귀납법칙 학습과 개체위주 학습의 결합방법

이 창 환†

요 약

대부분의 기계학습 방법들은 특정한 방법을 중심으로 연구되어 왔다. 하지만 두 가지 이상의 기계학습방법을 효과적으로 통합할 수 있는 방법에 대한 요구가 증가하며, 이 에따라 본 논문은 귀납법칙(rule induction) 방법과 개체위주 학습방법(instance-based learning)을 통합하는 시스템의 개발을 제시한다. 귀납법칙 단계에서는 엔트로피 함수의 일종인 Hellinger 변량을 사용하여 귀납법칙을 자동 생성하는 방법을 보이고, 개체위주 학습방법에서는 기존의 알고리즘의 단점을 보완한 새로운 개체위주 학습방법을 제시한다. 개발된 시스템은 여러 종류의 데이터에 의해 실험되었으며 다른 기계학습 방법과 비교되었다.

A Combined Method of Rule Induction Learning and Instance-Based Learning

Chang-Hwan Lee†

ABSTRACT

While most machine learning research has been primarily concerned with the development of systems that implement one type of learning strategy, we use a multistrategy approach which integrates rule induction learning and instance-based learning, and show how this marriage allows for overall better performance.

In the rule induction learning phase, we derive an entropy function, based on Hellinger divergence, which can measure the amount of information each inductive rule contains, and show how well the Hellinger divergence measures the importance of each rule. We also propose some heuristics to reduce the computational complexity by analyzing the characteristics of the Hellinger measure. In the instance-based learning phase, we improve the current instance-based learning method in a number of ways. The system has been implemented and tested on a number of well-known machine learning data sets. The performance of the system has been compared with that of other classification learning techniques.

1. Introduction

It is well known that acquiring expertise from experts causes serious problems. It causes serious bottlenecks due to the communications problem between the different backgrounds among people. Also, the knowledge obtained from humans is generally uncertain, inconsistent and sometimes even contradictory. Therefore, automatically generating knowledge provides significant potential as the number of available databases grow exponentially. For this purpose, most machine learning research has been primarily concerned with the development of systems

† 정 회 원:동국대학교 전산통계학과
논문접수:1997년 4월 3일, 심사완료:1997년 8월 8일

that implement one type of inference within a single computational paradigm. Such systems include those for empirical induction of decision trees [15], rule induction [9], instance-based learning [1], explanation-based learning [6], neural-net learning [16], and genetic algorithms [7]. These learning systems can be very effective and useful, if the learning problems they are applied to are narrowly defined. However, empirical comparison of these different approaches in a variety of application domains has shown that each performs best in some, but not in others. Even though many different approaches to inductive learning has been used in the machine learning literature, each has specific limitations that are hard to overcome. Many real-world applications pose learning problem beyond the capability of monostrategy learning methods. This raises a crucial question to knowledge system developed: which method is the most suitable for a certain domain?

Multistrategy learning is an attempt to tackle this problem by integrating multiple methods in one algorithm. Recent years have witnessed a growing interest in developing multistrategy systems that integrate two or more inference types and/or computational paradigms in one learning system. Among early well-known multistrategy systems are PRODIGY [10], OCCAM [12], and KBL [19]. Such multistrategy systems take advantage of the complementary, inferential representational mechanisms. Therefore, multistrategy systems have a potential to be more versatile and more powerful than monostrategy systems.

While most of these systems are concerned with integrating symbolic empirical induction with explanation-based learning in order to utilize the domain knowledge of explanation-based learning for empirical induction [11] [13], in this paper, we investigate the integration of rule induction system with instance-based learning, and show how this marriage shows overall better performance. The central idea of integrating rule induction with instance-based learning is the following. The proposed system is composed of two phases: rule induction learning phase and instance-based learning phase. In the rule induction learning phase, the proposed system generates a set of inductive rules which are sorted based on their rule strength. Apparently, all the rules generated from the system cannot be a member of the final rule set because most of the rules are very poor in their quality. Therefore, among the candidate rules generated from the rule induction learning phase, the system only selects rules whose strength quality is higher than a certain threshold value, and then the instance-based learning phase takes over the control and classifies the remaining instances which could not be processed by the rule induction learning phase.

