Journal of the Korea Convergence Society
Vol. 8. No. 2, pp. 27-33, 2017
https://doi.org/10.15207/JKCS.2017.8.2.027

확장된 표현을 이용하는 분류 알고리즘

이종찬
청운대학교 인터넷학과

A Classification Algorithm using Extended Representation

Jong Chan Lee
Dept. of Internet, Chungwoon University

요 약 인터넷을 통해 사용자에게 클라우드 컴퓨팅 서비스를 효율적으로 제공하기 위해서는 데이터 센터에 가상화와 분산 컴퓨팅 기술을 기반으로 하여 IT 자원을 구성해야 한다. 본 논문은 폭넓은 분야에서 새로운 훈련 데이터가 언제라도 추가될 수 있고, 또한 언제라도 훈련 데이터에 새로운 속성이 추가될 수 있다는 문제에 특별히 초점을 맞춘다. 이러한 경우, 기존 속성 집합들을 가지는 훈련 데이터로 생성된 규칙은 쓸모없게 된다. 더구나 새롭게 추가된 데이터나 속성을 가지는 새로운 데이터는 기존 규칙과 결합될 수 없다. 본 논문은 이와 같은 경우를 자연스럽게 처리할 수 있는 보다 진보된 새 추론 엔진을 제안한다. 이 방법에서 기존의 데이터로 부터 생성된 규칙은 개선된 규칙을 생성하기 위한 새로운 데이터 집합과 결합될 수 있다.

• **주제어** : 분류, 규칙 개선, UChoo, 결정 트리, 속성, 훈련 데이터, 융합

Abstract To efficiently provide cloud computing services to users over the Internet, IT resources must be configured in the data center based on virtualization and distributed computing technology. This paper focuses specifically on the problem that new training data can be added at any time in a wide range of fields, and new attributes can be added to training data at any time. In such a case, rule generated by the training data with the former attribute set can not be used. Moreover, the rule can not be combined with the new data set(with the newly added attributes). This paper proposes further development of the new inference engine that can handle the above case naturally. Rule generated from former data set can be combined with the new data set to form the refined rule.

• **Key Words** : Classification, Rule Refinement, UChoo, Decision Tree, Attribute, Training Data, Convergence

1. 서론

MLP, RBF, SVM, C4.5 등과 같은 많은 분류 알고리즘들이 결정 트리를 사용해 좋은 결과들을 산출해 왔다[1,2]. C4.5[3]는 Quinlan이 제안한 ID3로부터 확장된 분류 알고리즘으로, 결정 트리를 사용하여 메모리의 이용을 최소화하면서도 좋은 결과를 빠르게 낼 수 있음을 보여 왔다. 본 논문은 C4.5 분류 알고리즘을 기반으로 데이터 확장 기법을 적용할 수 있도록 확장한 추론 엔진 알고리즘인 UChoo를 사용하여, 기존의 분류기에 새로운 데이터를 결합하는 추론 방법을 제안한다. 데이터 확장 기법은 이산 데이터에 대해서는 각 속성 값 마다 확률 값으로 채워 넣는다. 그러나 연속 값을 가지는 데이터에 대해

*Corresponding Author : 이종찬(jclee@chungwoon.ac.kr)
Received January 6, 2017 Revised February 9, 2017
Accepted February 20, 2017 Published February 28, 2017

서 기존에 적절한 처리 기법이 없었으나 엔트로피를 이용해 처리할 수 있다[4]. 특히 제안된 확장 기법에서는 각 사건마다 중요도를 나타내는 가중치를 할당할 수 있어 전문가가 각 데이터의 평가를 통해 중요도를 할당할 수 있는 장점을 가지고 있다. 이를 응용하면 소실 데이터를 학습하는 모델로 변환이 가능하다[5].

