Learning Concepts by Arranging Appropriate Training Order

YAO-TUNG HSU[1], TZUNG-PEI HONG[2] and SHIAN-SHYONG TSENG[3]
[1]*Department of Information Management, Overseas Chinese Institute of Technology, Taichung, Taiwan, R.O.C.; E-mail: yth@vsz.ocit.edu.tw; and Department of Computer and Information Science, National Chiao-Tung University, Hsinchu, 30050, Taiwan, R.O.C.;* [2]*Department of Information Management, I-Shou University, Kaohsiung 84008, Taiwan, R.O.C.; E-mail: tphong@isu.edu.tw;* [3]*Department of Computer and Information Science, National Chiao-Tung University, Hsinchu, 30050, Taiwan, R.O.C.; E-mail: sstseng@cis.nctu.edu.tw*

Abstract. Machine learning has been proven useful for solving the bottlenecks in building expert systems. Noise in the training instances will, however, confuse a learning mechanism. Two main steps are adopted here to solve this problem. The first step is to appropriately arrange the training order of the instances. It is well known from Psychology that different orders of presentation of the same set of training instances to a human may cause different learning results. This idea is used here for machine learning and an order arrangement scheme is proposed. The second step is to modify a conventional noise-free learning algorithm, thus making it suitable for noisy environment. The generalized version space learning algorithm is then adopted to process the training instances for deriving good concepts. Finally, experiments on the Iris Flower problem show that the new scheme can produce a good training order, allowing the generalized version space algorithm to have a satisfactory learning result.

Key words: entropy, machine learning, noise, training instance, training order, version space

1. Introduction

Machine learning is currently in a period of rapid growth. The increasing success of expert systems has emphasized the importance of developing efficient methods for acquiring domain knowledge. Conventional knowledge acquisition methods by querying experts often result in inexact, incomplete, and inconsistent knowledge bases (Feigenbaum, 1977). The required time for development is also quite long. Obtaining domain knowledge through methods of machine learning offers an attractive alternative for quickly building a prototype knowledge base.

Among the topics in machine learning, developing concept descriptions from training examples is the most common. Given a set of examples and counter-examples of a concept, the learning program tries to induce general concept descriptions that describe all of the positive training instances and none of the negative ones (Hong and Tseng, 1994).

The validity and relevance of the final concept derived heavily depends on the training instances and their accuracy. In real applications, data provided to learning mechanisms by experts, teachers, or users usually contain noise, as follows (Kodratoff et al., 1987):

Minds and Machines **11:** 399–415, 2001.

1. *Wrong information:* Some values or classes in training instances may be given incorrectly. Incorrect training instances may originate from unreliable or incorrect information, or they may arise from input errors. There are two ways in which training instances may be classified incorrectly. An originally positive training instance which is wrongly classified into the negative class is called a *false negative training instance*. An originally negative training instance which is wrongly classified into the positive class is called a *false positive training instance*. Attribute values of training instances may also be given incorrectly. For example, "size=20" might be wrongly given as "size=10".

2. *Uncertain information:* Some training instances may be given with uncertainty. Uncertain training instances most commonly occur when the attributes applied are insufficient to appropriately describe the training instances, or when experts, teachers, or users are not quite sure in what class a given training instance should be placed.

Regardless of the types, noise will in general greatly influence the formation and use of the concepts derived. Modifying traditional learning methods to work well in noisy environments is then very important. Some successful learning strategies; suitable in noisy environments, have been developed over the past few years. Several successful learning strategies based on the 1D3 learning algorithm have been proposed (Clark and Niblett, 1989; Mingers, 1989; Quinlan, 1986 1987, 1992). Most of these use tree-pruning techniques to cope with the problem of overfitting. As for version-space-based learning strategies, Hirsh handled noisy information by assuming attribute values had a known bounded inconsistency (Hirsch, 1994). Hong and Tseng removed the assumption and proposed a generalized version space learning algoritlun to derive concepts in noisy environments (Hong, 1992; Hong and Tseng, 1997). Some other studies on effective noise management are still being developed.

It is well known that people can effectively learn in imperfect environments. Observing human learning bbhavior therefore provides a useful clue for machine learning in real-world applications (Chang et al., 1996). The order in which training instances are presented to people is important to the final learned results (Hall, 1989; Baddeley, 1990). Different orders of presentation of the same set of training instances may cause different learning results. This idea is used here for machine learning. Just like the process of human learning, typical and relevant training instances should be fed into a learning mechanism as soon as possible to quickly build core concepts. Once the core concepts are built, they cannot be easily changed by the noise which comes later.

In this paper, the effect of training order on human learning is first illustrated using six examples. A useful order arrangement scheme is then proposed to sort the given training instances for machine learning. The set of training instances are arranged according to their classes and the principle attribute, which is found by using the idea of entropy (Quinlan, 1983). These training instances are then fed into the simplified generalized version space learning algorithm (Hong, 1992;

Figure 1. An ambiguous picture in Example 1.

