정보이론을 이용한 K-최근접 이웃 알고리즘에서의 속성 가중치 계산

(Calculating Attribute Weights in K-Nearest Neighbor Algorithms using Information Theory)

이 창 환 †

(Chang-Hwan Lee)

요 약 최근접 이웃(k nearest neighbor) 알고리즘은 새로운 개체의 목표값을 예측하기 위하여 과거의 유사한 데이타를 이용하여 그 값을 예측하는 것이다. 이 방법은 기계학습의 여러 분야에서 그 유용성을 검증받아 널리 사용되고 있다. 이러한 kNN 알고리즘에서 목표값을 예측할 때 각 속성의 가중치를 동일하게 고려하는 것은 좋은 성능을 보장할 수 없으며 따라서 kNN에서 각 속성에 대한 가중치를 적절히 계산하는 것은 kNN 알고리즘의 성능을 결정하는 중요한 요소중의 하나이다. 본 논문에서는 정보이론을 이용하여 kNN 에서의 속성의 가중치를 효과적으로 계산하는 새로운 방법을 제시하고자한다. 제안된 방법은 각 속성이 목표 속성에 제공하는 정보의 양에 따라 가중치를 자동으로 계산하여 kNN 방법의 성능을 향상시킨다. 개발된 알고리즘은 다수의 실험 데이타를 이용하여 그 성능을 비교하였다.

키워드 : 최근접 이웃 알고리즘, 기계학습, 속성선택, 정보이론

Abstract Nearest neighbor algorithms classify an unseen input instance by selecting similar cases and use the discovered membership to make predictions about the unknown features of the input instance. The usefulness of the nearest neighbor algorithms have been demonstrated sufficiently in many real-world domains. In nearest neighbor algorithms, it is an important issue to assign proper weights to the attributes. Therefore, in this paper, we propose a new method which can automatically assigns to each attribute a weight of its importance with respect to the target attribute. The method has been implemented as a computer program and its effectiveness has been tested on a number of machine learning databases publicly available.

Key words : Nearest neighbor algorithm, Machine learning, Feature selection, Information theory

1. 서 론

최근접 이웃 알고리즘(k nearest neighbor algorithm, 이하 NN 알고리즘)은 이미 알려진 개체들을 훈련집합(training set)의 형태로 메모리에 기억한 다음 그 중 유사한 개체를 선택하여 선택된 개체의 값에 따라 새로운 개체의 값을 예측하는 방식의 분류 알고리즘이다. 이러한 NN 알고리즘은 다른 기계학습 알고리즘에 비하여 몇 가지 특징을 가진다. 첫째 학습되는 개념을 외부적으로 표현할 필요가 없다. 예를 들어 다른 학습 알고리즘들은 연역법칙 시스템(rule induction system)의 경우 정형화된 법칙으로 지식을 표현하고 있으며, C4.5[1] 혹은 CART[2] 등의 트리연역 시스템(tree induction system)은 결과를 결정트리(decision tree)등의 형태로 개념을 표현해야 한다. 이에 반하여 NN 알고리즘은 외부적으로 표현되는 개념은 존재하지 않으며 다만 도출된 결론에 대한 설명이 필요한 경우 유사한 개체를 제공함으로써 새로운 개체의 분류(classification)에 대한 설명의 기능을 제공할 수 있다. 이는 신경망(neural network) 알고리즘[3] 등 다른 기계학습 방법에서는 거의 제공할 수 없는 기능이다. NN 알고리즘은 최근 들어 여러 분야에서 적용되고 있으며 그 성능을 검증 받고 있는 기계학습 방법 중의 하나이다.

이러한 NN 알고리즘의 적용에 있어서 각 속성에 대하여 정확한 가중치 부여는 NN 알고리즘의 성능에 많은 영향을 미친다. 즉 모든 속성을 같은 비중으로 판단하게 되면 NN 알고리즘은 높은 성능을 제공할 수 없다.

† 종신회원 : 동국대학교 정보통신학과 교수
chlee@dgu.ac.kr
논문접수 : 2005년 1월 26일
심사완료 : 2005년 8월 1일

예를 들어서, 환자 데이타베이스에서 특정한 환자가 당뇨병을 가지고 있는지를 예측한다고 할 때 데이타베이스의 혈압 속성은 신장속성보다 더욱 당뇨병에 대하여 높은 연관을 가지며 따라서 높은 가중치를 부여하여야 한다. 반면에, 환자의 특정한 유전적인 병에 대하여 예측을 하는 경우, 그 환자의 혈압은 그의 신장보다 높은 가중치를 가지지 않을 수도 있다. 또 다른 예를 들면, 동물 데이타베이스에서 오리는 비행능력에 있어서는 거위와 많은 연관성을 가진다. 하지만, 동물의 목 길이가 예측하는 목적일 경우에는 오리는 거위와 거의 연관성을 가지지 않는다. 이와 같이, 데이타베이스에서 특정한 목적속성(target attribute)의 값을 예측/분류 하고자할 때 목적 속성을 제외한 다른 속성의 가중치는 목적 속성에 따라서 그 값을 달리하게 된다. 이와 같은 이유로 NN 알고리즘에서 속성의 가중치 계산은 중요한 문제중의 하나이며 이에 따라 많은 관련연구가 진행되어왔다.

