Journal of The Korea Convergence Society
Vol. 12. No. 11, pp. 91-97, 2021

I-SSN 2233-4890 / e-ISSN 2713-6353
DOI : https://doi.org/10.15207/JKCS.2021.12.11.091

데이터 확장 기법에서 손실값을 대치하는 확률 추정 방법

이종찬
청운대학교 컴퓨터공학과 교수

Probability Estimation Method for Imputing Missing Values in Data Expansion Technique

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

요 약 본 논문은 불완전한 데이터를 처리하기 위해 본래 규칙개선 문제를 위해 고안되었던 데이터 확장 기법을 사용한다. 이 기법은 사건마다 중요도를 의미하는 가중치를 가질 수 있으며 각 변수를 확률값으로 나타낼 수 있는 특징이 있다. 본 논문에서의 핵심 문제가 손실값과 가장 근사한 확률을 구하여 손실값을 확률로 대치하는 것이므로, 3가지 다른 알고리즘으로 손실값에 대한 확률을 구한 후 이 데이터 구조의 형식으로 저장한다. 그리고 각각의 확률 구조에 대한 평가를 위해 SVM 분류 알고리즘으로 각각의 정보 영역을 분류하는 학습을 한 후, 본래의 정보와 비교하여 얼마나 서로 일치하느냐를 측정한다. 손실값의 대치 확률을 위한 3가지 알고리즘들은 같은 데이터 구조를 사용하고 있으나 접근 방법에서는 서로 다른 특징을 가지고 있어 적용 분야에 따라 다양한 용도로 이용될 수 있기를 기대한다.

주제어 : 불완전한 데이터, SVM, 확장된 데이터 표현, 결정트리, 확률값

Abstract This paper uses a data extension technique originally designed for the rule refinement problem to handling incomplete data. This technique is characterized in that each event can have a weight indicating importance, and each variable can be expressed as a probability value. Since the key problem in this paper is to find the probability that is closest to the missing value and replace the missing value with the probability, three different algorithms are used to find the probability for the missing value and then store it in this data structure format. And, after learning to classify each information area with the SVM classification algorithm for evaluation of each probability structure, it compares with the original information and measures how much they match each other. The three algorithms for the imputation probability of the missing value use the same data structure, but have different characteristics in the approach method, so it is expected that it can be used for various purposes depending on the application field.

Key Words : Incomplete data, SVM, Extended data expression, Decision tree, Probability value

1. 서론

수집된 데이터에서 일부 변수가 손실된 불완전한 데이터는 유비쿼터스 등의 광범위한 분야에서 접하게 되며 이로 인해 시스템의 성능 저하와 연관되므로 반드시 해결해야 하는 문제[1,2]이다. 본 논문은 이러한 알고리즘 중에 데이터 확장 기법[3]을 사용하는 3가지 방법들을 소개하고 서로를 비교한다. 데이터 확장 기법은 일부

*This article is extended from the conference paper presented at ICCT2021.
*Corresponding Author : Jong Chan Lee(jclee@chungwoon.ac.kr)
Received October 6, 2021 Revised October 23, 2021
Accepted November 20, 2021 Published November 28, 2021

변수가 손상된 데이터를 복원하는 중에 사용되는 데이터 구조이며, 사건마다 가중치를 달리할 수 있고 속성값들을 확률로 표현하기 편하다는 특징을 가지고 있다. 본래 이 기법은 규칙 개선(rule refinement) 문제를 해결하기 위해 개발되었고 그동안 여러 문제에 적용되어 좋은 결과를 산출해 왔다[4]. 규칙 개선 문제는 학습 데이터를 이용해 지식습득(knowlege acquisition) 알고리즘들로 결정트리를 구성하며, 이에 따라 규칙을 산출하게 되는데 시간이 지남에 따라 학습 데이터가 추가 또는 변경되는 등의 변화가 있을 때 대처하기 위한 것이다. 따라서 본래의 데이터가 보존되지 않았다 하더라도 규칙만 가지고 본래 데이터와 근사한 정보를 복원해 낼 수 있어야 한다.

