DOI: 10.3745/KIPSTB.2010.17B.5.381

점진적 하강 방법을 이용한 속성값 기반의 가중치 계산방법

이 창 환[†] · 배 주 현[††]

요 약

나이브 베이시안 알고리즘은 데이터 마이닝의 여러 분야에서 적용되고 있으며 좋은 성능을 보여주고 있다. 하지만 이 학습 방법은 모든 속성의 가중치가 동일하다는 가정을 하고 있으며 이러한 가정으로 인하여 가끔 정확도가 떨어지는 현상이 발생한다. 이러한 문제를 보완하기 위하여 나이브 베이시안에서 속성의 가중치를 조절하는 다수의 연구가 제안되어 이러한 단점을 보완하고 있다. 본 연구에서는 나이브 베이시안 학습에서 기존의 속성에 가중치를 부여하는 방식에서 한걸음 나아가 속성의 값에 가중치를 부여하는 새로운 방식을 연구하였다. 이러한 속성값의 가중치를 계산하기 위하여 점진적 하강(gradient descent) 방법을 이용하여 가중치를 계산하는 방식을 제안하였다. 제안된 알고리즘은 다수의 데이터를 이용하여 속성 가중치 방식과 비교하였고 대부분의 경우에 더 좋은 성능을 제공함을 알 수 있었다.

키워드 : 기계학습, 데이터마이닝, 나이브 베이시안, 점진적 하강

Gradient Descent Approach for Value-Based Weighting

Chang-Hwan Lee[†] · JooHyun Bae[††]

ABSTRACT

Naive Bayesian learning has been widely used in many data mining applications, and it performs surprisingly well on many applications. However, due to the assumption that all attributes are equally important in naive Bayesian learning, the posterior probabilities estimated by naive Bayesian are sometimes poor. In this paper, we propose more fine-grained weighting methods, called value weighting, in the context of naive Bayesian learning. While the current weighting methods assign a weight to each attribute, we assign a weight to each attribute value. We investigate how the proposed value weighting effects the performance of naive Bayesian learning. We develop new methods, using gradient descent method, for both value weighting and feature weighting in the context of naive Bayesian. The performance of the proposed methods has been compared with the attribute weighting method and general Naive bayesian, and the value weighting method showed better in most cases.

Keywords : Machine Learning, Data Mining, Naive Bayesian, Gradient Descent

1. 서 론

나이브 베이시안(Naive Bayesian) 학습방법은 다양한 마이닝 분야에 적용되어 왔으며 알고리즘 수행의 간단함에 비하여 좋은 성능을 보여주고 있다. 하지만 나이브 베이시안의 기본적인 가정 중의 하나인 모든 속성이 같은 중요도를 가진다는 가정은 나이브 베이시안이 생성하는 이후(posterior) 확률의 정확도를 떨어지게 하는 원인으로 작용한다[4]. 예를 들어서 어떤 환자가 당뇨병이 있는지를 예측할 때, 그 환자의 혈압수치는 그 환자의 키보다 훨씬 높은 중요도를 가질 것이다. 따라서 속성별로 같은 가중치를 부여하는 것은 알고리즘을 간단하게하고 속도를 빠르게 하지만 경우에 따라서 정확도를 희생하는 경우가 있다. 따라서 나이브 베이시안에서 각 속성이 독립적이라는 가정을 변경하면 좀 더 좋은 성능을 보임을 보이는 연구도 제시되어 있다[9].

이와 같은 이유로 나이브 베이시안의 성능을 여러 방식으로 확장하는 시도들이 제안되어 왔다. 첫 번째 방식은 나이브 베이시안 학습을 속성선택(feature selection) 기능과 결합하는 방식이다[1, 10, 14]. 이 방식은 학습의 전처리 방식으로 수행되며 속성 중에서 중복되거나 학습에 의미가 없는 속성들을 제거하여서 나이브 베이시안 학습의 성능을 향상하고자 하는 방식이다. 가능한 속성공간의 크기가 천문학적으로 크기 때문에 모든 속성 선택 방법의 탐색 공간의 전체를 탐색하기는 현실적으로 불가능하며 따라서 이 방법은 주어진 속성 공간에서 적절한 속성의 집합을 찾기 위하여 주

※ 본 연구는 한국연구재단(NRF)의 중견연구자 사업(과제번호: 2009-0079025) 및 2009 년도 동국대학교 연구년 지원에 의하여 이루어졌음.
† 종신회원 : 동국대학교 정보통신학과 교수
†† 정 회 원 : 동국대학교 정보통신학과 강사
논문접수 : 2010년 5월 18일
수 정 일 : 2010년 7월 30일
심사완료 : 2010년 7월 30일

로 경험적 방식의 탐색을 이용하여 수행된다.

두 번째 방식은 나이브 베이시안의 모든 속성들에게 그 중요도에 따라 서로 다른 가중치를 부여하는 방법이다 [6, 7, 16]. 속성 가중치 부여 방식은 속성선택의 방식과 연관이 있는 것으로 볼 수 있다. 속성의 가중치를 0 혹은 1 로 제한하면 속성 가중치 방식은 속성선택방식과 같아진다. 속성의 가중치를 부여함으로써 속성선택방법보다는 훨씬 유연한 방식으로 사용될 수 있으며 높은 학습 정확도를 나타낼 수 있다. 이러한 속성의 가중치 계산 방식은 주로 근접이웃(nearest neighbor) 알고리즘에서 주로 연구되어 왔다[15].

