Journal of KIIT. Vol. 19, No. 1, pp. 55-61, Jan. 31, 2021. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2021.19.1.55

데이터 이산화와 계층형 클러스터링 알고리즘을 적용한 효과적인 머신 러닝 방법 관한 연구

이 영 호*

A Study on Analytical Machine Learning Method Applying Discretization and Hierarchical Clustering Algorithm

YoungHo Lee*

요 약

최근 산업 전반에서 머신 러닝 연구에 대한 요구가 증가함과 함께 공공데이터, 사물인터넷, SNS 등이 확산되면서 다량의 데이터 확보가 가능해졌다. 이를 통해 대규모로 축적된 데이터를 머신 러닝에 널리 활용할 수 있게 됨으로써 머신 러닝 연구를 위한 제반 여건이 점차 갖춰지고 있다. 한편으로, 머신 러닝 고도화를 위해서는 고품질의 학습 데이터가 확보되어야 하고, 데이터 분석을 통해 의미있는 규칙을 추출해 낼 수 있도록 데이터 전처리를 통한 최적화와 학습 알고리즘이 필요하다. 본 논문에서는 학습 데이터 집합에 대한 전처리 과정으로 학습 데이터를 범주형으로 변환하는 이산화 기법과 유클리드 거리 기반으로 유사한 객체들을 병합 군집화하는 계층형 클러스터링 알고리즘을 적용한 머신 러닝 방법에 대해 연구한다. 최종 테스트 과정에서 계층형 클러스터링을 통해 그룹화 된 데이터들로부터 의미있는 규칙을 발견해 낼 수 있었다.

Abstract

Recently, as the demand for machine learning research has increased across the industry, public data, Internet of Things, and SNS have spread, making it possible to secure a large amount of data. Through this, data accumulated on a large scale can be widely used for machine learning, thereby gradually providing conditions for machine learning research. On the other hand, in order to advance machine learning, high-quality learning data must be secured, and optimization through data preprocessing and learning algorithms are required so that meaningful rules can be extracted through data analysis. In this paper, we study a machine learning method that applies a discrete method that converts training data into a categorical type as a preprocessing process for a training data set and a hierarchical clustering algorithm that merges and clusters similar objects based on Euclidean distance. In the final test process, we were able to discover meaningful rules from the grouped data through hierarchical clustering.

Keywords
data discretization, hierarchical clustering, machine learning, data mining

* 성균관대학교 전자전기공학부 초빙교수
- ORCID: https://orcid.org/0000-0002-2875-2517

· Received: Jan. 13, 2021, Revised: Jan. 25, 2021, Accepted: Jan. 28, 2021
· Corresponding Author: YoungHo Lee
School of Electronic and Electrical Engineering, Sungkyunkwan University, 2066, Seobu-ro, Jangan-gu, Suwon, Gyeonggi-do, Korea,
Tel.: +82-31-290-5827, Email: brainfo@skku.edu

Ⅰ. 서 론

4차 산업 혁명으로 인해 디지털화, 네트워크화가 가속화 되면서 생성되고 수집되는 데이터의 양도 급증하게 되었다. 이로 인해 다량의 데이터 분석과 처리가 필요하게 되어 빅데이터 분석과 인공지능(AI)이 4차 산업 혁명의 핵심 기술로 주목 받게 되었다.

빅데이터 분석 기법으로는 데이터 마이닝, 패턴 인식, 머신 러닝(machine learning) 등이 있으며, 머신 러닝은 인간의 학습 능력을 모델링 한 인공지능의 한 분야로, 주어진 데이터 집합을 분석하여 의미있는 규칙을 추출함으로써 기계가 자가 학습할 수 있도록 알고리즘을 구현하는 분야이다.

