Journal of the Society of Korea Industrial and Systems Engineering
Vol, 29, No. 2, pp.13－19, June 2006.

엔트로피 기반 분할과 중심 인스턴스를 이용한 분류기법의 데이터 감소

손승현 · 김재련

한양대학교 산업공학과

Data Reduction for Classification using Entropy-based Partitioning and Center Instances

Seung-Hyun Son · Jae-Yearn Kim

Industrial Engineering, Hanyang University

The instance-based learning is a machine learning technique that has proven to be successful over a wide range of classification problems. Despite its high classification accuracy, however, it has a relatively high storage requirement and because it must search through all instances to classify unseen cases, it is slow to perform classification. In this paper, we have presented a new data reduction method for instance-based learning that integrates the strength of instance partitioning and attribute selection. Experimental results show that reducing the amount of data for instance-based learning reduces data storage requirements, lowers computational costs, minimizes noise, and can facilitates a more rapid search.

Keywords : Data mining, Data reduction, Instance-based learning

1. 서 론

기업간의 경쟁이 심화되고 정보의 중요성에 대한 인식이 커짐에 따라 대량의 데이터에서 유용한 정보를 캐내는 데이터 마이닝(data mining)이 주목을 받고 있다. 데이터 마이닝을 통해 얻은 정보는 여러 분야에서 현명한 의사결정을 할 수 있도록 도움을 준다. 데이터 마이닝의 여러 기법 중, 분류(classification)기법은 데이터의 클래스(class)를 구별하는 모델을 만들고, 그 모델을 이용하여 클래스의 값이 알려져 있지 않은 새로운 객체들의 클래스를 예측(prediction)할 수 있도록 하는데 그 목적이 있다.

예를 들어, 은행에서 기존 고객의 정보를 통해 어떠한 고객들에게 대출해 주는 것이 안전한지 위험한지를 구분하는데 이용될 수 있으며, 신용카드 회사의 고객 신용평가에도 사용될 수 있다. 그 외에도 많은 분야의 마케팅에서도 사용될 수 있다.

분류기법 방법 중, k-nearest neighbour(k-nn) 알고리듬은 가장 널리 알려진 인스턴스(instance) 기반 알고리듬 중의 하나이다[1]. 이 알고리듬은 전체 데이터 집합에서, 새로운 인스턴스에 대해 가장 가까운 k개의 인스턴스들을 탐색한다. 그리고 k개 인스턴스들의 다수 클래스를 새로운 인스턴스의 클래스로 예측하는 방법이다. 이 방법은 분류에 대한 높은 정확도를 가지고 있지만 많은 저장 공간과 계산시간이 걸리고, 잡음(noise)에 민감한 반응을 보일 수 있는 단점을 가지고 있다.

본 연구는 분류기법을 위한 데이터 감소(data reduction) 방법을 제시한다. 엔트로피 기반 분할방법(entropy-based partition)과 각 분할 집합의 중심 인스턴스(center instance)들을 이용한 데이터 감소를 통해, 분류 예측의 정확성(accuracy)을 높이는 방법이다. 또한 이 방법을 통해 전체 데이터 집합은 불필요한 속성들(irrelevant attrib-

utes)과 인스턴스들의 감소를 가져온다.

여러 가지 분류기법을 위한 데이터 감소 방법들이 제안되었다. Datta and Kibler는 각 클래스의 속성값으로 분할 한 후, 각 파티션에서 대표 인스턴스들을 찾았다[2]. 그리고 k-means clustering 방법을 이용하여 데이터를 분할하고, 클러스터 평균을 이용하여 인스턴스들을 찾는 방법도 제시하였다[3]. Wai Lam의 Prototype Generation and Filtering(PGF) 알고리듬은 각 인스턴스들의 거리를 계산하여 가장 가까운 인스턴스들을 우선적으로 합병(merge)하는 방식으로 파티션을 구하였다[4]. Sanchez가 제안한 방법에서는 가장 먼 거리에 위치한 두 개의 인스턴스를 클러스터의 중심으로 한 후, 가까운 클러스터에 각 인스턴스를 포함시키는 방법을 사용한다[5]. 미리 지정된 파티션의 개수가 나올 때까지 각 클러스터의 분할은 계속된다. 그리고 최종 파티션에서 중심값을 찾는다.

