한국 퍼지 및 지능시스템 학회 2003, Vol. 13, No. 4, pp. 391-396

데이터 분포를 고려한 연속 값 속성의 이산화

Discretization of Continuous-Valued Attributes considering Data Distribution

이상훈 · 박정은 · 오경환
Sanghoon Lee, Jung-eun Park and Kyung-whan Oh

서강대학교 컴퓨터학과

요 약

본 논문에서는 특정 매개변수(parameter)의 입력 없이 속성(attribute)에 따른 목적속성(class)값의 분포를 고려하여 연속형(continuous) 속성 값을 범주형(categorical)의 형태로 변환시키는 새로운 방법을 제안하였다. 각각의 속성에 대해 목적속성의 분포를 1차원 공간에 사상(mapping)하고, 각 목적속성의 밀도, 다른 목적속성과의 중복 정도 등의 기준에 따라 구간을 군집화 한다. 이렇게 생성된 군집들은 각각 목적속성을 예측할 수 있는 확률적 수치에 기반한 것으로, 각 속성이 제공하는 정보의 손실을 최소화 하는 이산화 경계선을 갖고 있다. 제안된 데이터 이산화 방법의 향상된 성능은 *C4.5* 알고리즘과 *UCI Machine Learning Data Repository* 데이터를 사용하여 확인할 수 있다.

Abstract

This paper proposes a new approach that converts continuous-valued attributes to categorical-valued ones considering the distribution of target attributes(classes). In this approach, it can be possible to get optimal interval boundaries by considering the distribution of data itself without any requirements of parameters. For each attributes, the distribution of target attributes is projected to one-dimensional space. And this space is clustered according to the criteria like as the density value of each target attributes and the amount of overlapped areas among each density values of target attributes. Clusters which are made in this ways are based on the probabilities that can predict a target attribute of instances. Therefore it has an interval boundaries that minimize a loss of information of original data. An improved performance of proposed discretization method can be validated using C4.5 algorithm and UCI Machine Learning Data Repository data sets.

Key Words : Discretization, Data Distribution, Density based Clustering, Decision Tree

1. 서 론

기계학습(machine learning) 알고리즘에 적용되는 실세계 데이터의 속성(attribute)은 연속형(continuous)과 범주형(categorical)의 혼합된 형태를 가지고 있다. 하지만 대부분의 기계학습 알고리즘은 한 가지 형태의 데이터만을 다룰 수 있기 때문에, 이러한 데이터를 기계학습 알고리즘에 적용시키기 위해서는 데이터 속성의 형 변환이 요구된다. 일반적으로 범주형 값을 연속형으로 변환시키는 문제의 복잡성, 그리고 범주형 값이 분류 규칙(classification rule)을 도출하기에 더 용이하다는 장점으로 인해, 연속형을 범주형으로 변환하는 기법인, 이산화(discretization) 방법을 많이 사용한다[1].

이산화를 하는 데 있어 중요한 기준은 크게 정보의 손실을 최소화 하는 것, 그리고 미지의 데이터에 대한 범주 값의 일반성을 최대화 하는 것의 두 가지로 나뉜다. 그러나 이 두 기준은 서로 상충되는데, 그것은 이산화 구간의 수가 많아질수록 정보의 손실은 적어지지만, 반대로 각 구간의 학습 집합(training set)에 대한 종속성이 증가(over-fitting)하여, 그 일반성이 감소하기 때문이다. 따라서 원래 데이터의 정보를 유지하면서 동시에 일반화된 대표 값을 산출할 수 있는 적절한 수준의 이산화 구간을 찾는 것이 이산화 알고리즘의 주된 목적이다.