먼저 훈련 데이터를 확장된 표현 형식으로 변환한 후, UChoo알고리즘을 이용해 결정 트리를 구축한 다음, 이 구축된 결정 트리로부터 규칙을 생성한다. 이로써 새롭게 추가된 센서로 부터의 데이터를 기존 시스템과 결합하여 새로운 규칙을 자동으로 산출해 내는 추론 엔진 알고리즘을 구성한다. 다시 말해 기존의 데이터를 통해 학습이 이루어져 규칙 데이터를 가지고 있는 상태에서 추가된 데이터가 수집 되었다고 할 때, 원 데이터로 부터의 규칙들과 새로운 데이터가 결합하여 새롭게 학습할 수 있는 방법을 제안한다. 무엇보다도 학습 데이터가 분실되거나 변형된 경우라도 규칙만 가지고 새로운 데이터와 결합하여 새로운 규칙을 산출할 수 있다고 볼 수 있다. 여러 분야에서 데이터들은 연속적으로 추가가 될 것이다. 이때마다 기존의 데이터들을 버리고 새로운 데이터로 새로운 학습을 할 수는 없다. 이를 규칙 개선 문제라 하며, 실험 과정을 통해 제안 방법이 유용함을 보인다.

2. 배경

2.1 확장된 데이터 표현

확장된 데이터 표현은 다음과 같이 설명된다. Table 1은 두 개의 이산 속성과 1개의 연속 속성으로 구성된다. 이산 속성을 가지는 “Home Owner”는 “Yes”와 “No” 두 개의 카디너리티(cardinarity)를 가지며, 두 번째 “Marital Status“ 속성은 ”Single“, ”Divorced“, ”Married“의 3개의 카디너리티를 가지는 이산 속성이다. 반면 ”Annual Income“ 속성은 연속된 값을 가진다. 마지막으로 부류(Class)는 ”행복하다”를 나타낸다고 할 때 두 개의 카디너리티를 가진다. 이 Table 1을 확장된 데이터 표현법으로 바꾸면 Table 2와 같이 된다.

〈Table 1〉 A common data set

Home Owner	Marital Status	Annual Income	Class
Yes	Single	120	No
No	Divorced	100	Yes
No	Single	80	Yes
Yes	Married	120	No
No	Married	100	No
No	Single	80	Yes

확장된 표현에는 두 가지 중요한 특성이 있다. 첫째는 속성들이 0과 1사이의 확률 값으로 채워진다는 것이고, 둘째는 각 사건들 마다 중요도를 나타내는 가중치 값을 가진다는 것이다. 예를 들어 Table 2에서 사건 1은 일반적인 레코드의 중요도가 1인데 반해 가중치 20을 가진다. 이는 사건 1이 가중치 1인 레코드의 20개에 해당하는 중요도를 가진 사건이라는 뜻이 된다. 따라서 전체의 사건의 개수와 레코드들의 개수는 서로 다른 값일 수 있다. Table 2에서 전체 레코드 수(T)는 25개인데 반해, 전체 사건 수(p)는 6개이다.

2.2 Weak 학습자로서의 UChoo 알고리즘

UChoo는 C4.5을 확장된 데이터 표현 방법에 맞게 변형한 알고리즘으로 결정 트리를 반복적으로 생성하는 과정을 가지고 있다. 결정 트리는 (1)식과 같은 측정 값에 따라 분류된다.

$$Gain\_ratio(A) = Gain(A) / Split\_info(A) \quad (1)$$

(1)식에서 Gain_ratio(A)는 임의의 속성(A)가 얼마나 부류를 분류해 낼 수 있느냐의 정도이다. 따라서 Gain_ratio(A)가 가장 큰 값을 가지는 A를 선택하게 된다. Gain(A)는 속성 A와 부류사이의 상호 정보 정보를

〈Table 2〉 Extended data representation of Table 1.

Event#	Weight(i)	Home Owner		Marital Status			Annual Income			Class	
		Yes	No	Single	Married	Divorced	80	100	120	Yes	No
1	20	1	0	1	0	0	0	0	1	0	1
2	1	0	1	0	0	1	0	1	0	1	0
3	1	0	1	1	0	0	1	0	0	1	0
4	1	1	0	0	1	0	0	0	1	0	1
5	1	0	1	0	1	0	0	1	0	0	1
6	1	0	1	1	0	0	1	0	0	1	0

나타내는 값으로 엔트로피를 사용하여 (2)식과 같이 나타내 진다. 또한 Split_info(A)는 속성 A에서 값들의 개수로 인해 측정에 영향이 가지 않도록 즉, 트리가 균등하게 되도록 일반화 해준다.

(2)식에서 사용하는 변수들은 다음과 같다.

