Journal of the Korea Convergence Society
Vol. 11. No. 3, pp. 1-7, 2020

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2020.11.3.001

SVM과 딥러닝에서 불완전한 데이터를 처리하기 위한 알고리즘

이종찬
청운대학교 컴퓨터공학과 교수

Algorithms for Handling Incomplete Data in SVM and Deep Learning

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

요 약 본 논문은 불완전한 데이터를 처리하기 위해 2가지의 서로 다른 기법과 이를 학습하는 알고리즘을 소개한다. 첫째방법은 손실변수가 가질 수 있는 균등한 확률로 손실값을 할당하여 불완전한 데이터를 처리하고, SVM 알고리즘으로 이 데이터를 학습하는 것이다. 이 기법은 임의의 변수에 손실 값의 빈도가 높을수록 엔트로피가 높도록 하여 이 변수가 결정트리에서 선택되지 않도록 하는 것이다. 이 방법은 손실 변수에 남아있는 정보를 모두 무시하고 새로운 값을 할당한다는 특징이 있다. 이에 반해 새로운 방법은 손실 값을 제외하고 남아있는 정보로 엔트로피 확률을 구하고 이를 손실 변수의 추정 값으로 사용하는 것이다. 즉, 불완전한 학습데이터로부터 소실되지 않은 많은 정보들을 이용해 소실된 일부 정보를 복구하고 딥러닝을 이용해 학습한다. 이 2가지 방법은 학습데이터에서 차례로 변수 하나를 선택하고, 이 변수에 손실된 데이터의 비율을 달리하면서 서로 다른 측정값들의 결과들과 반복적으로 비교함으로써 성능을 측정한다.

주제어 : SVM, 엔트로피, UChoo, 확장된 데이터 표현, 불완전한 데이터, 딥러닝

Abstract This paper introduces two different techniques for dealing with incomplete data and algorithms for learning this data. The first method is to process the incomplete data by assigning the missing value with equal probability that the missing variable can have, and learn this data with the SVM. This technique ensures that the higher the frequency of missing for any variable, the higher the entropy so that it is not selected in the decision tree. This method is characterized by ignoring all remaining information in the missing variable and assigning a new value. On the other hand, the new method is to calculate the entropy probability from the remaining information except the missing value and use it as an estimate of the missing variable. In other words, using a lot of information that is not lost from incomplete learning data to recover some missing information and learn using deep learning. These two methods measure performance by selecting one variable in turn from the training data and iteratively comparing the results of different measurements with varying proportions of data lost in the variable.

Key Words : SVM, Entropy, UChoo, Extended data expression, Incomplete data, Deep learning

*Corresponding Author : Jong Chan Lee(jclee@chungwoon.ac.kr)
Received December 9, 2019 Revised January 3, 2020
Accepted March 20, 2020 Published March 28, 2020

1. 서론

손실(missing) 변수가 포함되어 있는 데이터를 본 논문은 불완전한(incomplete)한 데이터라 정의한다. 이 불완전한 데이터 처리 문제는 여러 개의 센서나 정보를 사용하는 유비쿼터스 환경에서는 자주 발생하며, 이를 처리하는 연구는 높은 정확도를 가지는 분류 모델을 개발하기 위하여 매우 중요한 의미를 갖는다[1,2].

본 논문에서는 데이터 중에 임의 레코드에서 변수 값에 손실이 발생할 경우를 위해 2가지 방법을 제안한다. 데이터에 손실이 발생할 경우 사용하는 일반적인 기법은 손실치가 포함된 레코드를 무시하고 처리하는 방법과 여러 알고리즘을 이용해 손실치를 채우고 처리하는 방법으로 나누어진다. 본 논문에서는 후자에 속하는 방법으로 손실치가 포함된 전체 레코드를 학습에서 제외할 경우 이 레코드들에 포함된 정보들을 모두 잃어버리기 때문이다. 특히 학습 데이터 중에 손실 레코드가 많이 포함될수록 정보의 손실로 인해 성능에 커다란 영향을 미치게 된다. 따라서 불완전한 학습데이터로부터 대부분의 손실되지 않은 데이터들을 이용해 손실된 일부 정보를 복구하여 학습 성능을 향상할 목적으로 2가지 방법을 제안하고 이들의 학습 성능을 비교한다. 2가지 제안 방안은 모두 데이터 확장 기법을 이용해 학습데이터를 변환한 다음 서로 다른 확률 값들을 구하여 손실 값들을 채운다. 그리고 서로 다른 알고리즘인 SVM(Support Vector Machine)과 딥러닝(Deep Learning)을 이용해 학습한 후 그 결과들을 비교해 본다.

