Journal of the Korea Convergence Society
Vol. 12. No. 2, pp. 7-13, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.2.007

불완전한 데이터를 처리하기 위한 데이터 확장기법

이종찬
청운대학교 컴퓨터공학과 교수

A data extension technique to handle incomplete data

Jong Chan Lee
Professor, Dept. of Computer Engineering, Chungwoon University

요 약 본 논문은 학습 데이터에 손실값을 포함하고 있는 불완전한 데이터를 위하여 확률을 나타낼 수 있는 형식으로 변환한 후 손실값을 보상하는 알고리즘을 소개한다. 기존에 이러한 데이터 변환을 사용한 방법에서는 손실 변수가 가질 수 있는 균등한 확률로 손실값을 할당하여 불완전한 데이터를 처리하는 것이었다. 이 방법으로 많은 문제에 적용하여 좋은 결과를 얻었으나, 손실 변수에 남아있는 모든 정보를 무시하고 새로운 값을 할당한다는 점에서 정보의 손실이 있다는 지적이 있었다. 이에 반해 새로운 제안 방법은 손실값을 포함하지 않는 완전한 정보만을 잘 알려진 분류 알고리즘(C4.5)에 입력하고 학습하는 중에 결정트리가 구축된다. 그리고 이 결정트리로 부터 손실값에 대한 확률을 구하여 이를 손실 변수의 추정값으로 할당한다. 즉, 불완전한 학습 데이터에서 손실되지 않은 많은 정보들을 사용하여 손실된 일부 정보를 복구하는 것이다.

주제어 : 손실값, 보상 확률, 데이터 확장 표현식, 불완전한 데이터, 결정트리, 분류기

Abstract This paper introduces an algorithm that compensates for missing values after converting them into a format that can represent the probability for incomplete data including missing values in training data. In the previous method using this data conversion, incomplete data was processed by allocating missing values with an equal probability that missing variables can have. This method applied to many problems and obtained good results, but it was pointed out that there is a loss of information in that all information remaining in the missing variable is ignored and a new value is assigned. On the other hand, in the new proposed method, only complete information not including missing values is input into the well-known classification algorithm (C4.5), and the decision tree is constructed during learning. Then, the probability of the missing value is obtained from this decision tree and assigned as an estimated value of the missing variable. That is, some lost information is recovered using a lot of information that has not been lost from incomplete learning data.

Key Words : Missing value, Conservation probability, Data extension expression, Incomplete data, Decision tree, Classifier.

*This paper is sponsored by the academic research project of Chungwoon University in 2020. (No. 2020-53).
*Corresponding Author : Jong Chan Lee(jclee@chungwoon.ac.kr)
Received December 1, 2020 Revised January 4, 2021
Accepted February 20, 2021 Published February 28, 2021

1. 서론

인터넷의 발전과 함께 데이터의 양이 기하급수적으로 증가하면서 이들 중 유용한 정보만을 추출하는 문제는 중요한 이슈가 되었으며 데이터 마이닝이라는 학문으로 발전하였다[1,2]. 여기서 다루는 데이터들은 모든 요건을 갖춘 완전한 정보라기보다는 일부의 손실이 있는 불완전한 정보인 경우가 많다. 특히 서로 다른 운영체제를 가지는 여러 기기로부터 또는 원거리로부터 수집진 데이터를 취급하는 유비쿼터스 환경에서는 불완전한 데이터를 피할 수 없다.

이 문제의 해결을 위해 학습 데이터의 형식을 불완전한 데이터에 적합하게 변환하고 이를 학습하도록 하는 방법이 제안되었으며 여러 응용문제에 적용하여 좋은 결과를 얻어 왔다[3-6]. 이 데이터 형식 변환 방법을 데이터 확장기법이라 하며 2가지 특징을 가지고 있다 : 각각의 사건마다 중요도를 달리할 수 있고, 각 변수의 카디너리티 마다 확률값을 부여할 수 있다. 첫 번째 특징인 각 사건에 가중치를 부여하는 점은 Adaboost 알고리즘에서 다음 Weak Learner를 구성해 갈 때 학습을 위한 사건을 선택하는 기준으로 이용할 수 있었다. 그리고 두번째 손실값에 확률을 부여할 수 있는 특징은 손실 데이터에 보상값을 부여하는데 이용되었다[7].

