속성 값 빈도 기반의 전문가 다수결 분류기
(Committee Learning Classifier based on Attribute Value Frequency)

이 창 환 † 정 인 철 †† 권 영 식 †††
(Chang-Hwan Lee) (In Chul Jung) (Young S. Kwon)

요 약 센서 정보, 물류/유통정보, 신용 정보, 주식 정보 등이 과거보다 다양하면서 대용량의 연속 발생 형태 데이터가 발생하고 있다. 이러한 데이터는 대용량의 특성의 변화가 빠른 특징들을 가지고 있기 때문에 학습이 어렵다. 이러한 문제점을 해결하기 위해 일정 윈도우 크기의 최근 데이터를 연속적으로 학습시킴으로써 전체 모형을 새롭게 만들거나 모형의 일부분을 대체 하는 방법을 사용하여 왔다. 그러나 이러한 방법은 계속해서 새로운 학습모형을 만들어야 하므로 대용량의 연속 데이터를 학습시키는데 많은 시간과 비용이 든다. 따라서, 이러한 특성에 대비하기 위하여 추가적인 학습 데이터가 발생할 때 마다, 점진적이며 지속적으로 학습을 할 수 있는 학습 기법이 필요하다. 보다 빠른 속도로 학습 모형의 변화 없이 분류를 하기 위하여 대표적인 점진적 학습 방법으로 베이지안 분류기를 사용할 수 있지만, 사전확률을 알고 있다는 가정으로부터 시작을 하게 되어 일정량 이상의 학습데이터가 필요하다.

따라서 본 연구에서는 베이지안 분류기와 같이 점진적으로 학습을 할 수 있지만, 사전 확률을 알지 못하더라고 학습을 할 수 있는 새로운 점진적 학습 알고리즘을 제안하고자 한다. 본 연구에서 제안하는 알고리즘의 기본 개념은 여러 전문가의 의견을 종합하는 방식이다. 여기서는 속성값(attribute value)을 한명의 전문가로 보고 전문가 집단의 의사 결정이 맞을 경우에는 가점을 주고 틀릴 경우에는 감점을 하는 방식으로 학습을 하게 된다. 실험결과 이 방법은 의사결정나무나 베이지언 분류기와 비교해 비슷한 성능을 나타내었으며, 향후에 스트림 데이터 분석에 사용할 가능성을 보였다.

키워드 : 데이타마이닝, 학습기, 다수결원리, 전문가 학습기

Abstract In these day, many data including sensor, delivery, credit and stock data are generated continuously in massive quantity. It is difficult to learn from these data because they are large in volume and changing fast in their concepts. To handle these problems, learning methods based in sliding window methods over time have been used. But these approaches have a problem of rebuilding models every time new data arrive, which requires a lot of time and cost. Therefore we need very simple incremental learning methods. Bayesian method is an example of these methods but it has a disadvantage which it requries the prior knowledge(probabiltiy) of data.

In this study, we propose a learning method based on attribute values. In the proposed method, even though we don't know the prior knowledge(probability) of data, we can apply our new method to data. The main concept of this method is that each attribute value is regarded as an expert learner, summing up the expert learners lead to better results. Experimental results show our learning method learns from data very fast and performs well when compared to current learning methods(decision tree and bayesian).

Key words : Data Mining, Majority vote, Commitee Learning

· 본 연구는 한국연구재단(NRF)의 중견연구자 사업(과제번호: 2009-0079025) 및 2009년도 동국대학교 연구 년 지원에 의하여 이루어졌음

† 종신회원 : 동국대학교 정보통신공학과 교수
chlee@dgu.ac.kr
†† 학생회원 : 동국대학교 산업시스템공학과
uhhaha@gmail.com
††† 정 회 원 : 동국대학교 산업시스템공학과 교수
yskwon@dgu.edu

논문접수 : 2009년 11월 30일
심사완료 : 2010년 6월 28일

정보과학회논문지: 데이타베이스 제37권 제4호(2010.8)