대부분의 기존연구 방법들은 데이터의 동질성을 유지하는 여러 개의 파티션을 만들기 위해 군집화(clustering) 방법들을 많이 사용하였다[2-5]. 기존 방법에서 사용된 군집화 방법은 모든 인스턴스들의 거리를 반복적으로 계산하여 클러스터를 만드는 방법으로, 데이터의 크기가 커지면 계산시간이 크게 증가하였다. 또한 모든 속성들을 고려하기 때문에 계산시간의 증가와 함께 불필요한 속성까지도 사용되었다.

본 연구에서 제안하는 방법은 기존 방법보다 빠르게 각 파티션을 만들고, 불필요한 속성까지도 제거할 수 있는 특징을 가지고 있다. 또한 각 파티션의 대표 인스턴스를 보다 빠르게 찾을 수 있는 있는 방법을 제시한다.

2. 제안 알고리듬

2.1 엔트로피 계산

S를 s개의 인스턴스를 가지는 데이터 집합이라고 한다. 클래스 레이블(class label) 속성은 m개의 상이한 클래스 $C_i(i=1,2,...,m)$를 정의하는 m개의 상이한 값을 갖는다고 가정한다. s_i는 클래스 C_i에 있는 인스턴스의 개수라고 한다. 주어진 샘플을 분류하는데 요구되는 기대 정보량(expected information)은 다음과 같다[6].

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} p_i log_2(p_i), \quad (1)$$

p_i는 임의의 인스턴스가 클래스 C_i에 속할 확률이며 $\frac{s_i}{s}$으로 계산된다.

속성 A는 v개의 상이한 값 $a_1, a_2, ..., a_v$을 갖는다고 하자. 속성 A는 S를 v개의 부분집합 $S_1, S_2, ... S_v$로 분할하는데 사용될 수 있다. 여기에서 S_j는 A의 값 a_j를 갖는 인스턴스들을 포함한다. s_{ij}를 부분집합 S_j에 있는 클래스 C_i의 인스턴스의 개수라고 한다. 속성 A에 의해 부분집합으로 분할하는 경우의 엔트로피는 아래 식으로 얻어진다.

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + ... + s_{mj}}{s} I(s_{1j}, ..., s_{mj}). \quad (2)$$

$\frac{s_{1j} + ... + s_{mj}}{s}$ 항은 j번째 부분집합(즉, A의 a_j의 값을 갖는)의 가중치로 사용되고, 이는 부분집합의 인스턴스 수를 S의 총 인스턴스 수로 나눈 값이다.

엔트로피 값이 작으면 작을수록 분할된 부분집합의 순수도(purity)는 증가한다. 엔트로피는 제안 알고리듬의 데이터 분할 단계에서 어느 속성의 속성값을 기준으로 먼저 분할할 것인지를 결정할 때 사용된다.

2.2 거리 계산

각 파티션에서 중심 인스턴스를 구하기 위하여 사용하는 거리 척도는 유클리드 거리(euclidean distance)를 사용하며 다음과 같다[6].

$$ED(x, y) = \sqrt{\sum_{i=1}^{a} d(x_i, y_i)^2}, \quad (3)$$

x, y는 각각의 인스턴스이며, a는 속성의 수, x_i는 인스턴스 x의 i번째 속성값을 나타낸다. 속성이 수치형(numerical) 값인 경우의 $d(x_i, y_i)$는 속성들 간의 절대 값 차이로 정의된다 (즉, $x_i - y_i$ 이다). 속성이 범주형(categorical) 값인 경우는 다음과 같이 정의된다.

$$d(x_i, y_i) = 0 \text{ if } x_i = y_i \text{ and } 1 \text{ otherwise.} \quad (4)$$

중심 인스턴스는 유클리드 거리를 이용하여 각 파티션에 있는 인스턴스들의 거리의 합이 최소가 되는 인스턴스로 정한다. 즉, 각 파티션의 중심 인스턴스는 다음과 같다.