최근 지식정보 산업, 서비스 분야 등 산업 전반에서 머신 러닝 연구에 대한 요구가 증가하고 있는 추세이다. 그 이유는 각종 데이터의 양적 증가로 인해 수집된 데이터를 신속정확하게 유용한 정보와 지식으로 가공하여 생산해야 할 필요성이 있기 때문이다. 이렇게 생산된 정보와 지식은 기업 경영, 시장 분석이나 예측에서부터 과학 탐구와 공학 설계 등에 이르기까지 광범위한 분야에 유용하게 활용될 수 있다.

이러한 사회적, 시대적 요구와 함께 공공데이터, 사물인터넷(IoT), 사회 관계망 서비스(SNS) 등이 확산되면서 다량의 데이터 확보가 가능해졌고, 이를 통해 대규모로 축적된 데이터를 머신 러닝에 널리 활용할 수 있게 됨으로써 머신 러닝 고도화에 박차를 가할 수 있게 되었다.

머신 러닝 고도화를 위해서는 고품질의 학습 데이터(training data)가 확보되어야 하고, 학습 데이터 분석을 통해 의미있는 규칙을 추출해 낼 수 있도록 데이터 전처리를 통한 최적화, 그리고 학습 알고리즘이 필요하다.

그러나, 실제 현실 세계의 데이터베이스는 범주형과 연속형(수치형) 데이터가 혼재되어 있거나, 연속형 데이터로만 구성되어 있기도 하기 때문에, 이를 머신 러닝에 바로 적용하기에는 적합하지 않다. 이러한 데이터를 머신 러닝에 활용하기 위해서는 데이터를 동일한 유형으로 통일하기 위한 전처리 과정으로 데이터 형 변환이 이루어져야 한다.

범주형 데이터를 연속형 데이터로 변환시키는 문제의 복잡성, 그리고 연속형 데이터에 비해 범주형 데이터의 분류 규칙을 도출하기가 더 용이하다는 장점이 있다. 따라서 데이터 전처리 과정에서는 연속형 데이터를 범주형 데이터로 변환하는 기법인 이산화(discretization) 기법을 주로 사용한다[1]. 이러한 데이터 이산화 과정을 거치게 되면, 데이터가 간소화 되기 때문에 데이터 구조를 이해하기 용이해지고, 분류 알고리즘의 효율성도 향상된다[2].

본 논문에서는 학습 데이터 집합에 대한 전처리 과정으로 학습 데이터를 범주형으로 변환하는 이산화 기법과 유클리드(Euclidean) 거리 기반으로 유사한 객체들을 병합 군집화하는 계층형 클러스터링(hierarchical clustering) 알고리즘을 적용한 머신 러닝 방법에 대해 연구한다. 다량의 데이터로부터 의미있는 규칙을 체계적으로 추출해내기 위하여 이산화 기법을 이용한 데이터 전처리 과정, 데이터 집합에 대한 계층형 클러스터링 과정, 그리고 계층형 클러스터링을 통해 그룹화 된 데이터들로부터 의미있는 규칙을 추출해내는 과정을 테스트를 통해 진행한다.

Ⅱ. 관련 연구

2.1 이산화의 목적과 매핑 알고리즘

이산화란 효과적인 머신 러닝을 수행하기 위해 연속형 데이터를 범주형 데이터로 변환, 구간화 하는 데이터 전처리 방법의 하나이다[3]. 일반적으로 이산화의 목적은 데이터의 손실을 최소화하고 미지의 데이터에 대한 범주를 최대한 일반화 할 수 있는 구간을 찾는 것이다[4][5].

데이터 집합 S를 n개의 구간으로 분할하여 이산화 할 때, $_{i=1}^{n}S_i$는 다음과 같이 정의할 수 있다[6].

첫째, 분할된 각 구간(부분 집합)은 공집합이 될 수 없다($S_i \neq \varnothing$).

둘째, 분할된 각 구간의 합집합은 전체 도메인 구간 S가 되어야 한다($\bigcup_{i=1}^{n} S_i = S$).

셋째, 분할된 각 구간은 서로 겹치는 부분이 존재해서는 안 된다($s \in S_i$이면, $x \neq i$인 모든 x에 대하여 $s \notin S_x$).