데이터 확장 기법은 손실값을 가지는 경우 손실된 사건에 적합하도록 확률을 구하고 이를 손실값에 대체한다. 여기서 가장 중요한 부분이 손실값을 대체할 수 있는 확률을 구하는 것으로 균등값의 확률[4], 엔트로피 확률[5,6], 결정 트리로부터 확률[7]을 구하는 3가지 방법에 대해 서로 이론적인 배경을 살펴보고 각각의 실험 결과를 통해 서로의 성능을 비교해 본다. 첫째, 균등 분포(uniform distribution)의 확률로 대체하는 것은 변수의 값이 평균값을 가질 때 엔트로피가 가장 큰 값을 가진다는 성질을 이용하는 것으로 손실값을 많이 포함하는 변수일수록 결정트리의 상위 노드에서 선택되지 않도록 하여 분류 문제와 이에 따르는 규칙에서 중요도를 낮춘다는 것이다. 둘째, 엔트로피와 결정트리의 확률은 균등한 확률이 손실된 사건에 남아있는 정보를 무시하고 일정한 값으로 손실값을 대체한다는 점을 보완한 것으로 학습 데이터를 손실된 사건과 완전한 사건으로 분리한 후 완전한 사건에서 정보를 추출한 다음 이를 확률로 대체하는 방법을 사용한다. 완전한 사건에서 정보를 추출하는 방법에 따라 또 2가지로 나뉘는데, 엔트로피 방법은 엔트로피를 구해 테이블을 구하고 손실값에 따라 확률값을 이 테이블에서 찾아 대체한다. 그리고 결정 트리 방법은 분류 알고리즘인 C4.5[8]를 이용해 정보를 분류하는 중에 결정트리가 생성되고, 이 결정트리의 각각의 경로는 정보들의 구분된 영역을 의미한다는 것을 이용한다. 즉 손실 사건으로 결정트리를 순회하게 하여 자신과 가장 유사한 정보 영역을 찾아 손실된 정보를 확률로 표시한다.

이들 손실값을 복원하는 3가지 알고리즘으로 산출한 확률들의 성능을 측정하기 위해 Fisher의 선형분류식을 이용하는 SVM[9]으로 학습하여 그 결과를 비교한다. 물론 EBP나 딥러닝 알고리즘을 이용하면 SVM의 결과보다 좋은 결과를 얻을 수 있으나 이들 반복적 학습 알고리즘[10-12]은 손실값에 대한 복원 기능이 내재 되어있어 순수하게 3가지 확률의 성능을 비교하기에는 SVM 알고리즘이 적합하다.

2. 관련 연구

데이터 확장 기법은 본래 규칙 개선 문제를 해결하기 위해 개발되었는데, 각 속성값을 확률로 표현할 수 있고 각 사건의 중요도를 표시할 수 있는 데이터 구조이다. 그동안 이 구조를 이용하여 불완전한 데이터의 분류 문제는 물론이고 독성에 노출된 물고기들의 특성을 감지하여 오염된 환경을 조기에 감지하기 위한 생물학적인 조기경보 시스템, 그리고 앙상블 시스템에서 약한(weak) 학습자를 구성하기 위한 학습 데이터를 선택하기 위한 가중치를 관리하는 용도로 이용되어 좋은 결과를 산출해 왔다[4].

