled wrong information by assuming attribute values had a known bounded inconsistency [6]. In [7], Hong and Tseng proposed a generalized version space learning algorithm to manage both noisy and uncertain data.

Some kinds of problems for inductive learning arising in a vague environment were discussed in [5][8]. How to avoid the problems arising in learning concept descriptions affected by noise, uncertainty and vagueness is very important. Among these solutions, the fuzzy set theory seems as an appropriate approach to handle these problems. In this paper, we will propose a fuzzy learning algorithm based on the Version-Space strategy for inducing a fuzzy rule base from both numerical and linguistic information which are respectively obtained from sensor measurements and human experts. The learning approach is basically an extension of the generalized version space learning algorithm [7], and can overcome problems of inductive learning in the noisy, uncertain and vague learning environments.

The remaining part of this paper is organized as follows. The learning strategy of Version Space, is first reviewed in Section 2. Some related concepts of the fuzzy theory are reviewed in Section 3. The fuzzy Version Space learning algorithm is proposed in Section 4. Experiments are made on the IRIS flower classification problem in Section 5. Conclusions are finally given in Section 6.

2. Review of the Version-Space Learning Strategy

The Version-Space learning strategy [9][10], was proposed by Mitchell in 1978. It is mainly used for learning from training instances of only two classes: positive and negative. It then attempts to induce concepts which include all positive training instances and exclude all negative training instances. The term "***version space***" is used to represent all legal hypotheses describable within a given concept description language and consistent with all observed training instances. The term ***"consistent"*** means that each hypothesis includes all given positive training instances and excludes all given negative ones. A version space can then be represented by two sets of hypotheses: set S and dual set G, defined as:

S={s | s is a hypothesis consistent with observed instances. No other hypothesis exists which is both more specific than s and consistent with observed instances};

G={g | g is a hypothesis consistent with observed instances. No other hypothesis exists which is both more general than g and consistent with observed instances}.

Sets S and G, together, precisely delimit the version space in which each hypothesis is both more general than some hypothesis in S and more specific than some hypothesis in G. When a new positive training instance appears, set S is generalized for including this training instance; when a new negative training instance appears, set G is specialized for excluding this training instance

Unfortunately, Version Space only works well in the ideal domains where no noisy, uncertain, or linguistic information exists in the data. When the above information appears, the version space derived is usually null, thus providing no classification information at all. However, the effective use of learning systems in real world applications substantially depends upon their capability in handling noisy, uncertain, and linguistic information. In the paper, we then apply the concept of fuzzy sets to the version-space learning strategy for solving the above problem.

3. Review of Related Fuzzy Concepts

A fuzzy set is an extension of the 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 partially belong to a 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

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

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

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

where X refers to the universal set defined in 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 as follows:

(1)The *intersection* operator: $u_{A\cap B}(x) = u_A(x)\ \tau\ u_B(x)$, where τ : [0, 1] * [0, 1] $\rightarrow$ [0, 1] is a *t-norm* operator satisfying the following conditions [8]:

for each $a, b, c \in [0, 1]$:
(a) $a\ \tau\ 1 = a$
(b) $a\ \tau\ b = b\ \tau\ a$
(c) $a\ \tau\ b \geq c\ \tau\ d$ if $a \geq c, b \geq d$
(d) $a\ \tau\ b\ \tau\ c = a\ \tau\ (b\ \tau\ c) = (a\ \tau\ b)\ \tau\ c$

Some examples for 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 ρ: [0, 1] * [0, 1] $\rightarrow$ [0, 1] is a *s-norm* operator satisfying the following conditions [8]:

for each $a, b, c \in [0, 1]$:
(a) $a\ \rho\ 0 = a$
(c) $a\ \rho\ b \geq c\ \rho\ d$ if $a \geq c, b \geq d$
(b) $a\ \rho\ b = b\ \rho\ a$
(d) $a\ \rho\ b\ \rho\ c = a\ \rho\ (b\ \rho\ c) = (a\ \rho\ b)\ \rho\ c$

Some examples for a *s-norm* operator $a\,\rho\,b$ are $\max(a, b)$ and $a + b - a * b$.