중심 인스턴스 =

$$\min\{\sum_{k=1}^{n} ED(x^{1st}, y^{kth}), ..., \sum_{k=1}^{n} ED(x^{nth}, y^{kth})\}, \quad (5)$$

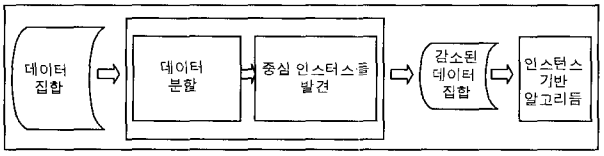
n은 각 파티션에 속해있는 인스턴스 개수이며, x^{ith} 과 y^{kth}는 각각 i번째와 k번째의 인스턴스들이다. 제안 알고리듬의 중심 인스턴스를 찾는 단계에서 위 공식들은 사용된다.

2.3 제안 알고리듬

제안 알고리듬은 데이터 분할과 중심 인스턴스들을 찾는 부분으로 이루어져 있다. 먼저 데이터 집합을 분할한다. 분할 기준으로는 각 속성에 대해서 엔트로피를 구한 후, 엔트로피 값이 가장 작은 속성의 속성값을 기준으로 우선적으로 분할하는 방법을 사용한다. 이 방법을 통해, 전체 데이터의 특성이 잘 파악된 동질의 데이터들이 같은 파티션에 모이게 된다.

파티션 집합(partition set)은 각 파티션의 특성을 가장 잘 대표할 수 있는 인스턴스들로 이루어진다.

제안 알고리듬을 도식화하면 <그림 1>과 같다. 제안 알고리듬을 통해 감소된 데이터 집합은 인스턴스 기반 알고리듬을 위한 데이터 집합으로 사용된다.



<그림 1> 제안 알고리듬

제안 알고리듬의 절차는 다음과 같다.

단계 1~단계 4 : 데이터 분할 단계

단계 1 : 식 (1)과 식 (2)을 이용하여 전체 데이터 집합을 고려한 각 속성에 대한 엔트로피를 계산한다. 가장 엔트로피가 작은 속성을 선택한다. 그 속성값을 기준으로 전체 인스턴스들을 분할한다.

단계 2 : 속성값을 기준으로 한 여러 파티션이 생성된다. 각 파티션에 대하여 나머지 속성들에 대해서도 엔트로피를 계산한다. 각 파티션에서 가장 작은 엔트로피 값을 가지는 속성을 찾아 다시 분할한다.

단계 3 : 분할된 파티션에서의 엔트로피 값이 0이거나 더 이상 분할할 속성이 없을 때까지 단계 2의 과정을 반복한다.

단계 4 : 최종 분할된 파티션 집합을 구성한다.

단계 5~단계 7 : 중심 인스턴스들 발견 단계

단계 5 : 각 파티션에 대한 중심 인스턴스를 구한다. 중심 인스턴스는 식 (3)과 식 (4)의 유클리드 거리와 식 (5)를 이용하여 각 파티션에 있는 인스턴스들의 거리의 합이 최소가 되는 인스턴스로 정한다.

이때 중심 인스턴스를 구하기 위해 고려할 속성은 전체 속성이 아니다. 일단 데이터 분할 단계의 각 파티션에서 고려된 속성들은 각 분할 집합에서 동일한 값을 가지고 있기 때문에 고려하지 않아도 된다. 또한 데이터 분할 단계에서 사용되지 않은 속성들은 불필요한 속성으로 간주하고 사용하지 않는다.

단계 6 : 각 파티션에서 중심 인스턴스를 기준으로 가장 가까운 k개의 인스턴스들을 구한다. k는 각 파티션에 있는 인스턴스 개수에 비례하여 설정한다. 각 파티션에 있는 인스턴스 개수에 비례하여 k를 설정함으로써, 보다 비중이 높은 파티션에서 많은 수의 인스턴스들을 선택한다.