1. 서 론

인터넷, RFID 및 USN기술 등의 발전과 함께 센서 정보, 물류/유통정보, 신용 정보, 주식 정보 등이 과거보다 다양하면서 대용량의 연속적인 데이터가 발생하고 있다[1,2]. 최근의 이러한 데이터는 크게 두 가지 특징을 가진다. 첫째는, 처리할 데이터량이 대용량(large-volume)이라는 것이다. IDC의 최근 보고서에 따르면, 2010년 국내에서 생성 및 복제되는 디지털 정보량의 규모는 2006년 대비 무려 약 7배 늘어난 1만 5700PB에 달할 것으로 전망하고 있다. 이는 11개월에 2배씩 늘어나던 데이터 증가 속도가 2010년이면 11시간마다 두 배씩 늘어날 것이라고 한다[3]. 그러나 데이터 처리 및 분석을 위한 리소스(예, 메모리, 저장 공간)는 한계가 있기 때문에, 분석을 위한 리소스 분배가 점점 어려워지고 있다. 비록 데이터 저장 기술의 발전으로 인해 메모리 가격이 하락하고 있다고 하지만 무한대의 저장 용량을 사용할 수는 없기에 지속적인 대용량 데이터 발생으로 인한 분석의 어려움이 있다. 두 번째로, 시간이 경과함에 따라 이전 데이터와 최신 데이터의 성격이 변화가 일어나는 컨셉트 변화(concept drift)이다[4]. 컨셉트 변화 현상 때문에 과거의 데이터로 학습을 한 모형을 이용하여 미래의 사건을 분류하거나 예측할 때 정확도가 떨어지게 된다. 일반적으로 학습을 통한 분류 및 예측 기법은 tradeoff를 갖고 있다. 예를 들어, 정확한 성능을 위해서는 충분한 학습 데이터가 확보되어 일정 시간이상을 들여 연산 학습을 수행하게 되며, 적은 량의 학습 데이터와 빠른 연산을 통한 예측 분류는 어느 정도의 성능 감소를 감수하게 된다. 특히, 최근의 대용량 연속 데이터는 다양한 컨셉트의 변화로 인해, 기존 정적인 형태의 데이터 학습을 통한 정확도를 기대하기 힘들다. 따라서, 이러한 특성에 대비하기 위하여 추가적인 학습 데이터가 발생할 때 마다, 점진적이며 지속적으로 학습을 할 수 있는 학습 기법이 필요하다.

그러나 이전의 대표적인 분류 예측 학습 알고리즘인 의사결정나무 기법의 경우, 연속적인 데이터를 학습하기 위해서는, 일정한 학습 윈도우 크기를 정하여 발생 데이터를 학습하여 분류 모형을 생성한다. 그 후 주기적으로 재학습 과정을 수행하거나, 컨셉트 변화가 발생할 때마다 기존 모형을 완전히 버리고 새로이 학습 모형을 생성 하면서 실시간의 새로운 데이터에 대해 적용을 한다. 이러한 경우에는 새로이 발생한 데이터가 추가될 경우 처음부터 다시 학습과정을 거치며 지속적으로 새로운 모델을 생성하기 때문에 많은 량의 계산과 시간을 필요로 하게 된다. 이를 해결하기 위해서 지속적으로 일정한 크기의 최근 발생 데이터를 이용하여 새로운 모델을 개발하면서 기존 모델을 부분적으로 대체 하는 방식을 취할 수 있다[4-7] 이에 비해서 보다 빠른 속도로 학습 모형의 변화 없이 분류를 하기 위하여 대표적인 점진적 학습 방법으로 베이지안 분류기를 사용할 수 있다. 그러나 베이지안의 경우, 사전확률을 알고 있다는 가정으로부터 시작을 하게 되어 일정량 이상의 학습데이터가 필요하다.