(3) The $\alpha-cut$ operator: $A_\alpha(x) = \{x \in X | u_A(x) \geq \alpha\}$, where A_α is an $\alpha-cut$ of a fuzzy set A. A_α contains all the elements of the universal set X that have a membership grade in A greater than or equal to the specified value of α.

In the next section, these fuzzy concepts and operators will be used in our learning algorithm to model the vague knowledge.

4. Fuzzy Version-Space Learning Strategy

Most inductive learning approaches begin with finding the appropriate training examples. For achieving this purpose, the attributes must first be determined, and their feasible values must be established. These values of attributes can be determined either by consulting the experts or from sensor measurements. The values of attributes can either be numerical (continuous or discrete) or symbolic (linguistic or crisp). If the values of attributes are numerical, it may be necessary to discretize the original values into a number of possible regions for traditional inductive learning methods to work with.

Recently, the fuzzy logic is commonly used to transform a numerical domain into a number of fuzzy regions (also called *fuzzy* or *linguistic labels*). The use of the fuzzy set theory to replace the traditional discretization techniques has at least the following advantages:

1. Each fuzzy region can be interpreted from a linguistic or semantic point of view; it can then be understood by the human.
2. It usually provides a small number of fuzzy labels in the transformed domain, thus making the learning and reasoning simple.
3. It can help us effectively deal with noise in the data.

In the next section, we will propose a fuzzy transformation technique which can divide the continuous values into a number of fuzzy labels.

4.1. Generate Linguistic Cases from Numerical Data

Suppose we obtain a set of desired input-output data pairs from sensor measurements as in Table 1.

Table 1 Input-output data pairs

attributes / examples	A_1	A_2	...	A_m	CLASS
instance e_1	$a_1^{(1)}$	$a_2^{(1)}$	...	$a_m^{(1)}$	$c^{(1)}$
instance e_2	$a_1^{(2)}$	$a_2^{(2)}$	...	$a_m^{(2)}$	$c^{(2)}$
			⋮		
instance e_n	$a_1^{(n)}$	$a_2^{(n)}$	...	$a_m^{(n)}$	$c^{(n)}$

In Table 1, $a_1^{(i)}, a_2^{(i)}, \ldots, a_m^{(i)}$ respectively being the values of attributes $A_1, A_2, \ldots, A_m$ with continuous domains form the input of the i-th example, and $c^{(i)}$ is the output part of the *i-th* example. The task here is to transform the numerical domain into a fuzzy domain. The approach applied here consists of the following four steps.

Step 1: Transform the numerical domain into a number of fuzzy regions

Assume that the domain interval of attribute A_i in the training data is $[a_i^-, a_i^+]$, where a_i^- is the minimum value of A_i and a_i^+ is the maximum value of A_i. Each domain interval is divided into K regions denoted by $R_{i,1}, R_{i,2}, \cdots, R_{i,K}$, where K is arbitrarily assigned by the builder and may be different for different attributes. The length of the regions is not necessarily equal. Each region is then assigned a fuzzy membership function. Let $D_i = \{R_{i,1}, R_{i,2}, \cdots, R_{i,K}\}$ be a fuzzy domain of attribute A_i, a_i^- and a_i^+ will then respectively fall in $R_{i,1}$ and $R_{i,K}$. Fig. 2 shows an example where the domain interval of attribute A_i is divided into five regions. These regions together form a fuzzy domain $D_i = \{R_{i,1}, R_{i,2}, \cdots, R_{i,5}\}$. Note that the shape of each membership function is not necessarily triangular. The Cartesian product $D = D_1 * D_2 * \ldots * D_m * D_{class}$ then forms the complete fuzzy domain of examples.

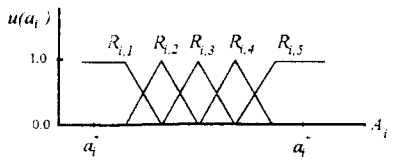

Fig. 2 Transforming the numerical domain of A_i into a fuzzy domain.