나이브 베이시안에서의 가중치 계산 방법은 근접학습 방법에 비하여 상대적으로 많은 연구가 되어 있지 않으며 최근 들어서 몇몇 방법들이 발표되고 있다. 하지만 근접 이웃 알고리즘에서도 속성의 가중치부여는 알고리즘의 정확도를 향상시키는 것으로 알려져 있으며[15] 아울러 나이브 베이시안에서도 속성의 가중치 부여는 정확도를 향상시키는 것으로 알려져 있다[6, 7, 16].

본 연구에서는 나이브 베이시안 환경에서 속성의 값에 대한 가중치를 계산하는 새로운 방식을 제안한다. 본 연구의 가중치 계산은 기존의 속성에 대하여 가중치를 부여하는 방식인데 반하여 본 연구는 속성의 값별로 별도의 가중치를 부여하는 새로운 방식이다. 본 논문에서는 이와 같은 속성값 가중치 방식을 위한 새로운 가중치 부여 방법을 제안하고 이를 기존의 속성 가중치 방식과 비교하여 서로의 성능을 비교하기로 한다. 나이브 베이시안에서 속성의 가중치를 부여하는 방법은 몇 가지 방법이 제안되어 있지만 속성값에 대한 가중치의 부여에 대한 방법은 아직 제안된 적이 없다. 지금까지의 방법인 속성에 대하여 가중치를 부여하는 방식을 속성 가중치 방식 이라고 하며 본 논문의 속성 값에 대한 가중치 계산 방식은 속성값 가중치 방식이라고 이름 부르기로 한다.

2. 가중치 나이브 베이시안 방법

분류학습에서 알고리즘은 학습 후에 새로운 데이터에 대하여 목적속성의 값을 부여한다. 새로운 데이터가 n 개의 속성을 가지고 이들 속성의 값을 $a_1, a_2, \ldots, a_n$ 이라고 하자. 속성 C 는 우리가 예측해야하는 목적 속성을 의미하고 c 는 목적속성 C 의 가능한 값들의 의미한다고 하면, 나이브 베이시안에서는 모든 속성들이 독립적이라는 가정에 의하여 다음의 관계가 성립한다.

$$P(a_1,a_2,\ldots,a_n|c)=\prod_{i=1}^{n}P(a_i|c)$$

새로운 데이터는 최대의 이후 확률값(posterior)을 가지는 속성 값을 자신의 목적속성 값으로 예측하게 된다. 다시 말해서 새로운 데이터 D 의 목적속성 값은 다음의 식에 의하여 결정된다. 이 식에서 a_{ij} 는 i 번째 속성의 j 번째 값을 의미한다.

$$V_{NB}(D)=\arg max_c P(c)\prod_{a_{ij}\in D}P(a_{ij}|c)$$

새로운 데이터의 목적속성 값은 최대의 사후확률 값을 가지는 목적속성의 값을 가진다.

나이브 베이시안에서 가정하는 모든 속성이 서로 독립적이라는 가정이 현실적으로 항상 만족하지는 않으므로 이러한 독립성 가정을 보완하는 연구가 진행되어 왔다. 첫 번째 방법은 속성 선택의 방법이고 두 번째 방법은 속성의 가중치 부여 방법인데, 속성선택은 속성 가중치 부여방법의 일부분으로 간주할 수 있다. 속성에 가중치를 부여한 나이브 베이시안의 방법은 아래와 같은 방법으로 목적속성의 값을 구한다.

$$V_{WNB}(D)=\arg max_c P(c)\prod_{a_{ij}\in D}P(a_{ij}|c)^{w_i}$$

여기서 w_i 는 i-번째 속성의 가중치를 의미하며 속성의 중요도를 0 과 1 사이의 숫자로 표현한다.

본 논문에서는 이와 같은 속성에 가중치를 부여하는 방법을 좀 더 세분화 하여서 속성의 값마다 가중치를 부여하는 방법을 제안한다. (그림 1)은 나이브 베이시안에서 속성마다 가중치를 부여하는 것의 의미를 설명하고 있다.((그림 1)의 #(Y) 는 데이터의 값이 Y 인 데이터의 개수를 의미한다.)

(그림 1)에서 목적속성의 이전(prior) 확률 분포는 P(Y)=9,000/10,000=0.9 이며 P(N)=1,000/10,000=0.1 이다. 성별속성에서 속성의 값이 'male' 인 경우에 이들의 사후(posterior) 확률을 구하면 P(Y|male)=100/1,000=0.1이며 P(N|male)=900/1,000=0.9이다. 이는 목적속성의 이전 분포와 비교하면 상당한 변화를 보이고 있으며 따라서 성별의 값이 'male' 인 경우는 목적속성의 분포에 상당한 영향을 미친다. 반면에 속성의 값이 'female'인 경우에 이들의 사후 확률을 구하면 P(Y|female)=8,900/9,000=0.989이며 P(N|female)=100/9,000=0.011 이다. 이는 목적속성의 이전 분포와 비교하면 어느 정도의 변화를 보이지만 성별의 값이 'male' 인 경우에 비하면 거의 미미한 변화를 보이고 있다. 따라서 성별의 값이 'female' 인 경우는 목적속성의 분포에 상대적으로 미미한 영향을 미친다는 것을 알 수 있다.