단계 7 : 각 파티션을 대표 할 중심 인스턴스들을 구한다. 대표 인스턴스들은 중심 인스턴스와 중심 인스턴스와 가장 가까운 k개의 인스턴스들로 이루어진다.

단계 8 : 각 파티션을 대표하는 중심 인스턴스들이 모여 감소된 데이터 집합을 이룬다. 이 감소된 데이터 집합은 <그림 1>에서 보듯이 인스턴스 기반 알고리듬에 사용된다.

3. 예제 및 분석

예제 데이터 집합은 총 16개의 인스턴스들과 6개의 속성, 1개의 목표속성으로 이루어져 있다. 다음 <그림 2>와 같다.

#	A1	A2	A3	A4	A5	A6	Class
1	0	0	0	0	0	1	0
2	0	0	0	1	1	1	0
3	0	0	1	0	0	1	0
4	0	0	1	1	0	0	1
5	0	1	0	0	0	1	0
6	0	1	0	1	1	1	0
7	0	1	1	0	1	0	0
8	0	1	1	1	0	0	1
9	1	0	0	0	1	1	0
10	1	0	0	1	1	0	0
11	1	0	1	0	0	1	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
14	1	1	0	1	0	1	1
15	1	1	1	0	1	0	1
16	1	1	1	1	0	0	1

<그림 2> 예제 데이터 집합

단계 1~단계 4 : 데이터 분할 단계

단계 1 : 모든 속성에 대한 엔트로피를 계산한다.

식 (1)과 (2)를 이용하여 속성 A1에 대한 엔트로피를 계산하면 다음과 같다.

A1 = 0에 대하여,

$s_{11} = 6,\ s_{21} = 2$이고,

$$I(s_{11}, s_{21}) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$

$$= 0.31 + 0.5 = 0.81$$

A1 = 1에 대하여,

$s_{12} = 3,\ s_{22} = 5$이고

$$I(s_{12}, s_{22}) = -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8}$$

$$= 0.53 + 0.42 = 0.95$$

$$E(A1) = \frac{8}{16}I(s_{11}, s_{21}) + \frac{8}{16}I(s_{12} + s_{22})$$

$$= (8/16)*0.81 + (8/16)*0.95 = 0.88$$

나머지 속성도 같은 방법으로 엔트로피를 계산한다. A2, A3, A4, A5, A6에 대한 엔트로피를 계산해 보면 다음과 같다.

E(A2) = 0.88, E(A3) = 0.88, E(A4) = 0.88,
E(A5) = 0.82, E(A6) = 0.6

속성 A6의 엔트로피 값이 가장 작기 때문에 가장 먼저 속성 A6을 기준으로 인스턴스들을 분할한다. 분할된 파티션을 P1, P2라고 한다. 1단계 후의 데이터 분할 모습은 <그림 3>과 같다. 분할 시, 고려된 속성값들은 회색으로 표시한다.

#	A1	A2	A3	A4	A5	A6	Class	
7	0	1	1	0	1	0	0	
4	0	0	1	1	0	0	1	
8	0	1	1	1	0	0	1	
10	1	0	0	1	1	0	0	P1
12	1	0	1	1	0	0	1	
13	1	1	0	0	0	0	1	
15	1	1	1	0	1	0	1	
16	1	1	1	1	0	0	1	
1	0	0	0	0	0	1	0	
2	0	0	0	1	1	1	0	
3	0	0	1	0	0	1	0	
5	0	1	0	0	0	1	0	P2
6	0	1	0	1	1	1	0	
9	1	0	0	0	1	1	0	
11	1	0	1	0	0	1	0	
14	1	1	0	1	0	1	1	

<그림 3> 1단계 분할 집합

단계 2 : 파티션 P1, P2가 생성된다. 각 파티션에 대하여 나머지 속성들에 대해서도 엔트로피를 계산한다.

계산 결과, 파티션 P1에서는 A5 속성이, 파티션 P2에서는 A1 속성의 엔트로피 값이 가장 작기 때문에 그 속성들을 기준으로 다시 분할이 이루어진다. 각 파티션에서 엔트로피 값을 고려한 후의 분할 모습은 <그림 4>와 같다. P1에서는 P11, P12가 만들어 지고, P2에서는 P21, P22의 파티션이 만들어 진다.