이러한 데이터 확장기법에서 손실값에 확률값을 할당하는 방법은 이용 목적에 따라 여러 가지가 제안되었다[5]. 이들 중 가장 간단한 것으로 손실값에 균등한 확률값을 할당하여 엔트로피 함수를 사용하는 분류 알고리즘에서 손실값이 많이 포함된 변수일수록 상위 노드에서 선택되지 않도록 하는 알고리즘이 이용되었다[3]. 그러나 이 방법은 본래 가지고 있던 정보를 무시하여 잃어버린다는 점을 가지고 있었다. 이에 대해 손실값을 제외한 정보로 엔트로피 확률을 구하고 이를 손실값에 채우는 방법도 제시되었다[6,8,9].

본 논문은 손실값을 보상하는 확률값을 분류 알고리즘을 통해 분류된 영역에서 찾을 수 있지 않을까 하는 기본적인 아이디어에서 출발하였다. 이를 구현의 면에서 보면, 학습 데이터를 손실된 사건과 손실되지 않은 사건으로 2등분 한 다음, 손실되지 않은 데이터를 가지고 C4.5 분류 알고리즘[10]을 이용해 분류하며 결정트리(decision tree)를 구성하게 되고 이에 따라 각각의 영역은 구분되어 간다. 그리고 손실 데이터들을 차례대로 이 분류 영역에 입력해 손실되지 않은 정보들을 가지고 가장 근접한 영역을 찾아가도록 한다. 마지막으로 이 영역의 값을 확률로 표현한 후, 손실값이 가지는 확률로 대치하여 보상이 이루어지도록 한다.

전체 손실값들을 보상한 후 정보의 손실이 얼마나 보전되었나 하는 성능 평가를 위해 복구된 학습 데이터를 SVM 알고리즘으로 학습하여 그 성능을 비교한다. 학습 알고리즘으로 SVM을 선택한 이유는 딥러닝과 같은 반복적으로 개선해 나가는 알고리즘은 기본적으로 손실된 정보를 보상하는 기능이 있어 손상된 정보의 성능을 비교하는 본 논문의 실험 알고리즘으로 적합하지 않기 때문이다. 각각의 변수들에 대해 손실의 정도를 달리한 실험 결과들로부터 정보의 손실을 최소화하여 의미 있는 보상 방법으로 이용될 수 있는지를 확인한다.

2. 관련 연구

2.1 데이터 확장기법

데이터 확장 표현식은 이산(discrete)값을 가지는 불완전한 데이터에서 각 변수의 손실값들을 확률로 나타내기 위하여 개발되었다[3]. 그리고 이전의 연구[3]에서 이산값 뿐만아니라 연속(continuous)값을 가지는 불완전한 데이터의 처리방법도 소개되었다. 우선 이 표현법의 방법적인 면을 살펴보면, 각 사건에서 손실되지 않은 변수들은 원-핫-인코딩으로 변환하고 즉, 각 변수에서 하나의 값만 1이 되도록 하고, 손실 변수들은 각 변수의 값마다 확률값을 구하여 저장한다. 이때 각각의 사건들 마다 가중치(weight-W) 값을 부여한다. 이 가중치는 학습 데이터에서 각각의 사건들에 대한 중요도를 달리하고자 할 때 이를 반영할 수 있게 한다. 이는 앙상블 알고리즘의 AdaBoost에서 가중치에 따라 다음 학습 데이터 집합을 선택하는 알고리즘에서도 유용하게 쓰일 수 있다.