규칙 개선 문제는 Fig. 1과 같이 과거에 사용하던 규칙(rule)들과 새로 수집된 데이터들을 합하여 새로운 규칙을 만드는 것을 말한다. Fig. 1에서 “Training 1”의 학습 데이터를 가지고 임의의 분류 알고리즘으로 학습을 하면 “Rule 1”의 규칙이 만들어진다. 이 규칙을 사용하는 중에 “Training 2”와 같은 학습 데이터가 추가/변경되었다면 이를 반영하기 위해 처음부터 학습하는 방법도 있을 수 있다. 그러나 이것보다는 기존 규칙(Rule 1)과 새로운 학습 데이터(Training 2)를 결합하여 새로운 규칙(Rule 2)을 산출하도록 하는 것이 정보의 유지비용 면에서 유리할 것이다. 이 문제는 본래의 규칙을 만들 때 사용한 원본 데이터가 심지어 없거나 손실되었는데 새롭게 수집된 데이터와 기존 정보를 결합하는(규칙+새 데이터) 융합도 가능하다는 장점을 가진다. 예를 들어 임의의 학습 데이터로 학습한 결과 Fig. 2와 같은 결정 트리를 얻었다고 가정할 때, 이 결정 트리에는 경로에 따라 5개의 규칙 정보가 포함되어 있다. 이 트리에서 동그라미 노드는 속성을 의미하고, 노드와 노드 사이의 수는 속성값(사건의 수)를 의미한다. 예를 들어 V2 노드에서 3(5)는 [V2=3]인 사건이 5개 있다는 것을 의미한다. 그리고 마지막의 말단 노드는 부류를 나타내고 “C2:2”는 부류 2가 2개의 사건을 갖는 말단 노드를 의미한다. 이러한 규칙 정보를 바탕으로

하는 확장된 데이터 표현은 Table 1과 같이 규칙 기반의 학습 데이터가 만들어진다. 즉, 규칙 1에서 V2는 1이고 V1은 1인데 V3의 값은 무정(don't care)이므로 V3의 카디너리티인 2를 가지고 균등하게 할당한다. 그리고 부류가 1이므로 사건(E) 1의 부류에 표시한다. 그리고 규칙 2는 V2, V1, 부류의 값을 사건 2에 할당하고 V3의 카디너리티에 따라 균등하게 할당한다는 것은 사건 1의 경우와 같다. 그러나 규칙 2에 해당하는 사건은 2개이므로 가중치(W)가 2가 된다. 따라서 가중치가 2라는 것은 가중치 1인 사건의 2배의 역할을 담당한다고 볼 수 있다.

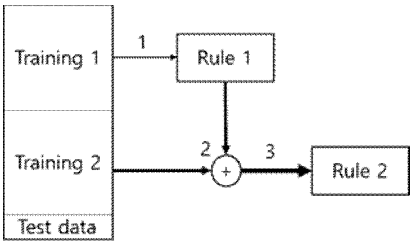


Fig. 1. Rule refinement problems

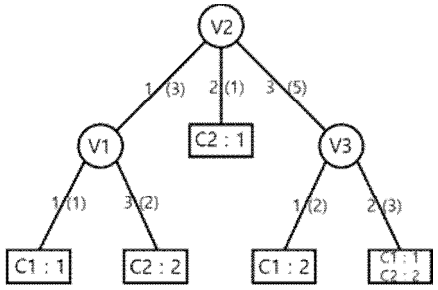


Fig. 2. Decision tree example

규칙 1 : [C1 : 1] = [V2=1][V1=1]
규칙 2 : [C2 : 2] = [V2=1][V1=3]
규칙 3 : [C2 : 1] = [V2=2]
규칙 4 : [C1 : 2] = [V2=3][V3=1]
규칙 5 : [C1 : 1, C2 : 2] = [V2=3][V3=2]

본 논문에서 중점적으로 다루는 손실값을 가지는 불완전한 데이터 처리에도 Table 1의 무정값 처리 방법과 같이 손실되지 않은 속성값은 원-핫-인코딩으로 나타내고 손실된 속성값에 대해서는 손실값을 대치하는 값을 구하여 손실값에 확률 형식으로 채운다.

Table 1. Extended data expression according to the decision tree in Fig. 2.