데이터 확장 기법은 먼저 손실된 변수는 제외하고 손실되지 않은 변수들을 원-핫-인코딩으로 변환하고, 각 레코드들 마다 각각의 가중치 값을 부여한다. 이 가중치는 전체 데이터 집합에서 어느 특정한 일부 데이터에 전문가가 개입하여 특별히 중요한 의미를 부여하고자 할 때 이를 반영하여 학습이 가능하도록 해준다. 또한 AdaBoost 알고리즘과 같이 가중치에 따라 다음 학습 데이터 집합을 선택하는 알고리즘에서도 유용하게 쓰일 수 있다. 그 동안 이 방법은 C4.5와 같이 엔트로피를 기반으로 하는 알고리즘[3-5]에 손실 데이터를 위해 적용하여 좋은 결과를 산출해 왔다. 데이터 확장 기법을 실행한 후 소실된 변수들은 다음 과정을 통해 확률 값으로 변환된다. 첫째, 소실된 변수의 카디너리티(cardinality)를 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 UChoo 등[6-10]은 각 변수마다 엔트로피를 구한 다음, 이 정보를 이용해 결정트리를 구성해 간다[7]. 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. 이 방법은 딥러닝과 같이 엔트로피를 기반으로 하지 않는 알고리즘에서 불완전한 데이터를 처리하기에 적합하다.

2. 배경

2.1 FLDF을 사용하는 SVM 알고리즘[11,12]

SVM 알고리즘의 초평면(P)을 구하기 위해 Fisher의 선형분리함수(Fisher's Linear Discriminant Function (FLDF))를 이용하는데, 그 과정은 다음과 같다. k(k>2)개의 부류를 가지는 경우의 Fisher 선형분리함수는 (1)식에 나타나 있다. 이 식은 각 학습패턴들을 초평면에 투사할 때 투사점들로 이루어지는 각 부류의 그룹들 간의 거리(B)는 최대로 멀리 떨어지도록 하고(분모), 동시에 동일 부류에 속한 투사점(V)들은 가능한 모여 있도록 하는(분자) 직관을 반영한 것이다.

$$\frac{P^T \cdot B \cdot P}{P^T \cdot V \cdot P} = \frac{P^T[\sum_i (\overline{X_i} - \overline{X})(\overline{X_i} - \overline{X})^T]P}{P^T[\sum_i \sum_j (X_{ij} - \overline{X_i})(X_{ij} - \overline{X_i})^T]P} \qquad (1)$$

여기서 i=1,...,부류의 수(k), j=1,...,각 부류의 이벤트 개수

(1)식을 최대로 하는 해를 구하기 위해 Cauchy-Schwartz 부등식을 이용한다. 우선 $\alpha = V^{1/2}P$라 놓으면 (1)식은 $(\alpha^T V^{-1/2} B V^{-1/2} \alpha)/(\alpha^T \alpha)$이 된다. 여기서 $V^{-1/2}BV^{-1/2}$의 고유값을 구하고, 이 고유값들 중에 최대 고유값(λ_1)에 상응하는 고유벡터(e_1)를 구한다. Cauchy-Schwartz 부등식에 의해 $\alpha = e_1$일 때 $V^{-1}B$는 최대값을 가진다. 따라서 초평면 벡터(P)는 $V^{-1/2}e_1$로 구한다.

초평면을 구하고 입력패턴(X) 각각을 이 초평면에 투사(P^TX)한다. 그리고 이 투사점들을 분리하는 임계값은 (2)식의 엔트로피 함수를 이용한다.

$$E(C \mid \delta^d) = PL^* E(q_1) + PR^* E(q_2) \qquad (2)$$

우선 투사면상의 투사점들을 크기에 따라 정렬한다. 그리고 차례대로 각 투사점들 사이의 평균을 구해 d를

정한다. 즉 $d_s = (P^T X_{s-1} + P^T X_s)/2$, s=1,...,(패턴의 총갯수-1). d를 기준으로 엔트로피를 각각 계산해서 가장 작은 엔트로피를 가지는 d를 임계값으로 정한다.