즉 성별의 속성이 목적속성에 영향을 미치는 것의 대부분은 속성의 값이 'male'일 때에 발생하는 영향임을 알 수 있다.

성별	기타 속성들	목적속성
male : #(Y)= 100 #(N)= 900 female : #(Y)= 8,900 #(N)= 100	.....	#(Y)=9,000 #(N)=1,000
.....	.....	.....

(그림 1) 속성값 가중치의 예시

하지만 기존의 방법들은 가중치 계산에서 속성별로 가중치를 부여하므로 이러한 현상을 표현할 수 있는 방법이 제한되고 있다. 따라서 이러한 문제점을 보완하기 위하여 본 연구는 속성의 값 별로 가중치를 계산하는 방법을 제시한다. 이와 같이 속성의 값에 대하여 가중치를 부여하게 되면 나이브 베이시안의 분류식은 다음과 같아진다.

$$V_{VWNB}(D) = \arg max_c P(c) \prod_{a_{ij} \in D} P(a_{ij}|c)^{w_{ij}} \quad (1)$$

이 식에서 w_{ij}는 i번째 속성의 j번째 값의 가중치를 의미한다.

3. 관련 연구

속성의 가중치 계산은 속성의 가중치로써 0 과 1 사이의 실수를 부여함으로써 속성 선택의 방법보다도 좀 더 유연하며 속성선택은 속성 가중치 방법의 일부분으로 간주할 수 있다. 이러한 속성의 가중치 계산 방법은 속성에 대한 바이어스(bias)의 일종으로써 지금까지 대부분 근접이웃 알고리즘의 경우에 주로 사용되어 왔다[15]. 속성의 가중치 계산 방법은 여러 가지 방법이 제안 되어 있지만 이들은 크게 필터(filter) 방법과 래퍼(wrapper)두 가지의 종류로 구분할 수 있다. 이들 방법들은 가중치의 계산 방법과 분류학습의 상호 작용 방식에 따라서 결정된다.

첫째, 필터 방법은 속성의 가중치가 알고리즘의 수행 전에 계산되며 따라서 알고리즘의 전처리의 과정으로써 수행된다. 데이터의 특징 혹은 경험적인 측정치에 의하여 결정되는 방법으로써 대부분 정보량, 정보획득비율 등의 특정치를 사용하여 속성의 가중치를 계산한다. 두 번째의 래퍼 방식은 가설을 기준으로 가중치를 계산하는 방식인데, 먼저 각 속성에 임의의 가중치를 부여하고 이러한 가중치를 사용하여 분류학습을 진행한다. 다음으로 분류학습의 성능을 기준으로 기준의 조정한 후에 다시 분류학습을 수행한다. 이와 같은 과정을 속성 가중치의 변화가 없거나 특정한 종료조건이 만족할 때까지 반복해서 수행한다. 즉 래퍼 방법에서 속성의 가중치는 해당 가중치를 사용하는 경우에 알고리즘의 성능이 어떤가에 좌우되며 따라서 래퍼 방법은 항상 특정 학습 알고리즘과 연관되어 결정한다.

나이브 베이시안에서의 래퍼 방법의 예는 Langley 와 Sage 의 SBC(selective Bayesian Classifier)를 들 수 있다[9]. 이 방법은 속성의 가중치 계산이 아닌 속성 선택에 중점을 두고 있는데, 나이브 베이시안의 정확도를 기준으로 선택된 속성집합의 성능을 평가한다. Langley 와 Sage 는 전체 속성에서 알맞은 속성 부분집합을 구하는 SBC 라는 greedy 알고리즘을 제안하였다.

속성에 가중치를 부여하는 방법은 근접이웃 알고리즘에서는 많이 제안 되어 있으나 나이브 베이시안의 환경에서는 극히 제한적으로 연구가 진행되어 왔다. Hall[7]은 결정트리를 이용한 나이브 베이시안에서의 가중치 계산 방법을 제안하였다. 이 방법은 우선 가지치기전의 결정트리를 생성하고 각 속성이 결정트리의 어느 레벨에서 나타나는지를 검사한다. 즉 상위 레벨에 나타나는 속성 일수록 중요도가 크다고 할 수 있으므로 해당 속성의 레벨을 기준으로 속성의 가중치를 계산한다. 결정 트리의 생성 시에 배깅(baggin) 방법을 이용하여 오차를 줄이고 결정트리를 안정화 시켰으며 결정트리에 나타나지 않은 속성의 가중치는 0 으로 결정하였다. 이와 같은 방법으로 기존의 나이브 베이시안에 비하여 분류의 성능이 향상됨을 보였다.