2 Rule Induction Learning Phase

In recent years many systems for generating inductive rules from examples have been developed. The current rule induction system includes PRISM [3], CN2 [4], and AQ family of rule induction systems [9]. Although these rule induction methods differ with each other, the basic idea is to use the single-best-rule method which expands only one rule at a time and add one condition after another to the antecedent until the rule is consistent with the negative data.

The rule induction phase of this paper is primarily based on Smyth and Goodman's ITRULE rule induction algorithm [18]. The ITRULE algorithm and the rule induction method proposed in this paper have some similarities in the sense that both adopt an entropy function as the measure of the strongness of rules. However, as discussed in the following section, we point out some deficiencies of Kullback entropy function used in ITRULE algorithm, and propose a new improved entropy function, called Hellinger measure. The method is probabilistic because the significance of each rule is measured by the Hellinger measure.

2.1 Probabilistic Rule Induction

The format of the rules the system will generate is the following:

$A=a,\ B=b,\cdots\rightarrow T=t$ with s

where A, B, and T are attributes with a, b, and t being values in their respective discrete values. Each rule is associated with a value(s) which represents the significance of the rule. The right-hand expression is restricted to be a single value assignment while the left-hand side may be a conjunction of such expressions. The attribute appearing in the right-hand side is usually called 'target attribute.'

The basic idea of rule induction in the proposed system starts with the fact that the value assignments in the left hand side of each rule effects the probability distribution of the attribute of the right-hand side. The target attribute forms its a priori probabilities without presence of any conditions. It normally represents the class frequencies of the target attribute. However, its probability distribution changes when it is measured under certain conditions given as value assignments of other attributes. Intuitively speaking, if a certain value assignment has significantly changed the probability distribution of the target, it is clear that the given value assignment plays an important role determining the class values of the target attribute. On the other hand, if the probability distribution of the target attribute remains the same regardless of a value assignment of other attribute, those two attributes are independent with each other. Therefore, it is a natural definition, in this paper, that the significance of a rule is interpreted as the degree of dissimilarity between a priori probability distribution and a posteriori probability distribution of the target attribute.

Our next step is to define or select a proper measure which can correctly measure the dissimilarity (divergence) of these two probability distributions. Many studies regarding dissimilarity measure were conducted in the field of information theory. Among them, Kullback measure is one of the most widely used measure. For example, Smyth and Goodman [18] have developed the ITRULE algorithm using modified form of Kullback entropy function as the certainty factor of inductive rules. However, one of the problems that the Kullback entropy function used in the ITRULE algorithm is that as the function is originally defined for continuous variables, it is not applicable to all the cases of discrete probability distributions. For instance, if one value of an attribute takes the probability of unity, Kullback measure is not able to be defined in this case unless the original values are approximated. For more information about other measures, readers are referred to the work of Smyth and Goodman [18].

Throughout the classification learning system described in this paper, we employ a new measure, called Hellinger divergence, which was originally introduced by Beran [2]. Let t_i denote the value of attribute T. Also suppose $P(t_i)$ denotes a priori probabilities of attribute T and $P(t_i|a)$ means a posteriori probabilities under the condition $A=a$. The corresponding Hellinger divergence, is defined as

$$[\sum_i (\sqrt{P(t_i)}-\sqrt{P(t_i|a)})^2]^{1/2}$$

It can be interpreted as a distance measure where distance corresponds to the amount of divergence between a priori and a posteriori distribution. In words, it is a measure of how dissimilar a priori and a posteriori beliefs are about T-useful rules imply a high degree of dissimilarity. It is definable in every combination of a priori probabilities and a posteriori probabilities.

Suppose there is a rule with the event $A=a$ as the right-hand side and $B=b$ as the left-hand side, as shown in the following:

$$B=b\rightarrow A=a \tag{1}$$

According to the definition of Hellinger measure, the

significance is defined as follows.