$PL^* = n_1/(n_1+n_2)$, $PR^* = n_2/(n_1+n_2)$

$n_1(n_2)$: d의 왼쪽(오른쪽) 영역에 속한 패턴의 수

$E(q_i) = -\sum q_{ij} \log q_{ij}$,

여기서 i=d의 왼쪽/오른쪽, j=1,...,부류의 수. 또한 x_j를 각 영역(왼쪽, 오른쪽)에서 각 부류 j의 패턴개수라 정의하면 $q_{ij} = x_{ij} / n_i$가 된다.

이 알고리즘에 확장된 데이터 표현을 사용하는 데이터를 입력하려면 가중치 개념과 각 데이터 엔트리의 표현 방법의 차이로 인해, (1)식의 초평면과 (2)식의 엔트로피 함수에 반영해야 한다[11,12].

2.2 UChoo의 처리기법

2.2.1 확장된 데이터 표현[13]

UChoo는 C4.5 알고리즘에 확장된 데이터 표현을 더한 알고리즘으로 많은 손상된 데이터 실험을 하여 좋은 결과를 산출해오고 있다. 여기서 확장된 데이터 표현은 손실된 데이터에 대해 확률적으로 표현하는 방법을 사용하였고 각 사건(event)의 중요도를 달리할 수 있는 표현 기법을 고려하였다. 이를 Table 1과 같은 학습데이터를 이용해 설명한다. 7개의 사건으로 이루어졌으며, 3개의 변수(V1, V2, V3)와 1개의 부류(class)가 각각 (3, 3, 2, 2)의 카디너리티를 가지고 있다. 여기서 주목할 것은 사건 6의 V2와 사건 7의 V3가 손상되었다는 것이다. Table 1을 데이터 확장 표현으로 바꾼 것이 Table 2에 나타나 있다. 첫째, 손상되지 않은 각 변수의 엔트리들은 변수의 카디너리티에 따라 원-핫-인코딩한다. 둘째, 각 사건들의 중요도에 따라 가중치를 부여한다. 이 가중치는 알고리즘에 의해 할당될 수도 있고, 전문가의 경험에 의해 결정될 수도 있다. 예를 들어 사건 5는 가중치가 10이 할당되었는데 이 사건은 다른 사건에 비해 10배 만큼 중요한 사건임을 의미한다. 따라서 사건의 수와 인스턴스의 수는 같지 않은데 Table 2의 예에서 사건의 수는 7인데 반해 인스턴스의 수는 16이다.

Table 1. Example of training data containing missing data

Event	V1	V2	V3	Class
1	1	2	2	1
2	3	1	1	2
3	2	2	2	2
4	2	3	2	1
5	1	2	1	1
6	3	?	1	1
7	3	3	?	2

Table 2. The extended data expression of Table 1.

E	W	V1			V2			V3		Class	
		1	2	3	1	2	3	1	2	1	2
1	1	1	0	0	0	1	0	0	1	1	0
2	1	0	0	1	1	0	0	1	0	0	1
3	1	0	1	0	0	1	0	0	1	0	1
4	1	0	1	0	0	0	1	0	1	1	0
5	10	1	0	0	0	1	0	1	0	1	0
6	1	0	0	1	?	?	?	1	0	1	0
7	1	0	0	1	0	0	1	?	?	0	1