Table 1. An example of learning data set

Event	V1	V2	V3	Class
1	3	2	1	1
2	1	2	3	2
3	3	1	2	2
4	2	1	2	1
5	?	2	1	1
6	1	?	3	2

데이터 확장기법으로 변환하는 예를 들면, Table 1과 같은 불완전한 학습 데이터는 6개의 사건과 3개의 변수

(V1, V2, V3)와 부류(Class)를 가진다. 여기서 사건 5의 V1과 사건 6의 V2가 손실되었다. 이러한 Table 1을 데이터 확장기법으로 변환하면 Table 2와 같다. 완전한 사건들은 각 변수가 원-핫-인코딩으로 변환되었고 불완전한 사건에서 손실 변수는 별도의 방법에 따라 손실값에 가장 근접한 확률값을 구하여 채워진다. 또한 각 사건들은 가중치가 부여되는데 예를 들어 사건 4에서는 10은 다른 사건에 비해 10배 만큼의 중요도를 가진다는 것을 의미한다.

Table 2. Data extention expression of Table 1.

E	W	V1			V2		V3			Class	
		1	2	3	1	2	1	2	3	1	2
1	1	0	0	1	0	1	1	0	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	0	1
4	10	0	1	0	1	0	0	1	0	1	0
5	1	?	?	?	0	1	1	0	0	1	0
6	1	1	0	0	?	?	0	0	1	0	1

2.2 불완전한 데이터에서 손실값을 위한 확률값

정보이론[11-14]에서 Claude Shannon은 (1) 식과 같은 정보의 양을 나타내는 엔트로피 수식을 제안하였다.

$$H(X) = -\sum_{i=0}^{N} p_i \log p_i \quad (1)$$

여기서 p_i는 특정한 사건 i가 일어날 확률이다. 예를 들어 사건이 발생하는 경우의 수가 2이며 발생 확률이 동일하다고 가정하면 확률은 p, (1-p)이므로 1/2씩을 가진다. 이때의 엔트로피 값은 다음과 같이 최대값을 가진다.

$H(X)$ = - 1/2 $\log_2$(1/2) - 1/2 $\log_2$(1/2) = 1

또한 어느 사건이 발생할 확률이 1이라면, 다른 사건이 발생할 확률은 0이 되기 때문에 엔트로피 $H(X)$는 0이 된다. 만약 두 사건이 일어날 확률이 1/4, 3/4이라면 엔트로피 $H(X)$는 0.811이 된다. 이로부터 Shannon이 정의한 정보량으로써의 엔트로피란 결국 불확실성의 정도를 의미함을 알 수 있다. 다시 말해 불확실성이 커질수록 엔트로피는 증가하며, 반대로 불확실성이 감소할 경우 0에 가까워진다. 즉, 각각의 사건들이 일어나는 경우의 확률이 동일한 경우 가장 큰 엔트로피 값을 가지게 된다.

Table 2의 사건 5에서 V1의 손실값을 V1의 카디너리티가 3이므로 확률값을 1/3씩(1/3, 1/3, 1/3) 할당하고, 사건 6에서 V2 확률값을(1/2, 1/2)씩 동일한 값을 할당하면 엔트로피는 가장 높은 값을 가지게 된다. 이를 정리해 보면 Table 2에서 5, 6번째 사건에서 회색 컬럼의 손실된 값은 데이터 확장 기법을 실행한 후 다음 과정을 통해 확률값으로 변환되어 채워진다..

첫째, 소실된 변수의 카디너리티를 이용해 균등한 확률 값을 구하고 이를 손실 변수에 채워 넣는다. 이렇게 손실 변수에 균등한 값을 할당하게 되면, 손실 값이 많이 포함된 변수일수록 엔트로피가 증가하는 효과를 얻는다. C4.5와 UChoo는 각각의 변수마다 엔트로피를 구한 다음, 이 정보를 이용해 결정트리를 구성해 간다. 따라서 이들 알고리즘에서는 손실 변수들은 상위 노드에 배치되지 않게 된다. 둘째, 손실된 변수에서 손상되지 않은 나머지 정보를 가지고 확률을 구해 소실된 값을 채워가는 방식이다. 여기서 확률은 엔트로피를 기반으로 산출된다[6].