이러한 정의에 따라 n개로 분할된 각 구간의 경계를 결정짓는 기준점, 즉 분할 지점(split point)의 개수는 n-1개가 존재한다는 것을 알 수 있다. 예를 들어 임의의 연속형 변수 값의 구간이 [a, b]이고, 분할 지점을 p라 할 때(a<p<b), 구간 [a, b]는 분할 지점 p에 의해 [a, p]와 [p, b]로 분할될 수 있다.

데이터 집합 S의 인스턴스를 $s(s_1, s_2, \ldots, s_n \in S)$라 하고, 인스턴스 s의 연속형 변수를 A라 하자. 그리고, 연속형 변수 A의 값을 $val_A(s)$, A의 분할 지점 집합을 $\{p_1, p_2, \ldots, p_{n-1}\}$이라 가정할 때, $val_A(s)$를 범주형 값으로 매핑(mapping)하는 함수 mapf(s, A)를 그림 1과 같이 정의할 수 있다.

$$\text{mapf(s, A)} = \begin{cases} D_1, \text{ if } val_A(s) <= p_1 \\ D_i, \text{ if } P_{i-1} < val_A(s) <= p_i \\ D_n, \text{ if } val_A(s) > p_{n-1} \end{cases}$$

그림 1. 매핑 함수의 정의
Fig. 1. A definition of the mapping function

그림 1에서 매핑 함수에 의해 범주형 값으로 변환된 결과 값이 D_i(i=1, 2, ... , n)이며, mapf(s, A)$\in \{D_1, D_2, \ldots, D_n\}$이 성립된다.

2.2 이산화 기법의 분류

이산화 기법은 이산화 과정에서 학습 데이터 집합의 인스턴스의 목적 변수(종속 변수)의 존재 여부에 따라 지도(supervised) 이산화 방식과 비지도(unsupervised) 이산화 방식으로 분류될 수 있다[7]. 만일, 학습 데이터 집합에서 목적 변수가 모호하거나 목적 변수가 존재하지 않는 경우, 비지도 이산화 방식을 적용해야 한다. 일반적으로 이산화는 데이터를 범주화 하여 궁극적으로 분류 모델을 생성하는데 사용되기 때문에 대부분 목적 변수가 존재한다[8]. 따라서, 분류 모델을 생성할 때에는 주로 지도 이산화 방식을 적용하는 사례가 많다.

지도 이산화 방식은 입력 변수(설명 변수)와 목적 변수 값 간에 상관관계를 측정하기 위해 적용하는 평가 함수의 지표에 따라 엔트로피(entropy) 방식과 카이스퀘어(chi-square) 방식 등으로 분류되며, 비지도 이산화 방식은 등 간격(equal width) 이산화 방식과 등 빈도(equal frequency) 이산화 방식 등으로 분류된다[9][10]. 등 간격 이산화는 연속형 변수 값의 전체 도메인에 대하여 동일한 간격의 구간을 갖도록 균등 분할하는 방식으로, 연속형 변수 도메인의 최대값(A_{max})과 최소값(A_{min}) 사이의 구간을 동일한 간격으로 n개만큼 균등 분할한다. 구간 간격(거리)을 d라 가정할 때, $(A_{max}-A_{min})/n$을 통해 d를 도출할 수 있다. 이에 반해, 등 빈도 이산화는 분할된 각 구간별로 동일한 개수의 데이터를 가지도록 분할하는 방식으로, 학습 데이터 집합 S가 m개의 인스턴스를 가지고 있고 n개의 구간으로 분할한다고 가정할 때, 분할된 각 구간별 인스턴스의 개수는 m/n을 통해 도출할 수 있다[11]. 이때에 두 방식 모두 이산화를 수행하기 전에 미리 적절한 구간 수(arity)가 입력되어야 한다. 즉, 도메인 구간을 몇 개로 분할할 것인지를 사전에 결정해야 한다. 이산화하고자 하는 변수에 대한 정보를 알 경우, 각 변수의 구간 수는 변수가 갖는 인스턴스 값의 분포에 따라 다양한 값으로 결정될 수 있다. 반면에, 변수에 대한 사전 정보가 없는 경우, 이산화 할 변수의 모든 구간 수는 보통 단일 값으로 결정된다.