Hong and Tseng, 1997) to derive concepts. Finally, experiments on the Iris Flower problem will be made, with results showing that the new scheme can produce a good training order, which facilitates the generalized version space algorithm providing satisfactory learning results.

2. Effect of Training Order on Human Learning

In this section, we use several examples to describe the effect of training order on human learning. Examples 1 and 2 show that people can grasp the important characteristics of a concept from a sequence of presented data. Example 3 shows the importance of the training order. It also shows that the first several training instances are critical in forming a core concept. Interleaving different kinds of training instances thus makes classification become ambiguous, which is shown in Example 4. Examples 5 to 7 show that the classification accuracy decreases along with the increase of noise, but if a core concept has been formed, a small amount of noise has only a little effect on it. The order in which training data is presented is thus important to human learning. Especially, correct training instances should be provided to the learner as soon as possible for forming a core concept. Also, noisy instances should be removed as possible as they can.

Example 1: Given a cartoon picture shown in Figure 1, its contents are to be identified. Since it is a little ambiguous, some people may think that it is the face of a man, and some others may think that it is the body of a woman (Chang, 1992). Figure 1 was shown to fifty students for experiments. Among them, 80% thought it was the face of a man and 20% thought it was the body of a woman. From this only picture, we can not correctly classify the picture since the useful part is hard to distinguish from the noisy part.

Example 2: Given the four cartoon pictures shown in Figure 2 (a) to (d), the contents in (d) (which is the same as Figure 1) are to be identified. When these pictures are examined in the sequence of (a) to (d), the contents in (d) will be recognized very likely as the face of a man. Figure 2 was shown in the order of (a) to (d) to fifty students for experiments. All the students thought it was the face of a man.

Example 2 shows that people can grasp the important characteristics of a concept from a sequence of presented data and ignore the dispensable details. This is also

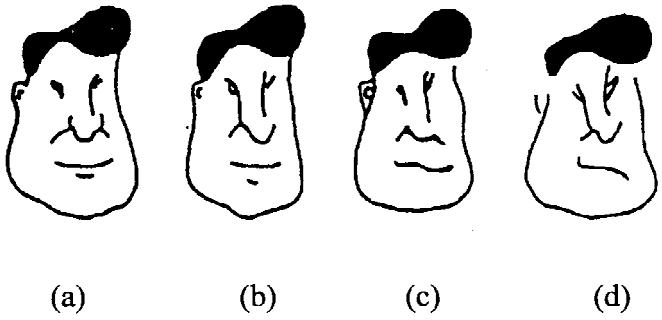

Figure 2. A sequence of pictures of a man's face in Example 2.

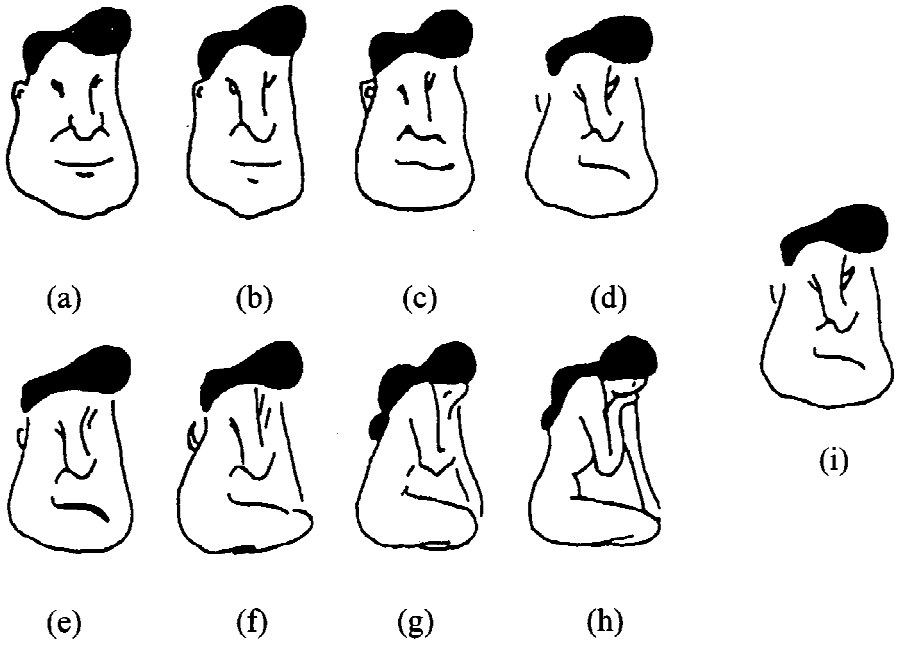

Figure 3. Effect of different orders in Example 3.

the principle for machine learning. An ambiguous or uncertain data thus needs additional information to make a good classification. Example 2 shows that the remaining pictures and their sequence provide such additional information in classifying the picture in Figure 1.