본 논문에서는 정보이론(information theory)을 이용하여 각 속성의 가중치를 자동으로 계산하는 새로운 방법을 제안한다. 본 연구에서 제안하는 가중치의 계산방법은 기존의 방법과 달리 정보이론의 엔트로피 함수를 이용하여 각 속성이 목적값에 미치는 영향을 정량적으로 분석하며 이에 따라서 각 속성의 가중치를 계산한다. 따라서 각 속성의 가중치에 대하여 이론적인 배경을 제공하며 좀더 정확한 가중치 값을 제공할 수 있다.

2. 관련 연구

NN 알고리즘은 최초로 Cover 와 Hart[4]에 의하여 각각 독립적으로 제안되었다. 이 후 Smith와 Medin[5] 등에 의하여 NN 알고리즘은 논리학적으로 그 타당성을 인정받았지만 실제 알고리즘을 위한 모델은 개발되지 않은 상태였다. 이후 Aha, Kibler and Albert[6]에 의하여 몇 개의 개체중심 학습(instance-based learning, IBL) 알고리즘이 개발되었으며, 이 들은 각각 IB1, IB2, IB3 로 알려져있다. 가장 기본적인 IB1 알고리즘은 NN 알고리즘을 바탕으로 하여 속성의 값의 정규화, 알고리즘의 순차적 처리, 누락정보의 처리 등의 기능을 추가한 내용이다. IB2 알고리즘은 IB1 알고리즘과 거의 유사하지만 잘못 분류된 개체만 저장함으로써 기억장치의 요구를 대폭 감소시켰다. IB3 알고리즘은 IB2 알고리즘의 확장으로서 저장된 개체들이 얼마만큼 정확한 예측을 할 수 있는지를 계산한 후, 한 개체가 현재의 다른 개체에 의하여 정확하게 분류될 수 있는 경우에는 그 개체를 삭제되는 방법을 사용하고 있다.

또한, Zhang[7]은 개체간의 유사도 와 개체내의 유사도를 정의하여 계산된 대표적인 개체를 개념설명의 형태로 처리하는 알고리즘을 발표하였으며 Romaniuk[8]는 IBL 알고리즘의 정확도와 기억장치 요구를 감소시킨 다중 패스 알고리즘을 개발하였고, Skalak[9]은 몬테 카를로 샘플링(Monte Carlo sampling) 방법을 이용하는 대표적인 개체의 선택 방법을 제안하였다.

그리고 NN 알고리즘에서 속성의 중요도에 따라 가중치를 계산하는 문제에 대하서도 여러 가지 방법들이 제안되어있다. Salzberg가 제안한 EACH[10]에서는 학습 데이타의 분류이후에 정확한 분류에 대해서는 속성의 가중치를 일정한 양만큼 향상시키고 틀린 분류에 대해서는 반대로 일정한 양만큼 가중치를 감소하는 방법을 사용하였다. 이 방법을 이용한 가중치의 변경은 학습 데이타의 수행 순서에 따라 다르게 계산될 수도 있는 문제점이 있다. 이 방법은 추후에 IB4[11]의 방법으로 발전하였다.

Creecy[12] 등은 조건 확률을 사용하는 가중치 계산법을 제안하였다. 이들은 전체 속성을 이진(binary) 속성으로 전환하였으며 각 이진 속성에 대하여 PCF (per-category feature importance)를 계산하여 가중치를 계산하였다. PCF는 주어진 목적속성의 값들과 높은 상관관계(correlation)를 가진 속성일수록 높은 속성의가중치를 부여하였다.

Cost와 Stanfill이 제시한 PEBLS[13]에서는 MVDM (Modified Value Difference Metric) 방법을 제시하였는데 속성 값의 분포가 한 곳으로 치우칠수록 높은 가중치를 가지도록 설계되어있다. MVDM 에서는 두 데이타간의 거리는 각 데이타에서의 목적값의 분포에 따라 결정된다. 즉, 두 데이타에서의 목적값의 빈도수 분포가 일치 할수록 유사도는 증가하며 분포가 불일치 할수록 유사도는 감소하게 된다.

유전자 알고리즘을 이용하여 각 속성의 가중치를 계산하는 방법들도 제안되었다([9,14]). 이 방법은 속성의 가중치값들을 문제의 공간으로 하고 정확도를 적합함수로 하여서 GA 유전자들을 반복적으로 적용하여 속성의 가중치를 계산하였다. Kelly와 Davis의 방법([14])은 GA-WKNN이라는 알고리즘으로 구현되었으며 5 개의 유전자 연산을 정의하였고 3 개의 데이타를 이용하여 기존 방법보다 높은 성능을 나타냄을 보였다.