한편으로, 구간 분할 수행 시점에 따라 이산화시킬 필요가 있는 데이터가 존재할 경우에만 변수 값을 동적으로 분할하는 동적 이산화 방식과 학습 데이터 집합의 모든 연속형 데이터를 미리 이산화시키는 정적 이산화 방식으로도 분류된다. 일반적으로 데이터 전처리 과정에서는 정적 이산화 방식이 사용된다.

일반적으로 목적 변수를 고려하지 않은 이산화 기법은 정확한 이산화 경계를 갖지 않기 때문에 데이터 손실이 많이 발생될 수 있다. 그리고, 최적의 구간 수를 결정하는 방법도 주관적으로 이루어지기 때문에 일반화 된 구간을 보장할 수 없다. 따라서, 본 논문에서는 학습 데이터 집합의 연속형 데이터를 미리 범주형으로 이산화시키는 정적 이산화 방식을 이용하고, 목적 변수가 모호하거나 존재하지 않는 문제점을 보완하기 위해 거리 값을 기반으로

유사도를 측정하여 유사한 객체들끼리 계층적으로 병합하여 클러스터링 하는 알고리즘을 적용한다.

2.3 클러스터링 알고리즘

하나의 객체가 여러 개의 속성(attribute)을 갖는다고 가정하고, 이러한 객체가 다수 존재하는 경우, 클러스터링이란 유사한 속성들을 갖는 객체들을 그룹으로 묶어 전체 객체들을 군집화 하는 것을 말한다. 클러스터링 알고리즘은 크게 계층형 클러스터링과 비계층형 클러스터링으로 분류된다.

비계층형 클러스터링의 대표적인 알고리즘으로는 사전에 클러스터의 개수를 지정해두고 그룹화 하는 K-means 알고리즘이 있다. K-means 알고리즘은 주로 연속형 데이터들을 그룹화 하는 데 사용되는 전통적인 알고리즘이다[12].

반면에, 계층형 클러스터링은 사전에 클러스터의 개수를 정하지 않고 단계적으로 서로 다른 클러스터링 결과를 제공하는 것으로, 이는 다시 응집 방식과 분리 방식으로 구분된다[3]. 응집(병합) 방식은 각 객체를 개별적인 하나의 클러스터로 간주함을 시작으로 서로 유사한 객체들을 묶어 클러스터로 만들고, 다시 유사한 객체들을 묶어 새로운 클러스터로 만들어나가는 과정을 전체 객체들이 하나의 클러스터로 군집화 될 때까지 반복하는 상향식(bottom-up) 방법이다[13][14]. 이와 반대로 분리 방식은 전체 객체들을 단일 클러스터로 간주함을 시작으로 유사성(연관성)이 떨어지는 객체들을 분리시켜 다른 독립적인 클러스터로 만들어나가는 과정을 각 객체가 개별적인 하나의 클러스터가 될 때까지 반복하는 하향식(top-down) 방법이다.

III. 알고리즘 설계

본 논문에서는 그림 2의 예시처럼 연속형 변수와 범주형 변수가 혼재하여 있고, 목적 변수가 존재하지 않는 데이터들을 입력 변수로 사용한다. 데이터 전처리 과정으로 연속형 변수를 범주형 변수로 이산화한다. 그리고, 응집(병합) 방식으로 계층형 클러스터링 알고리즘을 적용하여 의미있는 규칙들을 추출하는 테스트를 수행한다.