2.3 확장된 데이터 표현을 가지는 C4.5 알고리즘

UChoo는 확장된 데이터 형식으로 변환된 학습 데이터를 C4.5로 처리할 수 있도록 개발된 알고리즘[3]이다. C4.5 알고리즘은 (2) 식과 같은 Gain_ratio를 각각의 변수에 대해 계산하여 가장 작은 변수를 선택하고 이 변수의 속성값에 따라 하부 노드로 분기한다. 이러한 과정은 일정 영역에 하나의 부류 데이터만 있을 때까지 반복되며 이 과정에서 결정트리가 생성된다.

$$Gain\_ratio(A) = Gain(A) / Split\_info(A) \quad (2)$$

(2) 식을 설명하기 위해 다음과 같이 정의한다.

- S : 분류된 각 노드에서의 데이터 집합
- $|S|$: S의 데이터 개수
- S_{A_j} : S 집합 중에서 변수 A가 변수값 j를 가지는 S의 부분 집합
- $|S_{A_j}|$: S_{A_j}의 데이터 개수
- 부류 : $C_1, C_2, \ldots, C_k$
- $freq(C_i, S)$: S에서 부류가 C_i인 사건의 빈도수

(2) 식에서 Gain(A)는(3) 식과 같이 나타낼 수 있다.

$$Gain(A) = info(S) - info_A(S) \quad (3)$$

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

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

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

(2) 식에서 Split_info(A) 함수는 결정 트리가 균등하게 배분되도록 하는 역할을 담당하며 (4) 식에 나타나 있다.

$$Split\_info(A) = -\sum_{j=1}^{n}(|S_{A_j}|/|S|)\cdot\log_2(|S_{A_j}|/|S|) \quad (4)$$

Fig. 1은 간단한 분류와 결정트리의 생성 과정의 예를 보인다. (a)로 부터 학습 데이터가 2차원(V1, V2)으로 구성되며 3개의 부류를 가진다는 것을 알 수 있다. 여기서 (2) 식에 따라 V1과 V2의 엔트로피 값을 구했을 때 V2가 작았다면 V2의 속성값에 따라 평면(P1)이 구해지고 P1에 의해 부류 3영역과 부류 1, 2 영역으로 구분된다. 그리고 (b)의 결정트리에 이 정보를 저장한다. 이때 왼쪽의 부류 3 영역은 완전히 분류되었으므로 학습을 마치고, 부류 1, 2 영역은 다시 엔트로피를 계산하고 평면 P2를 구하여 결정트리에 저장하며 학습을 완성한다.

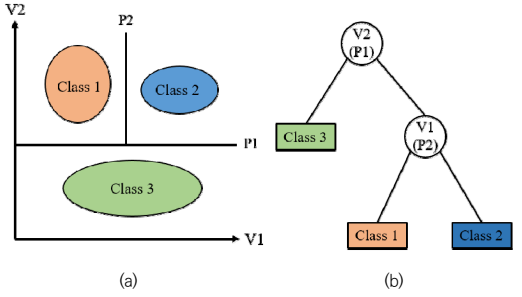


Fig. 1. Decision tree creation process as classification progresses

3. 분류 알고리즘을 이용해 손실값을 대치하는 확률을 구하는 알고리즘

UChoo의 손실값 처리기법은 원 데이터가 가지고 있는 정보를 무시하고 일정한 확률값을 손실값에 할당하는 방법으로 엔트로피를 척도로 하는 알고리즘에는 적합하였다. 그러나 딥러닝과 같이 엔트로피를 기반으로 하지 않는 학습에서는 학습 데이터 자체가 가지고 있는 정보로 확률을 구함으로써 손실값을 보상하는 방법이 보다 효율적일 것이다. 이에 따라 손실값을 처리하는 간단한 알고리즘은 다음과 같다.

1. 학습 데이터 집합을 손실 데이터와 비손실 데이터로 분리한다.
2. 비손실 데이터를 C4.5에 입력하여 학습 과정을 통해 분류하고, 이때 각 부류당 영역을 의미하는 결정트리가 구성되어 진다.
3. 손실 데이터의 각 사건들을 차례로 2번의 결정트리에 입력하여 가장 근접한 영역을 찾고 이 영역의 정보를 확률로 나타내어 손실값을 대체한다.