Example 3: Given the nine cartoon pictures shown in Figure 3 (a) to (i), the contents in (i) (which is the same as Figure 1) are to be identified. Figure 3 was shown in the orders of (a) to (h) and (h) to (a) to fifty students for experiments. When Figure 3 was shown in the order of (a) to (h), 88% of the students thought the contents in (i) were the face of a man. When these pictures are examined in the

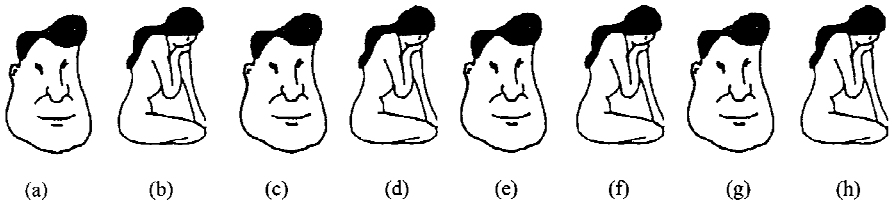

Figure 4. An interleaved sequence of pictures of a man's face and a woman's body in Example 4.

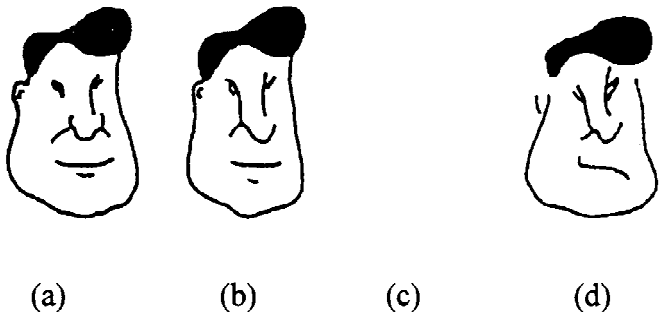

Figure 5. An incomplete sequence of pictures in Example 5.

sequence of (h) to (a), 64% of the students thought the contents in (i) were the face of a man.

Example 3 shows the training order is very critical to the final classification by people. Different orders of presentation of the training instances may cause different classification results and the first several training instances are critical in forming a core concept. Similar experimental results are also shown by Fisher (1967), and called the effects of context (Atkinson et al., 1990).

Example 4: Given the eight cartoon pictures shown in Figure 4 (a) to (h), the contents in Figure 1 are to be identified. Pictures in Figure 4 (a), (c), (e) and (f) are the same as that in Figure 3 (a). Pictures in Figure 4 (b), (d), (f) and (h) are the same as that in Figure 3 (h). Restated, the pictures for a man's face and for a woman's body are interleaved in the sequence. Figure 4 was then shown in the order of (a) to (h) to fifty students for experiments. Among them, 62% thought it was the face of a man, lower than 80% for Figure 1. Interleaving training examples of different concepts makes the recognition ambiguous. Psychologists have also shown that interleaving different kinds of examples is an important factor in the failure of recall (Solso, 1988).

Example 5: Given the three cartoon pictures shown in Figure 5, the contents in (d) (which is the same as Figure 1) are to be identified. These three pictures are the same as those in Figure 2 (a), (b), (d). Picture (c) is not included in Figure 5. Figure 5 was shown in the order of (a), (b), (d) to fifty students for experiments. All the students thought it was the face of a man even though picture (c) is not examined.

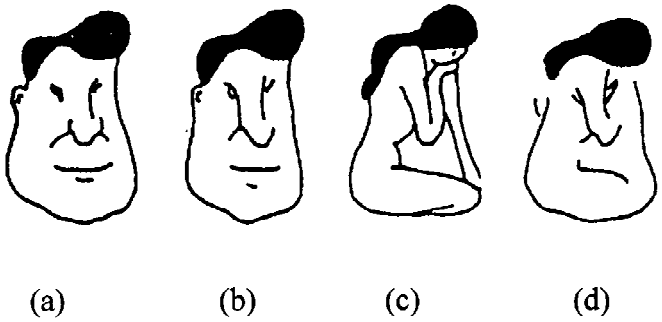

Figure 6. A noisy sequence of pictures in Example 6.

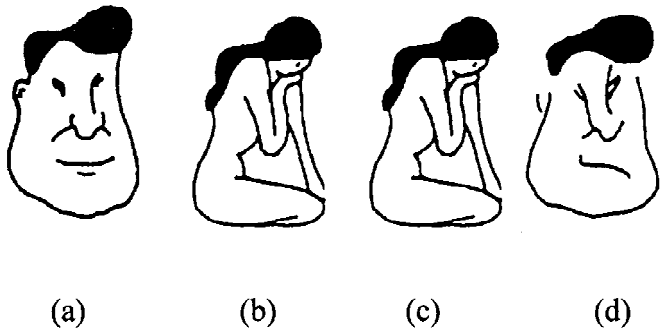

Figure 7. A noisy sequence of pictures in Example 7.

It is easily seen from Figure 3 that picture (c) is more noisy than pictures (a) and (b) in recognizing the face of a man. Since a core concept has been formed from pictures (a) and (b), the incomplete sequence presented by taking away picture (c) is still enough for the contents in (d) to be recognized as the face of a man.