단계 3 : 분할된 파티션에서의 엔트로피 값이 0 이거나 더 이상 분할할 속성이 없을 경우에, 종료하게 된다.

예제에서는 최종 7개의 파티션이 생성된다.

단계 4 : 최종 분할된 파티션 집합을 구성한다.

예제에서는 {P11, P121, P1221, P1222, P21, P221, P222}의 분할 집합이 만들어 진다. 분할 단계에서의 최종 분할 집합은 <그림 5>와 같다.

#	A1	A2	A3	A4	A5	A6	Class	
4	0	0	1	1	0	0	1	
8	0	1	1	1	0	0	1	
12	1	0	1	1	0	0	1	P11
13	1	1	0	0	0	0	1	
16	1	1	1	1	0	0	1	
7	0	1	1	0	1	0	0	
10	1	0	0	1	1	0	0	P12
15	1	1	1	0	1	0	1	
1	0	0	0	0	0	1	0	
2	0	0	0	1	1	1	0	
3	0	0	1	0	0	1	0	P21
5	0	1	0	0	0	1	0	
6	0	1	0	1	1	1	0	
9	1	0	0	0	1	1	0	
11	1	0	1	0	0	1	0	P22
14	1	1	0	1	0	1	1	

<그림 4> 2단계 분할 집합

#	A1	A2	A3	A4	A5	A6	Class	
4	0	0	1	1	0	0	1	
8	0	1	1	1	0	0	1	
12	1	0	1	1	0	0	1	P11
13	1	1	0	0	0	0	1	
16	1	1	1	1	0	0	1	
7	0	1	1	0	1	0	0	P121
10	1	0	0	1	1	0	0	P1221
15	1	1	1	0	1	0	1	P1222
1	0	0	0	0	0	1	0	
2	0	0	0	1	1	1	0	
3	0	0	1	0	0	1	0	P21
5	0	1	0	0	0	1	0	
6	0	1	0	1	1	1	0	
9	1	0	0	0	1	1	0	
11	1	0	1	0	0	1	0	P221
14	1	1	0	1	0	1	1	P222

<그림 5> 분할 단계에서의 최종 분할 집합

단계 5~단계 8 : 중심 인스턴스들 발견 단계

단계 5 : 식 (3), (4), (5)를 이용하여 각 파티션에 대한 중심 인스턴스를 구한다.

P11 집합에서 중심 인스턴스를 구하는 경우에 A1, A2 속성만을 고려하여 중심 인스턴스를 계산한다. 따라서 전체 속성이 아닌 일부 속성만을 사용하여 중심 인스턴스를 구하기 때문에 빠르게 중심 인스턴스를 구할 수 있는 장점이 있다.

중심 인스턴스는 각 파티션에 있는 인스턴스들의 거리를 계산하여, 인스턴스들의 거리의 합이 가장 작은 인스턴스를 중심 인스턴스로 정한다.

예를 들면, 파티션 P11에 있는 모든 인스턴스들에 대하여 각 인스턴스들 간의 거리의 합을 계산해 보면 다음과 같다.

4번 인스턴스와 나머지 인스턴스들과의 거리의 합 = $\sqrt{1}+\sqrt{1}+\sqrt{2}+\sqrt{2}=2+2\sqrt{2}$,

8번 인스턴스와 나머지 인스턴스들과의 거리의 합 = $\sqrt{1}+\sqrt{2}+\sqrt{1}+\sqrt{1}=3+\sqrt{2}$,

12번 인스턴스와 나머지 인스턴스들과의 거리의 합 = $\sqrt{1}+\sqrt{2}+\sqrt{1}+\sqrt{1}=3+\sqrt{2}$,

13번 인스턴스와 나머지 인스턴스들과의 거리의 합 = $\sqrt{2}+\sqrt{1}+\sqrt{1}+\sqrt{0}=2+\sqrt{2}$,

16번 인스턴스와 나머지 인스턴스들과의 거리의 합 = $\sqrt{2}+\sqrt{1}+\sqrt{1}+\sqrt{0}=2+\sqrt{2}$,

따라서 파티션 P11에서의 중심 인스턴스는 13번 또는 16번 인스턴스가 된다.