이를 확장된 데이터 기법을 반영한 단계별 과정으로 보면 그림 2와 같다. 우선 손실이 없는 완전한 사건들은 C4.5 분류기로 보내 점진적으로 결정트리를 완성하고, 데이터 확장 기법을 위해 원-핫-인코딩으로 변환한다. 그리고 손실 사건들에서 손실되지 않은 변수를 결정트리에 입력해 순회하며 확률값을 구하고 손실 변수에 채워 데이터 확장 기법의 변환을 완성한다.

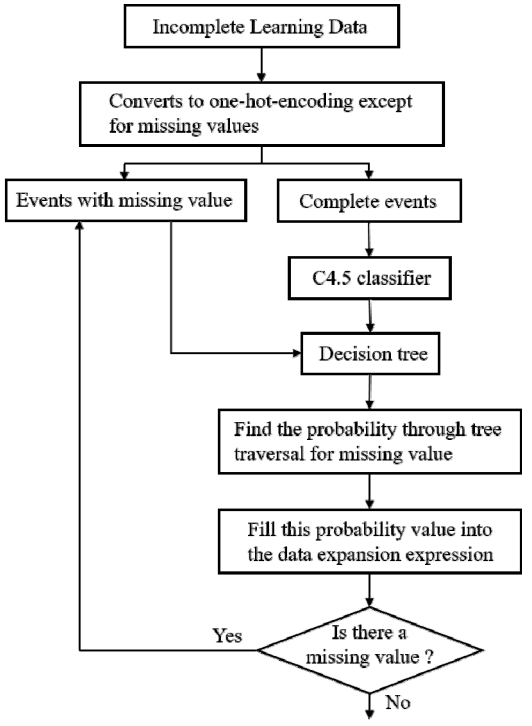


Fig. 2. The process of finding the probability of imputing missing values

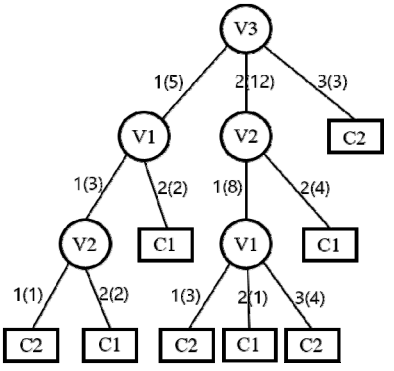


Fig. 3. Example of decision tree obtained through C4.5

Fig. 3은 제안 알고리즘을 설명하기 위하여 임의의 학습 데이터로 학습한 결정트리라고 가정한다. 여기서 동그라미는 선택된 변수를 의미하는 노드이며, 노드와 노드 사이의 수(ex: 2(5))는 변수값(분류된 영역에서 사건의 수)을 의미한다. 또한 네모 노드는 말단(leaf) 노드이며 각 부류의 영역을 의미한다. 이 결정트리에 손실값을 포함하는 사건들을 입력한다. 손실값을 포함하는 사건의 예로 {-, 1, 2} 사건의 경우 VL 표현식으로 나타내면 [V1=?] [V2=1][V3=2]가 된다. 이 사건을 그림 3의 결정트리에 입력하여 트리를 순회하면 [V3=2]이고 [V2=1]인데 V1 값이 손실되었다. 그러므로 말단 노드의 각 부류 영역으로 진전하지 못하기 때문에 확률로 손실값을 나타내야 한다. 다시 말해 V1 노드와 말단 노드 사이의 정보를 이용해 P(V1==1)=3/8, 즉 V1이 1일 확률은 3/8이다. 그리고 P(V1==2)=1/8, P(V1==3)=4/8로 나타내진다. 즉, P(V1) = {3/8, 1/8, 1/3}의 확률값으로 표현된다. 다른 사건의 예들을 보면 {1, -, 1}의 경우 P(V2) = {1/3, 2/3}이고, {1, -, 2} 사건은 P(V2) = {8/12, 4/12}가 된다. 또한 {-, 2, 3} 사건의 경우 결정트리에서 [V3=3]로 이동했을 때 C2 부류를 가지는 3개의 사건이 있음을 알 수 있다. 이 사건들 중 변수 [V1=1]인 사건이 1개이고, [V1=2]인 사건이 2개라고 한다면 P(V1) = {1/3, 2/3, 0}이 된다.