E	W	V1			V2			V3		Class	
		1	2	3	1	2	3	1	2	1	2
1	1	1	0	0	1	0	0	1/2	1/2	1	0
2	2	0	0	1	1	0	0	1/2	1/2	0	1
3	1	1/3	1/3	1/3	0	1	0	1/2	1/2	0	1
4	2	1/3	1/3	1/3	0	0	1	1	0	1	0
5	3	1/3	1/3	1/3	0	0	1	0	1	1/3	2/3

3. 불완전한 데이터에 확장된 데이터 표현을 사용한 알고리즘들

3.1 균등한 확률

샤논은 열역학 문제를 위해 개발된 엔트로피 개념을 정보 이론에 도입하여 (1)식과 같이 정보의 양을 측정하도록 새롭게 해석하였다.

$$H(P)=\sum_{i=0}^{E}-P_i\log P_i \quad (1)$$

(1) 식은 임의의 사건 i가 일어날 확률을 P_i라 할 때 정보의 양을 표현한 것이다. 즉, 정보의 양은 사건 i의 확률(P_i)에 반비례하므로 $1/P_i$이 된다. 여기에 log를 사용하여 0과 1사이의 값으로 변환하고, 모든 경우에 대한 기댓값을 구한 것이다. (1) 식의 엔트로피는 Fig. 3과 같이 확률과의 관계를 가지고 있다. 예를 들어 특정 사건이 일어나는 경우의 수가 2(E=2)일 때, 두 사건이 일어날 확률이 같다면(P=1/2) 엔트로피는 1이다. 반면 둘 중의 한 사건이 반드시 일어난다면(P=1) 엔트로피는 0이다. 그리고 한 사건이 일어날 확률이 1/4(P=1/4)이라면 엔트로피는 0.811이다. 따라서 엔트로피는 불확실성이 커질수록(P=1/2) 증가하고 불확실성이 감소할 경우(P=0, P=1) 0에 가까워진다.

이와 같은 엔트로피의 성질들을 손실값이 포함된 불완전한 데이터에 적용하는 방법은 다음과 같다. Quinlan의 ID3, C4.5, C5.0은 버전에 따라 약간의 차이는 있지만 기본 아이디어는 학습 데이터의 속성들에 엔트로피를 각각 적용하여 가장 작은 값을 가지는 속성의 순서대로 분류하며 결정트리를 구성해가는 알고리즘들[3,4]이다. 이들 중에 C4.5를

2장의 확장된 데이터 표현 방법에 맞게 변형한 알고리즘이 UChoo이다. 여기서는 손실값에 균등한 확률값을 할당하여 손실값을 보상하도록 한다. 손실값에 균등한 확률값을 할당하는 근거는 엔트로피의 성질에서 찾을 수 있다. 즉, 각 변수가 중간값을 가질 때 엔트로피는 가장 크기 때문에 각 변수의 카디너리티에 따라 균등한 값을 손실값에 할당하면 손실값이 많은 변수일수록 엔트로피가 높아져 결정 트리의 상위 노드에서 이 변수가 선택되지 않도록 한다는 것이다. 결정 트리에서 가장 많은 정보를 가지고 있는 변수가 상위 노드를 차지하게 되므로 손실값을 많이 가지고 있는 변수일수록 상위 쪽에서 밀려나도록 한다는 것이다.