단계 6 : 각 파티션에서 중심 인스턴스를 기준으로 가장 가까운 k개의 인스턴스들을 구한다. k는 데이터를 분할하는 단계에서 미리 정의된다.

위 예제에서의 파티션 P11, P21에서는 1개의 k 인스턴스가 선택되고, 나머지 5개에서는 인스턴스의 개수가 1개 또는 2개기 때문에 0개의 k 인스턴스를 선택한다.

단계 7 : 각 파티션을 대표 할 중심 인스턴스들을 구한다.

파티션 P11에서는 1개의 중심 인스턴스와 중심 인스턴스와 가장 가까운 1개의 k 인스턴스가 모여 P11을 대표하는 중심 인스턴스를 구성한다.

단계 8 : 각 파티션을 대표할 중심 인스턴스들은 불필요한 속성이 제거된 데이터 감소 집합을 구성한다.

예제에서 최종 감소된 데이터 집합은 <그림 6>과 같다.

#	A1	A2	A5	A6	Class
13	1	1	0	0	1
16	1	1	0	0	1
7	0	1	1	0	0
10	1	0	1	0	0
15	1	1	1	0	1
1	0	0	0	1	0
3	0	0	0	1	0
9	1	0	1	1	0
14	1	1	0	1	1

<그림 6> 최종 감소된 데이터 집합

4. 실험 결과

새롭게 제안한 방법은 k-nn 알고리듬[1], 분류기법을 위한 데이터 감소 방법인 PGF 알고리듬과 비교하였다[4]. k-nn 알고리듬은 모든 데이터 집합을 학습 집합(training set)으로 사용하였으며, 제안한 방법에서는 불필요한 속성제거와 대표 인스턴스들로 이루어진 감소 데이터 집합을 학습 집합으로 사용한다.

실험에서 사용된 데이터들은 UCI Database Repository에 있는 데이터 집합이며, 모든 속성과 클래스는 범주형 속성값을 가지고 있다[7]. UCI의 Machine Learning 그룹에서는 금융, 카드, 제조, 의류 등의 다양한 분야에서 수집된 100여종의 실제 데이터 집합과 가상 데이터 집합을 제공해 주고 있으며, 이곳은 데이터 마이닝 기법을 시험해 보기 위한 데이터를 제공해 주고 있는 가장 대표적인 곳이다. 본 실험은 펜티엄 IV 3.0 GHz, 512 MB 메모리를 가진 컴퓨터에서 Visual C++ 컴파일러를 사용하여 수행하였다.

평균 분류 정확도(average classification accuracy)를 추정하기 위해서 10-fold cross-validation 방법을 사용한다[6]. 10-fold cross-validation은 전체 데이터 집합을 크기가 대략적으로 같은 10개의 부분 집합으로 나눈다. 9개의 부분집합을 학습 집합으로, 나머지 1개의 부분집합을 검정 집합(testing set)으로 설정한다. 학습과 검정을 k번 반복 수행하게 된다. 분류 정확도의 추정은 k번의 반복으로부터 정확하게 분류가 된 인스턴스 개수와 초기 데이터의 인스턴스 개수 비율로 나타낸다. 제안 알고리듬의 분류 정확도를 측정하기 위해서 사용된 인스턴스 기반 알고리듬은 k-nn 알고리듬을 사용한다. 데이

터 감소 비율은 초기 데이터 집합의 인스턴스 개수와 속성 개수의 곱과 감소된 데이터 집합의 인스턴스 개수와 속성 개수의 곱의 감소 비율로 나타낸다. 분류 정확도와 데이터 감소 비율이 높을수록 성과(performance)가 좋다는 것을 의미한다.

사용된 데이터 집합은 <표 1>과 같다.