Zhang과 Sheng[16]은 나이브 베이시안에서 정보획득비율(gain ratio)을 이용한 속성 가중치 계산 방법과 아울러 다른 피드백 방법을 사용하여 속성의 가중치를 계산하였으며 성능 평가의 기준으로 AUC 기준을 적용하였다. Gartner [6]는 SVM 을 이용하여 속성의 가중치를 계산하는 방법을 제안하였다. 이 방법은 가상공간을 최적으로 이 등분 할 수 있는 hyperplane 을 찾아내고 이 hyperplane 을 결정하는 가중치를 나이브 베이시안에서 해당 속성의 가중치로 계산하는 방식이다. 각 가중치들은 과적합(overfitting) 문제를 피하기 위하여 최적화 되었고 해당 방법은 다른 기계학습의 알고리즘보다 좋은 성능을 보인다고 기술하고 있다.

4. 속성값 기반의 나이브 베이시안 학습방법

앞에서 기술한 바와 같이 나이브 베이시안에서 속성의 가중치 계산은 몇 가지가 제안되어 있다. 하지만 지금까지 속성의 값 각각에 대하여 다른 기중치를 부여하려는 시도는 알려진것이 전혀 없다. 본 논문은 나이브 베이시안의 가중치 계산에서 새로운 시도를 하려한다.

기존의 가중치 계산 방법들은 각 속성에 대하여 한 개의 가중치를 부여한다. 따라서 해당 속성의 속성값마다 같은 가중치를 사용하게 된다. 하지만 나이브 베이시안에서는 각 속성의 값마다 분류학습에 영향을 미치는 중요도가 다르며 따라서 이들의 가중치를 조절하여야 한다. 예를 들어서 (그림 1)에서 {P(Y)=0.9, P(N)=0.1, P(Y|male)=0.1, P(N|male)=0.9, P(Y|female)=0.989, P(N|female)=0.011} 이다. 지금까지의 속성가중치 계산방법은 성별 속성에서의 'male', 'female' 속성값에 대하여 같은 가중치를 부여한다. 하지만 (그림 1)에서 보듯이 성별 속성의 값이 'male' 혹은 'female'인 경우에 목적속성에 미치는 영향은 많은 차이를 보인다. 속성값이 'male'의 경우에는 목적속성의 이전 값 분포와 많은 차이를 보이지만 속성값이 'female'의 경우에는 목적속성의 이전 값 분포와 많은 차이를 보이지 않는다. 따라서 각 속성의 값마다 같은 가중치를 부여하는 기존의 방법 보다는 서로 구분된 가중치를 부여하는 것이 더욱 정확한 학습을 가능하게 할 것이다. 이와 같은 이유로 본 연구에서는 각 속성의 값에 대하여 다른 속성 가중치를 부여하는 방법을 제안한다. 속성 전체에 같은 가중치를 부여하면 이러한 속성 값에 따른 영향을 탐지할 수 없으며 따라서 전체적인 성능에 영향

을 미칠 수도 있다.

앞에서 언급한바와 같이 속성의 가중치 계산 방법은 몇 가지 제안된 내용이 있지만 속성 값의 가중치 계산 방법은 지금까지 알려진 연구가 전혀 없다. 따라서 속성 값의 가중치 계산 방법은 본 연구에서 새롭게 고안하여야 한다. 또한 속성값 가중치 계산 방법과 더불어 속성 가중치의 계산 방법도 같이 개발하여 두 방법 간의 성능을 비교하려 한다.

본 연구에서는 점진적 하강(gradient descent)의 방법을 이용하여 속성값의 가중치를 계산하고자 한다. 점진적 하강 방법은 기계학습 혹은 마이닝의 최적화 알고리즘 등에서 다양하에 적용되고 있다.

5. 속성값 가중치의 계산

본 연구에서는 래퍼 방법을 기준으로 속성 값의 가중치를 계산하는 방법을 제안하고자 한다. 구체적으로 래퍼 방법 중에서도 점진적 하강의 방법을 사용하여 가중치를 계산하는 방법을 제시한다. 점진적 하강의 방법은 기계학습이나 최적화 등의 문제에서 널리 사용되는 대표적인 방법 중의 하나이다.

점진적 하강 방법을 사용하기 위하여 위하여 알고리즘이 최소화하는 에러의 양을 측정하는 에러 함수를 정의해야한다. t_k를 예측하고자 하는 데이터의 실제 목적속성 값이라고 하고, o_k를 알고리즘이 예측하는 값이라고 하면 학습시의 에러의 양은 다음과 같이 정의된다. (N 은 데이터의 개수)

$$\frac{1}{2}\sum_{k=1}^{N}(t_k-o_k)^2 \tag{2}$$

본 연구에서는 나이브 베이시안 방법을 적용하기 때문에 나이브 베이시안의 정의에 의하여 o_k의 값은 수식 (1)에 따라 다음과 같이 정의된다.

