Journal of the Korea Convergence Society
Vol. 9. No. 12, pp. 9-15, 2018

ISSN 2233-4890
https://doi.org/10.15207/JKCS.2018.9.12.009

Application Examples Applying Extended Data Expression Technique to Classification Problems

Jong Chan Lee
Professor, Deptartment of Computer Engineering, ChungWoon University

패턴 분류 문제에 확장된 데이터 표현 기법을 적용한 응용 사례

이종찬
청운대학교 컴퓨터공학과 교수

Abstract The main goal of extended data expression is to develop a data structure suitable for common problems in ubiquitous environments. The greatest feature of this method is that the attribute values can be represented with probability. The next feature is that each event in the training data has a weight value that represents its importance. After this data structure has been developed, an algorithm has been devised that can learn it. In the meantime, this algorithm has been applied to various problems in various fields to obtain good results. This paper first introduces the extended data expression technique, UChoo, and rule refinement method, which are the theoretical basis. Next, this paper introduces some examples of application areas such as rule refinement, missing data processing, BEWS problem, and ensemble system.

Key Words : Extended data expression, Classification, Learning, Rule refinement, Missing data

요 약 확장된 데이터 표현의 주요 목표는 유비쿼터스 환경에서 일반적인 문제에 적합한 데이터 구조를 개발하는 것이다. 이 방법의 가장 큰 특징은 속성 값을 확률로 표현할 수 있다는 것이다. 다음 특성은 훈련 데이터의 각 이벤트가 중요도를 나타내는 가중치 값을 갖도록 한다는 것이다. 데이터 구조가 개발된 후에 이를 학습할 수 있는 알고리즘이 고안된다. 그 동안 이 알고리즘은 여러 분야에서 여러 문제에 적용하여 좋은 결과를 산출해 왔다. 본 논문은 먼저 데이터 표현 기법인 UChoo를 소개하고 이론적인 배경이 되는 규칙 개선 문제를 소개한다. 그리고 규칙 개선, 손실 데이터 처리, BEWS 문제, 앙상블 시스템과 같은 응용 분야의 예를 소개한다.

주제어 : 확장된 데이터 표현, 분류, 학습, 규칙 개선, 손실 데이터

1. Introduction

A new problem arises in the ubiquitous environment where data collected in different environments should be used as learning data. First, if the data collected at location A has 3 attributes and the data collected at location B has 4 attributes, then the problem of learning these data together is an example. The second is a rule refinement problem that must deal with the "rule + new learning data" type. In other words, new data is added to information in the form of a rule. And as data is collected from remote sensors, it is a problem that must

*This paper is sponsored by the academic research project of Chungwoon University in 2018. (No. 2018-51)
*Corresponding Author : Jong Chan Lee (jclee@chungwoon.ac.kr)
Received October 5, 2018 Revised October 31, 2018
Accepted December 20, 2018 Published December 28, 2018

be learned with partially damaged loss data. However, no adequate method has been suggested to solve these problems. As a solution to this problem, a new learning data structure called extended data expression has been proposed, and UChoo, which is a modified version of this data structure to C4.5 [1], has been developed. In the meantime, UChoo [2-5] has been reported to produce very good results by applying it to various fields and application problems. This paper analyzes the algorithms, introduces the existing application fields, and tries to summarize the contents needed to deal with them incidentally.

A typical characteristic of the extended data expression is that the attribute values can be presented as a probability. For example, if the "headache" attribute has 3 cardinalities such as {high, medium, low}, then the previous method should choose one of 3 values. And in the case of class, you also had to choose one of {Flu, Normal}. However, the new data structure allows for probability values such as {High = 20%, Medium = 30%, High = 50%} in the headache attribute and {Flu = 40% and Normal = 60%} in the class. The advantage of this method is that the "Normal" value in the case of the class is 20% higher than the "flu", so if you judge it to be "normal", the information that the probability of "flu" is 40% is lost. Therefore, it has the advantage of preserving small information when expressed as probability.

The second characteristic has a weight for each event in the training data. This weight can have a probability value or mean the number of samples. For example, if the weight of A event is 0.2 and the B event is 0.6, then A event is three times as important as B event. This is a significant role in the voting system with the results of several classifiers, such as the ensemble system.

