분류학습을 위한 연속 애트리뷰트의 이산화 방법에 관한 연구

이창환[†]

요약

대부분의 기계학습 방법들은 이산형의 데이타를 학습에 사용되는 데이타의 형식으로 요구하고 있다. 따라서 연속형 데이타의 경우는 기계학습 방법들을 적용하기 전에 그 데이타를 이산형으로 바꾸어 주는 과정이 필요하다. 이러한 이산화 과정은 그 중요성에 비하여 상대적으로 관련 연구가 미비한 수준이다. 따라서 이 논문은 정보이론을 사용하여 연속형 자료를 이산형의 형태로 변환시키는 새로운 방법을 제안하였다. 각 애트리뷰트의 값들이 목적 애트리뷰트에 제공하는 정보의 량을 엔트로피 함수의 일종인 Hellinger 변량을 이용하여 계산하였으며, 각 애트리뷰트마다 제공하는 정보의 손실을 최소화할 수 있는 이산화 경계선을 계산하였다. 본 논문이 제안한 방법의 성능을 ID3 와 신경망 알고리즘을 사용하여 기존의 이산화 방법들과 비교하였으며 거의 대부분 우수한 정확성을 보였다.

Discretization of Continuous-Valued Attributes for Classification Learning

Chang-Hwan Lee[†]

ABSTRACT

Many classification algorithms require that training examples contain only discrete values. In order to use these algorithms when some attributes have continuous numeric values, the numeric attributes must be converted into discrete ones. This paper describes a new way of discretizing numeric values using information theory. Our method is context-sensitive in the sense that it takes into account the value of the target attribute. The amount of information each interval gives to the target attribute is measured using Hellinger divergence, and the interval boundaries are decided so that each interval contains as equal amount of information as possible. In order to compare our discretization method with some current discretization methods, several popular classification data sets are selected for experiment. We use back propagation algorithm and ID3 as classification tools to compare the accuracy of our discretization method with that of other methods.

1. Introduction

Discretization is a process which changes continuous numeric values into discrete categorical values. It divides the values of a numeric attribute into a number of intervals, where each interval can be mapped to a discrete categorical or nominal symbol. Most real-world

[†] 정회원:동국대학교 전산통계학과 전임강사

논문접수:1996년 11월 5일, 심사완료:1997년 5월 6일

applications of classification algorithm contains continuous numeric attributes. When the feature space of data includes continuous attributes only or mixed type of attributes(continuous type along with discrete type), it makes the problem of classification vitally difficult. For example, classification methods based on similarity-based measures are generally difficult, if not possible, to apply to such data because the similarity measures defined on discrete values are usually not compatible with similarity of continuous values. Alternative methodologies such as probabilistic modeling, when applied to continuous data, require an extremely large amount of data.

Despite the importance of the discretization issue, research in machine learning has not paid enough attention to discretizing numeric attributes and few algorithms perform discretization automatically. However, poorly discretized attributes prevent classification systems from finding important inductive rules. For example, if the ages between 15 and 25 mapped into the same interval, it is impossible to generate the rule about the legal age to start military service. In addition, poor discretization makes it difficult to distinguish the non-predictive case from poor discretization. In most cases, inaccurate classification caused by poor discretization is likely to be considered as an error originated from the classification method itself. In other words, if the numeric values are poorly discretized, no matter how good our classification systems are, we fail to find some important rules in databases.

In this paper, we describe a new way of discretizing numeric attributes. We discretize the continuous values using a minimum loss of information criterion. Our discretization method takes into consideration the class values of examples, and adopts information theory as a tool to measure the amount of information each interval contains. A couple of typical machine learning data sets are selected for discretization, and these are discretized by both traditional discretization methods and our proposed method. To compare the correctness of the discretization results, we use the back propagation algorithm and ID3 as the classification algorithms to read and classify data.