또한 정보 이론을 이용하여 속성의 가중치를 계산 방법이 일부분 제시된 바가 있다. Bosch와 Daelemans [15]은 정보획득량(information gain)을 이용하여 속성의 가중치를 계산하는 방법을 제시하였으며 단어의 하이펀(hyphenation)과 철자의 음소변환 등에 적용하여 좋은 결과를 보였다. 하지만 정보획득량 함수는 일부의 경우에 정확하지 않은 결과를 생성할 수 있는 문제점을 가지고 있다.

본 연구에서는 NN 알고리즘에서 속성의 가중치 자동 계산을 위하여 정보이론의 Hellinger 함수를 이용하며 다양한 종류의 속성에 대하여 가중치를 계산할 수 있는 새로운 방법을 제시한다.

3. NN 알고리즘의 내용

NN 알고리즘의 수행을 위해서는 속성의 가중치 계산, 속성값 간의 유사도 계산 등의 계산방법을 필요로 한다. 본 장에서는 논문에서 사용하는 NN 알고리즘의 기본적인 내용을 설명한다.

우선 X와 Y를 각각 k 개의 속성으로 구성된 데이타라고 하고, x_i와 y_i를 각각 X와 Y의 i 번째 속성값이라고 하자. 또한 T를 목적 속성이라고 가정하고 $\Delta_T(X,Y)$를 T에 대한 X와 Y의 유사도라고 할 때, $\Omega_T(X,Y)$는 다음과 같이 정의된다.

$$\Omega_T(X,Y)=\sum_{j=1}^{k}\omega_T(j)\cdot S_T(x_i,y_i)$$

여기서 $w_T(j)$는 i 번째 속성의 가중치이며 $S_T(x_i,y_i)$는 속성 값 x_i, y_i 간의 유사도이다. 위의 식에서 볼 수 있듯이 NN 알고리즘에서 유사도의 계산은 다음의 두 가지 단계로 구분된다.

1. 속성 값의 유사도($S_T(x_i,y_i)$)의 계산
2. 각 속성의 가중치($w_T(i)$) 계산

이와 같은 계산식들을 기반으로 새로운 데이타에 대하여 과거의 데이타에서 가장 유사도가 높은 데이타들을 선택하여 그들의 목적값을 이용하여 분류를 수행한다.

속성에 따른 유사도의 계산

속성 내에서 속성의 값들 간의 유사도를 계산하는 방법은 속성의 타입에 따라서 달라지게 된다. NN 알고리즘에서 사용하는 데이타의 속성은 다양한 종류로 표현되어있다. 크게 나누어서 데이타의 속성은 다음의 세 가지 타입중의 하나로 구성되어 있다: 이진(binary)속성, 연속(continuous)속성, 혹은 카테고리(category)속성. 본 논문의 주된 주제는 속성의 가중치 자동 계산에 대한 내용이지만 NN 알고리즘을 실제 구현하기 위해서는 데이타의 속성에 따른 유사도(혹은 거리)를 정의하여야한다. 아래에는 각 속성 타입에 따라 본 논문에서 그 유사도를 계산하는 구체적인 방법을 설명한다.

먼저 이진속성의 경우를 보면 임의 속성 A에서의 두 값을 x_i와 y_i라고하고 이 값들 간의 유사도를 $S_T(x_i,y_i)$라고할 때 이진 속성의 경우는 두 개체의 값이 같은 경우 1 그렇지 않은 경우 0을 부여한다.

$$S_T(x_i,y_i)=\begin{cases}1, & x_i=y_i\\ 0, & x_i\neq y_i\end{cases}$$

특정한 속성 A가 연속 변수인 경우는 두 변수를 n-차원의 유클리드 공간(Euclidean space, E^m)에서의 두 점으로 간주하는 방법이 가장 많이 사용한다. 따라서 두 개체간의 유사도는 두 점의 유클리드 거리에 반비례하는 개념을 사용하게 되는데 이 방법은 객체간의 유사도가 값의 범위(range)에 따라 큰 변동을 보이므로 유사도의 값을 정규화 시켜서 사용하게 된다.