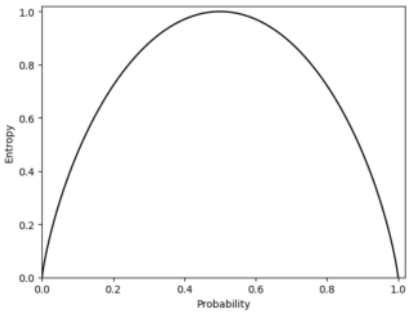


Fig. 3 Probability and Entropy

3.2 엔트로피 확률

3.1절의 균등한 확률을 손실값에 할당하는 것은 엔트로피를 기반으로 결정트리를 구성하는 분류 알고리즘에는 적합하나, 딥러닝과 같이 반복적으로 가중치를 개선해 나가는 알고리즘에서는 정보의 손실이 성능의 저하로 이어진다는 우려가 있을 수 있다[13,14]. 이에 반해 엔트로피 확률은 손실 사건에 남아있는 손실되지 않은 정보들로부터 손실값을 추정해 낼 수만 있다면 무조건 손실 속성의 카디너리티에 따라 균등확률을 할당하는 것보다는 효율적일 것이라는 아이디어에서 출발하였다[5]. 따라서 정보이론을 바탕으로 불완전한 학습 데이터로부터 손실값을 위한 확률을 추출해 내는 것이 이 절에서의 핵심이다.

이를 위해 먼저 엔트로피를 이용한 확률값을 구해 손실값에 대치하기 위해 사용하는 변수들을 정의한다. 그리고 이 변수들을 이용해 (2)식과 같은 조건부 확률값을 정의한다. 이 값은 전체 데이터 중에 손실값을 포함하지 않는 완전 데이터들 중에 변수값이 j이며 부류가 i인 사건에 대한 확률을 나타낸 것이다. 그리고 (2)식의 확률을 이용해 (3)식과 같은 성질을 가지는 손실값(V_{ij})에 채워 넣을 확률을 계산한다.

- n(·) : 함수 (·)에 적합한 사건 개수(가중치도 고려)
- i = 1,...,부류(C)의 갯수(k)
- j = 1,...,손실된 속성(A)의 카디너리티
-

$$P(C_i|A_j) = \frac{n((Class = i) \cap (attribute = j))}{n(attribute = j)} \quad (2)$$

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

$$For\ \forall_i,\ \sum_{j=1}^{cardinarity} V_{ij} = 1 \quad (3)$$

(3)식은 각 부류에 대해 손실 변수가 가질 수 있는 값들을 엔트로피를 이용해 확률로 나타낸 것이다. 다시 말해 이 식은 학습 데이터 중에 손실 부분을 제외한 완전한 데이터로 부터 엔트로피 정보를 구하고 이를 확률로 변환한 것이다. 이 식의 확률값은 불완전한 데이터로부터 구하여 검색표(lookup table)에 저장하며 손실값에 차례대로 대치한 후 분류 알고리즘에 따라 학습이 이루어지도록 한다.

3.3 결정 트리를 이용한 확률

이 방법의 기본 아이디어는 손실 데이터에 남아있는 정보들을 이용하여 손상된 부분과 가장 근접한 확률값을 찾아가는 것이다. 불완전한 데이터들이 일부 정보만이 손상된 것들이기 때문에 손상된 사건들에 아직 남아있는 많은 정보를 이용하여 유용한 정보로 복원하는 작업은 의미가 있을 것이다[7]. 이에 따라 손실값을 처리하는 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 결정 트리가 구성되어 진다.

3. 손실 데이터의 각 사건들을 차례로 2번의 결정 트리에 입력하여 가장 근접한 영역을 찾고 이 영역을 나타내는 경로의 정보를 확률로 나타내어 손실값을 대체한다.

이를 확장된 데이터 기법의 관점에서 단계별로 보면, 첫째 손실이 없는 완전한 사건들 만으로 C4.5 분류기로 분류하는 중에 결정트리가 완성된다. 그리고 이 완전한 사건들은 원-핫-인코딩으로 변환한 후 저장한다. 반면 일부가 손실된 불안전한 사건들을 결정트리에 입력하고 경로에 따라 순회하여 확률값을 구한다. 그리고 이 확률값을 손실 변수에 채워 데이터 확장 기법의 변환을 완성한다.