기존의 방법들은 보통 엔트로피(entropy), 카이스퀘어 통계학(chi-square statistics)등을 사용하여 각 속성의 이산화된 값과 목적속성(class)간의 상관관계를 구한다. 그리고 이 값이 최대로 되는 구간을 이산화 구간으로 결정한다. 그러나 이들 대부분은 초기 구간을 결정하는 과정, 그리고 생성된 이산화 구간을 병합(일반화)하는 과정에 필요한 임계값(threshold)으로 매개변수(parameter)를 요구한다[2][3][4]. 보통 이러한 매개변수의 값에 따라 이산화 알고리즘의 성능은 큰 영향을 받으므로 최적의 매개변수를 찾는 것은 결국 또 하나의 최적화 문제를 야기한다. 따라서 본 논문에서는 매개변수의 고려 없이 최적의 이산화 구간을 결정할 수 있는 새로운 이산화 방법을 제안한다. 각 속성에 따른 목적속성의 밀도, 분포에 따라 1차원 상에서 데이터를 군집화(clustering)하고, 이 때 결정되는 군집의 경계선에 따라 이산화

접수일자 : 2003년 3월 25일
완료일자 : 2003년 6월 25일
본 연구는 과학 기술부 주관 뇌신경 정보학 사업에 의해 지원 되었음.

경계선을 결정한다. 이 결정 과정에 데이터 분포만이 고려되기 때문에, 제안한 방법은 최적화(optimization)과정 없이 하나의 프로세스(process)로 일반화된 이산화 구간을 얻을 수 있다. 본 논문은 다음과 같이 구성되어 있다. 2절에서는 간략한 이산화 방법을 소개하고, 이와 관련된 기존 연구를 제시한다. 3절에서는 군집화와 이산화의 유사성을 알아보고, 이를 토대로 밀도기반 이산화 방법의 이론, 알고리즘을 소개한다. 그리고 4절에서는 실험을 통해 제안한 알고리즘의 성능을 평가하고, 마지막으로 5절에서는 결론 및 향후 과제에 대해서 논의한다.

2. 관련 연구

2.1 이산화의 정의

연속형 속성 값을 이산화 하는 문제는 다음과 같이 정의될 수 있다. 구간 [a,b]를 연속형 속성 A가 갖을 수 있는 값의 도메인(domain)이라고 하자. 이 때 이 구간을 m개의 중복되지 않는(disjoint) 구간으로 나눈다고 하면 다음과 같은 구간을 얻을 수 있다. $[a,c_1],[c_1,c_2],\cdots,[c_{m-1},b]$, $(a<c_1<c_2<\cdots<c_{m-1}<b)$. 각 구간에 속하는 값들을 m개의 범주 값으로 사상(mapping)하는 함수를 정의 할 수 있고, 그 사상 과정을 이산화, 그리고 사상된 결과 값을 이산화된 범주 값이라고 한다. 사상함수 DVA(inst)는 다음과 같이 정의된다.

$$DV^A(inst) = \begin{cases} V_1^A & if \quad a \le V^A(inst) \le c_1 \\ V_j^A & if \quad c_{j-1} < V^A(inst) \le c_j \\ V_k^A & if \quad c_{k-1} < V^A(inst) \le b \end{cases}$$

여기서 i=1,2,$\cdots$,m-1이다. V^A(inst)는 인스턴스 inst의 속성 A가 갖는 원래의 연속형 값이고, DV^A(inst)는 그것의 이산화된 값이다. 이산화 과정에서는 이러한 사상 함수를 얻기 위해 최적의 이산화 경계선 집합(cut point set), $\{c_1,c_2,\cdots,c_{m-1}\}$을 구한다. 그림 1은 이산화 과정의 간단한 예를 보이고 있다. 그림에서 화살표는 목적속성 정보를 최대한 유지시킬 수 있는 이산화 경계를 나타내며, 점선은 그것의 일반화된 이산화 경계이다.

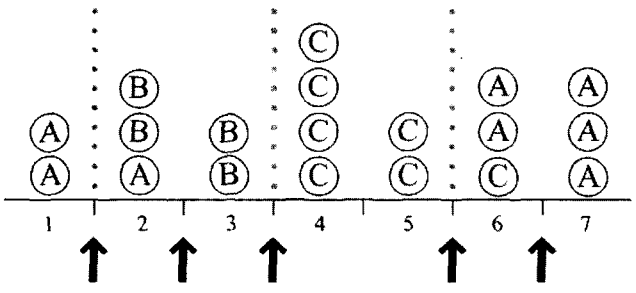

그림 1. 이산화 과정의 예

Fig. 1. An Example of Discretization Process

2.1 이산화 방법의 분류

이산화 방법은 크게 이산화 과정에서 목적 속성 값을 고려하는지 아닌지 여부에 따라 다음 두 가지로 분류될 수 있다[1].