$$[\sum_{a \in A} (\sqrt{P(a)} - \sqrt{P(a|b)})^2]^{1/2} \quad (2)$$

Equation (2) adds up, for all value of a in attribute A, the difference of probability $P(a)$ and $P(a|b)$. However, in rule generation environment, only one particular value of the class attribute appears in the right hand side of the rule, and thus the probabilities for the other values are accumulatively included in $1-P(a)$. In words, what is important in measuring the accuracy of the rule is to check whether or not an instance belongs to the class value of the right hand side of the rule. Suppose the target attribute(say A) has k values($a_1, a_2, \cdots, a_k$) and the rule has $A=a_1$ as the right hand side. The probability distribution of attribute A is transformed into a binary probability distribution form such as $(P(A=a_1), P(A \neq a_1))$. Therefore, the significance of the rule in formula (1) is transformed as follows.

$$(\sqrt{P(a|b)} - \sqrt{P(a)})^2 + (\sqrt{1-P(a|b)} - \sqrt{1-P(a)})^2$$

where $P(a|b)$ means the conditional probability of $A=a$ under the condition $B=b$.

Another problem we have to consider in rule induction is to decide how general a rule is. The basic idea behind the generality is that the more often the left-hand side occurs for a rule, the more useful the rule becomes. For example, suppose there are two cases of HIV positive patients in an imaginary hospital database, and the blood type of both patients. turned out to be 'A'. It is not a good idea for a rule induction system to generate a rule such as

If patient = HIV-positive, then Blood-type = D

This type of rule overfits a specific condition and thus lacks its generality. Therefore, in this paper, we use

$$\sqrt{P(b)}$$

as a measure of generality of hypothesis. Consequently, the H measure represents the significance of rule as a multiplicative measure of the generality and accuracy of a given rule. For a given rule in Equation (1), the final form of its significance(H measure) is defined in the following as the product term between the generality and accuracy of the rule.

$$\sqrt{P(b)}\,[\sqrt{P(a|b)} - \sqrt{P(a)})^2 + (\sqrt{1-P(a|b)} - (\sqrt{1-P(a)})^2]$$

3. Combining Instance-based Learning with Rule Induction

We have described how the system generates an initially large candidate set of rules that models the data. Our next objective is to find a rule set which can provide best classification results as it could. In the proposed system, we will consider a greedy search procedure for finding the best rule set. Unlike other rule induction systems, the system keeps a table of the most promising rules, which are sorted based on their significance. At each stage, the best rule of the list is added to the current rule set. If we simply add the best rule from the list until all the instances are covered by the rule set, which is similar to the method used in the AQ style rule induction systems, we can maximize only the number of covered instances and it is unlikely that the final rule set produces good classification accuracy. Besides, the final covering rule set often overfits the training data, performing poorly on new cases. Therefore, a second refinement step is needed to adjust the rule set to the right complexity fit.

The basic idea of combining these two methods is the following. Among the rules generated from the rule induction learning phase, a certain number of rules are selected for the final rule set, and the instances that could not be classified based on this rule set will be classified by the instance-based learn-

ing phase. Initially, the system begins with an empty rule set. In this case, the entire instances will be classified only by the instance-based learning phase. After that, the system adds the best element of the rule list to the current rule set, and the test instances will be classified by the rule induction learning phase first. And then, the remaining instances, those could not be classified by the rule induction learning phase, will be classified by the instance-based learning phase. As more rules are added, the performance of the rule set increases since all the rules added at early stages are to have high quality. However, after the size of the rule set reaches a certain point, the overall perfo-rmance decreases as the poor rules begin to be added. If the classification accuracy of the new rule set is worse than that of the previous rule set, the system stops and the previous rule set becomes the final rule set.

4 Instance-Based Learning Phase

After the rule induction learning phase produces inductive rules and thus classifies the test data set, it is the responsibility of instance-based learning to process the test data which could not be classified by the inductive rules. This section describes an information theoretic instance-based learning technique. Instance-based learning method is primarily based on the traditional k nearest neighbor(k-NN) algorithm. We have improved the k-NN algorithm in some ways. In the following section, we point out some problems that current similarity measures contain and propose a new improved method of measuring similarity between instances in databases. We define similarity measures for every possible attribute type in relational database, and provide methods to calculate the weights for both attributes and selected instances. The similarity measure used in the proposed method is designed to take into account all the knowledge expressed in relational databases.