T(|T|) : 각 노드에서의 데이터 집합(의 개수)

T_{A_j}(|T_{A_j}|) : 집합 T 중에 속성 A에서 속성 값 j를 가지는 데이터의 부분집합(개수)

|T_{A_j}| : S_{A_j}의 데이터 개수

부류 : $C_1, C_2, \ \cdots\ , C_k$

$freq(C_i, T)$: T에서 부류가 C_i인 레코드의 개수

속성 A의 값들의 집합은 $\{O_{A_1}, O_{A_2}, \ \cdots\ , O_{A_n}\}$이다. 여기서 n은 속성 A의 카디너리티이다.

$$Gain(A) = info(T) - info_A(T) \quad (2)$$

$$info(T) = -\sum_{i=1}^{k}(freq(C_i,T)/|T|) \cdot \log_2(freq(C_i,T)/|T|)$$

$$info_A(T) = \sum_{j=1}^{n}(|T_{A_j}|/|T|) \cdot info(T_{A_j})$$

$$info(T_{A_j}) = -\sum_{i=1}^{k}(freq(C_i,T_{A_j})/|T_{A_j}|) \cdot \log_2(freq(C_i,T_{A_j})/|T_{A_j}|)$$

$$Split\_info(A) = -\sum_{j=1}^{n}(|T_{A_j}|/|T|) \cdot \log_2(|T_{A_j}|/|T|) \quad (3)$$

가중치를 가지는 UChoo의 특성에 따라 C4.5는 다음과 같이 변형되어야 한다.

부류의 소속 값 : C_1(m),C_2(m),C_3(m),…,C_{k-1}(m),C_k(m)

C_i(m)는 m번째 사건이 C_i부류에 속한 정도를 나타낸다. 단, i는 부류 값, $\sum_{i=1}^{k} C_i(m) = 1$이다.

속성의 소속 값 :

$O_{A_1}(m), O_{A_2}(m), \ldots, O_{A_{n-1}}(m), O_{A_n}(m)$

$O_{A_j}(m)$는 m번째 사건의 속성 A에서의 속성 값 j가 가지는 값을 말한다. $\sum_{j=1}^{n} O_{A_j}(m) = 1$ 이다.

Weight(m,S) : 집합 S에서 m번째 사건의 가중치 값

freq(C_i,T) : 집합 T 안에서 부류 C_i에 속해있는 사건들의 개수이다.

$$freq(C_i,T) = \sum_{m=1}^{|T|} Weight(m,T) \cdot C_i(m) \quad (4)$$

$$freq(C_i,T_{A_j}) \sum_{m=1}^{|T|} Weight(m,T) \cdot C_i(m) \cdot O_{A_j}(m)$$

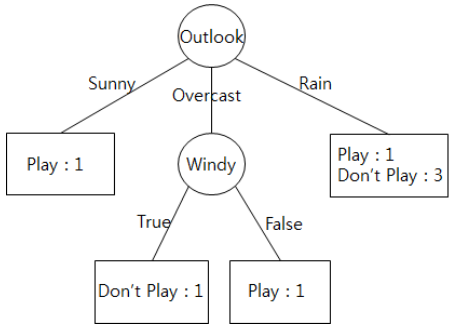
|T_{A_j}| : T_{A_j} 사건들의 개수.

$$|T_{A_j}| = \sum_{j=1}^{|T_{A_j}|} Weight(m,T_{A_j}) \cdot T_{A_j}(m)$$

3. 제안 알고리즘

3.1 데이터의 융합

이 절에서는 규칙의 생성과 규칙과 새로운 데이터와의 결합 방법에 대해 다룬다. 규칙 개선 문제는 기존의 규칙과 새롭게 수집된 데이터를 합하여 새로운 규칙을 만들어내는 것을 말한다. 이 문제는 원래의 규칙을 만들 때 사용한 데이터가 없거나 손실되었을 때 유용하게 쓰일 수 있다. 이 문제는 규칙으로부터 어떻게 정보를 추출하여 새롭게 수집된 데이터와 융합하는 것에 초점이 맞춰져 있다. UChoo에서 이 문제를 해결하기 위해 개발되었다.