$$S_T(x_i,y_i)=1-\left|\frac{x_i-y_i}{a_{\max}-a_{\min}}\right|$$

카테고리 속성에서의 유사도 계산을 위해서 지금까지의 전통적인 방법에서는 대체로 단지 두 값이 일치하는지 만을 검사하였다. 하지만 이 방법은 이진 속성과 유사한 방법을 사용함으로써 카테고리 속성의 특징을 제대로 활용하지 못하기 때문에 좋은 결과를 내지 못한다. 이해를 돕기 위해 다음의 예를 들어보겠다. 세 명의 학생이 각각 전산학, 전기공학, 음악을 전공한다고 가정하자. 수학적인 능력의 관점에서 전산학 학생은 음악 학생보다 전기공학 학생과 더욱 유사하다고 볼 수 있다. 전통적인 유사도에서는 단지 값의 일치만을 검사하기 때문에 이 경우 어느 누구도 다른 학생과 똑같은 유사도를 가지게 된다. 본 논문에서는 데이타값 들간의 유사도를 계산하기 위하여 현재 가장 많이 알려진 Stanfill과 Walz[16]의 VDM(Value Difference Metric)을 사용하여 데이타값들 간의 유사도를 계산한다. Stanfill과 Walz는 카테고리형(category type) 데이타에 대하여 유사도를 정의하였는데 그 들은 훈련집합(training set)으로부터 통계적인 방법을 사용하여 카테고리형 데이타의 모든 값의 조합에 대하여 유사도를 정의하였다. 이 들의 방법은 영어 발음인식(speech recognition)에 사용되어 인상적인 결과를 제공하였다. Stanfill과 Walz의 방법은 최근 Cost와 Salzberg[13]에 의하여 향상되었는데, 그들은 메모리에 있는 개체들에게 가중치를 부여함으로써 더욱 NN 알고리즘의 성능향상을 꾀하였다.

본 논문에서 사용하는 VDM 방법의 세부적인 내용은 다음과 같다. 목적속성은 t 개의 값(C_1, C_2, ... C_v)을 가지고 있고 각 속성은 p 개의 값을 가진다고 가정하자. 이 경우에 그림 1과 같은 도표를 가정할 수 있다.

이때 속성 A의 값 x_i와 y_i에 대한 거리 $S_T(x_i, y_i)$는 다음과 같이 정의된다.

$$S_T(x_i,y_i)=\sum_{k=1}^{t}\left|\frac{f_{ki}}{f_{*i}}-\frac{f_{kj}}{f_{*j}}\right|$$

지금까지 이진 속성, 연속속성, 카테고리 속성에 대하

	a_1	a_2	⋯	a_p	
C_1	f_{11}	f_{12}	⋯	f_{1p}	$f_{1\cdot}$
C_2	f_{21}	f_{22}	⋯	f_{2p}	$f_{2\cdot}$
⋮	⋮	⋮	⋯	⋮	⋮
C_t	f_{t1}	f_{t2}	⋯	f_{tp}	$f_{t\cdot}$
	$f_{\cdot 1}$	$f_{\cdot 2}$	⋯	$f_{\cdot p}$	$f_{\cdot\cdot}$

그림 1 VDM 방법의 계산 예제

여 유사도를 정의하였다. 비교되는 두 개체의 속성 값에 의해 정의된 유사도는 속성의 가중치에 의하여 곱하여지고 이 들이 각 속성에 대하여 합산된 값이 두 개체간의 최종 유사도가 된다.

4. 속성 가중치의 계산방법

이 장에서는 본 논문의 주된 주제인 NN 알고리즘의 속성가중치 계산방법에 대하여 자세히 설명하고자 한다. 본 논문에서 제안하는 속성의 가중치 계산 방법의 기본적인 내용은 다음과 같은 가정에서 출발한다. 데이타의 특정한 속성의 값은 목적속성에 정보를 제공하며 이와 같이 제공하는 정보의 양을 계산하여 가중치의 값을 자동으로 결정한다.

본 연구에서 제안된 속성의 가중치를 계산하는 방법은 다음의 세 가설에 바탕을 두고 있다: (1) 속성의 특정한 값이 정해지면 이는 목적 속성에 정보를 제공한다. (2) 제공되는 정보의 양은 엔트로피 함수(entropy function)에 의하여 정의될 수 있다. (3) 속성이 제공하는 정보의 양이 많을수록 엔트로피 함수의 값은 커진다. 이분(binary) 속성, 카테고리(category) 속성과 같은 이산형(discrete)의 속성에서 속성에 대한 가중치의 계산을 위해서는 먼저 속성의 각 이산형 값에 대한 정보량을 계산한 후 그 전체의 평균값을 해당 속성의 가중치로 사용하는 것이다. 이와 같은 방법의 문제점은 연속 속성의 경우 가중치를 계산하기 힘든 단점이 있다. 따라서 본 연구에서 제안된 방법은 연속 속성에 대하여 먼저 연속적인 수치데이타를 몇 개의 값의 범위로 분할하여 주는 이산화과정(discretization)을 거치고 난후 그 결과를 상대로 위의 방법을 적용하는 것이다.