$$o_k = \arg max_c P(c)\prod_{a_{ij}\in D_k} P(a_{ij}|c)^{w_{ij}}$$

점진적 하강의 방법을 사용하기 위해서는 o_k 의 값이 연속(continuous)이며 미분가능(differentiable)하여야 한다. 본 연구에서는 이 문제를 해결하기 위하여 우선 멀티클래스(multi-class)의 문제를 각 클래스 값의 이진클래스(binary class) 문제로 변환한다. 예를 들어서 원래의 데이터가 $(a_1,a_2,\cdots,a_n,c_p)$ 의 형태로 표현되어 있다고 가정하자. 여기서 a_i는 i 번째의 속성 값을 의미하며 c_p는 목적속성의 값 중에서 p번째의 값을 목적속성의 값으로 가진다고 하자. 데이터는 $(a_1,a_2,\cdots,a_n,t_1,t_2,\cdots,t_p,\cdots,t_L)$의 형태로 변환된다. 여기서 t_p의 값만 1/2 이 되며 나머지 t_i의 값들은 0을 가지게 된다. t_p의 값이 1/2 이 되는 이유는 이와 같이 이분 클래스로 문제를 변환하면 에러가 두 곳에서 발생하기 때문이다. 이와 같이 문제를 변환하면 수식 (2) 의 에러 함수는 다음과 같이 수정되어 정의 된다.

$$E=\frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{L}(t_{kl}-o_{kl})^2 \tag{3}$$

그리고 이와 같이 변환된 환경에서의 o_{kl} 는 다음과 같이 정의된다.

$$o_{kl}=\begin{cases}1/2 & \text{if } \ l=\arg max_l P(c_{kl})\prod_{a_{ij}\in D_k} P(a_{ij}|c_{kl})^{w_{ij}} \\ 0 & \text{otherwise}\end{cases} \tag{4}$$

여기서 c_{kl} 은 k 번째 데이터의 l 번째 클래스 값을 의미한다. 로그(logarithmic) 함수는 단조(monotonic)증가 함수이기 때문에 위의 식에 로그함수를 취하여도 부등호에 영향을 미치지 않는다. 따라서 A_{kl}와 A_{k^*}를 다음과 같이 정의하면

$$A_{kl}=\left(\ln P(c_{kl})+\sum_{a_{ij}\in D_k} w_{ij}\ln(P(a_{ij}|c_{kl}))\right),$$
$$A_{k^*}=\max A_{kl},$$

수식 (4) 의 o_{kl}의 정의는 다음과 같이 변경되어 표현할 수 있다.

$$o_{kl}=\begin{cases}1/2 & \text{if } \ A_{kl}=A_{k^*} \\ 0 & \text{if } \ A_{kl}<A_{k^*}\end{cases}$$

점진적 하강의 방법을 사용하려면 에러 함수가 연속이고 미분 가능하여야 하는데 위의 o_{kl} 식은 아직 이러한 조건을 만족하지 못하며 따라서 점진적 하강의 방법을 적용할 수 없다. 따라서 본 연구에서는 위의 o_{kl} 값을 위한 근사함수를 사용하여 이 문제를 해결한다. o_{kl}의 값으로 시그모이드(sigmoid) 함수를 사용하면 o_{kl}의 특징과 거의 유사한 성질을 가지며 따라서 근사값으로 사용할 수 있다. 또한 가장 중요한 장점은 이 함수가 연속이고 미분가능한 함수인 점이다. 따라서 o_{kl}의 값은 시그모이드 함수를 사용하여 다음과 같이 정의된다.

$$o_{kl}=\left[1+\exp(-(A_{kl}-A_{k^*}))\right]^{-1}$$
$$=\left[1+\exp\left(\ln\frac{P(c_{k^*})}{P(c_{kl})}+\sum_{a_{ij}\in D_k} w_{ij}\ln\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)\right]^{-1}$$

o_{kl}의 값이 위와 같이 정의 되면서 연속이며 미분가능한 함수로 정의 가 되었고 따라서 점진적 하강의 방법을 적용할 수 있는 준비가 되었다.

본 식에서 최소화하는 에러의 함수는 앞에서 기술한 바와 같이 다음과 같이 구성되어 있다.

$$E=\frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{L}(t_{kl}-o_{kl})^2$$

점진적 하강의 방법은 위의 에러함수를 기준으로 가중치 값의 벡터인 W의 값을 다음과 같은 방법을 이용하여 반복적으로 조정해 나간다. 여기서 η는 학습률(learning rate)을 의미한다.

$$W:=W-\eta\nabla E$$

W는 전체 속성 값 가중치의 벡터를 의미하므로 다음과 같이 구성되어 있다.

$$W=[w_{11},\cdots,w_{1|a_1|},w_{21},\cdots,w_{2|a_2|},\cdots,w_{M1},\cdots,w_{Ma_{M}|}]$$

에러 함수의 미분 결과는 다음과 같다.

$$\nabla E=\sum_{k}\sum_{l}(o_{kl}-t_{kl})\nabla o_{kl}$$

결과 예측치인 o_{kl}의 미분값은 다음과 같이 구성되어 있으며

$$\nabla o_{kl}=\left[\frac{\partial o_{kl}}{\partial w_{11}},\cdots,\frac{\partial o_{kl}}{\partial w_{1|a_1|}},\frac{\partial o_{kl}}{\partial w_{21}},\cdots,\frac{\partial o_{kl}}{\partial w_{2|a_2|}},\cdots,\frac{\partial o_{kl}}{\partial w_{M1}},\cdots,\frac{\partial o_{kl}}{\partial w_{Ma_{M}|}}\right]$$

따라서 o_{kl}의 각 가중치 별 미분값을 구하면 된다. 계산의 편의를 위하여

$$r_{kl}=\left(\ln\frac{P(c_{k^*})}{P(c_{kl})}+\sum_{a_{ij}\in D_k}w_{ij}\ln\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)$$

이라고 가정하면 o_{kl}의 값은 $o_{kl}=\frac{1}{1+\exp(r_{kl})}$으로 표현된다.