[Fig. 1] An example of decision tree

Fig.1은 임의의 데이터를 UChoo를 이용해 학습하여 얻어진 결정트리라 가정한다. Fig.1의 결정 트리를 복합 형태의 규칙으로 표현하면 다음과 같다.

규칙 1 : [Play:1] = [Outlook=Sunny]

규칙 2 : [Don't Play:1] = [Outlook=Overcast] [Windy=True]

규칙 3 : [Play:1] = [Outlook=Overcast] [Windy=False]

규칙 4 : [Play:1, Don't Play:3] = [Outlook=Rain]

Fig. 1의 결정트리가 나타내는 규칙을 이용해 Table 3과 같은 새로운 데이터 집합이 만들어 진다. 예를 들어 Table 3의 사건 1은 규칙 1을 나타내는데, 규칙 1에서 “Outlook“은 값이 지정되어 있으나 ”Temp“와 ”Windy“는 don't care이므로 ”Temp“(”Windy“)의 카디너리티인 3(2)를 균등하게 배분하여 1/3(1/2)씩 배정하였다. 규칙 4에서 “Play” 부류가 1개의 사건 “Don't Play” 부류가 3개의 사건이므로 이에 맞는 확률로 부류를 각각 채워 넣었다. 또한 규칙 4는 4개의 사건이 있으므로 가중치를 4을 할당한다.

Table 4는 새로운 데이터가 새로운 센서 등으로부터 생성되었다고 가정 할 때 기존 방법으로는 기존 학습 데이터와 새로운 데이터를 합하여 새롭게 학습을 해야 했다. 그러나 이것은 커다란 자원의 낭비가 발생하며 기존 데이터가 분실되었을 때는 불가능 하다. 이에 Table 5에 나타난 것과 같이 새로운 방법으로, 기존 학습 결과 생성된 규칙과 새로운 데이터가 결합하여 새로운 데이터를 생성할 수 있다면 새로운 데이터가 생성될 때마다 방대한 학습을 하지 않아도 되기 때문에 상당한 장점을 가진다고 할 수 있다. 또한 이를 응용하면 손실 데이터에도 응용할 수 있을 것이다.

3.2 앙상블 기반 알고리즘

앙상블 기반 알고리즘[6,7,8,9,10]은 여러 개의 약한 분류기로부터 강한 분류기를 만들어 가는 과정이다. 부트스트랩은 훈련 데이터 집합에서 샘플링 된 훈련 데이터의 부분 집합을 가지고 학습하는 과정을 반복하는 기법이다. 다음 샘플링 과정에서의 데이터 분포는 잘못 분류된 데이터가 선택될 확률이 높아지도록 결정되어진다.

약한 분류기로는 절단을 위한 매개 변수를 가지는 UChoo 알고리즘이 사용된다. 약한 분류기의 학습에서 결정트리의 가설은 각 부류를 가지는 말단 노드의 사건마다의 가중치들의 합으로 결정되어진다.

$$h_i = \frac{\sum_{j:y_j = C \& x_j \in S_{leaf}} Weight(j, S_{leaf})}{\sum_{j:x_j \in S_{leaf}} Weight(j, S_{leaf})}$$

데이터 집합 S_{leaf}는 말단 노드에서 훈련 집합이다. 훈련 데이터 $S = \{(Weight_i, x_i, y_i)\}_{i=1}^{n}$ 을 정의한다. $Weight_i \geq 0$는 인스턴스의 가중치이고, $x_i \in X$는 확장된 속성 공간 X에서 i번째 인스턴스이고, $y_i \in Y$는 확장된 부류 공간 Y에서 x_i의 부류 레벨이다. T는 UChoo 학습기의 수이다. UChoo의 출력은 가설 $h(x) = (h_1(x), ..., h_k(x))$이다. 여기

〈Table 3〉 An output data from the rule of Fig. 1

Event #	Weight	Outlook			Temp			Windy		Class	
		Sunny	Overcast	Rain	10	20	30	True	False	Play	Don’t Play
1	1	1	0	0	1/3	1/3	1/3	1/2	1/2	1	0
2	1	0	1	0	1/3	1/3	1/3	1	0	0	1
3	1	0	1	0	1/3	1/3	1/3	0	1	1	0
4	4	0	0	1	1/3	1/3	1/3	1/2	1/2	1/4	3/4