2.2.2 UChoo의 확장된 데이터 표현

UChoo는 학습 중에 결정트리를 구성해 나가는데, 다른 결정 트리를 이용하는 알고리즘에 비하여 손실 데이터를 처리하기에 유리하다. 각각의 데이터 엔트리들이 0과 1사이의 확률로 표현되기 때문이다. 훈련 데이터 집합에 손실 값이 존재하는 경우 그 변수의 카디너리티로 모든 엔트리에 대해 동일한 확률 값을 할당한다. 예를 들어 Table 2에서 사건 6의 V2의 손실 값은 카디너리티가 3이므로 1/3을 각 데이터 엔트리에 채워 (1/3, 1/3, 1/3)을 할당한다. 또한 사건 7의 V3는 카디너리티가 2이므로 1/2을 각 데이터 엔트리에 (1/2, 1/2)를 할당한다. UChoo는 엔트로피를 사용하여 분류하기 가장 적합한 즉, 엔트로피 값이 가장 작은 변수를 선택하고 이 변수의 값에 따라 트리를 반복적으로 구성해 나가는 알고리즘이다. 따라서 엔트로피는 각각의 사건이 일어날 확률이 동일할 때 가장 높아지므로, 손실 값이 많은 변수일수록 루트 쪽 보다는 리프 노드 쪽에 선택될 것이다. 다시 말해 손실 값을 포함하는 변수는 트리를 통한 분류에서 중요도를 떨어뜨리는 요소로 작용하도록 함을 알 수 있다.

3. 손실 값을 보상하기 위한 엔트로피 확률

앞 절에서 설명한 UChoo의 손실값 처리기법은 원 데이터가 가지고 있는 정보를 무시하고 일정한 확률값을 손실값에 할당하는 방법으로 엔트로피를 척도로 하는 알고리즘에는 적합하였다. 그러나 딥러닝과 같이 엔트로피

를 기반으로 하지 않는 학습에서는 학습 데이터 자체가 가지고 있는 정보를 보전하는 새로운 확률값을 이용해 손실값을 보상하기 위한 방법이 보다 효율적일 것이다 [14]. 따라서 본 논문은 정보이론을 바탕으로 학습 데이터에서 손실값을 위한 확률을 추출해 내는 것을 목적으로 한다.

정보이론(Information theory)은 정형화되지 않은 개념인 정보를 수량화 할 수 있다는 점에서 의미가 있다. 이와 관련된 연구들 중에 엔트로피는 본래 열역학에서 개발된 개념이었으나 Shannon과 Jaynes에 의해 재발견되어 정보이론 분야에 지대한 영향을 미치게 되었다 [15]. 정보 함수는 확률 실험에서 어떤 사건이 발생했을 때 이 사건에서 얻을 수 있는 정보의 양을 측정하는 척도이다. 정보함수가 지녀야 될 속성은 두 가지로 나누어지는데, 첫째는 사건이 일어날 확률과 정보의 양은 반비례한다는 것이고, 둘째는 독립된 사건을 동시에 관측했을 때 얻는 정보의 양과 각 사건을 따로 관측했을 때 얻는 정보의 양의 합은 같아야 한다는 것이다. 여러 학자들에 의해 이러한 속성을 만족하는 것은 로그 함수임이 밝혀졌다[16]. 확률 실험에서 관측자는 어떤 사건이 일어날 것인가는 라는 것 보다는 단지 각 사건이 일어날 확률만을 안다. 그러므로 주어진 확률로부터 정보의 양을 측정하기 위해서는 정보 함수의 기대치(probability density function)를 구해야 한다. 이러한 것들을 반영한 식이 (3)식에 있다.

$$H(P) = \sum_{i=0}^{E} -P_i \log P_i \tag{3}$$

이 식은 사건 i가 일어날 확률을 P_i라 할 때, 정보의 양은 확률 P_i에 반비례하므로 $1/P_i$이 된다. 그리고 log를 사용하여 0과 1사이의 값으로 변환한다. (3)식은 이 모든 경우에 대한 기댓값을 구한 것이다.

먼저 손실값을 채워 넣을 확률값을 산출하기 위한 변수들을 정의한다. 그리고 이 변수들을 이용해 (4)식과 같은 확률 값을 정의 한다. 이 값은 전체 학습 데이터 중에 손실값을 포함하지 않는 사건들 중에 변수값(attribute)이 j이며 부류가 i인 사건에 대한 조건부 확률이다. 그리고 (4)식의 확률을 이용해 (5)식과 같이 손실값(V_{ij})에 채워 넣을 확률을 계산한다.