1. **목적속성에 독립적인 방법(unsupervised method)**: 일정한 간격으로 구간을 정하는 *equal-width intervals* 방법, 일정한 데이터 빈도로 구간을 정하는 *equal- frequency intervals* 방법 등이 있다.
2. **목적속성에 의존적인 방법(supervised method)**: 속성 값과 목적속성 값과의 상관관계를 구하기 위해 사용된 평가 함수(evaluation function)에 따라 엔트로피 방법, 카이스퀘어 방법, 러프집합(rough set theory) 방법 등이 있다.

일반적으로 목적속성을 고려하지 않은 방법은 정확한 이산화 경계를 갖지 않기 때문에 손실되는 정보의 양이 많다. 그리고 최적의 구간 수를 결정하는 방법 또한 주관적으로 이루어지기 때문에 결정된 구간의 일반성을 보장할 수 없다. 따라서 이러한 문제점을 보완하기 위한 목적속성에 의존적인 이산화 방법들이 연구되어 왔다.

2.2 이산화 방법에 관한 기존의 연구

R. Kerber의 *ChiMerge* 알고리즘은 x^2 통계학에 기반해서 연속형 속성을 이산화시킨다[2]. *ChiMerge* 알고리즘은 크게 초기화와 병합의 두 단계로 구성되어 있는데, 먼저 1)초기화 단계에서는 각 속성 값을 정렬하고, 속성의 개별 값 자체를 이산화 경계로 하는 초기 구간을 생성한다(즉, 최대로 세분화된 구간을 생성). 그리고 2)병합단계에서는 모든 구간들에 대해 인접 구간과의 x^2값을 구하고, 그 값이 가장 작은 구간을 병합한다. 이 때, 병합은 모든 구간의 x^2값이 x^2-*threshold*인 α보다 작지 않을 때까지 반복되게 된다. 여기서 α는 종료조건을 결정짓는 매개변수로, 주관적으로 입력되는 값이다. 너무 큰 α값은 과도한 병합이 발생하여 구간의 수가 지나치게 적어지게 되고(over-discretization), 반대로 너무 작은 α값은 병합이 적게 수행되어 구간의 수가 지나치게 많아지게(under-discretization)된다. 따라서 적절한 α값을 찾는 것이 *ChiMerge* 알고리즘의 성능을 결정짓는 중요한 요인이다.

H. Liu는 앞선 *ChiMerge* 알고리즘의 문제점을 보완하기 위해 매개변수 α값을 자율적으로 결정할 수 있는 *Chi2* 알고리즘을 제안하였다[3]. *Chi2* 알고리즘의 기본적인 틀은 *ChiMerge* 알고리즘과 동일하지만, α값을 자율적으로 결정지을 수 있다는 점, 그리고 개별 속성 별로 서로 다른 α값을 사용한다는 점에서 *ChiMerge* 알고리즘과 차이점을 갖는다. 알고리즘은 크게 두개의 단계로 구성되어 있다. 1)첫 번째 단계에서는 *ChiMerge* 알고리즘을 루프(loop)로 씌워 반복하며, 적절한 α값을 찾는다(generalized version of *ChiMerge*). 이때, 큰 값에서 감소하며 적절한 α값을 탐색하는데, 종료 조건으로는 inconsistency rate δ를 사용한다. 이렇게 결정된 α값을 초기값으로 두 번째 단계가 수행된다. 2)두 번째 단계에서는 개별 속성마다 최적화된 x^2-*threshold*를 갖도록 개별 속성별로 α값을 조정한다(finer process). 이 과정은 과정 1)과 비슷한 방법으로 진행되며, 각 속성의 초기 α값은 모두 과정 1)에서 생성된 α값으로 시작된다. 비록 α에 비해 δ값을 다루는 것이 더 간단하지만, *Chi2* 알고리즘 역시 종료조건을 결정하기 위해 매개변수를 요구한다는 단점을 갖는다.

앞서 제시한 두 방법과 같이 대부분의 목적 속성에 의존적인 이산화 알고리즘은 병합의 종료 조건을 결정하기 위해 매개변수 값을 요구한다. 그런데 이러한 매개변수의 값은 데이터의 특성에 따라 결정 되는 것이기 때문에 이 값의 주관적인 결정은 해당 이산화 알고리즘의 최적화된 성능을 보장하지 못한다. 따라서 이러한 주관적인 매개변수 없이 데이터

분포를 통해 자율적으로 이산화 구간을 결정지을 수 있는 방법이 요구된다.