Step 2: Generate linguistic cases from given data pairs

This step transforms each training instance represented by numerical values into the ones represented by fuzzy terms. Assume one instance e_k is given :

$$e_k : (a_1^{(k)}, a_2^{(k)}, \cdots, a_m^{(k)}, c^{(k)}).$$

e_k is then transformed from the numerical domain into the fuzzy domain according to the fuzzy regions of each attribute as follows:

$$e_{kl}: ([R_{1,1}, u_{R_{1,1}}(a_1^{(k)})], \ldots, [R_{m,1}, u_{R_{m,1}}(a_m^{(k)})], c^{(k)}),$$

$$\vdots$$

$$e_{kd}: ([R_{1,|D_1|}, u_{R_{1,|D_1|}}(a_1^{(k)})], \ldots, [R_{m,|D_m|}, u_{R_{m,|D_m|}}(a_m^{(k)})], c^{(k)})$$

, where $d = |D_1| * \ldots * |D_i| * \ldots * |D_m|$, $R_{i,j} \in D_i$, $1 \le j \le |D_i|$, $c^{(k)} \in D_{class}$, and $u_{R_{i,j}}(a_i^{(k)})$ is the membership function of label $R_{i,j}$ for attribute A_i given $a_i^{(k)}$.

Step 3: Assign a membership value to each case

In this step, we will decide a membership value for an instance described by fuzzy terms according to its membership values of attributes. Assume that a linguistic case e_{kj} is given:

$$e_{kj}: ([R_{1,j}, u_{R_{1,j}}(a_1^{(k)})], \ldots, [R_{m,j}, u_{R_{m,j}}(a_m^{(k)})], c^{(k)}).$$

The fuzzy intersection operator is then used to find a membership value $u_c(k)(e_{kj})$ of each instance e_{kj} as follows:

$$u_c(k)(e_{kj}) = u_{R_{1,j}}(a_1^{(k)}) \ \tau \ \ldots \ \tau \ u_{R_{m,j}}(a_m^{(k)}),$$

where τ is a *t-norm*.

Step 4: Remove the redundancy and irrelevant cases

There are usually lots of data pairs in the original training set and several linguistic cases are generated for each data pair, so that in the fuzzy domain redundant cases (i.e., cases that have the same input and output) and ambiguous cases (i.e., cases that have the same input but a different output) may exist with quite a high probability. These cases respectively form a redundant training group ***R*** and an ambiguous training group ***G***. In order to remove the redundancy in the redundant group and

irrelevant cases in the ambiguous group, a solution applied here consists of the following two phases.

Phase 1 : Remove the redundancy:

If redundant groups exist in the linguistic training cases generated, the fuzzy union operator ρ is used to remove the redundancy. For example, if the maximum operator is chosen as the ρ operator, then in each redundant group only the case with the maximum membership value is kept, and the others in the redundant group are removed .

Phase 2 : Remove irrelevant cases.

After the process of removing redundancy, if ambiguous groups still exist in the linguistic training cases, then a fuzzy operation $\alpha - cut$ is used to remove irrelevant cases in each ambiguous group. Restated, in each ambiguous group only the cases with membership values greater than or equal to a predefined threshold α are kept.

These survived cases of each ambiguous group G form a new group G'. Although, the group G' may still contain some inconsistent cases, this way can prune some irrelevant cases and reduce the number of cases in the ambiguous group G. The inconsistency problem is then solved by the proposed fuzzy learning algorithm in the next section.

Through STEP 1 to STEP 4, the original input-output data pairs are then transformed into a set of linguistic cases with the membership values for belonging to the class. The *linguistic* case e_{kj} generated is then described as follows:

e_{kj}: $(R_{1,j}^{(k)}, \cdots, R_{i,j}^{(k)}, \cdots, R_{m,j}^{(k)}, c^{(k)})$

, with the membership value $u_c(k)(e_{kj})$.

4.2 Learning in Vague Environments

The previous section presented a preprocessing for generating linguistic cases from numerical data. This section then proposes a fuzzy version space learning algorithm that can successfully handle training instances with wrong, uncertain, and linguistic information. The proposed algorithm can also easily make a trade-off between including "*soft*" positive training instances and excluding "*soft*" negative training instances or a trade-off between the computational time consumed and accuracy of the final results. *Soft instances* differ from conventional instances in that they are with *degrees of truth* for positiveness and negativeness. The membership function $u_{c^+}(e)$ specifies the degree to which instance e belongs to the positive class c^+, and the membership function $u_{c^-}(e)$ specifies the degree to which instance e belongs to the negative class c^-. Sets S and G defined in the original version space learning strategy are modified to provide these additional functions. In the proposed method, the hypotheses in S/G no longer necessarily *include/exclude* all the *positive/negative* instances presented so far, since noisy, uncertainty and linguistic information exist in the training set. Instead, the most consistent i hypotheses are maintained in the S set and the most consistent j hypotheses are maintained in the G set (i and j are two parameters defined by the user). The fuzzy measure function H_S/H_g referred to as the *true of including the positiveness information/the true of excluding the negativeness* is attached to each hypothesis in S/G to summarize all positive/negative information implicit in the linguistic cases presented so far.