예를 들어 Fig. 2의 결정 트리에 {-, 1, 2} 사건이 입력되고 이를 VL 표현식으로 나타내면 [V1=?] [V2=1] [V3=2]가 된다. 이 입력값으로 결정트리를 순회하게 되는데 이때 순회 경로로 영역이 결정되고 결정된 영역에 따라 확률값이 산출될 수 있다. 즉, 루트 노드에서 V2가 1이므로 왼쪽 노드로 옮기면 다음 노드가 V1이다. 이때 V1이 손실되었으므로 이 노드에서 V1이 산출할 수 있는 확률에 따라 P(V1=1)=1/3, P(V1=2)=0, P(V1=3)=2/3가 된다. 다른 예로 {1, 3, -} 사건이 입력되었을 때 V2가 3이므로 오른쪽 노드로 이동하고 V3가 손실되어 더 이상 진전할 수 없다. 따라서 이 노드에서 V3가 가질 수 있는 확률을 구하면 P(V3=1)=2/5, P(V3=2) =3/5가 된다.

4. 실험

3가지 손실 확률을 적용한 보상의 성능 평가를 위해 UCI 기계저장소[15]에서 “Car Evolution”(1728 사건, 6변수, 4부류)와 “Balance and Scale”(625 사건, 4변수, 3부류) 데이터와 수면 중인 피실험자의 뇌파를 6단계로 수치화한 ”Sleep stage scoring“ (799 사건, 11변수, 6부류) 데이터를 사용하였다. 실험 방법은 10겹 교차 검증 방법(10-fold cross validation)을 사용했으며, 학습 데이터에서 각각 변수에 대해 4가지 손실 비율(손실률 - 5%, 15%, 30%, 45%) 만큼을 차례대로 손실시키며 결과를 알아보았다. 이러한 과정을 10번 반복한 평균결과가 변수의 손실률 별로 Table 2에 나타나 있다. 이들 결과는 앞에서 소개한 바와 같이 3가지 방법으로 손실된 변수에 손실값을 산출하여 보상한 후, Fisher의 선형분리 함수를 이용하는 SVM 알고리즘으로 학습한 결과이다. 여기서 각 변수마다 ”Uni“는 3.1절의 균등한 확률로 손실값을 할당한 방법을, ”Ent“는 3.2절의 손실된 정보를 위해 엔트로피를 이용하는 방법으로, 마지막으로 ”Prob“는 3.3절의 결정트리를 이용한 알고리즘의 결과이다. 마지막에 각 손실률 마다의 ”AVG“값 (평균값)으로 각각의 방법의 성능을 보이려 했다. 그리고 이들 평균들을 한눈에 볼 수 있도록 도표로 그린 것이 Fig. 4에 나타나 있다.

Table 2. Performance evaluation results of 3 algorithms handling missing values using extended data expression

(a) Balance & Scale data

		5	15	30	45
V1	Uni	89.66	89.36	85.47	82.68
	Ent	90.67	90.71	87.24	82.41
	Prob	90.75	90.00	89.59	85.09
V2	Uni	90.23	88.32	86.02	84.92
	Ent	91.17	89.79	89.27	85.67
	Prob	91.11	90.94	83.45	86.59
V3	Uni	89.33	86.93	83.68	81.95
	Ent	91.08	89.71	84.80	84.31
	Prob	92.11	89.84	87.53	83.65
V4	Uni	90.56	87.73	83.28	82.97
	Ent	90.94	88.89	87.60	85.14
	Prob	90.85	89.88	85.01	83.39
AVG	Uni	89.943	88.086	84.613	83.131
	Ent	90.965	89.775	87.227	84.382
	Prob	91.204	90.164	86.397	84.679