2. Previous Work

Although discretization influences significantly the effectiveness of classification algorithms, few studies have been done because it usually has been considered a peripheral issue. Among them, we describe two discretizing methods in machine learning literature. A simple method, called equal distance method, is to partition the range between the minimum and maximum values into N intervals of equal width. Thus, if L and H are the low and high values, respectively, then each interval will have width $W=(H-L)/N$. However, when the outcomes are not evenly distributed, a large amount of information may be lost after discretization using this method. Another method, called equal frequency method, chooses the intervals so that each interval contains approximately the same number of training examples; thus, if $N=10$, each interval would contain approximately 10% of the examples. These algorithms are very simple, easy to implement, and in some cases produce a reasonable discretization of data. However, there are many cases where they cause serious problems. For instance, suppose we are to discretize attribute age, and reason about the retirement age of a certain occupation. If we use the equal distance method, ages between 50 and 70 may belong to one interval, which prevents us from knowing what the legal retirement age is. Similarly, if we use the equal frequency method to discretize attribute weight, the weights greater than 180 pounds may belong to one interval, which prevents us to reason about the health problem of the persons who are overweight. With both of these discretizations it would be very difficult or almost impossible to learn certain concepts. The main reason for this is that they ignore the class values of the examples, making it very unlikely that the interval boundaries will just happen to occur in the places which best facilities accurate classification.

Some classification algorithms such as C4 [8], CART [2], and PVM [12] take into account the class information when constructing intervals. For example, in C4, a member of the ID3 [9] family of decision tree algorithms, an entropy measure is used to select the best attribute to branch on at each node of the decision tree. And that measure is used to determine the best cut point for splitting a numeric attribute into two intervals. A threshold value, T, for the continuous numeric attribute A is determined, and the test $A \leq T$ is assigned to the left branch while $A > T$ is assigned to the right branch. This cut point is decided by exhaustively checking all possible binary splits of the current interval and choosing the splitting value that maximizes the entropy measure. CART, developed by [2], takes into account the class information as well but it just splits the range into two intervals. It selects the interval boundary which makes the information gain gap between the two intervals maximum. This process is carried out as part of selecting the most discriminating attribute. Fayyad [5] has extended the method of binary discretization in CART and C4, and introduced multi-interval discretization using minimal description length technique. However, these algorithms differ in that discretization is performed dynamically as the algorithm runs, not as a preprocessing step. It is not obvious how such techniques should be used to perform static(non-dynamic) discretization when more than two intervals per attribute are desired. Some classification algorithm can be easily extended to discretize dynamically, but many can not.

Even for algorithms that could use a dynamic method, it might still be preferable to use static discretization. Using static discretization as a preprocessing step, we can see significant speed up for classification algorithm with little or no loss of accuracy [3]. The increase in efficiency is because the dynamic C4/CART algorithm must re-discretize all numeric attributes at every node in the decision tree while in static discretization all numeric attributes are discretized only once before the classification algorithm runs. One of the major problems in dynamic discretization is that it is expensive. Although it is polynomial in complexity, it must be evaluated $N-1$ times for each attribute where N means the number of distinct values. Since classification programs are designed to work with large sets of training sets, N is typically very large. Therefore, algorithms like ID3 runs very slowly when continuous attributes are present. In addition, the real performance of binary discretization is not proven when there are more than two classes in the problem. As the algorithm attempts to minimize the weighted average entropy of the two sets in the candidate binary partition, the cut point may separate examples of one class in an attempt to minimize the average entropy.

3. Information-Theoretic Discretization

With the traditional discretization methods, it is seldom possible to feel confident that a given discretization is reasonable because these methods do not provide any justifications for their discretizations. A classification algorithm can hardly distinguish a non-predictive case from a poorly discretized attribute and the user cannot do so without examining the raw data. In general, it is seldom possible to know what the correct or optimal discretization is unless the users are familiar with the problem domain. Another problem which complicates evaluation is that discretization quality depends on the classification algorithms that will use the discretization. Even though it is not possible to have an optimal discretization with which to compare results, some notion of quality is needed in order to design and evaluate a discretization algorithm. The primary purpose of discretization, besides eliminating numeric values from the training data, is to produce a concise summarization of a numeric attribute. An interval is essentially a summary of the relative frequency of classes within that interval. Therefore, in an accurate discretization, the relative class frequencies should be fairly consistent within an interval(otherwise the interval should be split to express this difference) but two adjacent intervals should not have similar relative class frequencies(otherwise the intervals should be combined to make the discretization more concise). Thus, the defining characteristic of a high quality discretization can be summarized as: maximizing intra-interval uniformity and minimizing inter-interval uniformity. Our method achieves this notion of quality by using an entropy function. The difference between the class frequencies of the target attribute and the class frequencies of a given interval is defined as the amount of information that the interval gives to the target attribute. The more different these two class frequencies are, the more information the interval is defined to give to the target attribute. Therefore, defining an entropy function which can measure the degree of divergence between two class frequencies is crucial in our method and will be explained in the following.