이제 가장 중요한 물음은 '속성의 특정 값이 목적 속성에 제공하는 정보의 양을 어떤 방법으로 측정할 것인가' 이다. 정보의 양을 측정하기위한 방법은 다른 기계학습방법에 사용된 경우가 있으며 예를 들어서 C4.5의 경우에는 속성이 제공하는 정보의 양을 측정하기 위하여 정보획득량(information gain)이라고 불리는 아래의 엔트로피 함수를 사용한다. 즉 특정한 속성 A의 목적속성 T에 대한 정보의 양은 다음과 같다(H는 엔트로피값을 의미함). Bosch와 Daelemans[15]는 이와 같은 정보획득량 함수를 이용하여 속성의 가중치를 계산하는 방법을 제시한 바 있다.

$$I(T/T=a)=H(T)-H(T/A=a)$$

하지만 이와 같은 정보획득량 함수는 속성값의 이전확률분포(prior distribution)와 이후확률분포(posterior distribution) 모두를 고려하지만 이전확률분포와 이후확률분포가 대칭일 때에는 이를 구분하지 못하여 함수의 값으로 0을 생성한다. 즉 이전확률분표와 이후확률분포가 많은 차이를 보이더라도 대칭인 경우에는 그 차이를 구분하지 못하는 단점이 있다.

따라서 이 논문에서는 정보의 양을 측정하는 엔트로피 함수로서 Hellinger 변량(divergence)을 사용한다. Hellinger 변량은 최초로 Beran[17]에 의하여 제안된 이후 여러 분야에서 사용되고 있다(e.g. [18]).

목적 속성을 T라고 하고 $A=a$를 특정 속성 A의 값이 a가 됨을 나타낸다고 하자. t_i를 목적 속성 T값 중의 하나로 가정하고, $p(t_i)$와 $p(t_i|a)$를 목적 속성 T의 사전 확률(a priori probabilities) 및 사후 확률(a posteriori probabilities)로 가정한다. 그러면 $A=a$가 T에게 제공하는 정보의 양(Hellinger 변량)을 $H(T|A=a)$로 표시할 때 그 변량은 다음과 같이 정의된다.

$$H(T|A=a)=\left[\sum_i\left(\sqrt{p(t_i)}-\sqrt{p(t_i|A=a)}\right)^2\right]^{\frac{1}{2}} \quad (1)$$

이 변량은 목적 속성 T의 사전 확률분포와 사후 확률분포간의 차이를 측정한 함수이다. 이 변량의 특성에 대하여 조사하여 보면 우선 모든 경우의 $p(t_i)$와 $p(t_i|a)$ 값에 대하여 그 값이 정의가능(definable)하고 연속(continuous)이다. 또한 위의 Hellinger 변량은 사전 확률분포와 사후 확률분포가 일치할 때만 값이 0이 되며 나머지 경우는 속성의 사전 확률분포와 사후 확률분포의 차이에 따라 항상 0과 1 사이의 값을 가진다. 특정 속성A의 가중치 계산을 위하여 속성 A가 가지는 모든 값에 대하여 위에서 정의된 Hellinger 변량 값을 구하고 그 합한 값을 속성 A의 가중치로 사용할 수 있을 것이다. 하지만 이 경우 속성이 가지는 값의 개수가 증가하게 되면 위 변량의 값도 더불어 증가하게 된다. 이 문제를 해결하는 첫째 방법은 위 변량에 각 속성 값의 발생확률 $p(a)$를 곱하는 것이다.

$$H(T|A)=\sum_a p(a)\cdot H(T|A=a)$$

따라서 $H(T|A)$의 값은 속성의 값의 개수에 영향을 받지 않게 된다. 마지막으로 위에서 정의된 가중치의 범위를 0과 1 사이로 제한하기 위하여 모든 속성의 가중치 값의 합에 대한 비율로써 표현하였다. 최종적인 가중치의 값의 식은 다음과 같이 표현된다.

$$\omega_T(A) = \frac{H(T|A)}{\sum_A H(T|A)} = \frac{\sum_a p(a) H(T|A=a)}{\sum_A H(T|A)} \quad (2)$$

본 논문에서 제시하는 가중치 계산의 방법은 비교적 빠른 시간에 속성에 대한 가중치를 계산할 수 있다. 즉 속성의 수를 a 라고 하고, 속성 내에는 평균적으로 c 개의 값을 가지고 있고, 목적 속성은 t 개의 값을 가진다고 가정하면 위의 식 (1)과 (2)를 참고할 때 전체 속성의 가중치를 계산하는 복잡도는 $O(act)$임을 쉽게 알 수 있다.

그림 2의 예를 들어 속성의 가중치를 계산하는 방법을 설명하자.

Name	Sex	Eye	Blood type	Blood pressure	Weight	Diabetes
John	m	black	a	120	150	n
Ruth	f	brown	o	130	130	n
Michael	m	black	a	190	180	y
Tom	m	brown	o	90	110	n
Sue	f	brown	a	168	175	y
Robert	m	black	o	155	170	y
David	m	black	b	89	150	n
Jane	f	brown	o	140	110	n
Richard	m	brown	a	120	140	n
Steve	m	black	b	150	115	n
Mary	f	brown	o	192	179	?