제안 방법의 장점은 손실 데이터들에 남아있는 정보들을 이용하여 손상된 부분에 가장 근접한 확률값을 찾아간다는 것이다. 대부분의 불완전한 데이터들이 일부 정보만이 손상된 것들이기 때문에 손상된 사건들에 남아있는 많은 정보를 이용하여 유용한 정보로 복원하는 작업은 의미가 있을 것이다.

4. 실험 결과

실험을 위해 UCI 기계 저장소[15]의 데이터 중에 “Balance & Scale(BS)”(625 사건, 3 변수, 3 부류)와 “Car Evolution(CE)”(1728 사건, 6 변수, 4 부류)를 사용하였다. 그리고 인간의 뇌파를 기록한 데이터인 “Sleep Stage Scoring(SSS)”(799 사건, 11 변수, 6 부류)데이터를 사용하였다. 성능 측정을 위해 전체의 10%는 테스터 데이터로 사용하는 10겹 교차 검증(10-fold cross validation)을 사용하였다. 그리고 학습 데이터는 손실비율을 각각 달리하여 5%, 15%, 30%, 45% 비율만큼 데이터를 선택한 다음, 이 선택된 사건의 각 변수 마다 교대로 변수값을 손실시키며 결과를 알아보았다. 손실율이 30%라는 것은 테스트 데이터를 제외한 학습 데이터 중에 30%는 손실 데이터로 나머지 70%를 비손실 데이터 랜덤하게 만든 후 실험이 이루어진다는 것이다. 실험 결과는 이러한 방법대로 10번을 실행하여 각 결과의 평균을 산출하였다. 그 결과가 변수와 손실률 별로 Table 3에 나타나 있다.

Table 3 결과의 의미를 알아보기 위해 각 손실률의 평균값을 가지고 그림 4에서 그래프로 나타내었다. 여기서 특이점은 (a)의 SSS와 (b)의 CE 결과에서는 손실 변수의 카디너리티에 일정한 확률을 할당한 실험 결과인 “Uniform”이 손실률이 증가함에 따라 급격한 성능변화가 있는 것에 반해, 제안된 알고리즘인 “Probability”의 결과에서는 전체 성능에 커다란 변화가 나타나지 않는 완만한 하락을 보이고 있다. 이러한 점은 (c)의 BS 데이터의 결과에서 손상률에 따른 결과 저하도가 비교적 명확하다는 것과 대비된다. 이러한 결과는 첫째, 제안 방법은 손실 정보를 확률값으로 보간하는 방법이므로 손실률이 클수록 그 효과가 나타나는 것이라고 본다. 둘째, (a) 데이터가 11개, (b) 데이터가 6개의 변수를 가지고 있어 어느 하나의 변수에 손실이 생겼다고 해서 전체 성능에 커다란 영향을 미치지 않는 것에 반해, (c) 데이터가 3개의 변수로 비교적 적기 때문에 이들 중 하나가 손상되면 영향이 큰 것이라고 해석된다. (c)의 경우에는, 우선 결정트리를 이용해 손실값이 속한 구역을 정하고, 이 구역의 중심점으로 손실값을 대체하면 성능이 개선될 수도 있을 것으로 본다.

Table 3. Results of applying two probability values to training data containing missing values

(White: Uniform, Gray: Proposed method)

	5%	15%	30%	45%
V1	86.017	85.110	83.331	79.635
	87.352	86.829	87.388	87.015
V2	87.343	86.430	84.161	84.006
	87.586	86.819	86.771	86.379
V3	86.817	87.331	84.452	80.583
	87.681	88.427	85.593	85.859
V4	87.139	86.201	84.450	80.098
	88.134	87.645	88.811	88.546
V5	85.843	85.147	82.829	81.001
	88.872	88.180	87.296	87.113
V6	87.417	86.393	83.670	79.332
	87.162	87.477	87.237	85.653
V7	86.634	85.939	83.597	83.063
	88.066	88.758	87.237	85.549
V8	87.217	86.411	84.154	82.992