Example 6: Given the four cartoon pictures shown in Figure 6 (a) to (d), the contents in (d) (which is the same as Figure 1) are to be identified. These pictures are the same as those in Figure 2 except that picture (c) is the one in Figure 3 (h), which is quite different from the face of a man and can be thought of as a noisy picture. Figure 6 was shown in the order of (a) to (d) to fifty students for experiments. Among them, 84% thought it was the face of a man.

It is easily seen from Figure 6 that although picture (c) is a noisy picture in recognizing the face of a man, the classification accuracy is still higher than that in Example 1 since a core concept has been partly formed from pictures (a) and (b). The accuracy is however lower than that in Example 2 due to the negative effect of the noise. Noise can indeed reduce the classification accuracy, but this effect can be reduced as long as the core concept is built. This is also very consistent with our intuition that most people are flexible enough to not allow a small amount of noisy data instances to change their classification.

Example 7: Given the four cartoon pictures shown in Figure 7 (a) to (d), the contents in (d) (which is the same as Figure 1) are to be identified. These pictures are the same as those in Figure 6 except that picture (b) is the same as picture (c), which is quite different from the face of a man and can be thought of as a noisy picture. There is thus more noise in Figure 7 than there is in Figure 6 for recognizing the face of a man. Figure 7 was shown in the order of (a) to (d) to fifty students for experiments. Among them, 72% thought it was the face of a man, lower than 80% for Figure 1. Much noise will thus make the recognition accuracy decrease.

The experiences from the above examples will be used below to improve the performance of a machine learning algorithm in processing noisy training instances.

3. Machine Learning in Noisy Environments

Machine learning strategies usually apply some form of generalization to obtain the desired concepts. This is usually accomplished by identifying subsets of training instances that share a common property. Noise in the training instances will tend to confuse any learning mechanism of this type (Quinlan, 1986). Even if the learning mechanism can learn some concepts from the noisy examples, the learned concepts may contain errors and the results given by the incorrect concepts in classifying objects in question might well be incorrect also. Noise is however unavoidable in real application domains. Reducing the effect of noise on the learned concepts is then an important task in machine learning. Two main steps are adopted here to solve this problem:

Step 1. Appropriately arranging the training order of the instances collected;

Step 2. Modifying a conventional noise-free learning algorithm, thus making it suitable for noisy environments.

For Step 1, an order arrangement scheme will be proposed here for evaluating typical and important training instances. These training instances will be processed earlier than non-typical ones to quickly derive the core concepts, thus reducing the effect of noise. For Step 2, the generalized version space learning algorithm (Hong, 1992; Hong and Tseng, 1997) will be adopted and modified to process the noisy examples one after another according to the order of arrangement.

4. The Order Arrangement Scheme

As mentioned in Section 2, the training order is very important to human learning. As long as the core concept is formed at the beginning of learning, the effect of noise can be reduced. This idea is used here for machine learning. An effective order arrangement scheme is proposed for feeding typical training instances into the learning mechanism early. Processing training instances into an appropriate order can not only take less computational time and space, but can also achieve better results. The scheme is outlined as follows.

4.1. THE ORDER ARRANGEMENT SCHEME:

Step 1. Find an appropriate way to decide the principal attribute.
Step 2. Sort the training instances first by their classes and then by the principal attribute.
Step 3. Design an evaluation process to determine confused training instances.
Step 4. Move these confused training instances to the final part of the training sequence.

In the proposed scheme, the classes of training instances are thought of as the most important characteristic in classification. This is apparently true since if the class for a training instance is known, this instance can be recognized correctly. In addition to the class, a principle attribute, which represents another important characteristic in the training set, is chosen to help the sorting. Processing together the training instances with the same classes and close principle attribute values tends to quickly derive the core concepts of this class. This is just like impressing a student's learning by repeatedly teaching him or her the same or similar materials.

The principal attribute corresponds to the most important characteristic of the objects described. Depending on different application domains and concept description languages, there may exist different ways to decide which attribute is the most principal. Several methods have been proposed for feature-based classification problems (Breiman, 1984; Hond and Tseng, 1997; Quinlan, 1983). Among them, the entropy method (Quinlan, 1983) is widely used for determining the distinguishing capability of an attribute. Assume attribute A with m possible values $\{(A_1, A_2, \ldots, A_m\}$ can partition the set of training instances S into $\{S_1, S_2, \ldots, S_m\}$, where S_i contains those instances in S with value A_i. Also assume there are k classes to be distinguished. Let the number of training instances of class O_j in S_i be n_{ij}. Then the entropy for Attribute A is calculated as follows:

$$E(A) = -\sum_{i=1}^{m}\sum_{j=1}^{k}\left[\left(n_{ij}/\sum_{r=1}^{k} n_{ir}\right) * \log\left(n_{ij}/\sum\sum_{r=1}^{k} n_{ir}\right)\right]$$

Among all of the feasible attributes, the one which causes the minimum entropy will be chosen as the principle attribute. Besides the principle attribute, no other attributes are chosen in our scheme since it is enough for the training instances to be roughly ordered. The second or the third attribute, of course, can be used with the principle attribute together to order the training instances accurately. Computational costs however increase.