그림 2 속성 가중치 계산의 예제

예를 들어 속성 Sex에 대한 가중치를 계산하여보자. 속성 Sex의 가중치를 $w_D(S)$라고 하면 식 (1)에 의하여 다음과 같이 정의된다.

$$w_D(S) = \frac{\sum_{a \in M,F} P(S=a) H(D|S=a)}{\sum_A H(D|A)}$$

위의 식을 계산하기 위한 조건확률들을 그림 3으로부터 계산하면 다음과 같다.

$P(S=m)=0.7$	$P(S=f)=0.3$
$P(D=y)=0.3$	$P(D=n)=0.7$
$P(D=y\|S=m)=0.29$	$P(D=n\|sex=m)=0.71$
$P(D=y\|S=f)=0.33$	$P(D=n\|sex=f)=0.67$

그림 3 조건확률의 예

따라서

$$H(D|S=m) = \sqrt{\sum_{t \in y,n} [\sqrt{P(D=t)} - \sqrt{P(D=t|S=m)}]^2}$$

$$= \sqrt{(\sqrt{0.3} - \sqrt{0.29})^2 + \sqrt{0.7} - \sqrt{0.71})^2} = 0.0109$$

같은 방법으로 $H(D|S=f) = 0.0323$ 으로 계산된다. 따라서

$$H(D|S) = 0.7*0.0109 + 0.3*0.0323 = 0.0173$$

비슷한 방법으로 다른 속성의 가중치도 계산할 수 있으며 최종적으로 정규화된 속성들의 가중치를 사용한다.

누락치의 처리

NN 알고리즘을 이용한 분류학습에 있어서 흔하게 발생하는 문제 중의 하나는 데이터에 누락치들이 존재하는 문제이다. 본 연구에서는 속성의 가중치를 계산할 때의 기본 원칙은 누락치도 한 개의 독립된 속성값으로 간주하여 계산하는 것이다. 즉 누락치가 가지고 있는 정보의 양을 계산한 다음 이를 속성내의 다른 값들에게 빈도수에 비례하여 분배하는 방식을 사용하였다. 예를 들어 특정한 속성 A가 k 개의 값$(a_1, a_2, \cdots, a_k, a_{k+1})$을 가지고 있으며 a_{k+1}을 누락치라고 가정하자. 누락치 가지고 있는 정보의 양은 $H(T|A=_{k+1})$이며 특정한 값 a_i이 가지는 정보의 양은 다음과 같이 추가된다.

$$H(T|A=a_i) = H(T|A=a_i) + H(T|A=a_{k+1}) * P(A=a_i)$$

따라서 위의 식과 수식 (1)을 이용하여 누락치가 포함된 속성의 가중치를 계산하게 된다.

선택된 개체들의 가중치 계산

제안된 알고리즘은 분류하고자하는 개체를 훈련 집합 내의 각 개체에 대하여 유사도를 계산한 후 가장 유사도가 높은 몇 개의 개체를 선택하여야한다. 선택되는 개체의 개수에 대하여 고려해 볼때, 오직 한 개의 가장 유사한 개체만을 선택하면 데이타베이스에 내재할지 모르는 에러나 노이즈(noise)에 심하게 영향을 받는 결과를 가져오므로 두 개 이상의 개체를 선택하여 그 들의 결과를 종합적으로 비교하는 것이 바람직하다. 그리고 여러 개의 선택된 개체 중에서 그 들의 유사한 정도에 따라서 가중치를 다르게 부여함으로써 좀더 분류의 정확성을 높일 수 있다. 대개의 경우 선택되는 개체의 개수는 사용자에 의하여 결정되며, 본 연구에서 제안된 알고리즘은 각 개체의 유사도를 각 개체의 가중치로 사용하여서 유사한 개체가 더욱 높은 가중치를 가질 수 있게 하였다.

5. 실험 결과

본 연구에서 제안된 가중치 계산의 방법은 NN 알고리즘의 기타 기능과 함께 C 언어로 구현되었다. 또한 그 성능을 평가하기 위하여 동일 데이타에서 (1) 가중치를 고려하지 않는 방법 (2) Bosch와 Daelemans[15]의 가중치 계산방법 들과 그 성능을 비교하였다. 가중치를 고려하지 않는 NN 알고리즘은 동일한 유사함수와 유사데이타 선택방식을 사용하며 유일한 차이는 모든 속성의 가중치를 1.0으로 동일하게 설정한 알고리즘이다.