3. 밀도 기반의 이산화 알고리즘 DENDIS algorithm : DENsity based DIScretization algorithm

연속된 속성값의 도메인(domain)을 이산화된 m개의 구간으로 자르는 문제는 목적속성의 분포에 따라 속성값의 도메인을 m개의 군집으로 군집화 하는, 군집화의 특별한 경우로 해석할 수 있다. 이산화 문제에서 각각의 이산화 구간은 최대의 유사성(similarity)을 가진 목적 속성의 분포를 갖도록 조정되어야 하며, 이러한 기준은 군집 내 유사성(intra-similarity)을 그 평가 척도로 삼는 군집화의 기본 개념에 대응될 수 있다. 따라서 군집화 방법을 사용해 이산화 구간의 경계선을 찾는 것이 가능하다.

본 논문에서 제안한 밀도 기반의 이산화 알고리즘(*DENDIS algorithm*)은 이러한 군집화 방법에 기반하여 이산화를 수행한다. 일반적인 군집화 문제에서는 군집화 하고자 하는 인스턴스(instance)에 대한 목적속성의 정보가 은폐되어 있거나, 또는 아예 목적속성 자체가 존재하지 않는다. 따라서 단순히 인스턴스의 거리(distance)나 밀도(density)등과 같은 측도(measure)만을 사용하여 유사성을 측정한다. 그러나 이와 달리 이산화 구간을 결정하는 문제에서는 목적속성에 관한 정보가 구간을 결정하는 데에 결정적인 요소이기 때문에 각 구간의 목적속성에 관한 정보를 유지해야 한다. 따라서 군집 내 유사성을 판단하기 위한 기준으로 분포의 거리나 밀도 이외에 목적속성 밀도의 비율이 추가로 고려되어야 한다. 이러한 이유로 *DENDIS* 알고리즘에서는 유사성을 판단하기 위한 측도로 다음의 두 가지 기준을 사용한다.

1) 목적속성의 밀도
2) 해당 밀도에서 각 목적속성이 차지하는 비율의 순위값

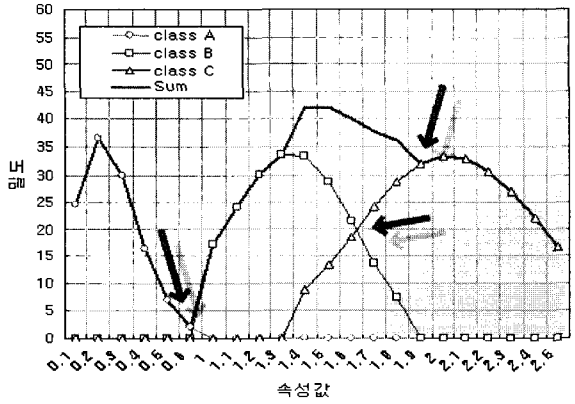

그림 2. 목적속성 분포의 예
Fig. 2. An Example of Class Distribution

밀도를 통해서는 목적속성의 분포가 비교적 명확히 구분되어 있는 지점을 군집화 할 수 있으며, 각 목적속성 밀도 비율의 순위 값을 통해서는 분포가 섞여있는 구간에 대해 목적속성의 정보 손실을 최소화하는 군집을 결정 할 수 있다. 그림 2에 이러한 두 가지 기준을 적용한 예가 나타나 있다. 그림에서 오른쪽과 왼쪽의 화살표는 목적 속성의 밀도 값을 통해 군집화 할 수 있는 지점을 나타내며, 가운에의 화살표는 각 목적속성이 차지하는 비율의 순위값을 통해 군집화 할 수 있는 지점을 나타낸다. *DENDIS* 알고리즘은 이러한 유사성 측도로 각 속성에 대해 군집화 하고, 그 군집의 경계를 이산화 경계선으로 사용한다.