<표 1> 데이터 집합

데이터 집합	인스턴스 수	속성 수
Zoo	101	16
Credit	690	5
Mushroom	8124	22

기존 방법 *k*-nn 알고리듬의 경우, *k*값을 k=1, 2, 3, 4, 5까지 변화시키면서 실험을 하였고 가장 좋은 분류 정확도를 채택하였다.

분류 정확도와 데이터 감소 비율의 실험 결과는 <표 2>와 같다. <표 2>에 있는 *k*-nn과 PGF의 실험결과는 Wai Lam의 논문에서 참고하였으며, 3개의 PGF 변형 알고리듬 중에서 가장 좋은 성과를 채택하였다[4]. 기호 "---"는 데이터 감소 비율이 0%를 의미하며 제안 알고리듬의 평균 정확도와 평균 데이터 감소 비율은 각각 91.43%와 96.10%를 나타내었다.

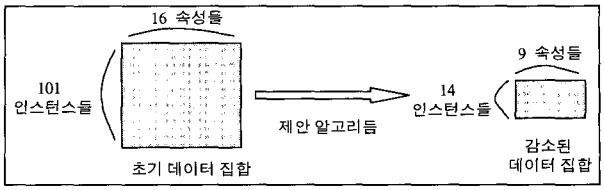
<표 2> 각 데이터 집합의 분류 정확도와 데이터 감소 비율

(단위: %)

데이터 집합	분류 정확도	*k*-nn	PGF	제안 알고리듬
Zoo	분류 정확도	97.00	90.00	92.10
	데이터 감소 비율	---	91.10	92.20
Credit	분류 정확도	80.70	84.50	86.70
	데이터 감소 비율	---	97.70	97.80
Mushroom	분류 정확도	99.90	99.60	95.50
	데이터 감소 비율	---	99.10	99.40
Average	분류 정확도	92.53	91.37	91.43
	데이터 감소 비율	---	95.97	96.10

Zoo 데이터 집합의 경우, 제안 알고리듬을 통하여 16개의 속성 가운데 7개의 불필요한 속성이 제거되고 총 14개의 파티션 중심 인스턴스들로 이루어진 데이터 집합을 구할 수 있었다. 제안 알고리듬을 통해 Zoo 데이터 집합의 데이터의 감소 비율을 그림으로 표시하면 <그림 7>과 같다.

또한 각 파티션에서 중심 인스턴스들을 구하기 위해 사용된 속성은 전체 16개의 속성 중에서, 2~7개의 속성을 사용했으며, 평균적으로 3.8개의 속성만을 사용했다.



<그림 7> Zoo 데이터 집합의 감소 비율

Credit 데이터 집합의 경우, 높은 데이터 감소 비율을 가져왔다. 초기 690개의 인스턴스들이 15개의 인스턴스들로 감소되었다.

Mushroom 데이터 집합의 경우, 높은 데이터 감소 비율과 함께 많은 속성들이 불필요한 속성으로 간주되어 제거되었다. 초기 8124개의 인스턴스들은 178개의 인스턴스들로 감소되었으며, 22개의 속성들이 6개의 속성들로 감소되었다.

<표 3> 각 데이터 집합의 계산시간

(단위: 초)

데이터 집합	*k*-nn	제안 알고리듬
Zoo	0.90	0.07
Credit	1.57	0.12
Mushroom	146.05	5.35

계산시간의 실험 결과는 <표 3>과 같다. *k*-nn 알고리듬의 경우, 계산시간은 초기 데이터 집합을 사용하여 정확도를 측정한 시간이다. 제안 알고리듬의 경우, 계산시간은 초기 데이터 집합을 감소시키는데 걸린 시간과 감소시킨 데이터 집합을 사용하여 정확도를 측정한 시간을 합하였다.