따라서 본 연구에서는 베이지안 분류기와 같이 점진적으로 학습을 할 수 있지만, 사전 확률을 알지 못하더라고 학습을 할 수 있는 새로운 점진적 학습 알고리즘을 제안하고자 한다. 본 연구에서 제안하는 알고리즘의 기본 개념은 여러 전문가의 의견을 종합하는 방식이다. 여기서는 속성값(attribute value)을 한명의 전문가로 보고 전문가 집단의 의사 결정이 맞을 경우에는 가점을 주고 틀릴 경우에는 감점을 하는 방식으로 학습을 하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 데이터 분류를 위한 관련 연구를 소개한다. 3장에서는 제안하고자 하는 알고리즘의 기본 아이디어를 기술한다. 4장에서는 제안한 알고리즘과 기존의 분석 알고리즘과의 성능을 비교하였다. 마지막 5장에서는 결론 및 향후 연구방향에 대해서 기술한다.

2. 관련연구

기존 마이닝 관련 연구에서는 지속적으로 일정크기의 최근 발생 데이터를 새롭게 학습하면서 기존 모델을 전체 혹은 부분적으로 대체 하는 방식이 연구되었다. 이때 의사결정나무 기법이나 다중 계측 네트워크를 이용한 연구 기법들이 있으나, 이러한 학습 알고리즘은 학습 모형이 완성된 후, 새롭게 발생한 데이터를 추가할 경우 처음부터 다시 학습과정을 거쳐야 하므로 학습 시 많은 량의 계산이 필요로 하게 된다.

2.1 베이지안 분류기

베이지안 분류기는 베이즈 정리(Bayes' theory)에 기반 하여 속성들 간의 독립성을 가정한 확률적인 모델이다. 단순한 특성을 사용하지만 널리 사용 하고 있는 전통적인 분류방법으로, 텍스트 문서 분류에 주로 사용되어 왔다. 베이지안 분류기는 통계적인 알고리즘으로 학습문서의 여러 통계 정보를 학습하고, 이렇게 얻은 통계 정보를 이용하여 입력 문서 스트림으로부터 문서를 분류한다. 확률이론을 기계학습에 적용한 것으로, 특정 데이터 집합 D를 조사했을 때, 가설 h가 사실일 확률은 P(h|D)가 된다. 그리고 가설이 사실일 경우 데이타 D의 확률이 P(D|h)일 때, 베이즈 정리는 다음 식과 같다.

$$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

위 식에서 P(h)는 데이터에 관한 정보가 주어져 있지 않을 때, 가설이 사실일 사전확률(prior probability)이다. 기계학습에서 관심을 가지는 값은 P(h|D)인데, 베이지안 학습방법은 가설집합 H에 포함된 가설 중 최대 확률을 가지는 가설 h를 구하는 것이다.

최대 확률을 구하기 위해서는 최대사후확률을 계산하면 된다. 이 확률은 데이타를 조사 했을 때 가장 가능성이 높은 가정으로서 다음 식을 이용한다.

$$h = \arg\max_{h \in H} P(h|D)$$
$$= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

위의 식의 argmaxP(h|D)는 확률 P(D|h)가 최대가 되는 H내에서 가설 h를 나타내는데 이때 분모 P(D)는 h와 무관하기 때문에 상수로 간주 하여 다음 식과 같이 표현된다.

$$h = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

2.2 VFDT와 CVFDT

스트림 데이터의 분류 기법으로 대표적으로 트리기반의 VFDT 기법이 연구되었다. VFDT는 트리기반의 구조를 사용함으로써 스트림 데이터를 분류 한다. 그러나 VFDT 는 컨셉트 변화가 갑자기 발생할 경우 정확도가 떨어지는 문제가 발생을 하였고 이에 대한 해결을 위해, Hulten은 스트림 데이터의 컨셉트 변화에도 민감한 VFDT를 확장한 CVFDT(Concept adapting VeryFast Decision Tree)를 제시하였다. CVFDT의 학습 방법은 새로운 컨셉트 변화가 발생할 경우 데이터의 패턴을 반영한 새로운 부분 트리를 새롭게 만들고 기존의 트리와 주기적인 비교를 통해 정확률을 평가한다[8,9] 이를 기준으로 하여 기존의 부분 트리보다 새로 구축한 부분트리가 최근 데이터 패턴을 더 잘 분류할 때, 기존의 부분트리를 새로 구축한 부분트리로 교체를 한다. 이와 같은 방식은 때에 따라서는 부분 트리 뿐만이 아니라 모든 트리 전체가 바뀌게 될 수 있는 단점이 있다. 이는 많은 량의 계산 량을 필요로 하기 때문에 긴급을 요하는 실시간의 데이터에 적용하기에는 문제가 될 수 있다.