따라서 첫 번째의 경우로 어느 속성값 a_{ij}가 현재 예측하고자하는 k번째 데이터의 속성 값에 포함되어 있으면 $(a_{ij}\in D_k)$, $\frac{\partial o_{kl}}{\partial w_{ij}}$의 값은 다음과 같다.

$$\begin{aligned}\frac{\partial o_{kl}}{\partial w_{ij}}&=\frac{\partial o_{kl}}{\partial r_{kl}}\frac{\partial r_{kl}}{\partial w_{ij}}\\&=\frac{-\exp(r_{kl})}{[1+\exp(r_{kl})]^2}ln\left(\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)\\&=\frac{\exp(r_{kl})}{[1+\exp(r_{kl})]^2}ln\left(\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\right)\\&=o_{kl}(1-o_{kl})\ln\left(\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\right)\end{aligned}$$

두 번째의 경우로 어느 속성값 a_{ij}가 현재 예측하고자하는 k번째 데이터의 속성 값에 포함되어 있지 않으면 $(a_{ij}\notin D_k)$, $\frac{\partial o_{kl}}{\partial w_{ij}}$의 값은 다음과 같다.

$$\frac{\partial o_{kl}}{\partial w_{ij}}=\frac{\partial o_{kl}}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial w_{ij}}=\frac{\partial o_{kl}}{\partial r_{ij}}\cdot 0=0$$

이들 경우를 결합하면 $\frac{\partial o_{kl}}{\partial w_{ij}}$는 아래와 같이 표현된다.

$$\frac{\partial o_{kl}}{\partial w_{ij}}=o_{kl}(1-o_{kl})\ln\left(\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\right)\cdot I(a_{ij}\in D_k)$$

여기서 $I(c)$ 함수는 c의 조건이 참인 경우에 1을 거짓인 경우에 0의 값을 가지는 함수이다. 따라서 점진적 하강의 최종적인 가중치 조정 수식은 다음과 같다.

$$\begin{aligned}w_{ij}&:=w_{ij}-\eta\sum_{k}\sum_{l}(o_{kl}-t_{kl})\frac{\partial o_{kl}}{\partial w_{ij}}\\&:=w_{ij}-\eta\sum_{k}\sum_{l}(o_{kl}-t_{kl})o_{kl}(1-o_{kl})\\&\ln\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\cdot I(a_{ij}\in D_k)\end{aligned}$$

따라서 본 연구의 속성값의 가중치는 다음과 같은 방법으로 계산되어진다. 각 속성값마다 자신의 가중치를 초기에 임의로 설정하며 그 다음부터 학습데이터로부터 위의 식의 값을 계산하여 속성값의 가중치들을 조정해 나간다.

6. 속성 기반의 가중치 계산방법

앞 절에서는 점진적 하강방법을 이용한 속성 값의 가중치 계산 방법을 제시하였다. 본 논문의 주제가 속성에 가중치를 주는 방법과 속성의 값마다 다른 가중치를 주는 방법의 차이를 분석하는 것이 주된 목적이므로 본 절에서는 속성에 대한 가중치를 계산하는 방법을 제안한다. 앞 절의 속성 값 가중치 계산에서 사용한 방법과 동일한 방법을 이용하여 속성의 가중치를 계산하는 방법을 제시하고자 한다. 에러 함수는 속성값의 가중치 계산에서와 같은 함수를 사용한다.

$$E=\frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{L}(t_{kl}-o_{kl})^2$$

속성기반의 가중치 계산에서 사용하는 방법은 속성 값은 계산할 때에 사용하는 방법과 거의 동일하며 차이점은 속성에 대하여 하나의 가중치만 존재한다. 즉, i 번째 속성은 w_i의 가중치를 가지며 이는 그 속성의 모든 속성 값에 적용된다. 따라서 예측치 o_{kl}는 다음과 같이 정의된다. 이 식에서는 속성 값의 가중치 w_{ij} 대신에 w_i가 사용되고 있음을 알 수 있다.

$$o_{kl}=\begin{cases}1/2 & \text{if } l=\arg max_l \mathrm{P}(\mathrm{c_l})\prod_{\mathrm{a_{ij}}\in \mathrm{D_k}}\mathrm{P}(\mathrm{a_{ij}}|\mathrm{c_{kl}})^{\mathrm{w_i}}\\ 0 & \text{otherwise}\end{cases}$$

여기서 $B_{kl}=\left(\ln P(c_{kl})+\sum_{a_{ij}\in D_k} w_i \ln(P(a_{ij}|c_{kl}))\right)$,