2. Background

2.1 Extended Data Expression

Table 1. An example of a simple training data set

Age	Case History	Phenomenon	Class
50	Yes	Low	Observation
60	No	Low	Normality
50	No	Middle	Observation
70	No	High	Emergency
60	Yes	Middle	Emergency

Table 1 is an example data set collected for the prevention of sudden death. This table consists of one numerical attribute(“Age”), two nominal attributes(“Case History”, “Phenomenon”) and a class label. The difference between the numerical and normative attributes is whether or not there is an order in the constituent values. Among these attributes, "age" refers to the age at which sudden death begins to occur. The "Case History" attribute is categorized as "Yes" or "No" depending on whether a person has a genetic disease in your family or has had a similar illness in the past. Symptoms of suspected stroke include severe dizziness, sudden loss of vision, and confusion of consciousness. And sleep disorders, sleep apnea, and cardiac arrest are known to be directly related to sudden death. Therefore, the "Phenomenon" attribute is classified as "High" regardless of other conditions if there is a single symptom. Stroke is known to be an important factor of sudden death. The causes of stroke are family history, hypertension, diabetes, smoking, drinking and obesity. The "Phenomenon" attribute assigns a value of "Low(Lo)" if it has less than 2 elements, "Middle(Mid)" if it has 3 or 4 elements, and "High(Hi)" if it has more than 5 elements. Classes are classified as "Normality(Nor)", "Observation(Obs)", and "Emergency(Eme)" depend-ing on the values of the attributes that make up each record.

Table 2 shows the modified expression of the training data set of Table 1. It simply put 0 and 1 into the entry of the training data set in order to help understanding on the data expression. More precisely, it fills each entry with a probability value between 0 and 1. Using this method has the effect of saving some

Table 2. Transform Table 1 into Extended Data Expression.

Event	Weight	Age			Case History		Phenomenon			Class		
		50	60	70	Yes	No	Low	Middle	High	Nor	Obs	Eme
1	1	1	0	0	1	0	1	0	0	0	1	0
2	1	0	1	0	0	1	1	0	0	1	0	0
3	1	1	0	0	0	1	0	1	0	0	1	0
4	1	0	0	1	0	1	0	0	1	0	0	1
5	1	0	1	0	1	0	0	1	0	0	1	1
6	20	0	1	0	1/2	1/2	0	1	0	0	2/3	1/3

of the uninformed information instead of excluding it from learning when some attribute values are corrupted. In addition, it is possible to combine data composed of different attributes in a ubiquitous environment. In particular, note the event 6 in Table 2. First, the weight value of this event has 20. Here, the weight is a measure of how important this event is. This means that it has 20 times more importance than other events with a weight value of 1. This allows the expert to intervene and act as a bias to use critical material for learning. Second, notice that the "Case History" attribute and class fill the entry with probability values. This is a way to preserve this event if the attribute value is corrupted. In addition, when learning is complete, a decision tree is created to form the rules. This method is needed when combining this rule with new learning data.

2.2 UChoo

UChoo[2-5] is a classification algorithm that modifies C4.5[1] to fit the format of the extended data representation method.

- k(n) : the cardinality of the class(each attribute)
- $C_i(r)$: a value indicating the probability that the r^{th} event belongs to C_i. Where, $\sum_{i=1}^{k} C_i(r) = 1$
- $O_{Aj}(r)$: the probability that the attribute A has j in the r^{th} event. Where, $\sum_{j=1}^{n} O_{A_j}(r) = 1$
- T_{Aj} : the event number whose attribute A has a value j in the total event T.
- $e(T)$: the event number in the set T.
- $Weight(r,T)$: the weight value of the r^{th} event in the set T.
- $freq\#(C_i,T)$: the instance number in T whose class value is $C_{i..}$

$$freq\#(C_i,T) = \sum_{r=1}^{e(T)} Weight(r,T)\, C_i(r)$$

- $freq\#(C_i,T_{Aj})$: the instance number in T_{Aj}, whose class value is C_i

$$freq\#(C_i,T_{A_j}) = \sum_{r=1}^{e(T)} Weight(r,T)\, C_i(r)\, O_{A_j}(r)$$

- $|T_{Aj}|$: the instance number in the set T_{Aj}.

$$|T_{A_j}| = \sum_{r=1}^{e(T_{A_j})} Weight(r,T_{A_j})\, O_{A_j}(r)$$

2.3 Rule Refinement

Rule refinement refers to the problem of creating an improved rule by adding a new data set to an existing rule. It is actually used when there is only rule without original training data set. At this point, the key is how to extract information from the rule and merge it with the newly collected data set.