4.1 New Similarity Measure

Now we will elucidate the detailed method of calculating similarity metrics. Suppose we compare the similarity between two instance X and Y. Let x_i and y_i, for $i=1,\cdots,k$, be the values of the i-th attribute for X and Y, respectively. Let a_i be a value of an attribute A and T its target attribute. As we mentioned earlier, the importance of each attribute depends on the target attribute. Let $D_T(X, Y)$ denote the similarity function between two instances X and Y with respect to attribute T. $D_T(X, Y)$ will be defined in the following manner.

$$D_T(X, Y)=\sum_{i=1}^{k} \omega_T(i) \cdot d_T(x_i, y_i)$$

where $\omega_T(i)$ is the weight of attribute A_i with respect to T, and $d_T(x_i, y_i)$ denotes the similarity between values x_i and y_i.

The computation of similarity between two instances consists of two steps: calculating the weights of attributes and calculating the value similarity between two attribute values. Each step will be explained in detail in the following sections.

4.2 Assigning Weights to Attributes

Information theory serves as the theoretic background in calculating both the weights of attributes and the value similarities. The basic idea for calculating the weight of each attribute is that the more information an attribute gives to the target attribute, the more weight the attribute is to have. Information theory provides us with variety of tools which can measure the amount of information each value assignment gives to other attributes as a form of an entropy function.

For one value assignment of the attribute, its corresponding information content will be calculated using entropy function. Therefore, for discrete attribute types such as binary, categorical or pointer, we calculate the information content for each separate discrete value and their average value becomes the weight of

the attribute.

One problem of this approach is that we can not apply this method directly to numeric attributes because the fundamental theory is based on discrete entropy function. For numeric attributes, we discretize the numeric values first, and then follow the same procedure used for categorical attributes. We use the context-sensitive discretization which is described in [8]. This method is applied, as part of the preprocessing procedure, to each of the numeric attributes.

Now the critical part is how to measure the amount of information a value assignment gives to the target attribute. We use the Hellinger divergence again as the information measure. Suppose we are to estimate the amount of information that a value assignment of an attribute A gives to the target attribute T, and let t_i, $i=1,\cdots,k$, and a_j, $j=1,\cdots,l$, denote the values of T and A, respectively. Hellinger measure of the amount of information $A=a_j$ gives to attribute T, denoted as $Ent(T|A=a)$, is defined as

$$Ent(T|A=a_j)=[\sum_i(\sqrt{P(t_i)}-\sqrt{P(t_i|a_j)})^2]^{1/2} \quad (3)$$

Summation of formula (3) with respect to attribute A may serve as the weight of attribute A. However, in that case, the weight increases monotonically as the number of category increases. Therefore, attributes with a large number of category are to have larger weights. As a way to normalize this value, formula (3) is multiplied by the probability that each category can happen, $P(a_j)$.

$$Ent(T|A)=\sum_j P(a_j)\,Ent(T|A=a_j)$$

By doing so, $Ent(T|A)$ becomes independent of the number of category. However, since this measure can grow indefinitely, using $Ent(T|A)$ as the weight of an attribute may provide anomalous similarity values. By dividing the current $Ent(T|A)$ value by the total summation of $Ent(T|A)$, we have the following formula which ranges from zero to one.

$$\omega_T(A)=\frac{Ent(T|A)}{\sum_{A\neq T}Ent(T|A)}=\frac{\sum_j P(a_j)Ent(T|A=a_j)}{\sum_{A\neq T}Ent(T|A)} \quad (4)$$

4.3 Computing Value Similarity

The second step for calculating instance similarity is to compute the similarity between two attribute values. As we mentioned earlier, the proposed similarity measure is defined on every possible attribute type in a database. In the following, a new method for calculating similarity between two values is defined for each data type: binary, categorical, numeric, and pointer.

For binary values, it is straightforward to decide how similar two binary values are. We just check whether they match with each other.

If $x_i=y_i$,

$$d_T(x_i,\ y_i)=1$$

otherwise

$$d_T(x_i,\ y_i)=0$$

Similarity measures for numeric values have been de fined in a number of places before. As we mentioned earlier, Euclidean or absolute methods have been widely used. In the proposed method, the distance between values x_i and y_i is defined as the ratio of absolute value of $|x_i-y_i|$ to the total range of A. Therefore, the similarity between these two values is defined as

$$d_T(x_i,\ y_i)=1-\frac{|x_i,\ y_i|}{R}$$

where R denotes the range of attribute A.