$B_{k^*}=\max \mathrm{B_{kl}}$ 라고 정의 하면, 예측치 o_{kl}는 다음과 같이 정의된다.

$$o_{kl}=[1+\exp(-(B_{kl}-B_{k^*}))]^{-1}$$
$$=\left[1+\exp\left(\ln\frac{P(c_{k^*})}{P(c_{kl})}+\sum_{a_{ij}\in D_k} w_i \ln\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)\right]^{-1}$$

속성마다 한 개의 가중치가 존재하므로 가중치의 벡터는 다음과 같은 내용을 포함하며

$$W=[w_1, w_2, \cdots, w_M]$$

예측치 o_{kl}의 미분값도 다음과 같이 표현된다.

$$\nabla o_{kl}=\left[\frac{\partial o_{kl}}{\partial w_1},\frac{\partial o_{kl}}{\partial w_2},\cdots,\frac{\partial o_{kl}}{\partial w_M}\right]$$

여기서 $s_{kl}=\left(\ln\frac{P(c_{k^*})}{P(c_{kl})}+\sum_{a_{ij}\in D_k} w_i \ln\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)$로 정의하면 $o_{kl}=\frac{1}{1+\exp(s_{kl})}$로 표시할 수 있다. 따라서 $\frac{\partial o_{kl}}{\partial w_i}$의 값은 다음과 같이 정의된다.

$$\frac{\partial o_{kl}}{\partial w_i}=\frac{\partial o_{kl}}{\partial s_{kl}}\frac{\partial s_{kl}}{\partial w_i}$$
$$=\frac{-\exp(s_{kl})}{[1+\exp(s_{kl})]^2}\sum_{j\in i}\ln\left(\frac{P(a_{ij}|c_{k^*})}{P(a_{ij}|c_{kl})}\right)$$
$$=\frac{\exp(s_{kl})}{[1+\exp(s_{kl})]^2}\sum_{j\in i}\ln\left(\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\right)$$
$$=o_{kl}(1-o_{kl})\sum_{j\in i}\ln\left(\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}\right)$$

따라서 최종적인 가중치의 수정식은 다음과 같다.

$$w_i:=w_i-\eta\sum_k\sum_l(o_{kl}-t_{kl})o_{kl}(1-o_{kl})\sum_{j\in i}\ln\frac{P(a_{ij}|c_{kl})}{P(a_{ij}|c_{k^*})}$$

속성의 가중치 계산방법도 속성값 가중치의 계산 방법과 동일하게 이루어 진다. 속성마다 가중치를 초기에 임의로 설정하며 그 다음부터 학습데이터로부터 위의 식의 값을 계산하여 속성의 가중치들을 조정해 나간다.

7. 실험 결과

본 연구에서 제안한 속성값 가중치 기반의 나이브 베이시안의 성능을 속성 가중치의 나이브 베이시안의 성능과 비교하기 위하여 다수의 데이터를 이용하여 실험을 진행하였다. 본 연구에서는 UCI machine learning [12] 에서 7 개의 데이터를 사용하여 실험을 진행하였다. <표 1> 은 본 실험에 사용된 데이터의 특징을 설명하고 있다. 이들 데이터는 속성들이 범주형의 속성들만을 포함하고 있어서 이산화(discretize)의 전처리 과정이 필요없는 장점이 있다. 속성 혹은 속성 값의 가중치를 계산할 때에는 라플라스 스무딩(Laplace smoothing) 의 방식으로 확률을 계산하였다. 이는 각 가중치에서 사용하는 확률을 계산할 때 분모가 0 이 되는 경우를 방지하기 위하여 사용하며 본 논문에서 사용하는 라플라스 스무딩 방법은 다음과 같다.

$$P(c|a_{ij})=\frac{\#(\mathrm{a\_ij}\wedge \mathrm{c})+1}{\#(\mathrm{a\_ij})+\mathrm{L}},\quad P(c)=\frac{\#(\mathrm{c})+1}{N+L}$$

여기서 L 은 목적속성의 값의 수를 의미라며 N 은 전체 데이터의 수를 의미한다. 각 데이터에 대하여 학습을 위하여 임의로 선택된 70% 를 학습데이터로 사용하고 나머지 30% 를 테스트 데이터로 사용하였다. 이와 같은 과정을 5 회 반복하여 평균의 값을 데이터에 대한 알고리즘의 정확도로 계산하였다.

본 연구 알고리즘들의 성능을 평가하기 위하여 각 데이터에 대하여 기본 나이브 베이시안 방법(NB), 제5장의 속성 가중치 방법(WNB), 제4장의 속성 값 기반 가중치 방법

〈표 1〉 실험 데이터의 특성

데이터	속성수	목적속성값의 수	데이터의 개수
balance-scale	4	3	625
flare	10	3	1389
vote	16	2	435
kr-vs-kp	36	2	3196
nursery	8	3	12960
promoters	58	2	106
tic-tac-toe	9	2	958

〈표 2〉 알고리즘들의 성능비교

데이터	NB	WNB	VWNB
balance-scale	91 ± 2.3	89 ± 2.8	85 ± 3.0
flare	74 ± 4.2	77 ± 2.1	78 ± 4.5
vote	90 ± 2.8	92 ± 1.9	93 ± 2.1
kr-vs-kp	87 ± 2.3	78 ± 3.1	89 ± 3.6
nursery	92 ± 1.6	89 ± 2.0	86 ± 2.6
promoters	94 ± 1.7	97 ± 0.9	91 ± 1.1
tic-tac-toe	69 ± 5.4	66 ± 4.7	70 ± 5.3

(VWNB)을 독립적으로 수행하여서 성능을 비교하였다. η와 λ의 값은 각각 0.5을 사용하였다. 실험은 데이터의 샘플링을 바꾸면서 5 회 반복 수행하였고 알고리즘의 정확도들에 대한 평균과 범위를 <표 2>에서 요약하고 있다.