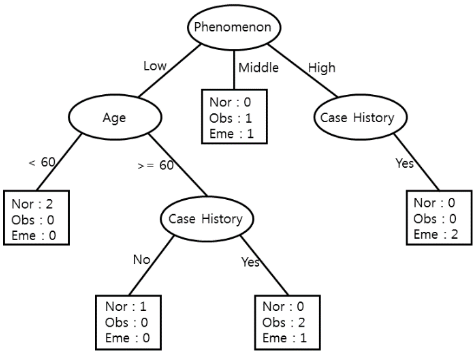


Fig. 1. An example of decision tree.

Table 3. The information derived from the Rules in Fig. 1.

Event	Weight	Age			Case History		Phenomenon			Class		
		50	60	70	Yes	No	Low	Middle	High	Nor	Obs	Eme
1	2	1	0	0	1/2	1/2	1	0	0	1	0	0
2	1	0	1/2	1/2	0	1	1	0	0	1	0	0
3	3	0	1/2	1/2	1	0	1	0	0	0	2/3	1/3
4	2	1/3	1/3	1/3	1/2	1/2	0	1	0	0	1/2	1/2
5	2	1/3	1/3	1/3	1	0	0	0	1	0	0	1

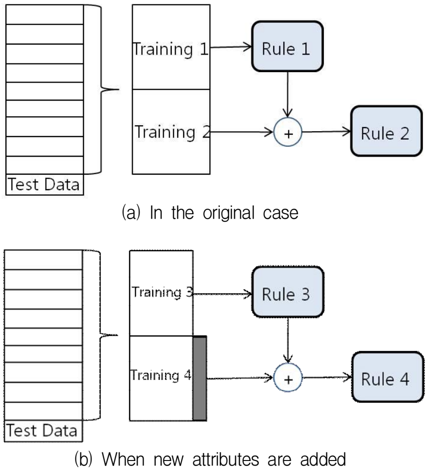
First, since the sample value of the first leaf node (Rule 1) in Fig. 1 is 2 (Nor: 2), the weight of Event 1 in Table 3 is 2. And since the "Case History" attribute does not exist in the path to the leaf, this attribute has no effect on the process of creating Rule 1. Therefore, since the cardinality value of this attribute is 2, it can be equally allocated with a probability of 1/2. In this way, information is extracted from the rule and then composed into a table, as shown in Table 3.

3. Application Examples

3.1 Rule refinement problem using Extended Data Expression

Assume that the original data has b attributes. Over time, new sensors were added to the network. That is, the attribute is increased by one from the original one, resulting in b + 1. In this case, will you discard the data that has been accumulating for a long time and accept new data? The loss of information will be too great. To solve this problem, UChoo [4,5] is proposed and experimented.

As shown in Fig. 2 (a), Training 1 changes data using the data extension method described in Section 2.1 and builds a decision tree through learning. Then, this decision tree is transformed into the data form of Rule 1. At this time, Training 2 is added as new data, and this new data is combined with this transformation data (Rule 1 + Training 2). Then, a new learning is performed and a decision tree of a new rule (Rule 2) is calculated. The final result is obtained from the average of these experimental results after 10 such experiments.



(a) In the original case

(b) When new attributes are added

Fig. 2. Rule refinement problem

For the experiment in which new attributes are added, Training 3 is created by deleting one attribute from Training1 as shown in Fig. 2 (b). From Training3, Rule 3 is created using UChoo and new data, Training4, is merged with Rule 3. These processes confirm that rules and new data can be combined. Rule on storage capacity is too simple to compare with data. This is because it is the rule that extracts the most important information from the data. First of all, even if the learning data is lost or modified, it can be seen that new rules can be generated by combining with new data with only rules. In many areas, the data will be added in succession. This is an appropriate algorithm for this situation.