계층형 클러스터링 알고리즘은 그림 3과 같이 m×n개의 데이터들로 구성된 데이터 집합에 대해 초기화 작업으로 각 데이터 X_j를 개별적인 하나의 클러스터 C_i로 두고, K개의 클러스터를 생성한다. 그다음 K×(K-1)/2개의 클러스터 간에 병합을 수행하게 되는데, 이때에 거리 값이 가장 작은 2개의 클러스터를 병합한다. 그다음 다시 (K-1)×(K-2)/2개의 클러스터 간에 병합을 수행하며, 이때에도 거리 값이 가장 작은 2개의 클러스터를 하나로 병합한다. 이후, 클러스터의 개수가 1개가 될 때까지 이 과정을 반복 수행한다.

```
// D: data set
// m: number of data objects(Number of rows)
// n: number of input variables(Number of columns)
// data X={X1, X2, ... , Xm}, Xj={xj1, xj2, ... , xjn}, j=1,2,3, ... , m
// cluster C={C1, C2, ... , CK}
1 for i ← 1 to K
2     for each (Xj in D)
3         Ci ← Xj
4 repeat
5 for Ci ← C1 to CK
6     for Cj ← C1 to CK
7         if (i≠j) then search Cj with the minimum distance between Ci
8     Ci ← merge(Ci, Cj)
9 until (|C| == 1)
```

그림 3. 계층형 클러스터링 알고리즘
Fig. 3. Hierarchical clustering algorithm

	PassengerId	Survived	Pclass	Fare_1	Age_1	Family	Sex_1	IsAlone	Title_Mr	Title_Mrs
0	1	0	3	7.2500	26	2	1	0	1	0
1	2	1	1	71.2833	37	2	0	0	0	1
2	3	1	3	7.9250	22	1	0	1	0	0
3	4	1	1	53.1000	37	2	0	0	0	1
4	5	0	3	8.0500	26	1	1	1	1	0
5	6	0	3	8.4583	26	1	1	1	1	0
6	7	0	1	51.8625	41	1	1	1	1	0
7	8	0	3	21.0750	26					

그림 2. 샘플 데이터 집합 예시
Fig. 2. An example of sample data set

Ⅳ. 실 험

본 논문에서는 그림 4와 같이 데이터 이산화를 통해 데이터 전처리 과정을 거치고, 범주형으로 이산화 된 데이터들을 계층형 클러스터링 하여 군집화 한 후, 이를 통해 최종적으로 의미있는 규칙을 추출하는 테스트를 수행한다.

테스트에 이용된 데이터 집합은 Kaggle에서 제공하는 타이타닉(Titanic) 데이터 집합을 이용하였고, 그림 5와 같이 총 200개의 레코드들로 구성되어 있다. 목적 변수는 존재 하지 않으며, 입력 변수만으로 이루어진 10개의 열(column)로 구성되어 있다.

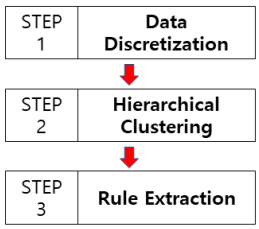

그림 4. 테스트 과정
Fig. 4. Test process

	PassengerId	Survived	Pclass	Fare_1	Age_1	Family	Sex_1	IsAlone	Title_Mr	Title_Mrs
0	1	0	3	7.2500	26	2	1	0	1	0
1	2	1	1	71.2833	37	2	0	0	0	1
2	3	1	3	7.9250	22	1	0	1	0	0
3	4	1	1	53.1000	37	2	0	0	0	1
4	5	0	3	8.0500	26	1	1	1	1	0
...	...	...	...	...	...	...	...	...	...	...
195	196	1	1	146.5208	37	1	0	1	0	0
196	197	0	3	7.7500	26	1	1	1	1	0
197	198	0	3	8.4042	26	2	1	0	1	0
198	199	1	3	7.7500	22	1	0	1	0	0
199	200	0	2	13.0000	27	1	0	1	0	0