After Step 2 of the scheme, training instances with the same class and close principle attribute values are grouped into a neighborhood. The evaluation process for determining questionable or confused data is then executed. Since different problem domains and different kinds of training instances may acquire different evaluation processes, the scheme allows users the flexibility to define the evaluation process. Below, a simple heuristic is proposed for evaluating confused data.

After the training instances are sorted by their classes and then by the principle attribute values, the ones with the same principle attribute values, but belonging to different kinds of classes are thought of as confused data. With these particular training instances, it is impossible to distinguish their classes using only their principle attribute values. They are then arranged at the back of the training sequence. Instead of being given up, the confused data will be processed later since they are not erroneous data, but confused data in terms of only the principle attribute. The data are still useful in deriving the final concepts. Processing training instances in this order will make the adopted learning algorithm more efficient than processing the training instances in a random order (later experiments will show this).

5. Adopting a Modified Machine Learning Algorithm

After the training order is rearranged using the above scheme, a conventional noise-free learning algorithm must be modified for application in noisy environments. Among the learning strategies, the version space strategy (Mitchell, 1978, 1982) [22] is one of the most famous. Hong and Tseng generalized the version space learning strategy to make some improvements (Hong, 1992; Hong and Tseng, 1997). The generalized version space learning algorithm can manage uncertain training instances, take into consideration the importance of given training instances, make a trade-off between including positive training instances and excluding negative ones, and find a maximally consistent version space in a noisy environment. It is then modified here to process noisy training instances.

The original version space learning strategy attempts to induce concepts which include all of the positive training instances and exclude all of the negative training instances. It uses two boundary sets S and G to effectively represent all of the possible candidates:

$S = \{s|$ *s is a hypothesis consistent with observed instances. No other hypothesis exists which is 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 set of all consistent concept definitions, each of which 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.

The version space learning strategy is however very sensitive to noise. If there exists any inconsistency in the training set, this method usually will not work. The sets S and G in the generalized version space learning strategy were modified to create the ability to manage noise. The hypotheses in S and G no longer necessarily include and exclude all of the positive and negative training instances presented so far, since noisy and uncertain data may be present. 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 users). A numerical measure referred to as the *count* is attached to each hypothesis in S and G to summarize all positive and negative information implicit in the training instances presented so far. A hypothesis with a higher count in S includes more positive training instances. A hypothesis with a higher count in G excludes more negative training instances. Sets S and G are then modified as follows:

$S = \{s | s$ *s is a hypothesis among the first i maximally consistent hypotheses. No other hypothesis in S exists which is both more specific than s and has an equal or larger count}.*

$G = \{g | g$ *is a hypothesis among the first j maximally consistent hypotheses. No other hypothesis in G exists which is both more general than G and has an equal or larger count}.*

Note that the most consistent i and j hypotheses in S and G are not necessarily the ones with the largest i and j counts. A hypothesis in S that includes many positive training instances may possibly also include many negative training instances. Which hypotheses in S and G are the first i and j maximally consistent then depends on both sets S and G (through the confidence values shown later), and not only on S itself or G itself.

The parameters i and j originally represent the maximum numbers of the hypotheses maintained in the sets S and G respectively. They were modified in this research as follows:

i: is the first i different confidence values for hypotheses maintained in S, and

j: is the first j different confidence values for hypotheses maintained in G.

For example, if $j = 2$, then all the hypotheses in G with the highest or the second highest confidence values are kept. The advantage of this modification is that the sizes of the hypotheses maintained in S and G can be altered dynamically in the learning process.

For the proposed learning algorithm to make a trade-off between including positive training instances and excluding negative training instances, a parameter called *the factor of including positive instances (FIPI)* is incorporated into the algorithm. The value of *FIPI* is 1 if the aim of the learning problem is only to include positive training instances and 0 if the problem aims only to exclude negative ones. *FIPI* is 0.5 if including positive training instances and excluding negative ones are of the same importance.

The generalized version space learning algorithm is simplified here to manage only noisy training instances, and is outlined as follows.

The simplified generalized version space learning algorithm:

INPUT: A set of n training instances, the parameter *FIPI*, and the maximum levels i, j of the hypotheses maintained in S and G.

OUTPUT: The hypotheses in sets S and G that are maximally consistent with the given training instances.

STEP 1: Initialize S to contain only the most specific hypothesis $\emptyset$ with *count* = 0 and initialize G to contain only the most general hypothesis with *count* = 0 in the whole hypothesis space.

STEP 2: For each newly presented training instance, do the following steps.

STEP 3: If the training instance belongs to the positive class, do STEP 4 to STEP 8; otherwise, Go to Step 9.