3.2 Handling of incomplete data

When one or more attribute values are missing from a particular attribute vector, it is called incomplete data.

The study of the processing of incomplete data can be divided into two major categories. First, Quinlan [1] and Fridman [6] performed a method to ignore and process incomplete data. Second, Hathaway [7], Dempster [8], and Hong [9] are examples of how to compensate lost values with appropriate alternatives. The proposed method belongs to the second method. When incomplete data is input into learning or testing process, the value of missing data is compensated first and then it is learned by using UChoo algorithm.

This method is simple, but it produces better results than ignoring the whole damaged event. It is a result of compensating for some of the damaged information. Another example is an algorithm that improves the method of compensating the missing value by a more statistical method. This method uses OCS algorithm among four algorithms proposed by Hathaway [7] (WDS, PDS, OCS, NPS) to fill the missing value. In this method, the cost problem of OCS is found by using MFA algorithm [11]. That is, the value that minimizes the objective function of OCS is repeatedly obtained using the MFA, and the minimum value is filled in the missing value. OCS uses FCM(Fuzzy c-Means) clustering algorithm to compensate missing data. FCM is a way of expressing the membership function as a continuous value between 0 and 1 in cases where the boundaries of the cluster are ambiguous.

3.3 The Biological Early Warning System (BEWS)

BEWS is a system that puts a fish in a given fish tank and observes the fish's behavior and generates an alarm in case of abnormal movement. BEWS continuously monitors the biological response of organics in the water to detect toxicity. Therefore, it has many merits such that it can grasp the water quality in real time and can detect the toxicity which was not found by the conventional chemical method. This is particularly useful for detecting intermittent toxic events in any environment.

The problem of extracting appropriate characteristics is the most important part of the classification. Let T be the sampling time, assuming that fish will visit point n for a given T time. In order to quantify the behavioral trajectory of a fish during T time, the attributes that can represent the trajectory are defined by 6 characteristic values [10] as x coordinate, y coordinate, distance, absolute distance, angle and fractal dimension. The vectors of such variables describe the behavior of the fish and use the vectors to create rules. One of these variables is fractals. The fish in the aquarium is observed from the front, so they show a trajectory moving on a 2-dimensional plane. Fractal variables are used to express how active they are at any given time. Feature extraction is used to analyze the data obtained from the monitor device and added to the new data for the classifier. A classifier generated from new data can be applied to detect toxicity in real time. During training, data is collected from clean and contaminated water. Put the toxicity into the water where the fish live and record the movement of the fish. Class 0 is clear water and class value is assigned according to degree of contamination. During the test, data is captured from the monitor device and input to the classifier obtained during the training. At each alarm time an average of the output values is calculated and this value determines if the environment is contaminated.

3.4 Ensemble System

Ensemble system [12-16] combines two or more classifiers to produce a result. This system is divided into Boosting and Bagging processes. Boosting refers to increasing the probability that an incorrectly classified event in the training data can be included in the next training data. The weight in Section 3.1 is appropriate for this Boosting function. Bagging is a technique for determining the results of weak classifiers through voting. First, the boosting algorithm uniformly initializes the database distribution to be used. That is, all events are selected with the same probability and included in the training set. Each weak classifier generates a hypothesis through learning from a set of training events. These hypotheses combine to produce results through voting. The weights of misclassified events are updated according to the performance of the hypothesis that is the result of the current weak classifier. In the next step, the training set is selected again using the distribution of weights. In this process, the weights of events that are difficult to classify are intensified and the probability of being selected increases accordingly.

Assume that there are 10 events at the terminal node of the decision tree, of which 6 are class 1 and 4 are class 2. The weak classifier assigns class 1 to this terminal node. Therefore 4 event information with class 2 is ignored. On the other hand, UChoo can reduce the loss of information because it can assign a weight to each event. Applying this example to an extended data representation gives the node a probability of 0.6 for class 1 and 0.4 for class 2. Since weak classifier is not completely classified and learning is stopped by the pruning process, the terminal node has several classes. Instead of selecting only the class information with the highest probability, it considers all the information of each class at the voting stage.