For categorical(nominal) values, the traditional similarity measuring technique is to check whether two values match or not. This measure produces poor results. Suppose x_i and y_i are two values of an attribute A, and T denotes the target attribute. We first

measure the amount of information that x_i or y_i gives about the target attribute. Their relative difference is divided by the total amount of information that A gives, and the result is subtracted from 1. Formally, the similarity is given as

$$d_T(x_i, y_i) = 1 - \frac{|Ent(T|A=x_i) - Ent(T|A=y_i)|}{Ent(T|A)}$$

There has been no attempt to define the similarity measure for pointer attribute type. A simple way to measure the similarity for this case is to check whether two values point to the same instance. As we discussed earlier, in this case, we are to lose the information embedded within the pointer values, which results in poor similarity measure. A second possible way is to regard pointer attribute as categorical attribute. This causes the overfitting problem because, unlike categorical values, pointer values have very few corresponding instances for each of its value assignment.

We adopt a more sophisticated way for the similarity measure of pointer type. Because a value of pointer type refers to another instance, the difference between two pointer values can be regarded as the difference between those two which are being referred to. Thus we can decide their similarity by recursively applying the similarity function to the instances being pointed to. The formal definition of similarity for pointer value is defined in a recursive form as follows. Suppose $\vec{x_i}$ and $\vec{y_i}$ represent the instances referenced by x_i and y_i respectively.

If $\vec{x_i}$ or $\vec{y_i}$ does not exist in the database,

$$d_T(x_i, y_i) = 0$$

otherwise

$$d_T(x_i, y_i) = \begin{cases} D_T(\vec{x_i}, \vec{y_i}), & x_i \neq y_i \\ 1, & x_i = y_i \end{cases}$$

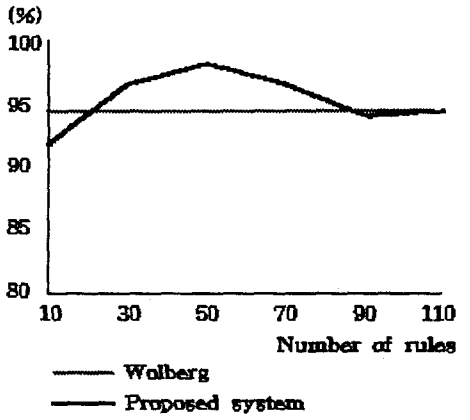
We have defined the similarity for each data type: binary, numeric, categorical, and pointer. Similarity value for each attribute is multiplied by its corresponding weight and the results are added attribute by attribute.

〈Table 1〉 Characteristices of data sets

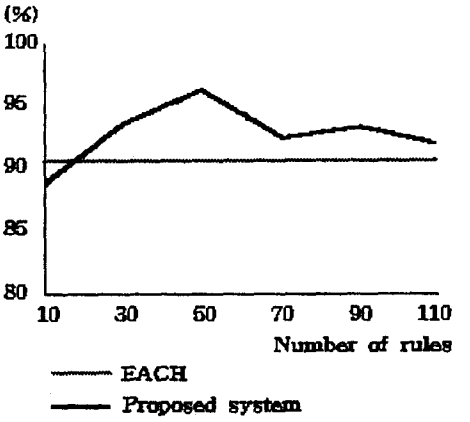
Data set	No. of class	No. of attribute	No. of instance
Breast cancer	2	10	699
Echocardiogram	2	8	132
Thyroid	3	5	215