성능 비교를 위하여 실험에 사용한 데이타는 University of California Irvine의 데이타 집합소([19])에서 다음 5 개의 데이타 집합을 선택하였다: Breast Can-

cer, Echocardiogram, Liver Disorders, Pima Diabetes, Voting. 알고리즘의 테스트는 훈련집합(training set)/테스트집합(test set)의 방법을 이용한다. 첫째 전체 데이타의 70%를 임의로 선택하여 훈련집합으로 하고 나머지 30%를 테스트집합으로 한다. 둘째로 훈련집합에 대하여 위에서 설명한 두개의 알고리즘들을 각각 수행한 후 테스트집합의 각 개체에 대하여 목적 속성의 값을 예측한다. 셋째로 테스트집합의 모든 개체에 대하여 예측결과를 실제 결과와 비교하여 그 정확도를 계산한다. 이 실험 과정은 편차를 줄이기 위하여 훈련집합과 테스트집합의 분할을 10 번 반복하여 수행하였다. 또한 데이타가 연속 속성을 포함하고 있을때에는 속성의 가중치 계산을 위하여 연속속성 값을 구간 값으로 변환하는 이산화를 필요로 한다(속성 값의 유사도 계산에서는 연속속성의 값을 그대로 사용함). 본 실험에서는 연속속성의 이산화를 위하여 편의상 동일빈도(equal frequency)방법을 사용하였다. 연속속성의 이산화 방법은 동일빈도 방법 외에도 많은 방법들이 제시되어있지만 본 연구의 실험비교에는 서로 같은 이산화 방법을 사용하는 한 영향을 주지 않으므로 동일빈도 이산화 방법을 사용하였다. 편의상 각 연속속성의 값은 5 등분하여서 이산화 하였다. 또한 유사 데이타의 개수(k 값)는 5를 사용하였다.

표 1은 Breast cancer 데이타에 대하여 계산된 속성의 가중치를 보여주고 있으며 표 2는 Echocardiogram 데이타의 각 속성에 대한 가중치를 보여주고 있다. Breast cancer에서는 cell의 uniformity가 cancer의 발생에 가장 큰 영향을 미치는 것을 보여주고 있으며 Echocardiogram 데이타에서는 Wall-motion이 높은 가중치를 가지고 있음을 보여주고 있다. 표 3은 각 데이타에 대하여 속성의 가중치를 고려한 알고리즘과 고려하지 않는 알고리즘 및 Bosch 방법의 정확도를 표준편차의 범위와 함께 보여주고 있다. 각 실험의 정확도 값은 데이타를 70/30 비율로 분할하는 경우를 10번 반복하여서 수행한 평균값을 의미한다. 표에서 보는바와 같이 모든 경우에 대하여 속성의 가중치를 자동 계산하는 알고리즘은 속성의 가중치를 고려하지 않는 방법에 비하여 높은 정확도를 보임을 알 수 있다. Bosch 값들에 비해서는 일부 Bosch 방법이 좋은 결과를 보이지만 대체적으로 비슷하거나 더 나은 성능을 보이고 있음을 알 수 있다.

이들 데이타 중에서 Breast cancer, Echocardiogram, Voting 데이타는 누락치를 포함하고 있으며 따라서 이들 데이타의 처리 결과는 본 연구의 계산방법이 누락치를 잘 처리하고 있음을 보여준다. 본 연구에서는 실험을 위하여 유사한 개체의 개수 k를 5로 사용하여 실험하였다. k 값의 변화에 따른 정확도의 변화를 관찰하기 위하여 Breast cancer 데이타에 대하여 k의 값을 1에서 8까지 변화시키면서 정확도의 변화를 관찰한 결과는 그림 4에서 보여준다. k의 값이 너무 작거나(예: 3 이하)

표 1 Breast cancer 데이타의 속성가중치

속성	가중치
Clump thickness	0.0988
Uniformity of cell size	0.1444
Uniformity of cell shape	0.1443
Marginal adhesion	0.0986
Single epithelial cell size	0.1127
Bare nuclei	0.1305
Bland chromatin	0.1207
Normal nucleoli	0.1040
Mitoses	0.0460

표 2 Echocardiogram 데이타의 속성 가중치

속성	가중치
Age-at-heart-attack	0.1410
Pericardial-effusion	0.0272
Fractional-shortening	0.0891
Epss	0.1868
Lvdd	0.1634
Wall-motion-score	0.1869
Wall-motion-index	0.2056

표 3 가중치 계산방법의 정확도 비교

데이타	가중치 사용(%)	가중치 미사용(%)	Bosch 방법(%)
Breast cancer	96.6 ± 1.3	94.3 ± 2.7	95.8 ± 1.2
Echocardiogram	90.7 ± 2.5	81.6 ± 2.9	92.3 ± 1.8
Liver Disorders	69.4 ± 3.2	62.5 ± 3.8	65.7 ± 2.9
Pima Diabetes	74.6 ± 1.7	70.3 ± 2.4	76.4 ± 2.2
Voting	96.2 ± 2.6	89.8 ± 1.9	95.6 ± 1.8

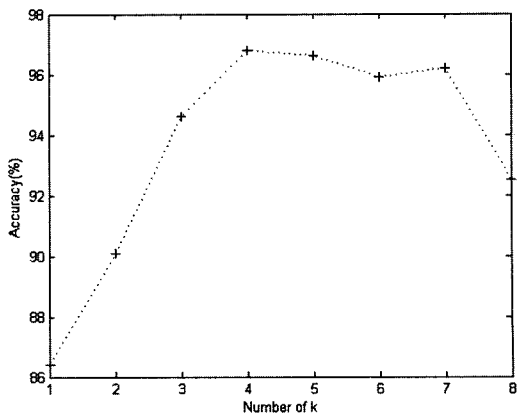

그림 4 k 값에 따른 정확도의 변화

너무 크면(예: 7 이상) 정확도가 상대적으로 감소하는 것을 알 수 있다.