Calculating Information Content of Intervals

The basic principle of our discretization method is to discretize numeric values so that the information content of each interval is as equal as possible. Therefore, the critical part of our method is to select or define an appropriate measure of the amount of information each interval gives to the target attribute. In our approach, the interpretation of the amount of information is defined in the following. For a given interval, its class frequency distribution is likely to differ than that of the target attribute. The amount of information an interval provides is defined as the dissimilarity(divergence) between these two class frequencies. We employ an entropy function in order to measure the degree of divergence between these two class frequencies. Some entropy functions have been used in this direction in machine learning literature. However, the purpose of these functions are different from that of ours. They are designed to decide the most discriminating attributes [9] or generate inductive rules from examples [4]. Suppose X is the target attribute and it has k discrete values, denoted as $x_1, x_2, \ldots, x_k$. Let $p(i)$ denote the probability of x_i. Assume that we are going to discretize an attribute A with respect to the target attribute X. Suppose $A = a_i$ and $A = a_{i+1}$ are boundaries of an interval, and this interval is mapped into a discrete value a. Then the probability distribution of X under the condition that $a_i < A < a_{i+1}$ is different from a priori distribution of X. We will introduce several studies for measuring divergence from the information theory literature.

In information theory literature, several studies are done about divergence measure. Kullback [7] derived a divergence measure, called I-measure, defined as

$$\sum_i p(x_i|a) \log \frac{p(x_i|a)}{p(x_i)}$$

This measure is the average mutual information between the attributes X and A with the expectation taken with respect to the a posteriori probability distribution of X. This measure appears in the information theoretic literature under various guises. It can be viewed as a special case of the cross-entropy or the discrimination, a measure which defines the information theoretic similarity between two probability distributions. Another group of divergence widely used in information theory literature are Bhattacharyya divergence [1] and Renyi divergence [10], and these are defined, respectively, in the following.

$$-\log \sum_i \sqrt{p(x_i)p(x_i|a)} \quad \text{and} \quad \frac{1}{1-\alpha}$$

$$\log \sum_i p(x_i)^{\alpha} p(x_i|a)^{1-\alpha}$$

where $\alpha > 0$. In Renyi divergence, the range of function can be changed depending on the value α. These measures including Kullback divergence become zero if and only if $p(x_i)p(x_i|a_i)$ for all i, and have been used in some statistical classification problems. However, since these measures are originally defined on continuous variables, there are some problems when these are applied to discrete values. These measures are not applicable in case one or more than one of the $p(x_i)$s are zero. Suppose that one class frequency of a priori distribution is unity and the rest are all zero. Similarly, one value of a posteriori distribution is unity and the rest are all zero. Then Kullback divergence, Renyi divergence and Bhattacharyya divergence are not defined in this case, and we cannot apply these directly without approximating the original values. Therefore, in this paper, we adopt a new entropy function, called Hellinger divergence [6], which is defined as

$$|\sum_i \sqrt{p(x_i)} - \sqrt{p(x_i|a)})^2|^{1/2}$$

Unlike other divergence measures, this measure is applicable to any case of probability distribution. In other words, Hellinger measure is continuous on every possible combination of a priori and a posteriori values. It can be interpreted as a distance measure where distance corresponds to the amount of divergence between a priori distribution and a posteriori distribution. It becomes zero if and only if both a priori and a posteriori distributions are identical, and ranges from 0 to $\sqrt{2}$. Therefore, we employ Hellinger divergence as a measure of divergence, which will be used as the information amount of intervals. The entropy of an interval described above, say I, is defined in the following.