	87.794	87.792	85.966	87.309
V9	85.687	84.805	82.697	78.779
	87.279	87.795	87.611	87.376
V10	87.016	83.778	81.185	74.897
	88.951	86.999	85.477	86.999
V11	87.101	84.560	82.567	79.352
	89.979	89.072	87.908	88.104
Average	86.813	85.646	83.372	80.340
	88.078	87.800	87.027	86.900

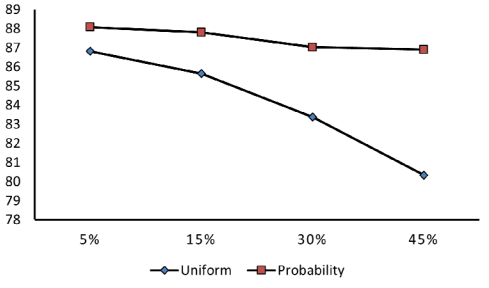
(a) Sleep Stage Scoring data

	5%	15%	30%	45%
V1	88.947	87.764	85.584	83.633
	89.893	87.366	87.543	84.731
V2	87.508	86.188	83.371	80.463
	88.201	87.603	87.368	85.541
V3	88.053	87.341	87.032	83.255
	89.127	87.965	88.763	86.139
V4	90.206	87.416	86.995	85.052
	89.406	88.698	88.296	88.345
V5	90.328	88.579	86.385	82.463
	87.513	87.324	88.088	87.235
V6	88.807	86.025	86.461	80.301
	88.199	85.643	83.229	84.241
Average	88.975	87.219	85.971	82.528
	88.723	87.433	87.215	86.039

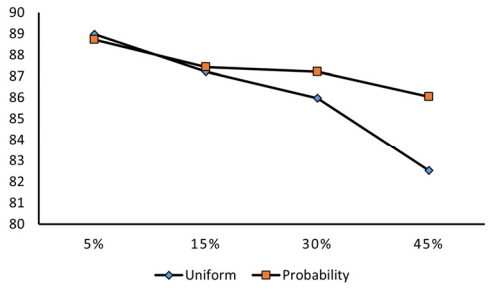
(b) Car evolution data

	5%	15%	30%	45%
V1	91.168	91.209	88.780	88.583
	90.684	91.701	89.143	84.036
V2	93.159	90.391	86.723	84.428
	90.067	90.316	89.307	87.922
V3	92.533	90.138	90.038	88.807
	91.556	91.695	87.420	82.799
Average	92.287	90.573	88.514	87.273
	90.769	91.237	88.623	84.919

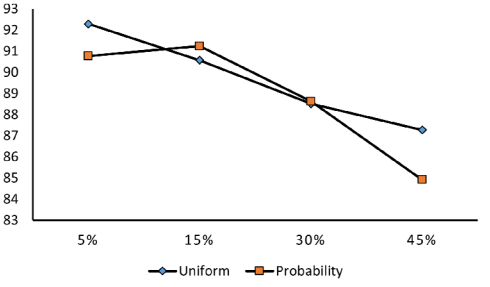
(c) Balance & Scale data



(a) Sleep Stage Scoring data



(b) Car evolution data



(c) Balance & Scale data

Fig. 4. Comparison of results in Table 3.

5. 결론

지금까지 손실값을 포함하는 불완전한 데이터를 위한 데이터 확장기법에 대해 알아보았다. 이와 관련 기존의 방법은 학습 데이터가 가지고 있던 정보들을 무시하고 일률적인 값을 할당하였다. 그리고 딥러닝과 같은 반복적 학습 모델에서는 적절한 방법이 아니라는 점이 지적 되었다. 이에 대해 제안 알고리즘은 손실값이 존재할 때 손실되지 않은 정보들을 이용해 확률을 계산하고 이를 손실된 정보의 추정값으로 할당하였다.