<표 2>에서 보는바와 같이 본 연구의 속성값 가중치 방법(VWNB)은 속성 가중치 방법(WNB)와 비교하여 7개의 데이터 중에서 4 개의 경우에 더욱 좋은 성능을 보이고 있다. 그리고 일반적인 나이브 베이시안 방법(NB)과 비교하여도 7개의 데이터 중에서 4개의 경우에 높은 정확도를 보이고 있다. 속성가중치 방법을 일반 나이브 베이시안 방법과 비교한 경우 3가지의 경우에 좋은 성능을 보이고 있다. 즉 일반 나이브 베이시안의 방법이 조금 좋은 성능을 보이는데 이는 기존의 다른 속성 가중치 방법과 비교하면 큰 차이를 보이지 않는다.

8. 결 론

본 연구에서는 나이브 베이시안 학습에서 기존의 속성에 가중치를 부여하는 방식에서 한 걸음 더 나아가 속성의 값에 가중치를 부여하는 새로운 방식을 연구하였다. 이러한 속성 값의 가중치를 계산하기 위하여 점진적 하강의 방법을 사용하여 가중치를 계산하는 방식을 제안하였다. 아울러 속성의 가중치를 계산하는 방법도 동시에 제안하여 두 가지 방법을 비교할 수 있게 하였다. 제안된 알고리즘은 다수의 데이터를 이용하여 속성 가중치 방식과 비교하였고 많은 경우에 더 좋은 성능을 제공함을 알 수 있었다. 본 연구는 나이브 베이시안의 가중치 부여에서 새로운 연구의 방향을 제시한다고 볼 수 있다. 추후연구로는 더욱 정밀한 속성값 가중치 계산 방법을 개발하여서 나이브 베이시안에 적용할 수 있는 기술을 개발할 계획이다.

참 고 문 헌

[1] Claire Cardie and Nicholas Howe. Improving minority class prediction using case-specific feature weights. In Proceedings of the *Fourteenth International Conference on Machine Learning*, pp.57-65, 1997.

[2] Peter Clark and Robin Boswell. Rule induction with CN2: some recent improvements. In EWSL-91: Proceedings of the *European working session on learning on Machine learning*, pp.151-163, 1991.

[3] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. Wiley-Interscience, New York, NY, USA, 1991.

[4] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3), 1997.

[5] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of *Thirteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1993.

[6] Thomas Gartner and Peter A. Flach. Wbcsvm: Weighted bayesian classification based on support vector machines, *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[7] Mark Hall. A decision tree-based attribute weighting filter for naive bayes. *Knowledge-Based Systems*, 20(2), 2007. 13

[8] P. Henrici. Two remarks of the kantorovich inequality. *American Mathematical Monthly*, 68:904-906,1961.

[9] Pat Langley and Stephanie Sage. Induction of selective bayesian classifiers. In Proceedings of the *Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399-406, 1994.

[10] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273-324, 1997.

[11] S. Kullback and R. A. Leibler. On information and suciency. *The Annals of Mathematical Statistics*, 22(1):79-86, 1951.

[12] C. Merz, P. Murphy, and D. Aha. UCI repository of machine learning databases. 1997.

[13] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[14] C. A. Ratanamahatana and D. Gunopulos. Feature selection for the naive bayesian classier using decision trees. *Applied Artificial Intelligence*, 17(5-6):475-487, 2003.

[15] Dietrich Wettschereck, David W. Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273, 1997.

[16] Harry Zhang and Shengli Sheng. Learning weighted naive bayes with accurate ranking. In ICDM '04: Proceedings of *the Fourth IEEE International Conference on Data Mining*, 2004.

이 창 환

e-mail : chlee@dgu.ac.kr

1982년 2월 서울대학교 계산통계학과(학사)

1988년 8월 서울대학교 계산통계학과(석사)

1994년 8월 University of Connecticut, Dept. of Computer Science(박사).

1982년 3월~1987년 2월 한국기계연구소

1994년 12월~1996년 2월 AT&T Bell Laboratories, Middletown, USA

1996년 3월~현 재 동국대학교 정보통신학과 교수

관심분야 : 기계학습, 마이닝, 인공진화 등

배 주 현

e-mail : baegop@pusan.ac.kr

1994년 2월 부산대학교 대기과학과(학사)

1999년 2월 부산대학교 환경시스템학과(석사)

2006년 2월 부산대학교 대기과학과(박사)

2006년 3월~현 재 동국대학교 정보통신학과 강사