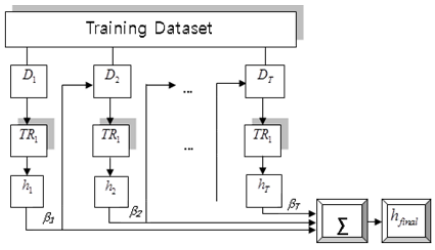
〈Table 4〉 New data set

Event #	Weight	Outlook			Temp			Windy		Class	
		Sunny	Overcast	Rain	10	20	30	True	False	Play	Don’t Play
1	1	0	1	0	1	0	0	1	0	1	0
2	1	1	0	0	0	1	0	0	1	0	1
3	1	0	0	1	1	0	0	1	0	1	0

〈Table 5〉 Connect with old rule and new data

Event #	Weight	Outlook			Temp			Windy		Class	
		Sunny	Overcast	Rain	10	20	30	True	False	Play	Don’t Play
1	1	1	0	0	1/3	1/3	1/3	1/2	1/2	1	0
2	1	0	1	0	1/3	1/3	1/3	1	0	0	1
3	1	0	1	0	1/3	1/3	1/3	0	1	1	0
4	4	0	0	1	1/3	1/3	1/3	1/2	1/2	1/4	3/4
5	1	0	1	0	1	0	0	1	0	1	0
6	1	1	0	0	0	1	0	0	1	0	1
7	1	0	0	1	1	0	0	1	0	1	0

서 $h_i(x)$는 사건 x가 부류 C_i에 얼마나 속해있는지를 말하며 $\sum_{i=1}^{k} h_i(x) = 1$이다. 이 과정들을 그림으로 표시하면 Fig. 2와 같이 나타내진다. 또한 학습을 위한 알고리즘은 다음과 같다.



[Fig. 2] AdaBoost Diagram

초기화 $w_i^1 = D_i^1 = \frac{1}{n}$

for $t = \overline{1, T}$

1. $D_t(i) = \frac{w_t(i)}{\sum_{j=1}^{n} w_t(j)}$
2. D_t에 따르는 훈련 데이터 부분 집합 TR_t을 선택
3. TR_t을 가지고 UChoo 학습기를 훈련하고, 가설 $h^t = \{h^t(x_i) | x_i \in S\}$를 얻는다.
4. 복합된 가설 $H_i^t(x) = \frac{\sum_{j=1}^{t} h_i^j(x)}{\sum_{l=1}^{k}\sum_{j=1}^{t} h_l^j(x)}$
5. H^t의 오류를 계산 :

$$e^t = \sum_{i : \arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)} D_i^t$$

6. 가중치 변경 :

$$w_i^{t+1} = \begin{cases} w_i^t, & \text{if}(\arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)) \\ \frac{e^t}{1-e^t} \cdot w_i^t, & Otherwise \end{cases}$$

End for

$$H_i^{final}(x) = \frac{\sum_{j=1}^{T} H_i^j(x)}{\sum_{l=1}^{k}\sum_{j=1}^{t} H_l^j(x)}$$

$$H(x) = \arg\max_{C_i} H_i^{final}(x)$$

처음에는 모든 사건들이 똑같은 확률 분포를 가지고 훈련 데이터 부분 집합 TR_t가 선택되는 일양분포를 가지도록 한다. UChoo 학습기가 반복적으로 학습되면서 가설 $h^t = \{h^t(x_i) | x_i \in S\}$를 얻도록 훈련한다. 부분 집합 TR_t의 확장된 데이터 표현에서 가중치 {Weight(S)}는 가설 h^t를 정의하기 위해 사용되고, 반복과정 중에 변함이 없다. AdaBoost 규칙[7,8]은 오분류된 사건의 가중치가 증가 되도록 가중치 $\{w_i\}_{i : x_i \in S}$를 변경하기 위해 적용된다.