STEP 4: Generalize each hypothesis in S to include the new training instance, and set its new *count* = old *count* + 1. Call the newly formed set as S' for convenience.

STEP 5: Find the set S'' including only the new training instance itself, and set the *count* of the hypothesis in S'' to be 1.

STEP 6: Combine the original S, S' and S'' together to form a new S. If identical hypotheses with different counts are present in the combined set, only the hypothesis with the maximum count is retained. If a particular hypothesis is both more general than another and has an equal or smaller count, discard that hypothesis.

STEP 7: For each hypothesis s with count c_s in the new S, find the hypothesis g in G that is more general than s and has the maximum value of count c_g. Calculate the confidence value as $FIPI \times c_s + (1\text{-}FIPI) \times c_g$.

STEP 8: Retain the hypotheses with the first i levels of confidence values in the new S and discard the others; Go to STEP 14.

STEP 9: (The training instance belongs to the negative class) specialize each hypothesis in G to exclude the new training instance, and set its new *count* = old *count* + 1. Call the newly formed set as G' for convenience.

STEP 10: Find the set G'' excluding only the new training instance itself, and set the *count* of each hypothesis in G'' to be 1.

STEP 11: Combine the original G, G' and G'' together to form a new G. If identical hypotheses with different counts are present in the combined set, only the hypothesis with the maximum count is retained. If a particular hypothesis is both more specific than another and has an equal or smaller count, discard that hypothesis.

STEP 12: For each hypothesis g with count c_g in the new G find the hypothesis s in S that is more specific than g and has the maximum value of count c_s. Calculate the confidence as $FIPI \times \mathrm{c}_s + (1\text{-}FIPI) \times c_g$.

STEP 13: Retain the hypotheses with the first j levels of confidence values in the new G and discard the others.

STEP 14: When there are still new training instances to be processed, go to STEP 2; otherwise, stop the learning process.

The simplified generalized version space learning algorithm contains user- specified parameters (i, j and *FIPI*) to increase the flexibility of learning.

6. The Experiments

Fisher's Iris Flower data (Fisher, 1936) containing 150 training instances are used here for demonstrating the effectiveness of the proposed leaming mechanism. Since the set of data are inconsistent, the original version space learning algorithm can't be successfully applied. The problem is described below.

6.1. THE PROBLEM

Three species of Iris Flowers exist: setosa, viginica, and versicolor to be distinguished. There are 150 instances in total, with 50 for each class. Each instance is described by four features: sepal width, sepal length, petal width, and petal length. The units for all f the four attributes are centimeters, measured to the nearest millimeter. For example, one flower of versicolor is described by a sepal length of 6.3 cm, a sepal width of 3.3 cm, a petal length of 4.7 cm, and a petal width of 1.6 cm.

Each legal hypothesis is restricted to conjunctions of the form $a \leq x \leq b$ for each attribute, where a and b are limited to multiples of 8 millimeters. For example, a legal concept description can be "sepal length is greater than 0.8cm and less than 1.6 cm".

6.2. THE EXPERIMENTAL METHOD

A method called *N-fold cross-validation* is adopted here (Breiman et al., 1984) for the small set of training instances. This method randomly partitions all training instances into N subsets of nearly equal size. For each $n (n = 1, \ldots, N)$, the n-th subset is put aside and the learning algorithm is trained by the other $N - 1$ subsets. The n-th subset is then used as testing data. For the Iris Flower classification problem, the instances are partitioned into ten subsets, each with fifteen instances including five for each class. Nine of the ten data subsets are then used as the training data and the remaining subset is used as testing data. When the concept of setosa is learned, training instances belonging to setosa are considered to be positive. Training instances belonging to the other two classes are considered to be negative. The resulting classification rates are averaged across all the ten runs.

6.3. RESULTS

Experiments were implemented using C language at a SUN 3/260 workstation. The parameters i, j of the generalized version space algorithm were assigned to 1 and the parameter *FIPI* was assigned to 0.5. The accuracy and time using our order arrangement scheme and those using a random order are shown in Tables 1 to 3, respectively for a class of iris flowers.

Table I. Experimental results for setosa

Set	Order by our scheme		Random order	
	accuracy	time (sec)	accuracy	time (sec)
1	100	226	100	2165
2	100	249	100	1307
3	100	234	80	38252
4	100	223	100	2912
5	100	231	100	34550
6	100	219	100	63459
7	100	219	100	72333
8	100	241	73.7	8472
9	93.33	207	93.33	124664
10	100	215	100	44839
Average:	99.3	226.4	94.7	39295.3

Table II. Experimental results for vignica

Set	Order by our scheme		Random order	
	accuracy	time (sec)	accuracy	time (sec)
1	100	102	93.33	5511
2	93.33	104	93.33	10171
3	100	104	100	11571
4	93.33	101	80	2566
5	93.33	186	93.33	60284
6	93.33	63	86.7	1168
7	80	98	80	2297
8	93.33	543	100	11506
9	100	107	73.3	1115
10	100	46	66.7	3093
Average:	94.65	145.4	86.66	10928

Results show that the learning mechanism using a random order spent much more time and got a lower accuracy rate than that using the arranged order. The rationale of the experimental results is discussed as follows.