2.3 정보 퍼지 네트워크(IFN)와 온라인 정보 네트워크(OLIN)

정보 퍼지 네트워크(IFN; Info-Fuzzy Network)는 정보 이론(Information Theory)을 배경으로 속성들 간의 다중 계층 네트워크를 구축하는 데이터 분류 모델로써 IFN은 정보 이론을 사용한 방법들 중 하나로서 개발 되었다[7]. IFN은 Mutual Information이라는 입력속성들과 출력 속성 사이의 상호정보를 계산하기 위하여 다중 계층 네트워크(multi-layered network)를 구축하여 학습을 수행한다. 입력속성들과 출력속성 사이의 상호 정보 값(MI)을 구하여 가장 높은 값을 갖는 속성이 첫 번째 층이 되고, 이후 값이 0이 될 때까지 조건적인 상호 정보 값이 높은 순서대로 층을 구성한다. 상호 정보 값이 0이라면 노드는 더 이상 분리되지 않고 어느 한 클래스로 분류 된다. 그렇기 때문에 모델은 조건적인 상호 정보값이 0이 되거나 더 이상 계층을 만들 수 있는 속성이 없을 때까지 모델을 생성하게 된다. 이 시스템은 네트워크로서 의사결정트리(decision tree)와 같이 목표 속성(target attribute)을 예측하는데 사용되지만, 모든 단말노드(leaf node)들은 목표 층(target layer)의 모든 노드들과 네트워크로 연결되는 차이점을 갖고 있다. 그림 1은 3계층 구조의 IFN을 보여준다. 단말 노드 2, 1.2, 1.1.1,1.1.2는 목표 노드 0, 1과 연결됨으로써 네트워크 구조를 이룬다. 이러한 점이 의사결정트리 구조와의 차이점이다. IFN은 분류를 결정하는데 필요한 입력 속성들의 수가 다른 알고리즘들에 비해 상대적으로 적어 빠른 데이터의 처리가 가능하다. 그 이유는 분류를 결정짓는 속성들을 잘 선별함으로써 입력 속성의 수를 줄이기 때문이다.

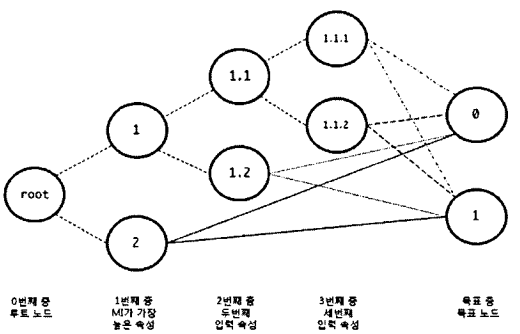

그림 1 3개층 구조의 정보 퍼지 네트워크

Last는 IFN을 온라인 상에서 구현한 OLIN 시스템을 제안하였다[6]. IFN을 기반으로 한 OLIN 시스템은 컨셉트 변화에 따라 윈도우의 크기를 동적으로 조절하여 연속적인 스트림 데이터를 처리를 하고 있다. 데이터 스트림 상에서 컨셉트 변화가 발생한다면, 윈도우의 크기를 줄임으로써 컨셉트 변화에 따른 오류율을 줄이고 새로운 IFN을 구축한다. 정확률이 높으면 컨셉트 변화가 발생하지 않았다고 판단하여 윈도우 크기를 확대하고, 새로운 IFN을 구축한다. 이러한 동적인 윈도우 크기의 변화는 고정된 크기를 가진 윈도우보다 더 높은 정확률을 보여 주었다[10]. 그러나 이 역시 데이터 변화에 따라 계속적인 모델의 변형 및 대치가 필요하다.