4. 실험

실험에 사용된 데이터는 UCI 기계 저장소[11]의 데이터 집합들을 사용하였다. 실험은 크게 두 가지 방법으로 실행되었는데 첫 번째는 규칙 개선 문제에 관한 것이고, 두 번째는 새로운 속성이 포함된 데이터에 관한 것이다. 데이터를 10개의 블록으로 나눈 후 9개는 훈련을 위해 1개는 실험을 위해 사용한다. 이때 9개의 훈련 데이터를 하나로 합친 다음, 이를 다시 Training 1과 Training 2와 같이 무작위로 2개의 블록으로 나눈다. Training 1은 2.1절에서 설명한 데이터 확장 방법을 이용해 데이터를 변경하고 학습을 통해 결정트리를 구축한다. 그리고 이 결정 트리를 3.1절의 규칙의 생성 방법에 따라 Rule 1의 규칙을 데이터 형태로 변형한다. 이때 Training 2를 새로 추가된 데이터로 하여 이 변형 데이터와 합친 후(Rule1 + Training2), 새로운 학습이 이루어져 새로운 규칙(Rule 2)의 결정 트리가 산출된다. Table 6에 첫 번째 실험에서 두 알고리즘의 실험 결과가 유사하다는 것은 규칙을 데이터로 변형하는 것이 가능하다는 것을 알 수 있다. 무엇보다도 학습 데이터가 분실되거나 변형된 경우라도 규칙만 가지고 새로운 데이터와 결합하여 새로운 규칙을 산출할 수 있다고 볼 수 있다. 여러 분야에서 데이터들은 연속적으로 추가가 될 것이다. 이때마다 기존의 데이터들을 버리고 새로운 데이터로 새로운 학습을 할 수는 없다. 기존의 정보들을 이용해야 한다.

다음으로 속성이 추가된 데이터에 관한 실험은 4가지 방법으로 이루어졌고 결과가 Table 7에 나타나있다. 첫째로 C4.5는 본래의 데이터(Training1+Training2)를 사용하여 학습하였고, 둘째로 Training2+Rule1은 Training1 데이터를 학습하여 Rule1의 규칙을 얻은 후 이를 Training2와 결합하여 실험하였다. 셋째 실험 Training3

〈Table 6〉 Experimental results of rule refinement problems

Data	Data Number	Attribute Num		Class Num	Error (%)	
		Disc.	Cont.		C4.5	UChoo
splice	3,190	62		3	5.68	6.33
vehicle	846		18	4	31.99	31.43
waveform	300		21	3	28.81	30.21

〈Table 7〉 Experimental results on data with new attributes added

Data	Experimental Methods			
	C4.5	Training2+Rule1	Training3	Training4+Rule3
Letter	26.12	28.30	30.89	27.75
Iris	10.01	8.00	22.67	8.67

는 변수가 추가되기 전과 후를 비교하기 위해, 전의 규칙과 변수가 추가된 새로운 데이터를 결합하는 실험이다.

C4.5와 Training2+Rule1의 결과는 Letter 데이터에서 C4.5가 다소 성능이 좋게 나타났고, Iris 데이터에서는 Training2+Rule1이 다소 낮은 것으로 나타났다. 여기서 C4.5는 본래의 데이터를 사용하였지만 Training2 +Rule1는 절반의 데이터를 이용해 규칙을 산출하고 이 규칙과 나머지 절반의 데이터를 새로 학습하여 얻어진 결과를 주목한다면 상당히 좋은 결과라고 생각할 수 있다. 실험에서 보이듯이 Training2 +Rule1의 결과가 좋을 수도 있는 것은 복잡한 데이터일수록 데이터에 잡음이 포함될 수 있어 본래의 데이터를 간략화 한 규칙의 성능이 더 좋을 수 있다 해석된다. 따라서 여러 가지 융합 데이터[12,13,14,15,16]에 응용될 수 있을 것으로 본다. 또 다른 실험으로 Table 8과 같이 UChoo와 Adaboost의 실험 결과를 비교하였다. 결과에서 보듯이 Adaboost의 결과가 다소 우수한 것으로 나타났다.