6. 결 론

본 논문에서는 NN 알고리즘의 성능에 많은 영향을 미치는 속성의 가중치 계산방법을 새롭게 제시하였다. 새로운 속성 가중치 계산 방법은 정보이론을 바탕으로 엔트로피 함수의 일종인 Hellinger 변량을 사용하여 개발되었다. 제안된 방법의 효율성을 검정하기 위하여 우리는 5 가지의 데이타베이스를 선택하여 실제 적용하여 보았고 그 결과를 가중치를 고려하지 않는 방법 및 다른 가중치 계산 방법과 비교하였다. 제안된 가중치 계산 방법은 가중치를 고려않는 경우에 비하여는 모든 경우에 대하여 더욱 나은 성능을 보이고 Bosch 방법에 대하여도 좋은 성능을 보임을 알 수 있었다.

본 연구에서 개발된 알고리즘 개발의 특징적인 요소인 속성 가중치의 자동부여 방법은 NN 알고리즘의 응용범위를 대폭 확장시킨다. 이는 다른 기계학습 방법에 비하여 적은 비용으로 비교적 높은 정확성을 보이는 NN 알고리즘의 중요성에 비추어볼 때, 기계학습 분야에 향상을 기대할 수 있을 것으로 기대한다.

참 고 문 헌

[1] Quinlan, J. R. *"C4.5: Programs for Machine Learning,"* San Mateo, CA:Morgan Kaufmann, 1993.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *"Classification and Regression Trees,"* Monterey, CA: Wadsworth International Group, 1984.

[3] S. Haykin, "Neural Networks: A Comprehensive Foundation," Prentice Hall, 1999.

[4] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Vol. 13, 1967.

[5] E. E. Smith and D. L. Medin, *"Categories and Concepts,"* Cambridge, MA: Harvard University Press, 1981.

[6] D. Aha, D. Kibler and M. Albert, "Instance-based Learning Algorithms," Machine Learning, 6(1) pp. 37-66, 1991.

[7] J. Zhang, "Selecting typical instances in instance-based learning," *Proceedings of the Ninth Int. Machine Learning Conference*, pp. 470-479, Aberdeen, Scotland: Morgan Kaufmann, 1992.

[8] S. Romaniuk, "Efficient Storage of Instances: The Multi-pass Approach," *Proc. of the Seventh Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert systems*, Austin, TX, 1994.

[9] D. Skalak, "Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms," *Proc. of the 11th International Conference on Machine Learning*, New Brunswick, NJ, 1994.

[10] S. Salzberg, "A Nearest Hyperrectangle Learning Method," *Machine Learning*, 6, pp. 251-276, 1991.

[11] D. Aha, "Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms," *Int'l Journal of Man-Machine Studies*, 36, pp. 267-287, 1992.

[12] R. Creecy, B. Masand, S. Smith, and D. Waltz "Trading MIPS and Memory for Knowledge Engineering," *Communications of the ACM*, 35, pp. 48-64, 1992.

[13] S. Cost and S. Salzberg, "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features," *Machine Learning*, 10, pp. 57-78, 1993.

[14] J. D. Kelly and L. Davis, "A Hybrid Genetic Algorithm for Classification," *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence*, pp. 645-650, Sydney, Australia: Morgan Kaufmann, 1991.

[15] van den Bosch, A. and Daelemans, W. "Data Oriented-Method for Grapheme-to-Phoneme Convertsion" Technical Report, Tilburg, Netherlands, Tilburg University: Institute for Language Technology and Artificial Intelligence, 1993.

[16] C. Stanfill and D. Waltz, "Toward Memory-based Reasoning," *Communications of the ACM*, 29(12), pp. 1213-1228, 1986.

[17] R. J. Beran, "Minimum Hellinger Distances for Parametric Models," *Ann. Statistics*, Vol. 5, pp. 445-463, 1977.

[18] Z. Ying, "Minimum Hellinger Distance Estimation for Censored Data," *Annals of Statistics*, Vol. 20, No. 3, pp. 1361-1390, 1992.

[19] P. Murphy and D. Aha, *UCI Repository of Machine Learning Databases*, Irvine, CA:University of California Irvine, Department of Information and Compter Science, 1993.

이 창 환

1982년 2월 서울대학교 계산통계학과 졸업(학사). 1988년 8월 서울대학교 계산통계학과 졸업(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~현재 동국대학교 정보통신학과 부교수. 관심분야는 기계학습, 마이닝, 생물정보학 등