$$E(I) = |\sum_i (\sqrt{p(x_i)} - \sqrt{p(x_i|a)})^2|^{1/2}$$

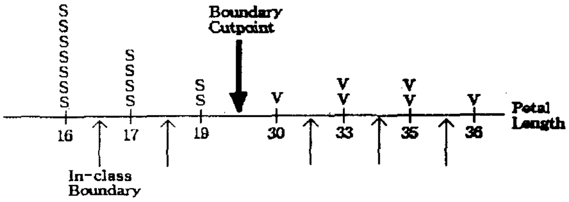
```
Input : $a_1, a_2, \cdots, a_N$ (sorted and distinct numeric values)
$a_0 = a_1$;  $a_{N+1} = a_N$
K:=maximum number of interval;
/* Initialization step */
for i=1 to N do
        $INTVL \leftarrow \{I_i = (p_i, q_i) | p_i = (a_{i-1} + a_i)/2, q_i = (a_i + a_{i+1})/2\}$
   end
/* Entropy of each interval */
for each I_i ∈ INTVL do
        $E(I_i) = |\sum_j (\sqrt{p(a_j)} - \sqrt{p(a_j|I_i)})^2|^{1/2}$
   end
/* Entropy of each cutpoint */
for i=1 to N-1 do
        $E(p_i) = E(I_i) - E(I_{i+1})$;
   end
repeat N-K times do
        MERGE=cutpoint with least value of {E};
        merge two intervals of MERGE;
   end
return INTVL;
```

(Figure 1) Discretization Algorithm

4. Discretizing Algorithm

The algorithm consists of an initialization step and a bottom up combining process. The training examples are sorted according to their values for the attribute being discretized and then each example becomes its own interval. In combining process, the amount of information that each interval gives to the target attribute is calculated using Hellinger divergence. For each pair of two adjacent intervals, the system computes the informational difference between them. The least value of difference will be selected and its corresponding pair of intervals will be merged. Merging process continues until the system reaches the maximum number of intervals usually given by users. The value of K, maximum number intervals, is determined by selecting a desired precision level the user wants. The standard recommended value of K is to set the value between 5 to 10 depending on the domain to prevent an excessive number of intervals from being created. Figure \ref{abstract-algorithm} shows the abstract algorithm of the discretization method.

As part of the initialization step, the numeric values are first sorted by increasing order, and the midpoint between each successive pair of values in the sorted sequence is called potential *cutpoint*. Each cutpoint associates two adjacent intervals(or point values). If the class frequency of these two intervals are exactly the same, the cutpoint is called *in-class cutpoint*, and if not, the cutpoint is called *boundary cutpoint*. In other words, if two adjacent point values or intervals have different class frequencies, their midpoint(cutpoint) is defined as boundary cutpoint. Figure 2 shows the example of cutpoints and boundary cutpoints of petal



(Figure 2) In-class cutpoints and boundary cutpoints

length attribute in iris data set. Intuitively, discretization at in-class cutpoints are not desirable because it separates examples of one class. Therefore, boundary cutpoint must have high priority to be selected for discretization. We have the following theorem which shows the correctness of our discretization algorithm.

Theorem 1 *The in-class cutpoints are not to be selected for discretization unless all boundary cutpoints are exhausted for discretization.*

Proof is omitted due to space limitation. This theorem implies that in our algorithm discretization keeps occurring only at boundary cutpoint unless it exhausts all boundary cutpoints. By doing so, it prevents the in-class cutpoints from being selected for discretization.

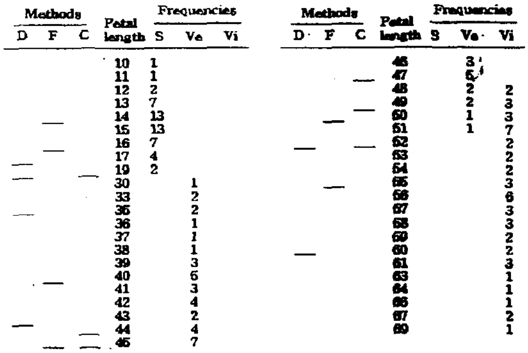
Another advantage of our method is that our discretization method has very low computational complexity. Its computational complexity is given as $O(n)$, where n is the number of examples. Therefore, suppose a training database has l numeric features, our discretization method will take $O(ln)$ time complexity to discretize the features.