200 rows × 10 columns

그림 5. 테스트용 데이터 집합
Fig. 5. Data set for testing

	PassengerId	Survived	Pclass	Fare_1	Age_1	Family	Sex_1	IsAlone	Title_Mr	Title_Mrs	Fare_Class	Age_Categories
0	1	0	3	7.2500	26	2	1	0	1	0	1	20대
1	2	1	1	71.2833	37	2	0	0	0	1	4	30대
2	3	1	3	7.9250	22	1	0	1	0	0	2	20대
3	4	1	1	53.1000	37	2	0	0	0	1	4	30대
4	5	0	3	8.0500	26	1	1	1	1	0	2	20대
5	6	0	3	8.4583	26	1	1	1	1	0	2	20대
6	7	0	1	51.8625	41	1	1	1	1	0	4	40대
7	8	0	3	21.0750	26	5	1	0	0	0	3	20대
8	9	1	3	11.1333	22	3	0	0	0	1	2	20대
9	10	1	2	30.0708	27	2	0	0	0	1	3	20대
10	11	1	3	16.7000	22	3	0	0	0	0	3	20대
11	12	1	1	26.5500	37	1	0	1	0	0	3	30대
12	13	0	3	8.0500	26	1	1	1	1	0	2	20대
13	14	0	3	31.2750	26	7	1	0	1	0	4	20대
14	15	0	3	7.8542	22	1	0	1	0	0	1	20대
15	16	1	2	16.0000	27	1	0	1	0	1	3	20대
16	17	0	3	29.1250	26	6	1	0	0	0	3	20대
17	18	1	2	13.0000	31	1	1	1	1	0	2	30대
18	19	0	3	18.0000	22	2	0	0	0	1	3	20대
19	20	1	3	7.2250	22	1	0	1	0	1	1	20대

그림 6. 데이터 이산화 결과
Fig. 6. Data discretization result

입력 변수 중 Fare_1과 Age_1은 연속형 데이터가 저장되어 있으므로, 범주형 데이터로 변환시켜 줄 필요가 있다. 따라서, 이를 위해 Python Jupyter Notebook 환경에서 Pandas 라이브러리를 이용하여, 그림 6과 같이 범주형 데이터로 변환시켜 줌으로써 이산화를 수행하였다.

나이와 요금과의 상관 규칙을 발견하기 위해 Matplotlib 라이브러리를 이용하여 산점도를 출력해 보았다. 그림 7과 같은 결과를 얻을 수 있었고, 나이와 상관 없이 전반적으로 저렴한 요금을 선택했다는 것 외에는 의미있는 규칙을 발견하지 못했다.

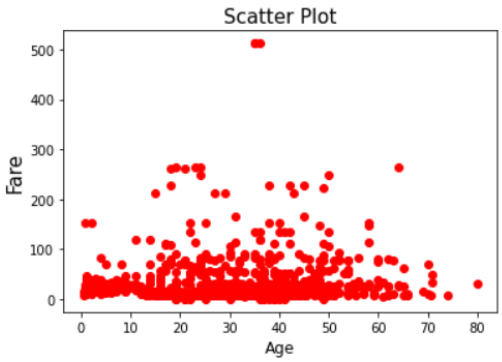

그림 7. 산점도 차트
Fig. 7. Scatter plot chart

나이와 요금에 대한 계층형 클러스터링 수행 결과를 시각화하기 위해 Matplotlib 라이브러리를 이용하였고, 그림 8과 같은 결과를 얻을 수 있었다. 이를 통해 30대 초반까지는 저가의 요금을 주로 선택하고, 30대 중반 이후부터는 고가의 요금을 다소 포함한 다양한 요금을 선택하는 경향이 있다는 규칙을 발견해낼 수 있었다.