4. Conclusion

The algorithms developed so far in machine learning field are divided into two kinds. The first is a method of extracting feature values from training data and then learning them. This method determines the overall performance of how to extract features. Another method is an algorithm that does not require feature value extraction. That is, if the training data set is input to the algorithm, the algorithm extracts the feature values by itself and executes the learning. Deep learning [13,14], which is in the spotlight nowadays, belongs to this method, but performance depends on how to adjust various learning parameters. So far we can not conclude which method is better and it is a tendency to choose algorithms suitable for learning according to research field. The algorithm introduced in this paper belongs to the former and has been applied to a wide range of applications and has shown excellent results.

Recent studies have attempted to combine extended data expression with deep learning and have confirmed the feasibility of the initial stage. Therefore, extended data expression technique can be applied to application problems that have not been realized until now by merging with various classification algorithms.

REFERENCES

[1] J.R.Quinlan. (1993) C4.5 : Program for Machine Learning, *San Mateo, Calif, Morgan Kaufmann*

[2] D. Kim, D. Lee, & W. D. Lee. (2006) Classifier using extended data expression, *IEEE Mountain Workshop on Adaptive and Learning Systems*, 154-159

[3] D. Kim, D. Seo, Y. Li, & W. D. Lee.(2008) A classifier capable of rule refinement, *International Conference on Service Operations and Logistics, and Informatics*, 168-173.

[4] J. M. Kong, D. H. Seo, & W. D. Lee.(2007) Rule refinement with extended data expression, *Sixth International Conference on Machine Learning and Applications*, 310-315

[5] D.H.Lee, C.Song, & W.D.Lee.(2007), A classifier capable of handling new attributes, *IEEE Symposium on Computational Intelligence and Data Mining*, 323-327.

[6] J. W. Friedman. (1977), A recursive partitioning decision rule for non parametric classification, *IEEE Transaction on Computer Science*, 404- 408.

[7] R. J. Hathaway, & J. C. Bezdek. (2001) Fuzzy c-means clustering of incomplete data, *IEEE Transaction on systems, Man and Cybernetics -part B: Cybernetics*, 31(5).

[8] J. Han, & M.Damber.(2001) Data mining : concept and techniques, *Morgan Kaufmann Publishers*

[9] T.P.Hong, L.H.Tseng, & B.C.Chien.(2002) Learning fuzzy rules from incomplete numerical data by rough sets, *IEEE international conference on Fuzzy Systems*, 1438-1443

[10] J. C. Lee, & W.D.Lee.*(2012)* Biological early warning system using UChoo algorithm, *Journal of Information and Communication Convergence Engineering*, 16(1)

[11] J.Wu, Y.S.Kim, C.H.Song, & W.D.Lee.(2008) A new classifier to deal with incomplete data, *International Conference on Software Engineering, Artificial Intelligence, Networking* , 105-110

[12] K.Yang, A.Kolesnikova, & W.D.Lee.(2013) A new incremental learning algorithm with probabilistic weight using extended data expression, *Journal of Information and Communication Convergence Engineering*, 11(4), 258-267

[13] Y. L. Cun, Y. Bengio, & G. Hinton.(2015) Deep learning. *Nature, 521(7553)*, 436−444. DOI : 10.1038/nature14539

[14] J. Lee. (2018) A method of eye and lip region dectection using faster R-CNN in face image, *Journal of the Korea Convergence Society,* 9(1), 1-8, https://doi.org/10.15207/JKCS.2018.9.8.001

[15] J. Z. Kolter & M. A. Maloof(2003), Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift, *Proceedings of the Third International IEEE Conference on Data Mining*, 123 – 130.

[16] J. Z. Kolter, & M. A. Maloof. (2007). Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts, Journal of Machine Learning Research 8 (2007) 2755-2790.

이 종 찬(Jong Chan Lee) [종신회원]

- 1988년 2월 : 충남대학교 (학사)
- 1990년 2월 : 충남대학교 대학원 (석사)
- 1996년 2월 : 충남대학교 대학원 (박사)
- 1996년 3월 ~ 현재 : 청운대학교 컴퓨터공학과 교수

- 관심분야 : 신경회로망, 패턴분류, 정보보호, 데이터압축
- E-Mail : jclee@chungwoon.ac.kr