3. 제안 알고리즘

3.1 기본 아이디어

제안하는 알고리즘은 다수 전문가들의 판단을 종합할 경우 더 나은 의사결정을 할수 있다는 전제에 기반을 두고 있다. 학습모형 하나하나를 전문가로 두는 앙상블 기법(ensemble technique)과 비슷하나 차이점은 제안한 알고리즘은 속성값(attribute value) 하나 하나를 전문가로 둔다는 점이다. 속성값에 해당하는 목적 변수값(target value)이 정분류될 경우 그 속성을 강화하고, 오분류될 경우 약화시키는 방식으로 학습을 진행한다. 다시 말해 각 속성의 속성값을 전문가로 보아 각각의 전문가가 판단한 다수의 결과를 최종결과로 산출하는 것이다. 그림 2는 이에 대한 설명을 표현한 그림이다.

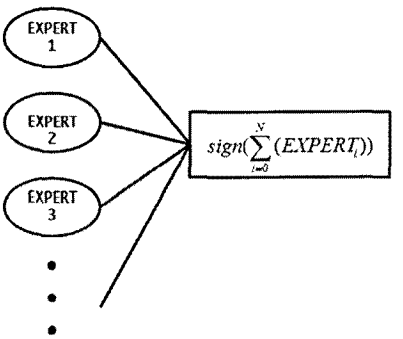

그림 2 전문가 학습기 결합

예를 들어, X와 Y를 속성(attribute)으로 구성된 데이터라 하고, x_i와 y_i를 각각 속성 X와 Y의 i번째 속성값이라고 하자. k개로 구성되어있는 목적 속성(target class)을 T이라고 하고, t_k를 k번째 목적 속성 값을 가졌다고 하자. $E_X(x_i)$와 $E_Y(y_i)$를 각각 X와 Y의 속성값마다 생성된 전문 분류기로 정의 한다. 모든 학습데이터를 학습을 하고 난후에 $E_X(x_i)$는 속성 값 x_i마다 목적 속성 T의 값 t_k에 대한 발생 빈도수를 유지한다. $E_Y(y_i)$의 경우도 마찬가지로 y_i마다 학습데이터 학습 후의 T의 값 t_k에 대한 빈도수를 유지 한다. 이처럼 모든 속성 값 기반으로 하여 학습한 전문 분류기들을 기반으로 하여 새로운 데이터에 대하여 분류를 수행하게 된다.

새로운 데이터에 대한 분류를 위해, 분류를 하고자 하는 새로운 데이터인 x_i와 y_i가 제시 되게 되면 학습을 통해 완성이 된 전문가 분류기들 중에 해당하는 $E_X(x_i)$와 $E_Y(y_i)$를 선택하고, 각각의 전문가 분류기가 분류한 목적 속성인 t_k의 빈도수를 합한다. 모든 t_k에 대한 총합한 결과 값 중 가장 큰 값을 가지는 목적 속성 t_k를 최종의 분류 결과로 도출을 한다.

예를 들어 목적 속성 T가 (t_1,t_2)와 같이 2개의 값으로 이루어진 목적 속성이라고 한다면,

$E_X(x_i)$에서 t_1의 빈도수 + $E_Y(y_i)$에서의 t_1의 빈도수 (1)

$E_X(x_i)$에서 t_2의 빈도수 + $E_Y(y_i)$에서의 t_2의 빈도수 (2)

(1)식과 (2)의 결과 값 중 가장 큰 값을 가지는 t_i를 새로운 데이터에 대한 분류 결과로써 수행을 한다.