알고리즘은 다음과 같이 크게 세 개의 처리 단위로 구성되어 있다. 첫 번째 단계에서는 각 지점의 밀도함수를 구하고, 두 번째 단계에서는 이 밀도의 지역 최소값을 통해 초기 이산화 경계를 생성한다. 마지막 단계에서는 앞서 생성된 구간 내에서 밀도 비율을 기준으로 구간을 세분화 시킨다.

3.1 목적속성의 분포를 반영한 밀도 함수

밀도 기반의 군집화 방법은 군집의 밀도가 주변의 밀도보다 높다는 사실에 근거하여 군집화를 수행한다. 대표적인 밀도 기반 군집화 알고리즘 중 하나인 *DENCLUE* 알고리즘은 밀도함수(density function)를 통해 n-차원 속성 공간의 밀도를 구하고, 그 값을 이용해 군집화를 수행한다[5]. 밀도 함수는 해당 좌표의 밀도 값을 구하는 함수로 다음과 같이 정의된다.

$$f_{Gauss}^{D}(x) = \sum_{i=1}^{n} e^{-\frac{d(x,x_i)^2}{2\sigma^2}} \qquad \text{식(1)}$$

이 때 $f_{Gauss}^{D}(x)$는 x에서의 밀도 값이고, n은 전체 인스턴스의 수, x_i는 좌표의 i번째 인스턴스, $d(x,x_i)$는 x와 x_i 사이의 유클리드 거리(Euclidian distance)이다.

전체 구간에 대한 목적속성의 밀도를 각각의 목적속성 값과 관계없이 전체적으로 구할 경우, 각 목적속성의 값에 따른 밀도 변화는 전체 밀도 값에 잘 반영되지 않는다. 따라서 *DENDIS* 알고리즘에서는 전체 구간의 밀도를 구하기 위해 각 목적속성의 밀도를 독립적으로 구한 후 각각을 누적한 값을 사용한다. 개별 목적속성 C의 밀도를 구하는 식은 식(2)와 같고, 전체 누적밀도를 계산하는 식은 식(3)과 같다.

$$f_{Gauss}^{D_C}(x) = \sum_{i=1}^{n_C} e^{-\frac{d(x,x_i)^2}{2\sigma_C^2}} \qquad \text{식(2)}$$

$$f_{Gauss}^{D_{total}}(x) = \sum_{C=1}^{k} f_{Gauss}^{D_C}(x) \qquad \text{식(3)}$$

식(2)는 전체 분포에서 목적속성 C값의 분포만을 독립적으로 고려했을 때의 밀도 값을 나타내며, 식(3)은 식(2)의 밀도 값들을 누적, 즉 개별 목적속성(k개)의 밀도 값들을 누적한 값이다. 만약 각 목적속성의 분포가 명확히 구분되어 있다면, 식(3)의 값은 그 구분된 지점에서 밀도가 낮아지게 되어 지역최소값(local minima)을 갖게 된다. 반대로 여러 목적속성 값이 섞여 있을 경우에는 그 분포들이 누적되어 큰 밀도 값을 갖게 된다.

3.2 지역 최소값을 통한 이산화 경계 생성

계산된 밀도 값을 사용하여 군집화 하는 방법에는 크게 두 가지가 있는데, 그것은: 1)지역최대값(local maxima)을 갖는 점들을 군집의 중심(cluster center)으로 선택하고 그

점을 기반으로 군집화를 수행하는 방법, 2)임계값 ξ이상의 밀도 값을 갖는 연속된 점들을 군집화하는 방법이다.

본 논문에서 제안한 *DENDIS* 알고리즘은 탐색 공간이 1차원이기 때문에 일반적인 n-차원 공간에서와는 다른 방법을 사용하여 군집화를 수행할 수 있다. 그것은 군집의 중심으로부터 군집의 경계를 결정하는 것이 아닌, 지역최소값을 통해 직접적으로 경계를 결정하는 것이다. 이러한 방식으로 군집을 결정할 경우 계산상의 간결성 뿐 아니라 군집의 수가 자율적으로 결정된다는 장점을 갖는다. 군집의 경계가 이산화 경계와 일대일로 대응되기 때문에, 지역최소값을 사용하여 군집을 결정하는 것은 직접 이산화 경계점으로 지역최소값을 선택하는 것과 동일한 의미를 갖는다. 따라서 *DENDIS* 알고리즘은 우선 전체 누적 밀도가 지역최소값을 갖는 점을 초기 이산화 경계선으로 결정한다.