제안 알고리듬은 기존 방법인 *k*-nn 알고리듬과 비교했을 때, 분류의 정확성이 거의 비슷함을 알 수 있었다. Credit 데이터 집합의 경우에서는 제안 알고리듬의 분류 정확성이 더 높았다. 또한 제안 알고리듬은 기존 데이터 감소 알고리듬인 PGF와의 분류 정확도와 데이터 감소 비율을 비교했을 때, 분류 정확도는 비슷하거나 약간 높은 결과를 나타냈으며 데이터 감소 비율은 전체적으로 높은 결과를 나타내었다. 또한 제안 알고리듬을 통해, 계산시간 단축의 효과를 가져 올 수 있었다.

5. 결 론

본 알고리듬은 인스턴스 분할과 속성 선택의 장점을 통합하여 인스턴스 기반 학습을 위한 새로운 데이터 감소 방법을 제시하였다. 분류 기법을 위한 데이터 감소는 데이터 저장 공간의 절약을 가져올 수 있으며, 새로

운 객체의 클래스를 예측하기 위해서 검색할 인스턴스와 속성의 수를 줄임으로써 인스턴스 기반 알고리듬에서의 예측시간을 단축시킬 수 있다.

본 알고리듬을 통해 전체 데이터 집합은 여러 개의 파티션으로 분할된다. 각 파티션은 엔트로피 값이 가장 작은 속성의 속성값을 기준으로 우선 분할이 되며, 엔트로피 값이 0이거나 더 이상 분할할 속성이 없을 때 종료된다.

최종 파티션 내에서의 주요 속성값들은 거의 같은 값을 가지게 된다. 결국 같은 파티션 내에서의 인스턴스들은 동질성 유지효과를 가져올 수 있다.

전체 데이터 집합을 분할한 후, 전체 속성 중에서 각 파티션에서 사용하지 않은 속성은 불필요한 속성으로 간주하고 제거한다.

각 파티션에 포함된 인스턴스들의 개수에 비례하여 각 파티션을 대표할 중심 인스턴스들을 구하게 된다. 이 방법을 통해 이상치(outliers)나 예외성(exception)이 있는 인스턴스들은 제거될 수 있다. 그리고 인스턴스들의 거리를 비교하여 각 파티션의 중심 인스턴스들을 구할 때에도 분할 단계에서 이미 고려된 속성과 불필요한 속성으로 간주되어 제거된 속성들은 고려할 필요가 없다. 따라서 기존 방법보다 빠르게 각 파티션의 중심 인스턴스들을 찾을 수 있다.

실험을 통해, 제안한 알고리듬은 불필요한 속성제거와 중심 인스턴스들로 이루어진 데이터 집합으로 정확한 분류 예측을 할 수 있었다. 또한 기존 분류기법을 위한 데이터 감소 알고리듬에서는 불필요한 속성 제거 없이 인스턴스만을 고려한 데이터 감소 방법들을 제시하고 있으나, 본 연구에서는 인스턴스와 함께 속성제거를 같이 고려함으로써 데이터 감소 비율을 높일 수 있었다. 제안한 알고리듬은 분류기법 뿐만 아니라 데이터 마이닝의 전처리(preprocessing) 과정에서도 유용하게 사용될 수 있다.

참고문헌

[1] Dasarath, B. V., "Nearest Neighbor Norms : NN Pattern Classfication Techniques," *IEEE Computer Society Press*, Los Alamitos, CA, 1991.

[2] Datta, P. and Kibler, D., "Learning prototypical concept description," Proceedings of the 12th International Conference on Machine Learning, pp. 158-166, 1995.

[3] Datta, P. and Kibler, D., "Symbolic nearest mean classifier," Proceedings of the 14th National Conference of Artificial Intelligence, pp. 82-87, 1997.

[4] Lam, W., Keung, C. K., and Ling, C. X., "Learning good prototypes for classification using filtering and abstraction of instances," *Pattern Recognition*, 35 : 1491-1506, 2002.

[5] Sanchez, J. S., "High training set size reduction by space partitioning and prototype abstraction," *Pattern Recognition*, 37 : 1561-1564, 2004.

[6] Han, J. and Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001.

[7] Merz, C. J. and Murphy, P. M., *UCI Repository of Machine Learning Databases*, Irvine, CA : Department of Information and ComputerScience. Internet:http://www.ics.uci.edu/~mlearn/MLRepository.html.