〈Table 2〉 Rules from breast cancer data

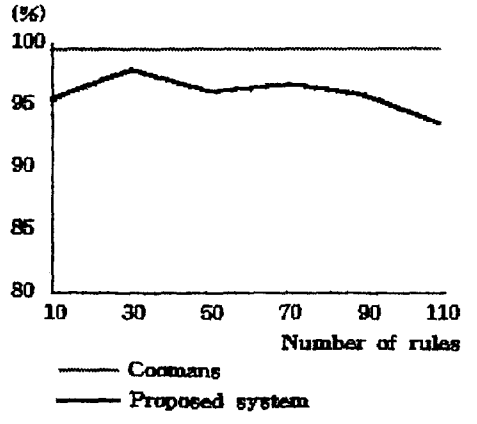
Rule	H
4.5 < SHAPE < 10, 55 < BN < 10 → Class = Malignant	0.2749
1.0 < SECS < 2.5, 1.0 < BN < 1.5, 1.0 < MT < 1.5 → Class = Benign	0.2741
1.0 < SECS < 2.5, 1.0 < BN < 1.5 → Class = Benign	0.2741
1.0 < SIZE < 2.5, 1.0 < BN < 1.5, 1.0 < NN < 2.5 → Class = Benign	0.2737
5.5 < BN < 10 → Class = Malignant	0.2737
6.5 < CT < 10, 4.5 < SHAPE < 10, 5.5 < BN < 10 → Class = Malignant	0.2687
6.5 < CT < 10, 4.5 < SHAPE < 10 → Class = Malignant	0.2685
1.0 < MA < 1.5, 1.0 < SECS < 2.5, 1.0 < NN < 2.5, 1.0 < M < 1.5 → Class = Benign	0.2640
1.0 < MA < 1.5, 1.0 < BN < 1.5, 1.0 < N < 2.5 → Class = Benign	0.2584
4.5 < SHAPE < 10 → Class = Malignant	0.2583



(Fig. 1) Success rates for breast cancer data



(Fig. 2) Success rates for echocardiogram data



(Fig. 3) Success rates for thyroid data

The system assigns different weights, the similarity value defined as above, to each instance selected from the system. It is obvious that the more similar an instance is, the larger the weight is. For each instance selected by instance-based learning, their values are accumulated based on the class categories, and the class category with the largest collected weight value will be selected as the final class value.

5. Evaluation

The classification results are analyzed and compared with other methods. The experiments consist of databases obtained from the University of California Irvine machine learning database repository. To evaluate the performance of the system, three data sets were considered: breast cancer, echocardiogram, and thyroid database. In this section, we describe the characteristics of these databases and explain the performance of the system against these databases. Two data sets, breast-cancer and echocardiogram, contain a number of missing values. Table \ref{data-set} summarizes the characteristics of these data sets. Every database contains numeric attributes, which have been discretized in advance. Throughout the entire experiments in this section, each numerical attribute will be discretized into seven intervals in advance. Too small a number of intervals for discretization degrades the performance and too large a number of intervals increases the complexity of the learning process. Based on our empirical experiments, seven intervals provided us with reasonable accuracy without sacrificing complexity.

Breast Cancer Dataset: This dataset is to predict whether or not breast cancer would recur during five year period. This dataset was provided and tested by Wolberg [20]. Wolberg applied his multisurface method of pattern separation to this data set. Table 2 shows the top 10 rules generated from the rule induction phase of the proposed system. Figure 1 shows that the proposed system produces better perfo-

rmance during the number of rules is between 30 and 70.

Echocardiogram Dataset: This dataset represents a set of people who had recently suffered acute heart attack. The dataset includes several measures taken from echocardiograms, which are ultrasound measurements of the heart itself. The test result is compared with EACH algorithm developed by Salzberg [17], and Figure 2 summarizes the results. The proposed system shows the better results in most cases except the number of rules is 10.

Thyroid Dataset: The objective of this dataset is to predict whether a patient's thyroid belongs to the class euthyroidism, hypothyroidism, or hyperthyroidism. Figure 3 shows the summary of the test results. The test result is compared with that of Coomans's algorithm [5] and his method showed better regardless of the number of rules.

6. Conclusion

The results presented in this paper demonstrate that hybrid algorithms combining rule induction learning and instance-based learning can be tractable, that resulting rules can capture complex interrelationships among attributes in a variety of domains, and that these combined method can thus classify new cases with high accuracy. The system displayed robustness in the face of both noise and incomplete data. Comparison with other classification techniques were quite favorable, and in some domains, the program performed better than any results published before.

The principal goal of performing this work is to allow users to use available databases to understand and predict more accurately the outcomes of future examples of the database. The results presented in this paper suggest that the system constitutes significant progress toward achieving this goal and generates classification rules that classify subsequent cases with high accuracy. The work in this paper indicates that information theoretic multistrategy methods can be applied to this problem, and that resulting rules may be used for classifying new cases, or providing insight into the interrelationships among variables in that domain. The results we presented in this paper demonstrates that the application of information theoretic multistrategy method is promising, however, much work remains to be done.