3.3 밀도 비율을 통한 이산화 경계 생성

3.2절에서 목적속성의 분포가 명확히 구분되는 점은 지역최소값을 통해 결정될 수 있음을 보았다. 그런데, 지역최소값에 의해서는 명확한 분포에 대해서만 이산화 경계를 결정지을 수 있을 뿐 그 분포가 섞여있는 구간에 대해서는 이산화 경계를 생성하지 못한다. 이러한 이유로 앞서 생성된 이산화 구간은 정보의 손실을 초래하는 큰 구간을 포함할 수 있다. 따라서 손실되는 정보의 양을 줄이기 위해 이러한 구간을 좀 더 세분화하는 과정이 필요하다.

각 목적속성 밀도 비율에 따라 구간을 세분화 하면, 목적속성의 값을 예측할 수 있는 확률적 수치가 보존된 구간을 얻을 수 있다. 일반적으로 범주형 속성값을 다루는 분류 학습 알고리즘은 목적속성 분포의 미세한 변화보다 어떤 목적속성이 다른 목적속성보다 더 주도적으로 분포하는가에 따라 분류 규칙을 생성시킨다. 즉, 각 목적속성 비율의 순위 값이 규칙 생성에 결정적인 영향을 미치게 된다. 따라서 *DENDIS* 알고리즘은 이러한 목적속성 밀도 비율의 순위 값을 판단하기 위해 앞서 생성된 모든 구간에 대해 각 목적속성 밀도 값이 교차하는 점을 찾는다. 개별 목적속성 밀도 값이 교차할 때, 그 점을 기준으로 양 쪽은 서로 다른 목적속성의 주도적인 영향을 받기 때문에 이러한 점들을 경계로 구간을 나누는 것은 정보 손실을 최소화하는 일반화된 이산화 구간을 보장한다.

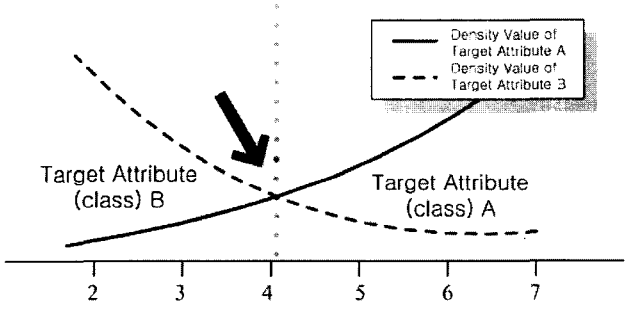

그림 3. 목적속성 밀도 값 교차 지점의 예
Fig. 3. An Example of Crossing Point of Target Attributes' Density Value

그림 3에 개별 목적속성의 밀도 값이 교차하는 지점에 대한 예가 나와 있다. 그림에서 좌표 값 4 이하의 구간에서는 목적속성 B의 비율이 A의 비율보다 더 높기 때문에 B가 해당 구간에 대해 주도적인 영향을 갖고, 반대로 4 이상의 구간에서는 A가 주도적인 영향을 갖는다.

지금까지 설명한 방법을 간략한 알고리즘으로 표현하면 그림 4와 같다. 알고리즘은 크게 밀도 계산, 지역최소값 검사, 각 밀도의 교차점 검사 부분으로 구성되어 있다.

DENDIS algorithm

```
For each continuous attribute a {
    For each target class c {
        calculate_each_density d(c,a);
        //density of each class
    }
    total_den(a):=TOTAL(d(c,a));
    //total density of all class
}
For each continuous attribute a {
    cut_point set(a):=LOCAL_MIN(total_den(a));
    //step1: create initial cut point set
    For each cut_point set(a) {
        ADD(set(a),CROSS(d(c,a));
        //step2: add cut points
    }
}
```

그림 4. 간략화한 *DENDIS* 알고리즘
Fig. 4. A Simplified *DENDIS* Algorithm

4. 실험 및 결과

제안한 *DENDIS* 알고리즘의 성능을 검증하기 위해 *UCI Machine Learning Data Repository* [6]의 데이터를 통해 예측 정확도를 측정하였다. 이산화의 결과가 얼마나 원래의 정보를 잘 보존하고 있는가, 그리고 얼마나 적절하게 일반화된 구간을 산출하고 있는가 하는 정도는 범주형 분류 알고리즘(본 실험에서는 *C4.5* 알고리즘)을 통해 그 예측 정확도를 측정함으로써 평가가 가능하다. 즉, 서로 상반되는 관계에 있는 이 두 기준의 적절한 지점은 예측 정확도로서 반영되기 때문이다. 실험은 Iris, Breast cancer, Heart diseases, Balance 등의 데이터를 사용해 수행되었다. 각 데이터의 특성은 표 1과 같다.