〈Table 8〉 Comparison of Uchoo and Adaboost Results

Data	UChoo	Adaboost
Glass	31.90	25.95
Vehicle	22.72	22.49
Balance-Scale	17.72	16.98

5. 결론

본 논문은 데이터를 확장해 표현하는 기법을 기본으로 하고 있다. 여러 분야에서 데이터들은 연속적으로 추가가 될 것이다. 이때마다 기존의 데이터들을 버리고 새로운 데이터로 새로운 학습을 할 수는 없다. 기존의 정보들을 이용해야 한다. 확장된 데이터 표현 방법은 이러한 면을 해결하기 위한 하나의 방법을 제시하고 있다. 이를 통해 주어진 데이터를 학습하여 결정 트리를 산출해 내는 것은 물론이고, 과거에 만들어진 규칙과 새로운 데이터가 결합할 수 있음을 보였다. 본래 데이터에 손실이나 손상이 있었을 경우 성능에 심각한 영향이 없이 규칙만으로 복원이 가능할 수 있음을 보았다. 이 과정에서 규칙과 데이터가 같은 형식을 취하는 데이터 확장 기법을 소개하였다.

확장된 데이터 표현을 이용해 속성 값들에 손실이 있는 데이터를 대상으로 규칙 개선을 하는 것이 가능하다. 따라서 이 연구로부터 데이터에 약간의 손상이 있다하더라도 성능에 많은 영향을 주지 않을 수 있는 분류 알고리즘이 완성될 것으로 본다.

REFERENCES

[1] P. N. Tan, M. SteinBach, V. Kumar,"Introduction to data mining", 2005

[2] M. Kantardzic,"Data Mining : Concepts, Models, Methods, and Algorithms", Wiley-IEEE Press, 2002.

[3] J.R.Quinlan, "C4.5 : Program for Machine Learning," San Mateo, Calif, Morgan Kaufmann, 1993.

[4] P. E. Utgoff, "Incremental Induction of Decision Trees", Machine Learning, Vol. 4, No. 2, pp. 161-186, 1989.

[5] J.C.Lee, D.H.Seo, C.H.Song, W.D.Lee, "FLDF based Decision Tree using Extended Data Expression", The 6th Conference on Machine Learning & Cybernetics, Hong Kong, pp. 3478- 3483, Aug. 2007.

[6] T. S. Lim, W. Y. Loh, Y. S. Shih,"A Comparison of

Prediction Accuracy, Complexity, and Training Time of Thirty-Tree Old and New Classification Algorithms", Machine Learning, Vol. 40, No. 3, pp. 203-228, 2000.

[7] R. Kohavi, J. R. Quinlan, "Data Mining Task and Methods: Classification: Decision-tree Discovery", Handbook of data mining and knowledge discovery press, pp. 267-276, 2002.

[8] H. Schwenk, Y. Bengio,"Boosting neural networks" ,Neural Computation, Vol. 12, pp1. 869-1887, 2000.

[9] R. Polikar, "Bootstrap-Inspired Techniques in Computational Intelligence", IEEE Signal Processing Magazine, pp. 59-72, 2007.

[10] J. R. Quinlan,"Bagging, Boosting, and C4.5", AAAI/ IAAI, Vol. 1, 1996.

[11] E. Keogh, C. Blake, C. J. Merz,"UCI Repository of Machine Learning Databases",http://www.ics.uci.edu / ~mlearn/MLRepository.html, 1989.

[12] K. Ryu, "Convergence Research for Implementing NC Postprocessor Based Cloud Computing", Journal of the Korea Convergence Society, Vol. 7, No. 1, pp. 17-23, 2016.

[13] D. Kim, N Kim, "Design of Mixed Reality based Convergence Edutainment System using Cloud Service", Journal of the Korea Convergence Society, Vol. 6, No. 3, pp. 103-109, 2016

[14] H. Lee, K. Park, D. Kim,"A Study on Possible Construction of Big Data Analysis System Applied to the Offline Market", Journal of Digital Convergence, Vol. 14, No. 9, pp. 317-323, 2016.

[15] G. Kim, S. Jeong, H. Mun, C. Kim, "Design of Curve Road Detection System by Convergence of Sensor", Journal of Digital Convergence, Vol. 14, No. 8, pp253-259, 2016.

[16] Y. Jung, J. Jeon, "A Fusion of the Period Characterized and Hierarchical Bayesian Techniques for Efficient Cluster Analysis of Time Series Data", Journal of Digital Convergence, Vol. 13, No. 7, pp. 169-175, 2015.

저자소개

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 인터넷학과 교수

<관심분야> : 신경회로망, 패턴분류, 정보보호, 데이터 압축