As described in Section 5, the simplified generalized version space learning algorithm consists of two main phases: generating and pruning, as shown in Figure 8.

The generating phase collects possible candidates (Steps 4 to 6 and Steps 9 to 11) into a large set; the pruning phase then prunes this set according to the

Table III. Experimental results for versicolor

Set	Order by our scheme		Random order	
	accuracy	time (sec)	accuracy	time (sec)
1	93.33	54	86.7	25283
2	93.33	72	93.33	9385
3	100	31	100	61902
4	93.33	32	86.7	22465
5	93.33	68	93.33	26659
6	93.33	62	86.7	6260
7	86.7	65	86.7	25160
8	93.33	67	80	117163
9	100	32	73.3	2297
10	93.98	31	60	344
Average:	93.98	51.4	84.6	9691.8

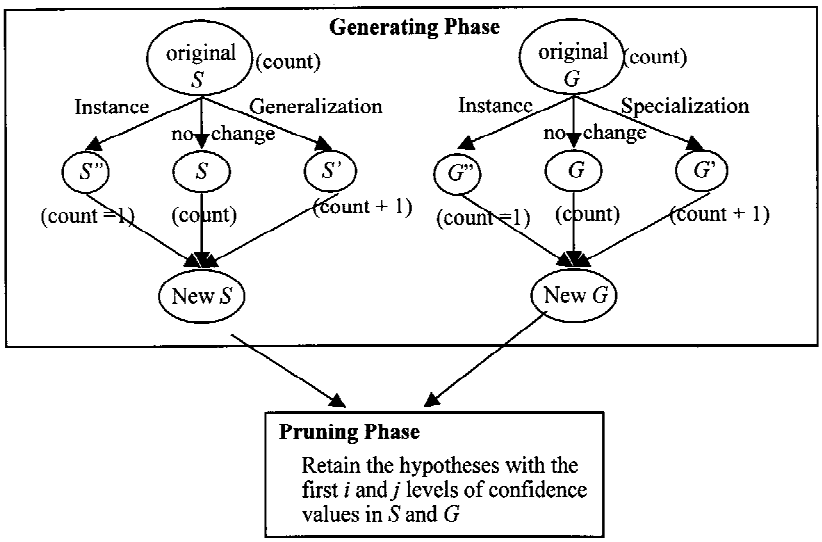

Figure 8. The generating and pruning phases in the learning algorithm.

degree of consistency of the hypotheses in the boundary sets (Steps 8 and 13). Candidate hypotheses with high confidence values are kept in the boundary sets with a high priority. If the core training instances can be found and fed into the leaming mechanism at the beginning, then the desired hypotheses formed from these instances have larger counts than other hypotheses have due to the generalization or the specialization process (in Figure 8). Note that a hypothesis is both more general and specific than itself. Thus each core training instance adds one to the

Table IV. Predictive accuracy of five learning strategies

class algorithm	Setosa	Viginica	Versicolor	Average
Our scheme	99.3	94.65	93.98	95.98
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

count of a desired hypothesis. The desired hypotheses also have larger confidence values since their corresponding hypotheses in the other boundary set (G or S) are consistent with them. These hypotheses will then be kept after the pruning phase and other undesired hypotheses will be pruned. Our proposed algorithm can thus effectively learn desired concepts as long as the concepts can be formed in the boundary sets at the beginning.

The values of i and j also influence the pruning effects in our algorithm. If the i, j values are small, then in a random order of training instances, the desired hypotheses may be discarded or even not formed for the first several coming training instances. Later, whenever the desired hypotheses are generated in the generating phase, they are immediately pruned due to their initial low confidence values. The incorrect hypotheses formed from non-core training instances can not thus be replaced since the i, j values are not large enough to keep the desired hypotheses in the boundary sets. Therefore, whether the final output hypotheses are maximally consistent with the training instances highly depends on the choice of i, j values and the order scheme. The larger the i, j values are, the more consistent the final output hypotheses are. For small i,j values as in the experiments, the accuracy by our scheme is thus better than that by a random order since the core training instances can be found and fed into the learning mechanism at the beginning. Also, if the core training instances are not given at the beginning of the learning process, the generating and pruning phases will process much more candidate hypotheses than actually needed, causing that the time complexity increases a lot. The proposed order arrangement scheme is then useful in improving the learning performance and reducing the effects of noise.

The accuracy of some other learning algorithms on the Iris Flower Classification Problem was examined by Hirsh (1989). The methods studied were Hirsh's Incremental Version-Space Merging (Hirsh, 1989, 1994), Kibler's noise-tolerant NTgrowth (Kibler and Langley, 1988), Dasarathy's pattern-recognition approach (Dasarathy, 1980), and Quinlan's C4 (Quinlan, 1987, 1992). Table 4 compares the accuracy of our learning scheme with that of the others. It can easily be seen that our method is as accurate as Hirsh's IVSM, even though our method does not

assume that knowledge of the bounded inconsistency is available. The learning mechanism proposed here is thus satisfactory.