A hypothesis s with a higher $H_S(s)$ in S represents more the truth to include "*soft*" positive training instances; a hypothesis g with a higher $H_g(g)$ in G represent more the truth to exclude "*soft*" negative training instances. The new sets S and G are then redefined as follows:

S={s | s is a hypothesis among the first i maximally consistent hypotheses. No other hypothesis s' in S exists which is both more specific than s and $H_S(s') \geq H_S(s)$};

G={g | g is a hypothesis among the first j maximally consistent hypotheses. No other hypothesis g' in G exists which is both more general than g and $H_g(g') \geq H_g(g)$}.

The maximum number of hypotheses maintained in S here is i, and the number in G is j. The first i/j maximally consistent hypotheses in S/G, however, are not necessarily the ones with the largest H_S/H_g. A hypothesis in S that includes much positive information may possibly include much negative information. A hypothesis in G that

excludes much negative information may also possibly exclude much positive information. Clearly these kinds of hypotheses are not necessarily better than the ones in S, which include both less positive information and less negative information. Which hypotheses in S/G are the first i/j maximally consistent then depends on both sets S and G (and not only on S itself or on G itself).

For the proposed learning algorithm to make a trade-off between including "*soft*" positive training instances and excluding "*soft*" negative ones according to the requirements of specific learning problems, two parameters respectively called the *factor of including positive instances* (*FIPI*) and the *factor of excluding negative instances* (*FENI*) are incorporated into the algorithm. The values of *FIPI* is 1 if the aim of the learning problem must be to include all of "*soft*" positive training instances and 0 if the aim of the learning problem is not to include any of "*soft*" positive ones. The values of *FENI* is 1 if the aim of the learning problem must be to exclude all of "*soft*" negative training instances and 0 if the aim of the learning problem is not to exclude any of "*soft*" negative ones. The values of *FIPI* and *FENI* are usually between 0 and 1. We then define the fuzzy measure function H_{sg}/H_{gs} to evaluate the *truth of classification* for each hypothesis in S/G according to the requirements of specific learning problems.

The fuzzy version space learning strategy consists of two main phases: searching and pruning. The searching phase generates and collects possible candidates into a large set; the pruning phase then prunes this set according to the *truth of classification* of each hypothesis in the boundary sets. The same procedure is repeated until all training instances have been processed. Finally, two sets of hypotheses S and G are obtained, and the hypotheses with the first k higher *truth of classification* in S and G are chosen as desired hypotheses. Here, k is determined by users, and k must smaller than or equal to the minimum value of i and j. Below, we shall introduce the fuzzy version space algorithm as follows:

Fuzzy Version Space Learning Algorithm:

INPUT: A set of n linguistic training instances, each one with $u_c(e_i)$, $i = 1, \cdots, n$, the values of the parameters *FIPI* and *FENI*, and the maximum numbers i, j of the hypotheses maintained in S and G.

OUTPUT: The hypotheses with the first k highest *truth of classification* in sets S and G are output as desired rules.

1: If appropriate human knowledge expressed by linguistic hypotheses is available, then adds the linguistic hypotheses to both S and G. The initial the *truth of including the positiveness/truth of excluding the negativeness* of each newly added hypothesis h_{sj} / h_{gj} to S/G is assigned p_{sj} / p_{gj}.

2: If human knowledge does not exist in this domain, then initialize S to contain only the most specific hypothesis ϕ with $H_S(\phi) = 0$ and initialize G to contain only the most general hypothesis ∞ with $H_g(\infty) = 0$ in the whole hypothesis space.