5. Experimental Results

The behavior of the algorithm will be demonstrated using an example. To show the validity of our discretization algorithm, we selected iris data as a test data. This well-known data set has been used for many previous classification algorithms. Fisher's paper is a classic in the field and is referenced frequently to this day. Each of the iris data consists of four integer-valued variables plus a known assignment of the example to a particular species of iris. The data covers three different species: setosa, versicolor, and virginica. The four variables measured are sepal length, sepal width, petal length, and petal width. The ranges of these variables are 43-79, 20-44, 10-69, and 1-24, respectively. For the purpose of test, we discretized the values of petal length into seven intervals using equal distance method, equal frequency method and our context-sensitive discretization method, respectively. The discretization results are shown in Figure 3. It is not easy to completely assess the correctness of discretized intervals because its real validity does not show up until the discretized intervals are used by a classification algorithm. However, we can notice that all plants with petal length less or equal to 19 belong to setosa while all plants greater than 51 belong to virginica. If both 19 and 30 belong to the same interval or both 51 and 52 belong to the same interval, we cannot derive such possibly important rules as

If petal-length < 30, then species = setosa.
If petal-length > 51, then species = virginica

Figure 3 also shows the results of discretization carried out by equal distance method and equal fre-



Methods			Petal	Frequencies		
D	F	C	length	S	Ve	Vi
			10	1		
			11	1		
			12	2		
			13	7		
			14	13		
	—					
			15	13		
			16	7		
	—					
			17	4		
—						
			19	2		
—		—				
			30		1	
			33		2	
—						
			35		2	
			36		1	
			37		1	
			38		1	
			39		3	
			40		6	
	—					
			41		3	
			42		4	
			43		2	
—						
			44		4	
		—				
			45		7	
	—	—				
			46		3	
			47		5	
		—				
			48		2	2
			49		2	3
		—				
			50		1	3
	—					
			51		1	7
—		—				
			52			2
			53			2
			54			2
	—					
			55			3
			56			6
			57			3
			58			3
			59			2
—						
			60			2
			61			3
			63			1
			64			1
			66			1
			67			2
			69			1

(Figure 3) Petal length data for iris

quency method for the purpose of comparison. We can see that these methods do not guarantee to cut at both between (19 and 30) and (51 and 52).

Comparison

Because our discretization method is not itself a classification algorithm it cannot be tested directly for classification accuracy, but must be evaluated indirectly in the context of a lassification algorithm. Therefore, our discretization method, the equal-distance intervals and equal-frequency intervals will be used to create intervals for two well-known classification systems: back propagation neural network classifier [11] and ID3 [9]. These system are chosen because they are widely known, thus requiring no further description.

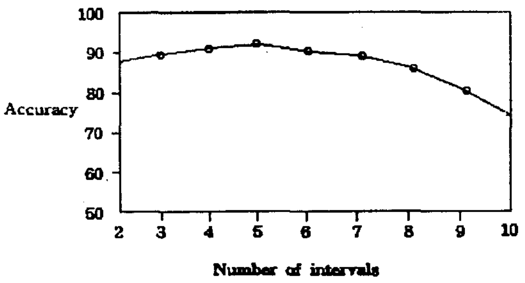
For the test data set, we have chosen two data sets: iris flower data and Indian diabetes. (Obtained from the University of California-Irvine machine learning database repository: ics.uci.edu.) These are chosen because all attributes of these data sets are numeric and these data sets have been used by a number of other classification algorithms. Iris flower database is already described above. Indian diabetes data is a data set about whether patients show signs of diabetes according to World Health Organization criteria. All patients in this data set are selected from the females at least 21 years old of Indian heritage.