7. Conclusion

In this paper, we have provided a two-step solution to the problem of learning in noisy environments. Based on the effect of training order on human learning, we have proposed an order arrangement scheme to improve the performance of learning and reduce the interference of noise. We have also simplified the generalized version space strategy to learn concepts from noisy training examples. Finally, experimental results on Iris data have shown our learning scheme can get a satisfactory result, quite consistent with our idea.

Order-dependent problems often occur in many learning systems which are incremental and memory-limited. Our order arrangement scheme can serve as a preprocessing step for incremental learning. In the future, we will continuously attempt to apply (or modify) this scheme to different domains and to other learning systems.

Acknowledgements

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

References

Atkinson, R.L., Atkinson, R.C., Smith, E.E. and Bem, D.J. (1990), *Introduction to Psychology*, Tenth Edition, Harcount Brace Jovanovich, Inc.

Baddeley, A. (1990), *Human Memory Theory and Practice*. MA: Allyn and Bacon.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984), *Classification and Regression Trees*. CA: Wadsworth.

Chang, F. (1992), 'From artificial intelligence to cognitive science,' *Science Monthly* 23(2), pp. 108–113.

Chang, K.C., Hong, T.P. and Tseng, S.S. (1996), 'Machine Learning by Imitating Human Learning,' *Minds and Machines* 6(2), pp. 203–228.

Clark, P. and Niblett, T. (1989), 'The CN2 induction algorithm,' *Machine Learning* 3, pp. 261–283.

Dasarathy, B.V. (1980), 'Noise around the neighborhood: a new system structure and classification rule for recognition in partially exposed environments,' *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2(1), pp. 67–71.

Feigenbaum, E.A. (1977), 'The art of artificial intelligence: themes and case studies of knowledge engineering,' *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, pp. 1014–1029.

Fisher, G.H. (1967), 'Preparation of ambiguous stimulus materials,' *Perception and Psychophysics* 2, pp. 421–422.

Fisher, R.A. (1936), 'The use of multiple measurements in taxonomic problems,' *Annual Eugenics* 7, pp. 179–188.

Hall, J.F. (1989), *Learning and Memory*. MA: Allyn and Bacon.

Hirsh, H. (1989), *Incremental Version-Space Merging: A general Framework for Concept Learning*. Ph.D. Thesis, Stanford University.

Hirsh, H. (1994), 'Generalizing version space,' *Machine Learning* 17, pp. 5–46.

Hong, T.P. (1992), *A Study of Parallel Processing and Noise Management on Machine Learning*. Ph.D. Thesis, National Chiao Tung University, Taiwan, R.O.C.

Hong, T.P. and Tseng, S.S. (1994) 'Learning concepts in parallel based upon the strategy of version space,' *IEEE Transactions on Knowledge and Data Engineering* 6(6), pp. 857–867.

Hong, T.P. and Tseng, S.S. (1997), 'A generalized version space learning algorithm for noisy and uncertain data,' *IEEE Transactions on Knowledge and Data Engineering* 9(2), pp. 336–340.

Hong, T.P. and Chen, J.B. 'Finding relevant attributes and membership functions,' accepted and to appear in *Fuzzy Set and Systems*.

Kibler, D. and Langley, P. (1988), 'Machine learning as an experimental science,' *Proceedings of the European Working Session on Learning*, pp. 87–92.

Kodratoff, Y., Manago, M.V. and Blythe, J. (1987), 'Generalization and noise,' *International Journal of Man-Machine Studies* 27, pp. 181–204.

Mingers, J. (1989), 'An empirical comparison of pruning methods for decision tree,' *Machine Learning* 4, pp. 319–342.

Mitchell, T.M. (1978), *Version Space: an Approach to Concept Learning*. Ph.D. Thesis, Stanford University.

Mitchell, T.M. (1982), 'Generalization as search,' *Artificial Intelligence* 18, pp. 203–226.

Quinlan, J.R. (1983), 'Learning efficient classification procedures and their application to chess end games,' in R.S. Michalski, J.G. Carbonell and T.M. Mitchell, eds., *Machine Learning: An Artjficial Intelligence Approach*, Vol. 1, Morgan Kaufmann, pp. 463–482.

Quinlan, J.R. (1986), 'The effect of noise on concept learning,' in R.S. Michalski, J.G. Carbonell and T.M. Mitchell, eds., *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, Morgan Kaufmann, pp. 463–482.

Quinlan, J.R. (1986), 'Induction of decision trees,' *Machine Learning* 1(1), pp. 81–106.

Quinlan, J.R. (1987), 'Simplifying decision trees,' *International Journal of Man- Machine Studies* 27(4), pp. 221–234.

Quinlan, J.R. (1992), *C4.5 Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Solso, R.L. (1988), *Cognitive Psychology*. MA: Allyn and Bacon.