즉, 학습데이터를 이용하여 각각의 모든 속성 값마다 목적 속성 값의 빈도를 누적하면서 유지하는 전문 분류기를 가지고 있으며, 새로운 데이터에 대한 분류 시 해당 데이터에 대응하는 학습된 전문 분류기를 합산 조합하여 가장 많은 영향을 미치고 있는 목적 속성을 분류하는 것이다.

그런데, 설명한 바와 같이 속성값을 기반으로 전문가 분류기를 각각 생성을 하여 분류를 할 경우, 학습 데이터의 량에 따라 분류의 성능이 좌우된다. 따라서, 학습도중 정분류를 한 경우에는 전문가 분류기의 예측 값에 강화를 시켜주고, 오분류를 했을 경우에는 전문가 분류기의 예측값에 약화를 시켜주는 최적화 기법이 필요하다. 예를 들면 정분류나 오분류에 따라 일정한 값, β 값을 이용하여 가감을 하는 방안이 필요하다.

$$E(x) = \text{frequence of } y \pm \beta$$

그러나, 이경우에도 데이터 집합의 량이 무한대로 커지면 빈도수가 너무 큰 값이 형성이 되게 되므로, β를 통해서 보강이 되는 것이 미미할 수 있다. 따라서 판단식을 일반화 시킬 필요가 있는데, 이를 위해 단순빈도수를 다음과 같이 변형을 할 수 있다.

$$E(x) = \sqrt{frequency\ of\ y} \pm \beta$$

빈도 값에 루트를 사용한 이유로는 측정된 빈도수에 따라 미치는 영향을 보정하기 위해서 사용하였다. 즉, 분류기가 얼마나 일반적인지를 보여주는 항목으로서 빈도수를 그냥 사용할 경우 빈도 값 n은 그냥 선형적으로 계속 증가하지만 sqrt(n) 은 데이터의 갯수가 초기인 경우에 더욱 많은 가중치를 제공한다(그림 3 참조). 따라서 좀 더 현실의 의미에 적합하다. sqrt(n)에서 데이터의 갯수의 중요성은 처음에 발생할 때 가장 중요성이 높으며, 그 다음으로 발생할 때 조금씩 중요도가 떨어진다. n은 무조건 데이터의 갯수에 비례해서 가중치를 부여하므로 sqrt(n)이 좀더 현실적인 가중치를 제공한다.

본 논문에서 사용한 방식 대신 다양한 방식의 최적화 기법을 사용할 수 있으나 본 논문에서는 전문가 집단의 다수 판단을 통해 결정을 하는 것을 보여주고자 가장 단순한 방식으로, 정확히 분류하였을 경우에 β 더해주

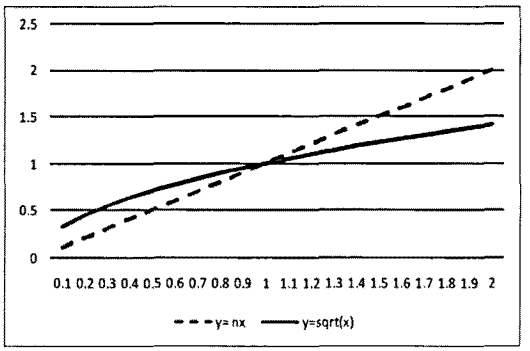

그림 3 선형그래프와 sqrt 그래프의 차이

```
input : trains, tests, beta = 0.08

loop (attribute X, attribute Y, target T) in trains :
    E_X(x_i).T_k ++
    E_Y(y_i).T_k ++
    E_X(x_i).val_k = sqrt(E_X(x_i).T_k)
    E_Y(y_i).val_k = sqrt(E_Y(y_i).T_k)

loop (attribute X, attribute Y, target T) in tests :
    loop (target T) :
        tmp_k = E_X(x_i).val_k + E_Y(y_i).val_k
    select k of tmp_k in tmp_k values
    if (k == T) :
        E_X(x_i).val_k = E_X(x_i).val_k + β
        E_Y(y_i).val_k = E_Y(y_i).val_k + β

return E_X(x_i), E_Y(y_i) | i = 0,1,....,n
```

그림 4 의사 코드

는 방식을 사용 하였다. 그림 4는 이에 대한 의사 코드로써 표현을 할 것을 보여주고 있다.