표 1. 실험 데이터의 특성
Table 1. Databases Characteristics

Database	Number of Instances	Attributes	Classes	Majority Class(%)
Iris	150	4	3	33.3%
Breast cancer(W)	699	9	2	65.5%
Heart diseases(H)	294	13	5	63.9%
Balance	625	4	3	46.1%

각각의 데이터를 학습 집합(training set) 60%와 검증 집합(validation set) 40%로 분리하였으며, 이 때 편중성(biasness) 문제를 제거하기 위해 각 학습 집합과 검증 집합의 목적속성 비율을 전체의 데이터와 동일한 비율이 되도록 임의 추출하였다.

의사결정나무(decision tree)는 범주형 데이터만을 처리할 수 있는 분류 방법이지만, 이를 구현한 알고리즘 중 하나인 *C4.5* 알고리즘은 연속형 속성값을 처리하기 위한 *binarization*이라는 이산화 방법을 알고리즘 안에 포함하고 있어 연속형 속성 값을 갖는 데이터를 바로 처리할 수 있다. 본 실험에서는 제안한 알고리즘의 성능 비교를 위해 원래의 연속형 값과 *DENDIS* 알고리즘을 통해 나온 범주형 값을 각각 *C4.5* 알고리즘을 통해 학습(training)한 후, 그 예측 정확도를 측정하였다. *C4.5*에서 사용한 *binarization* 방법은 다른 이산화 방법과 비교했을 때, 거의 비슷한 정도의 정확도를 보여주기 때문에 이를 통한 비교로 제안한 방법의 성능을 검증할 수 있다[7]. 그림 5에 *C4.5*를 사용한 예측 정확도가 나타나 있다. 그래프의 막대 옆에 표시된 정확도는 제안한 알고리즘의 예측 정확도를 나타낸다.

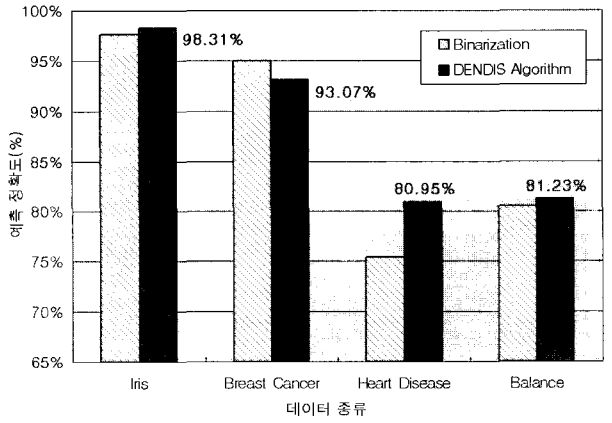

그림 5. C4.5의 예측 정확도
Fig. 5. Prediction Accuracies of C4.5 algorithm

그림에서 볼 수 있는 것처럼 *DENDIS* 알고리즘을 통해 이산화된 값은 복잡한 최적화 과정이 없었음에도 불구하고 전반적으로 좋은 예측 정확도를 보이고 있다. Breast Cancer 데이터를 제외한 나머지 데이터들에 대해 모두 *binarization* 방법보다 높은 예측 정확도를 보이고, 특히 Heart disease 데이터에 대해서는 5%이상의 정확도 향상을 볼 수 있다. 그 이유는 이산화 과정에서 목적속성을 예측하기 위해 필요한 속성값의 정보를 각 이산화 구간이 적절히 분리해 주었기 때문이다. 이 때 데이터의 분포가 비교적 잘 분리되어 있는 영역 뿐 아니라 그 분포가 섞여 있는 영역에 대해서도 각 목적속성 값의 밀도 순위를 고려해줌으로 인해 정보의 손실을 줄일 수 있는 이산화 경계를 결정지을 수 있었다.