3: If the presented instance e_i is "*soft*" positive, then do the following steps:

a: Generalize each hypothesis s with $H_S(s)$ in S to include the new training instance e_i; attach the new *truth of including the positiveness* to each newly formed hypothesis s' in S, where $H_S(s') = H_S(s) \rho\, u_{C+}(e_i)$; call the newly formed set S' for convenience.

b: Find the set S'' including only the new training instance e_i itself, and set the *truth of including the positiveness* of each hypothesis s'' in S'' to be $u_{C+}(e_i)$; i.e. $H_S(s'') = u_{C+}(e_i)$.

c: Combine the original S, S', and S'' together to form a new S. If identical hypotheses with different *truth of including the positiveness* are present in the combined set, only the hypothesis with the *maximum truth of including positiveness* is retained. If a particular hypothesis is both more *general* than another and has an equal or smaller *truth of including the positiveness*, discard that hypothesis.

d: For each hypothesis s with $H_S(s)$ in the new S, find the hypothesis g in the new G that is more *general* than s and has the *maximum truth of excluding negativeness* $H_g(g)$. Calculate the *truth of classification* as $H_{sg}(s) = (H_s(s) \tau FIPI) \rho (H_g(g) \tau FENI)$ for each hypothesis s in S.

e: Retain the hypotheses with the first i highest *truth of classification* H_{sg} in the new S, and discard the others.

4: If the presented instance e_i is "*soft*" negative, then do the following steps:

f: Specialize each hypothesis g with $H_g(g)$ in G to exclude the new training instance e_i ; attach the new *truth of excluding the negativeness* to each newly formed hypothesis g' in G, where $H_g(g') = H_g(g)\ \rho\ u_c\text{-}(e_i)$; call the newly formed set G' for convenience.

g: Find the set G'' excluding only the new training instance itself, and set the *truth of excluding the negativeness* of each hypothesis g'' in G'' to be $u_c\text{-}(e_i)$; i.e. $H_g(g'') = u_c\text{-}(e_i)$.

h: Combine the original G, G', and G'' together to form a new G. If identical hypotheses with different *truth of excluding the negativeness* are present in the combined set, only the hypothesis with the *maximum truth of excluding the negativeness* is retained. If a particular hypothesis is both more *specific* than another and has an equal or smaller *truth of excluding the negativeness*, discard that hypothesis.

i: For each hypothesis g with $H_g(g)$ in the new G, find the hypothesis s in the new S that is more *specific* than g and has the *maximum* truth of including the positiveness $H_s(s)$. Calculate *the truth of classification* as $H_{gs}(g) = (H_s(s)\,\tau\,FIPI)\,\rho\,(H_g(g)\,\tau\,FENI)$ for each hypothesis g in G.

j: Retain the hypotheses with the first j highest *truth of classification* H_{gs} in the new G, and discard the others.

5: When there are still new training instances to be processed, go to STEP 3; otherwise, stop the learning process, and choose the hypotheses with first k highest *truth of classification* in S and G as the desired rules.

When the learning process terminates, the hypotheses in sets S and G that result in the first k highest *truth of classification* are output to form the version space, which can then be thought of as being maximally consistent with the training instances.

5. Experiments

To demonstrate the effectiveness of the proposed fuzzy version space learning algorithm, we applied it to classify Fisher's Iris Data [4] which contains 150 training instances. The data are inconsistent, so the original version space learning algorithm yield a null final version space, providing no information at all.

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. All values of these four attributes are continuous, and the ranges of values for the attributes *S.L*, *S.W*, *P.L*, and *P.W* are respectively [4.3, 7.9], [2.0, 4.4], [1.0, 6.9], and [0.1, 2.5]. Assume the domain interval of each attribute is divided into three regions defined as follows:

$D_{S.L}$ ={ *Short, Medium, Long* }
$D_{S.W}$ ={ *Narrow, Medium, Wide* }
$D_{P.L}$ ={ *Short, Medium, Long* }
$D_{P.W}$ ={ *Narrow, Medium; Wide* }.

Fig. 3 shows an example of membership functions for each attribute (here we use triangular membership functions).