The proposed system seems to have wide area of applications as we have seen in Section 5. The only requirement the users have to consider before running the system is that the databases must be represented in the form of relational database tables. Besides, the system does not allow the values of attributes form a hierarchical structure.

References

[1] D. Aha, D. Kibler and M. Albert. Instance-based Learning Algorithms. *Machine Learning*, 6(1) pp. 37-66, 1991.

[2] R. J. Beran. Minimum Hellinger Distances for Parametric Models, *Ann. Statistics*, Vol. 5, pp. 445-463, 1977.

[3] J. Cendrowska. PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 1987, 27, pp. 349-370, 1987.

[4] P. Clark and T. Niblett. The CN2 Induction Algorithm, *Machine Learning*, Vol. 3, pp. 261-283, 1989.

[5] D. Coomans, M. Broeckaert, M. Jonckheer, and D. L. Massart, Comparison of Multivariate Discriminant Techniques for CLinical Data-Application to the Thyroid Functional State, Meth. Info. Med. 22, pp. 93-101, 1983.

[6] G. DeJong and R. Mooney. Explanation-Based Learning: An alternative view, *Machine Learning*, 1, pp. 145-176, 1986.

[7] J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. R. Thagard. *Induction: Processes of Inference,*

Learning, and Discovery, Cambridge, MA:MIT Press, 1986.

[8] C. H. Lee and D. G. Shin. A Context-Sensitive Discretization of Numeric Attributes for Classification Learning, *the 11th European Conference on Artificial Intelligence*, Amsterdam, Netherlands, 1994.

[9] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *the Fifth National Conference on Artificial Intelligence, Philadelphia*, PA:Morgan Kaufmann., pp. 1041-1045, 1986.

[10] S. Minton, J. G. Carbonell, O. Etzioni, C. A. Knoblock, and D. R. Kuokka. Acquiring Effective Search Control Rules:Explanation-based Learning in the PRODIGY System, *the 4th International Machine Learning Workshop*, 1987.

[11] D. Ourston and R.J. Mooney. Theory Refinement Combining Analytical and Empirical Methods, *Artificial Intelligence*, 66, pp. 273-309, 1994.

[12] M. J. Pazzani. Integrating Explanation-based and Empirical Learning Methods in OCCAM, *Proceedings of EWSL*, 1988.

[13] M. J. Pazzani and D. Kibler, The Utility of Knowledge in Inductive Learning, *Machine Learning*, 9, pp. 57-94, 1992.

[14] J.R. Quinlan. Generating Production Rules from Decision Trees, the *Int'l Joint Conf. Artificial Intelligence*, pp. 304-307, 1987.

[15] J.R. Quinlan. Induction of decision trees, *Machine Learning*, 1, pp. 81-106, 1986.

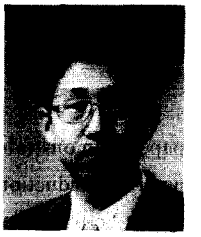
[16] D. E. Rumelhart, J. L. McClelland, and The PDP Research Group. Parallel *Distributed Processing: Explorations in the microstructure of cognition*, Vol. 1, Cambridge, MA:MIT Press, 1987.

[17] S. Salzberg. Exemplar-based learning:Theory and implementation (*Technical Report TR-10-88*). Harvard University, Center for Research in Computing Technology, Aiken Computation Laboratory, 1988.

[18] P. Smyth and R. M. Goodman. An Information Theoretic Approach to Rule Induction from Databases, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 4, 1992.

[19] B. L. Whitehall. Knowledge-based Learning: Integration and Deductive and Inductive Learning for Knowledge Base Completion, Ph.D. Thesis, Computer Science Department, University of Illinois, 1990.

[20] W. H. Wolberg, and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *the National Academy of Sciences, 87*, pp. 9193-9196, 1990.



이 창 환

1982년 서울대학교 계산통계학과 졸업(학사)
1988년 서울대학교 계산통계학과 대학원 졸업(이학석사)
1994년 University of Connecticut 졸업(공학박사)
1994년~1995년 AT&T Bell Laboratories 위촉연구원
1996년~현재 동국대학교 전산통계학과 전임강사
관심분야:기계학습, 인공지능, 인공생명