5. 결론 및 향후 연구과제

본 논문에서는 연속형 속성 값을 범주형 값으로 변환시키기 위한 새로운 이산화 방법을 제안하였다. 기존의 평가함수 기반 이산화 방법은 보통 구간 병합을 통해 최적화를 수행하므로, 최적의 구간을 결정하기 위해 매개변수를 요구한다는 단점을 갖는다. 본 논문에서는 이러한 문제를 해결하기 위해 데이터 분포만을 통해 적절한 이산화 경계를 결정지어 줄 수 있는 새로운 알고리즘을 제안하였다.

제안한 밀도 기반의 이산화 알고리즘 (*DENDIS algorithm*)은 목적속성의 밀도와 그 분포 비율을 기준으로 이산화 경계를 결정한다. 이 과정에서 구간을 조정하는 단계 없이 최적의 이산화 구간이 결정되기 때문에 어떠한 매개변수 값도 요구되지 않는다. 실험 결과를 통해 제안한 알고리즘이 적절한 이산화 구간을 산출하고 있음을 확인할 수 있었다. 그러나 목적 속성의 수가 많을 경우, 또는 전체 구간에 대해 목적속성의 분포가 혼재할 경우에 구간이 과도하게 많이 생성된다는 점, 그리고 어떤 목적속성의 분포가 전체에서 적은 부분을 차지할 때, 이 값이 전체밀도에 잘 반영되지 않는다는 점 등은 향후 연구되어야 할 과제이다. 또한, 더 많은 데이터에 대해, 그리고 다양한 이산화 방법과의 비교 실험을 통해 제안한 알고리즘의 좀 더 객관적인 성능 평가가 이루어져야 할 것이다.

참 고 문 헌

[1] Ian H. Witten, Eibe Frank, "Data Mining", Morgan Kaufmann Publishers, 2000, page(s): 238-246
[2] Ren-Pu Li, Zheng-Ou Wang, "An entropy-based discretization method for classif-ication rules with inconsistency checking", Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference, On page(s): 243- 246
[3] R. Kerber. "ChiMerge: Discretization of numeric attribute." In Proc. Tenth National Conf. on Artificial Intelligence (AAAI-92), San Jose, CA, 123-127, 1992.
[4] H. Liu, R. Setiono, "Feature selection via discretization", IEEE Transactions on Knowledge and Data Engineering, vol.9, page(s): 642-645, 1997
[5] J. Han and M. Kamber, "Data Mining Conceip and Techniques", Morgan Kaufmann Publishers, 2001, page(s): 363-369
[6] http://www.ics.uci.edu/~mlearn
[7] T. Elomaa, J. Rousu, "General and Efficient Multisplitting of Numerical Attributes", Kluwer Academic Publishers, 1999

저 자 소 개

이상훈(Sanghoon Lee)
2002년 : 서강대학교 컴퓨터학과 졸업(학사)
2002년~현재 : 동 대학원 컴퓨터학과 석사과정

관심분야 : 기계학습, 데이터마이닝, 인지과학

Phone : 02-703-7626
Fax : 02-704-8278
E-mail : sadclan@ailab.sogang.ac.kr

박정은(Jung-eun Park)
2001년 : 성공회대학교 컴퓨터학과(학사)
2003년 : 서강대학교 컴퓨터학과(공학석사)
2003년~현재 : 서강대학교 대학원 컴퓨터학과 박사과정

관심분야 : 데이터마이닝, 시맨틱웹, 기계학습
Phone : 02-703-7626
Fax : 02-704-8278
E-mail : fayemint@ailab.sogang.ac.kr

오경환(Kyung-Whan Oh)
1978년 : 서강대학교 수학과 졸업(학사)
1985년 : Florida State University Computer Science(공학석사)
1988년 : Florida State University Computer Science(공학박사)
1989년~현재 : 서강대학교 컴퓨터학과 교수

관심분야 : 퍼지로직, 인공지능, 다중에이전트
Phone : 02-703-7626
Fax : 02-704-8278
E-mail : kwoh@ccs.sogang.ac.kr