- E : 학습데이터에서 손실값을 포함하지 않는 사건
- n(·) : 가중치를 고려한 함수 (·)의 수
- i = 1,...,부류(C)의 수(k)
- j = 1,...,손실된 어트리뷰트(A)의 카디너리티

$$P(C_i|A_j) = \frac{n((Class = i) \cap (attribute = j))}{n(attribute = j)} \tag{4}$$

$$V_{ij} = P(A_j) \times \frac{-P(C_i|A_j)\log_2 P(C_i|A_j)}{\sum_{i=1}^{k} -P(C_i|A_j)\log_2 P(C_i|A_j)},$$

$$For\ \forall_i,\ \sum_{j=1}^{cardinarity} V_{ij} = 1 \tag{5}$$

(5)식은 각 부류에 대해 손실 변수가 가질 수 있는 값들을 확률로 나타낸 것이다. 그리고 이 식은 학습 데이터 중에 손실 부분을 제외한 데이터의 정보를 이용해 엔트로피를 구하고 이를 확률로 변환하였다. 이 식의 이해를 돕기 위해 Table 2를 학습 데이터의 예로 들어 가중치를 고려하여 (5)식에 따라 확률값들을 계산하였다. 이 값들은 Table 3와 같이 검색표(lookup table)에 저장된다. 이 검색표는 학습 전에 만들어 지고 학습 데이터에 손실값이 발견되면 부류와 카디너리티에 따라 값을 찾아 할당 한다. 예를 들어 Table 2의 사건 6에서 손실값(V3)은 부류값으로 1을 가진다. 따라서 Table 3의 검색표에서 부류 1이 (0, 0.7694, 0.2306)이므로 이 값을 손실 엔트리에 채우면 된다. 즉 UChoo의 방법에서 (1/3, 1/3, 1/3) 대신에 (0, 0.7694, 0.2306)로 대치된다. 또한 사건 7에서 손실값(V3)은 부류값으로 2를 가지므로 (0.8339, 0.1661)을 엔트리에 채운다.

이 모델의 학습[16,17]은 각 변수의 카디너리티의 합과 같은 입력노드와 부류의 카디너리티와 같은 출력노드, 그리고 은닉노드들로 구성된다. Table 2의 학습 데이터의 경우 9개의 입력노드(8개의 입력 1개의 편기노드)와 2개의 출력노드들로 구성된다.

Table 3. Lookup table for missing variable in Table 2.

(a) Missing values for V2

		Cardinality		
		1	2	3
Class	1	0	0.7694	0.2306
	2	0.2320	0.5361	0.2320

(b) Missing values for V3

		Cardinality	
		1	2
Class	1	0.7237	0.2763
	2	0.8339	0.1661

4. 실험

실험을 위해 UCI 기계 저장소[18]의 데이터 중에 “Balance & Scale(BS)”(625 사건, 3 변수, 3 부류)과 “Car evolution(CE)”(1728 사건, 6 변수, 4 부류)을 사용하였다. 그리고 인간의 뇌파를 기록한 데이터인 “Sleep stage scoring(SSS)”(799 사건, 11 변수, 6 부류)데이터를 사용하였다. 성능 측정을 위해 10%는 테스터 데이터로 사용하는 10-fold cross validation을 사용하였다. 그리고 학습 데이터는 손실비율을 각각 달리하여 0%, 5%, 15%, 30%, 45% 비율만큼 임의로 데이터를 선택한 다음, 이 선택된 사건의 각 변수 마다 교대로 값을 손실시키며 결과를 알아보았다. 실험 결과는 이러한 방법대로 10번을 실행하여 각각의 결과의 평균을 산출하였다. 그 결과가 변수와 손실률 별로 Table 4에 나타나 있다. 여기서 변수와 손실률이 만나는 위 줄(하얀색)은 UChoo의 데이터 확장방법을 사용하고 이를 SVM 알고리즘으로 학습한 결과를 나타내고, 아래 줄(회색)은 본 논문에서 제안하는 엔트로피 확률을 사용하고 딥러닝으로 학습한 결과이다. 이들 결과를 좀 더 알아보기 좋게 평균값(Avg)들을 그래프로 나타낸 것이 Fig. 1에 나타나 있으며 빨간색과 네모 포인트가 제안한 엔트로피를 사용한 결과이고, 파란색에 다이아몬드 포인트가 UChoo의 데이터 확장 기법을 사용한 결과이다. 데이터 마다 결과에 약간의 차이는 있으나 제안한 방법의 결과가 다소 나은 결과를 보이고 있다.