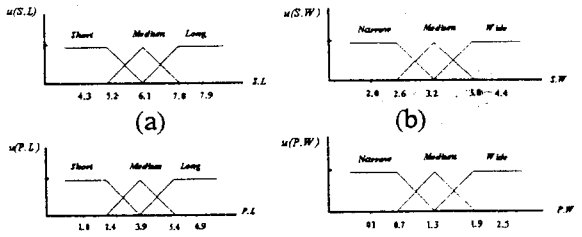

Fig. 3 Transforming the input spaces of Iris data into fuzzy regions

Since the training set includes only 150 instances, a method called *N-fold cross validation* [2] is adopted for this small set of samples. All instances are randomly divided into N subsets of as nearly equal size as possible. For each n, n=1, .., N, the n-th subset is used as a test set, and the other subsets are combined as a training set. In this domain, the data are partitioned into ten subsets, each with fifteen instances composed of five positive training instances and ten negative training instances.

Since all values of four attributes are continues, all training instances in the nine of ten data subsets need to be fuzzified into a set of linguistic cases by a fuzzy transformation technique. The fuzzy learning algorithm is then trained using the linguistic cases, and it tests the most promising version space derived on the remaining data subset. This is done for each set of linguistic cases generating from randomly nine data subsets, and classification rate are averaged across all then ten possible groups.

The fuzzy version space learning algorithm was implemented in C language on an SUN SPARC/2 workstation. The algorithm was run 100 times, using different random partitions of the sample set. The accuracy of some other learning algorithms on the Iris Flower Classification Problem was examined in [6] by Hirsh. The methods studied were Hirsh's Incremental Version Space Merging [6], Aha and Kibler's noise-tolerant NT-growth [1], Dasarathy's pattern-recognition approach [3], and Quinlan's C4 [12]. Table 2 compares the accuracy of our learning algorithm with that of the others. It can easily be seen that our method is as accurate as GVS, and is higher than other learning methods. Beside, the rules induced from the proposed algorithm are easily understood by the human since they are linguistic rules. Experimental results show that our method yield a high accuracy.

Table 2. Predictive accuracy of six learning algorithms

algorithm \ class	Setosa	Viginica	Versicolor	Average
FVS (I=10,J=25)	100	92	98	96.66
GVS (I=5, J=20)	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	94.67
C4	100	91.07	90.61	93.89

6. Conclusion

In this paper, we propose a fuzzy version space learning algorithm to generate a version space from numerical data and linguistic information. Since we can easily transform a fuzzy version space to a set of fuzzy rules, this can be considered as a method to generate fuzzy rules from a set of numerical data by the inductive learning approach. This approach can overcome problems of the inductive learning caused by noisy, uncertain and linguistic information. It can manage noisy, uncertain, and linguistic instances and finds a maximally consistent version space. Experimental results show that our method yield accuracy as high as that of some other learning algorithms. It is a flexible and efficient fuzzy inductive learning method.

REFERENCE

[1] D. W. Aha and D. Kibler, "Detecting and removing noisy instances from concept descriptions," *Technical Report University of California*, 1989, pp. 12-88.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. stone, *Classification and Regression Tree*, 1984.

[3] B. V. Dasarathy, "Noising around the neighborhood: a new system structure and classification rule for recognition in partially exposed environments," *PAMI*, Vol. 2, 1980, pp. 67-71.

[4] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annual Eugenics*, Vol. 7, 1936, pp. 179-188.

[5] A. Gonzalez, "A learning methodology in uncertain and imprecise environments," *Int. J. of Intelligent Systems*, Vol. 10, 1995, pp. 357-371.

[6] H. Hirsh, "Generalizing version spaces, *Machine Learning*, Vol. 17, 1994, pp. 5-46.

[7] T. P. Hong and S. S. Tseng, "A generalized version space learning algorithm for noisy and uncertain data," *IEEE Transactions on Knowledge and Data Engineering* (accepted).

[8] J. Kacprzyk and C. Iwanski, "Fuzzy logic with linguistic quantifiers in inductive learning," *Fuzzy Logic for the Management of Uncertainty*, 1992, pp. 465-478.

[9] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, Vol. 18, 1982, pp. 203-226.

[10] T. M. Mitchell, "Version Space: an approach to concept learning," *Ph.D. Thesis*, Stanford University, 1978.

[11] J. R. Quinlan, *C4.5: programs for machine learning*,1993.

[12] J. R. Quinlan, "Decision tree as probabilistic classier," *Proceeding of the Fourth International Machine Learning Workshop*, 1987, pp. 31-37.

[13] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, Vol. 69, 1995, pp. 125-139.