이 접근 방법은 손실되지 않은 정보들을 이용해 분류 영역을 구분해 놓고 손실된 정보를 이 영역에서 찾아서 확률로 변환하고 이를 손실값의 보간값으로 보상하였다. 손실된 정보의 보상된 성능을 확인하기 위해 SVM 알고리즘으로 학습하여 기존의 방법에 비해 정보의 손실을 최소화하는 의미 있는 결과를 보았다.

제안된 알고리즘은 학습 데이터의 손실값 만을 다루었으나 앞으로 부류 값이나 테스트 데이터에도 적용해 볼 수 있을 것으로 본다. 또한 손실값을 대치해 가는 다양한 방법이 개발되고 여러 분야에 적용해 보아 기법을 발전시킬 필요가 있다고 본다.

REFERENCES

[1] J. Han, J. Pei & M. Kamber. (2011). Data Mining: Concepts and Techniques, *Waltham : Elsevier*

[2] R. Kohavi & J. R. Quinlan. (2002). Data mining tasks and methods: Classification: Decision-tree discovery, *Handbook of data mining and knowledge discovery, New York : Oxford University Press,* 267-276.

[3] D. Kim, D. Lee & W. D. Lee. (2006). Classifier using Extended Data Expression, *IEEE Mountain Workshop on Adaptive and Learning Systems.* DOI : 10.1109/SMCALS.2006.250708

[4] J. C. Lee, D. H. Seo, C. H. Song & W. D. Lee. (2007). FLDF based Decision Tree using Extended Data Expression, *The 6th Conference on Machine Learning & Cybernetics,* 3478-3483

[5] J. C. Lee. (2018). Application Examples Applying Extended Data Expression Technique to Classification Problems, *Journal of the Korea convergence society*, 9(12), 9-15. DOI : 10.15207 /JKCS.2018.9.12.009

[6] J. C. Lee. (2019). Deep Learning Model for Incomplete Data, *Journal of the Korea Convergence Society, 10(2)*, 1-6. DOI : 10.15207 /JKCS.2019.10.2.001

[7] J. C. Lee & W. D. Lee. (2010). Classifier handling incomplete data. *Journal of the Korea Institute of Information and Communication Engineering, 14(1),* 53-62.

[8] A. McCallum, D. Freitag & F. Pereira. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. *Proc. Of 17th International Conference on Machine Learning*, 591-598

[9] T.Delavallade & T.H.Dang.(2007). Using Entropy to Impute Missing Data in a Classification Task. IEEE International Fuzzy Systems Conference. DOI : 10.1109/FUZZY.2007.4295430

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

[11] A. Sportisse, C. Boyer, A. Dieuleveut & J. Josse. (2020). Debiasing Averaged Stochastic Gradient Descent to handle missing values, 34th Conference on Neural Information Processing Systems, Vancouver, Canada, 1-11

[12] T. F. Johnson, N. J. B. Isaac, A. Paviolo, M. González-Suárez. (2020). Handling missing values in trait data, Global Ecology & Biogeography, 1-12. DOI : 10.1111/geb.13185

[13] S. Huang & C. Cheng. (2020). A Safe-Region Imputation Method for Handling Medical Data with Missing Values, *Symmetry 2020, 12, 1792;* DOI : 10.3390/sym12111792

[14] J. You, X. Ma, D. Y. Ding, M. Kochenderfer & J. Leskovec. (2020). Handling Missing Data with Graph Representation Learning, *34th Conference on Neural Information Processing Systems, Vancouver, Canada.* 1-13

[15] Center for Machine Learning and Intelligent Systems, University of California, Irvine, (2020). UCI Machine Learning Repository. https:// archive.ics.uci.edu/ml/datasets.php

이 종 찬(Jong Chan Lee) [종신회원]

· 1988년 2월 : 충남대학교 계산통계학과 (학사)
· 1990년 2월 : 충남대학교 대학원 전산학과(석사)
· 1996년 2월 : 충남대학교 대학원 전산학과(박사)
· 1996년 3월 ~ 현재 : 청운대학교 컴퓨터공학과 교수
· 관심분야 : 딥러닝, 패턴분류, 정보보호, 데이터압축
· E-Mail : jclee@chungwoon.ac.kr