이 결과에서 특이점은 다음과 같이 2가지를 볼 수 있다. 첫째 Table 4의 (C)에서 SSS 데이터는 변수가 11개나 되어 어느 하나의 변수에 손실이 생겼다고 해서 전체 성능에 커다란 영향을 미치지 않는다. 이러한 해석은 (a)의 BS 데이터의 결과에서 손상률에 따른 결과 저하도가 비교적 명확하다는 것과 대비된다. BS는 3개의 변수만으로 구성되기 때문에 이들 중의 하나가 손상되면 영향이 가장 클 것이다. 두 번째 제안된 결과가 좋았던 것은, 손실된 데이터를 복구한 후 성능평가를 위해 딥러닝을 이용해 학습했다는 점에서 원인을 찾을 수 있다. 일반적으로 손실된 데이터를 복구 했을 때 원본 데이터 보다는 정보가 손실되어 있을 것이라고 생각할 수 있다. 그러나 학습하는 중에 딥러닝의 특성인 오류를 복구하는 성질로 인해 정보 손실의 차이를 줄였다고도 볼 수 있다.

Table 4. The experiment results

(a) Balance Scale Weight & Distance Database

	45%	30%	15%	5%	0%
V1	88.58	88.78	91.21	91.17	93.08
	87.27	90.17	92.09	97.99	
V2	84.43	86.72	90.39	93.16	
	85.55	88.13	95.33	96.62	
V3	88.81	90.04	90.14	92.53	100.0
	83.97	87.22	91.81	96.55	
Avg	87.27	88.51	90.57	92.29	
	84.98	88.27	92.97	97.05	

(b) Car Evaluation Database

	45%	30%	15%	5%	0%
V1	83.63	85.58	87.76	88.95	92.81
	98.24	98.24	98.43	99.41	
V2	80.46	83.37	86.19	87.51	
	97.84	98.72	98.27	99.35	
V3	83.25	87.03	87.34	88.05	
	94.82	96.85	98.04	99.59	
V4	85.05	86.99	87.42	90.21	
	97.81	99.01	99.46	99.77	99.88
V5	82.46	86.39	88.58	90.33	
	94.63	95.55	97.74	98.94	
V6	80.30	86.46	86.02	88.81	
	96.82	98.79	99.32	99.54	
Avg	82.53	85.97	87.22	88.97	
	96.69	97.86	98.54	99.43	

(c) Sleep Stage Scoring Data

	45%	30%	15%	5%	0%
V1	79.64	83.33	85.11	86.02	87.63
	88.90	90.59	89.08	90.38	
V2	84.01	84.16	86.43	87.34	
	89.41	90.37	89.39	89.48	
V3	80.58	84.45	87.33	86.82	
	89.75	89.04	90.64	90.79	
V4	80.10	84.45	86.20	87.14	
	89.34	89.14	89.81	89.87	
V5	81.00	82.83	85.15	85.84	
	88.91	89.54	89.94	90.26	
V6	79.33	83.67	86.39	87.42	
	90.29	90.11	90.29	89.18	
V7	83.06	83.60	85.94	86.63	89.35
	89.58	89.91	89.39	89.52	
V8	82.99	84.15	86.41	87.22	
	89.21	90.08	89.36	89.06	
V9	78.78	82.70	84.81	85.69	
	89.93	88.67	89.59	90.25	
V10	74.90	81.19	83.78	87.02	
	89.56	90.71	90.14	90.53	
V11	79.35	82.57	84.56	87.10	
	89.46	89.31	89.27	89.85	
Avg	80.34	83.37	85.65	86.81	
	89.49	89.77	89.72	89.92	