(b) Car evolution data

		5	15	30	45
V1	Uni	88.99	85.38	86.25	84.52
	Ent	89.19	87.09	86.24	83.89
	Prob	90.61	87.81	85.74	86.30
V2	Uni	88.87	85.62	88.37	85.58
	Ent	88.71	86.27	86.26	85.76
	Prob	88.65	87.09	87.74	85.75
V3	Uni	88.41	87.10	87.03	86.15
	Ent	88.43	87.92	87.61	87.35
	Prob	88.39	87.83	86.88	86.09
V4	Uni	87.94	86.34	88.54	87.58
	Ent	89.70	88.17	87.94	88.89
	Prob	88.11	86.82	87.02	87.58
V5	Uni	88.59	86.15	85.57	86.84
	Ent	87.68	87.62	86.25	86.33
	Prob	89.35	87.18	86.82	85.41
V6	Uni	88.21	86.99	85.71	85.21
	Ent	89.84	87.17	88.20	87.08
	Prob	87.32	86.59	85.16	84.12
AVG	Uni	88.501	86.263	86.912	85.979
	Ent	88.926	87.374	87.082	86.550
	Prob	88.737	87.219	86.562	85.873

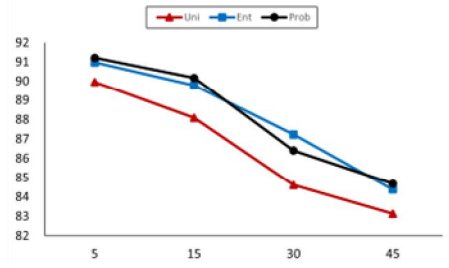
(c) Sleep Stage Scoring data

		5	15	30	45
V1	Uni	86.95	86.02	86.84	84.77
	Ent	87.75	86.91	84.98	86.97

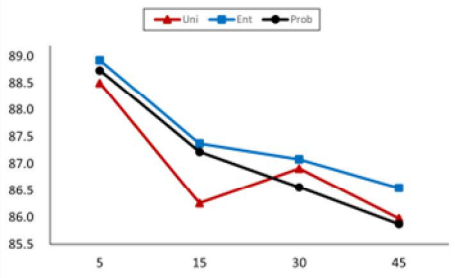
	Prob	88.05	88.16	88.77	87.19
V2	Uni	88.63	87.29	87.70	89.18
	Ent	87.60	87.85	87.87	88.34
	Prob	86.49	87.53	87.69	87.29
V3	Uni	85.64	87.47	84.30	85.16
	Ent	87.30	87.85	87.68	85.25
	Prob	87.19	86.67	86.55	84.57
V4	Uni	86.79	88.32	87.87	88.23
	Ent	88.14	87.77	85.39	88.48
	Prob	89.52	87.51	87.97	85.05
V5	Uni	87.27	87.65	87.72	86.12
	Ent	86.53	87.65	87.15	88.36
	Prob	88.62	89.70	87.03	87.39
V6	Uni	85.26	87.29	85.70	86.16
	Ent	87.04	86.86	85.73	85.98
	Prob	87.41	85.59	84.80	85.86
V7	Uni	87.65	87.33	87.33	87.44
	Ent	88.31	86.82	87.80	86.05
	Prob	87.54	89.12	85.84	88.21
V8	Uni	86.87	87.09	87.24	85.45
	Ent	88.73	88.60	87.31	87.59
	Prob	87.33	86.75	87.82	88.07
V9	Uni	88.39	87.65	87.85	85.49
	Ent	87.81	88.00	88.81	87.63
	Prob	86.49	85.88	86.58	86.44
V10	Uni	88.09	87.83	86.92	87.47
	Ent	88.72	89.01	86.96	85.77
	Prob	87.39	88.99	85.75	86.63
V11	Uni	88.45	88.00	88.25	87.15
	Ent	90.03	87.12	89.34	85.43
	Prob	88.76	87.14	86.95	86.61
AVG	Uni	87.274	87.452	87.065	86.603
	Ent	87.997	87.676	87.184	86.895
	Prob	87.709	87.549	86.887	86.664

Fig. 4의 결과에서 2가지 특이점을 발견할 수 있었다. 첫째 Balance & Scale는 사건과 변수의 수에서 다른 데이터에 비해 규모가 작은 편이다. 따라서 이 데이터의 결과는 손실률의 변화에 따라 성능 저하의 영향이 가장 큰 것으로 나타났다.