〈Table 1〉 Classification results using back propagation algorithm

Database	Equal distance	Equal frequency	Context-sensitive
Iris flower	68.6 ± 5.3%	89.1 ± 6.4%	95.5 ± 3.4%
Indian diabetes	69.5 ± 6.4%	66.9 ± 5.8%	77.3 ± 4.7%

〈Table 2〉 Classification results using ID3

Database	Equal distance	Equal frequency	Context-sensitive
Iris flower	87.8 ± 4.1	86.6 ± 6.2	91.2 ± 6.8
Indian diabetes	67.1 ± 3.1	68.8 ± 2.3	71.1 ± 2.9



(Figure 4) Classification accuracy versus number of intervals

For the sake of simplicity, all attributes are discretized into seven intervals. For each data set, the 2/3 of data set is selected randomly and used as training data and the rest 1/3 is used as test data for classification. For back propagation algorithm, the neural network has one hidden layer with 6 hidden units, and the initial weights and biases are decided randomly between −0.5 and 0.5. Also each data set is read 500 times(epoches) for learning. Table 1 shows the classification results of each discretization method. As we can see, our context-sensitive discretization shows better results for both data sets. For iris flower data, equal frequency discretization shows better performance than equal distance discretization, while equal distance discretization is a little better than equal frequency discretization in Indian diabetes data. However, in both cases, our discretization is superior to other methods. The same data set is used for ID3 algorithm for classification. Table 2 shows the results of classification for each data set using ID3, and we can easily see that our discretization method shows the better classification accuracy than other methods.

Determining the right value of maximum number of interval significantly effects the correctness of discretization. Too small number of interval prevents important cutpoints from being discretized while too many cuts produce unnecessary intervals. In order to see the effect of the number of intervals, we applied back propagation algorithm to iris data set with different number of intervals, and the results are shown in Figure 4. For iris data set, when the attribute is discretized into 3-5 intervals, its classification result shows best accuracies while the number of interval is greater than 8 or less than 3, the classification accuracy drops significantly.

6. Conclusion

In this paper, we proposed a new way of discretizing numeric attributes, considering class values when discretizing numeric values. Using our discretization method, the user can be fairly confident that the method will seldom miss important intervals or choose an interval boundary when there is obviously a better choice because discretization is carried out based on the information content of each interval about the target attribute. In contrast, the equal-distance interval and equal-frequency interval methods can produce extremely poor discretization. Our algorithm is easy to apply because all it requires for users to do is to provide the maximum number of intervals. Our method can be applied virtually to any domain. It is applicable to multi-class learning(i.e. domains with more than two classes-not just positive and negative examples). Another benefit of our method is that it provides a concise summarization of numeric attributes, an aid to increasing human understanding of the relationship between numeric features and the class attributes.

One problem of our method is the lack of ability to distinguish between true correlations and coincidence. In general, it is probably not very harmful to have a few unnecessary interval boundaries; the penalty for excluding an interval is usually worse, because the classification algorithm has no way of making a distinction that is not in the data presented to it.

References

[1] T. Kadota and L. A. Shepp, On the Best Finite Set of Linear Observables for discriminating two Gaussian signals, *IEEE Transactions on Information Theory*, Vol. 13, pp. 278-284, 1967.

[2] L. Breiman, J. H. Fiedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, CA, 1984.

[3] J. Catlett, On changing continuous attributes into ordered discrete attributes. In *European Working Session on Machine Learning*, 1991.

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

[5] U. M. Fayyad and K. B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *13th International Joint Conference of Artificial Intelligence*, pp. 1022-1027, 1993.

[6] Z. Ying, Minimum Hellinger Distance Estimation for Censored Data, *The Annals of Statistics*, Vol. 20, No. 3, pp. 1361-1390, 1992.

[7] S. Kullback, *Information Theory and Statistics*, New York: Dover Publications, 1968.

[8] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus, Inductive knowledge acquisition: a case study. In J. R. Quinlan, editor, *Applications of Expert Systems*, Addison-Wesley, Sydney, pp. 157-173, 1987.

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

[10] A. Renyi, On Measures of Entropy and Information, *Proceedings of Fourth Berkeley Symposium*, Vol. 1, pp. 547-561, 1961.

[11] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, 1986.

[12] S. M. Weiss, R. S. Galen, and P. V. Tapepalli, Maximizing the predictive value of production rules, *Artificial Intelligence*, 45: 47-71, 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