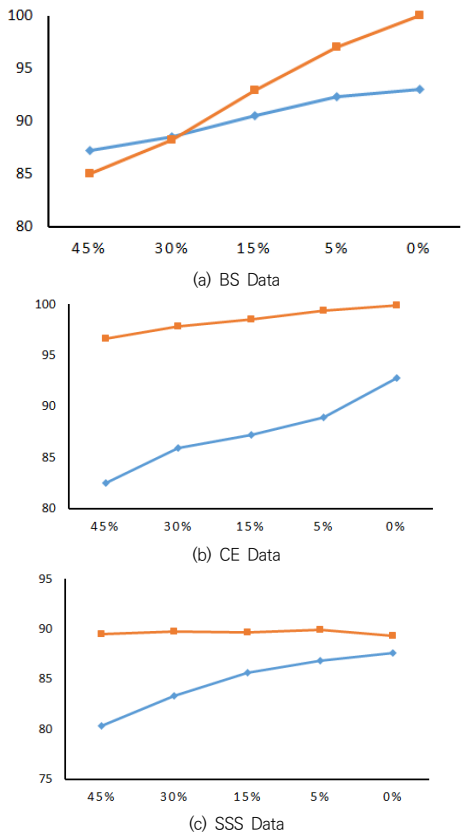

Fig. 1. Graph for the results in Table 4.

5. 결론

지금까지 손실 값을 포함하는 불완전한 데이터의 처리 방법에 대해 2가지 방면으로 알아보았다. 첫 번째는 C4.5, UChoo와 같이 엔트로피를 기반으로 결정트리를 구성해 가며 학습이 이루어지는 알고리즘에서의 처리방법이었다. 여기서는 학습 중에 손실값이 있는 경우 이 변수의 엔트로피가 가장 높도록 하는 방법을 사용하였다. 즉, 이 알고리즘은 손실된 변수가 결정트리의 상단에서 선택되지 않도록 확률이 일양분포를 가지도록 할당했다. 이 방법을 사용하여 많은 문제를 해결할 수 있었으며 좋은 결과를 산출해 왔다. 그러나 이 방법은 기존의 학습 데이터가 가지고 있던 정보들을 무시하고 일률적인 값을 할당한다는 문제가 있었다. 또한 딥러닝과 같은 반복적 학습 모델에서는 적절한 방법이 아니라는 것이다.

이에 따라 새로운 방법이 제안되었는데 이는 손실값이 존재할 때 손실되지 않은 정보들을 이용해 엔트로피 확률을 계산하고 이를 손실된 정보의 추정값으로 할당한다. 실험은 학습데이터에 대해 손실률을 달리해가며 결과를 측정하고, 데이터가 가지고 있는 변수를 변경해가며 실험을 하여 좋은 결과를 산출함을 보았다.

앞으로 손실된 변수의 평균값으로 손실 값을 대치해 가는 방법 등 다양한 방법이 개발되고 여러 분야에 적용해 보아 기법을 발전시킬 필요가 있다고 본다. 예를 들어 유비쿼터스 분야에서는 서로 다른 환경과 센서 기기들로부터 데이터가 수집되고, 원격지로부터 데이터들이 모아지는 관계로 불완전한 데이터가 수집되는 경우가 많으므로 본 논문에서 제안한 방법이 유용하게 쓰일 수 있기를 기대한다.

REFERENCES

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

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

[3] T. Maszczyk & W. Duch. (2008). Comparison of Shannon, Renyi and Tsallis Entropy used in Decision Trees. *Lecture Notes in Computer Science, 5097*, 643-651.

[4] M. Srednicki & Theoretical Physics Group.(1993) *Entropy and Area.* Physical Review letters. DOI : 10.1103/PhysRevLett.71.666

[5] R.M.Gray.(2011).*Entropy and Information Theory.* Springer 2ED. DOI : 10.1007/978-1-4419-7970-4

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

[7] P. E. Utgoff. (1989). Incremental Induction of Decision Trees. *Machine Learning, 4(2),* 161-186.

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

[9] H. Lee, S. Chung & E. Choi. (2016). A Case Study on Machine Learning Applications and Perfor- mance Improvement in Learning Algorithm, *Journal of Digital Convergence, 14(2),* 245-258.

[10] Y. Jeong. (2019). Machine Learning Based Domain Classification for Korean Dialog System, *Journal of Convergence for Information Technology, 9(8),* 1-8.

[11] 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.

[12] 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.

[13] 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

[14] 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

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

[16] 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

[17] 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

[18] E. Keogh, C. Blake & C. J. Merz. (1998). *UCI Repository of Machine Learning Databases,* UCI Machine Learning Repository. http://www.ics .uci.edu/~mlearn/MLRepository.html

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