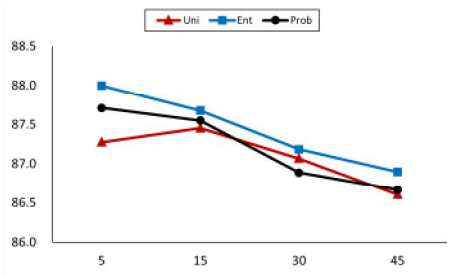
그리고 Car evolution에서도 사건의 수는 가장 많으나 변수의 수가 비교적 작아 각 변수에서의 손실이 전체 성능에 커다란 변화를 일으켰다. 그러나 Sleep Stage Scoring는 변수의 수와 사건의 수가 가장 많아서 각 변수에서의 변화와 성능의 관계가 가장 적은 것으로 나타났다. 이로서 학습 데이터에서 변수의 수가 작을수록 변수의 손실에 따르는 영향이 클 수밖에 없음을 알 수 있다. 둘째 3가지 소개된 손실 보상 알고리즘 중에서 "Uni"보다는 나머지 2 방법("Ent", "Prob")의 성능이 다소 높은 것으로 나타났다. 이것은 손실된 정보를 보다 근사하게 보상한 결과로 평가하다. 특히 "Ent"의 결과가 3가지 학습 데이터에서 모두 우수한 것으로 나타났는데 학습했던 데이터에 의존했다고 볼 수도 있지만 원 데이터에 가장 근접하게 보상한 결과로도 생각할 수 있다.



(a) Balance & Scale data



(b) Car evolution data



(c) Sleep Stage Scoring data

Fig. 4 Experiment results

5. 결론

본 논문은 불완전한 데이터를 처리하기 위해 확장된 데이터 표현 기법을 사용하는 3가지 알고리즘에 대해 살펴보고 실험 결과를 통해 성능을 비교해 보았다. 균등한 확률을 할당하는 방법은 C4.5와 같이 엔트로피를 노드 선택 기준으로 사용하는 알고리즘에서는 유용하여 좋은 결과를 얻었으나, SVM 등의 성능실험에서는 가장 낮은 결과를 나타내었다. 이는 데이터에 남아있는 정보들을 무시하고 일률적으로 같은 확률값을 할당했기 때문이라고 판단된다. 엔트로피와 결정트리를 이용한 확률로 손실

값을 보상하는 방법은 손실 사건에서 손실 변수를 제외한 나머지 정보를 이용한다는 점에서 장점을 가지며 실험에서도 다소 우수한 결과를 확인할 수 있었다.

다음 연구에서는 첫째, 3.3절에서 사용한 C4,5 대신에 SVM 분류 알고리즘을 사용했을 때 분류 선이 좌표와 평행하지 않고 자유롭게 변할 수 있어 보다 좋은 결과를 얻을 것으로 본다. 둘째, 딥러닝으로 학습한 결과와 성능 비교도 의미가 있을 것으로 본다. 셋째, 이미지 인식의 성능향상을 위해 사용되는 데이터 증대(data augmantation) 기법을 적용해 볼 수 있을 것이다. 이는 이번 실험 과정에서 일부 성능이 향상되는 가능성을 확인하였다. 마지막으로 3.2절의 엔트로피 대신에 교차 엔트로피(cross-entropy)나 쿨백-라이블러 발산(KL-divergence) 기법으로 확률식을 업그레이드할 수 있을 것이다.

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. (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

[5] J. C. Lee. (2020). Algorithms for Handling Incomplete Data in SVM and Deep Learning, *Journal of the Korea convergence society*, *11(3),* 1-7. DOI : 10.15207/JKCS.2020.11.3.001

[6] 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

[7] J. C. Lee. (2021). A data extension technique to handle incomplete data. *Journal of the Korea Convergence Society, 12(2),* 7-13. DOI : 10.15207 /JKCS.2021.12.2.007

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

[9] 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.

[10] 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.

[11] 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

[12] 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

[13] 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

[14] 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.

[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