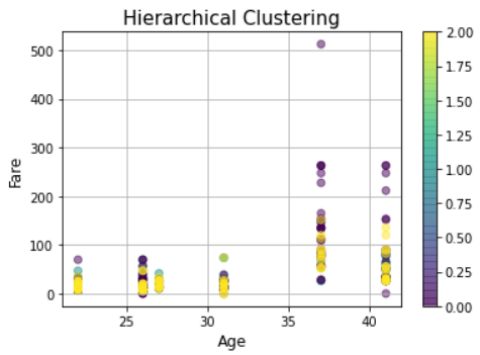

그림 8. 계층형 클러스터링 결과
Fig. 8. Hierarchical clustering result

V. 결론 및 향후 연구

본 논문에서는 학습 데이터를 머신 러닝에 효과적으로 적용할 수 있도록 Python Pandas를 이용하여 데이터 이산화를 수행하였고, 유의미한 규칙을 발견하기 위해 계층형 클러스터링 수행 결과를 Python Matplotlib로 시각화하였다. 실험 결과, 산점도를 통해서는 발견하지 못했던 유의미한 규칙을 발견할 수 있었다. 따라서, 이를 머신 러닝에 활용하여 숨은 규칙을 추출하게 함으로써 보다 스마트한 판단 및 예측이 가능해질 것으로 기대된다.

향후 연구 방향으로는 세계 주요 여행지들의 월평균 기온과 월평균 항공 비용 데이터 집합을 수집, 정제하여 학습 데이터로 만들고, 이를 기반으로 월별 최적의 여행지를 자동 추천해주는 머신 러닝 모델에 관한 설계가 될 것이다.

References

[1] Ian H. Witten and Eibe Frank, "Data mining: Practical machine learning tools and techniques [3rd edition]", Morgran Kaufmann, Jan. 2011.

[2] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features", Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, pp. 194-202, Jul. 1995.

[3] Jiawei, Han, Micheline Kamber, and Jian Pei, "Data mining: Concept and techniques[3rd edition]", Morgan Kaufmann, 2012.

[4] B. S. Choi, H. J. Kim, and W. O. Cha, "A comparative study on discretization algorithms for data mining", Communications for Statistical Applications and Methods, Vol. 18, No. 1, pp. 89-102, Jan. 2011.

[5] C. J. Tsai, C. I. Lee, and W. P. Yang, "A discretization algorithm based on class-attribute contingency coefficient", Information Sciences, Vol. 178, No. 3, pp. 714-731, Feb. 2008.

[6] T. Elomaa and J. Rousu, "General and efficient

multisplitting of numerical attributes", Machine Learning, Vol. 36, pp. 201-244, Sep. 1999.

[7] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization : An enabling technique", Data Mining and Knowledge Discovery, Vol. 6, pp. 393 -423, Oct. 2002.

[8] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank, "Classifier chains for multi-label classification", Machine Learning, Vol. 85, No. 3, pp. 333-359, Jun. 2011.

[9] Sotiris Kotsiantis and Dimitris Kanellopoulos, "Discretization techniques: A recent survey", GESTS International Transactions on Computer Science and Engineering, Vol. 32, No. 1, pp. 47-58, 2006.

[10] Kavita Das and O. P. Vyas, "A suitability study of discretization methods for associative classifiers", International Journal of Computer Applications, Vol. 5, No. 10, pp. 46-51, Aug. 2010.

[11] David K. Y. Chiu, Andrew K. C. Wong, and Benny Cheung, "Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis", Knowledge Discovery in Databases, pp. 125-140, 1991.

[12] X. Wu, V. Kumar, J. R. Quinlan, and etc., "Top 10 algorithms in data mining", Knowledge and Information Systems, Vol. 14, pp. 1-37, Dec. 2007.

[13] Johnston, B., Jones, A., and Kruger. C, "Applied Unsupervised Learning with Python", Packt, May 2019.

[14] I. Jonyer, D. J. Cook, and L. B. Holder, "Graph-based hierarchical conceptual clustering", Journal of Machine Learning Research, Vol. 2, pp. 19-43, Oct. 2002.

저자소개

이 영 호 (YoungHo Lee)

2013년 8월 : 수원대학교 컴퓨터학과 이학박사

2020년 3월 ~ 현재 : 성균관대학교 전자전기공학부 초빙교수

2020년 6월 ~ 현재 : CGSIT(주) 이사

관심분야 : AI, 빅데이터, IoT