제안한 알고리즘은 이와같이 한번 형성된 학습 모형이 시간의 지남에 따라 변형이 이루어지지 않고, 가볍고 간단한 판별식을 통해 판단을 할 수 있기 때문에, 비록 단순한 판별식으로 인해 정밀한 정확도는 약간 떨어질 수도 있지만, 실시간의 대용량의 스트림데이터의 경우 적합한 이점을 갖고 있다. 이는 기존의 알고리즘은 데이터의 학습을 하는 도중 몇 번의 모형 변화로 인한 계산 시간이 필요하지만 제안한 방법의 경우 그럴 필요가 없기 때문에 보다 빨리 분류를 할 수 있을 것으로 판단된다.

3.2 제안 학습 방법의 AND 연산 데이터 적용 예

간결한 설명을 위하 다음과 같은 학습 데이터가 있다고 가정을 하겠다. 표 1는 2개의 속성값으로 이루어진 AND연산을 표현한 데이터이다. AND 연산은 2개 속성

표 1 AND 연산 학습 데이터

No	1st attribute	2st attribute	Target Class
1	True	True	True
2	True	False	False
3	False	True	False
4	False	False	False

모두 True인 경우에 결과 값이 True이며, 그 외의 경우는 모두 False이다. 따라서 총 4개의 인스턴스를 갖고 있고 결과 값은 True 혹은 False 2가지의 경우만 존재하는 집합이다.

제안한 알고리즘은 모든 속성의 실제 값마다 전문가 분류기를 생성하기 때문에, 첫 번째 속성에 생성될 전문가는 True에 대한 전문가 1과 False에 대한 전문가 2를 생성하게 된다. 두 번째 속성 역시 True와 False에 대한 전문가 3과 전문가 4를 생성한다. 그림 5는 데이터 속성 값들의 전문가 집단이 생성되어 초기화 되어 있는 모습을 나타낸다. 그림 5에서와 같이 속성의 실제 값(True or False)에 대해서 전문가들이 생성이 되고, 학습 진행하면서 전문가에는 해당 속성 실제 값일 경우의 결과 값이 True인지 False인지에 따라 발생 빈도수를 가중하게 된다.

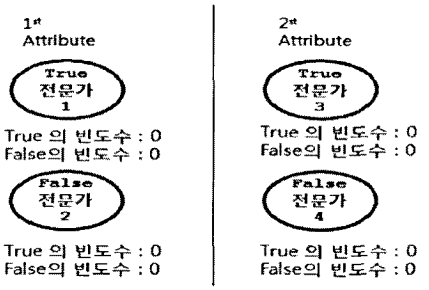

그림 5 제안 알고리즘의 전문가 생성 초기

표 1의 데이터 집합을 학습함에 따라 학습 모형의 변화 과정을 그림 6~그림 10으로 나타내었다. β는 0.08로 하였고 정확히 분류하였을 경우에만 강화를 시키는 것으로 예시를 하도록 하겠다.

그림 9와 같이 최종의 학습 결과를 얻게 되면, 이를 기반으로하여 분류기 성능을 실험해 볼 수 있다. 예를 들어 True & False인 경우의 예측을 위해서 전문가 1과 전문가 4를 이용해서 판단을 하게 된다. True로 판단되기 위해서는 전문가 1의 True일 경우의 f값 $\sqrt{1}+0.08$와 전문가 4의 f 값 0의 합이, False로 판단될 경우 전문가 1의 f값 $\sqrt{1}+0.08$과 전문가 4의 f값 $\sqrt{2}+0.08$ 합보다 작기 때문에 False로 분류하게 된다.

전문가 1	전문가3
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 0	False의 빈도수 : 0
False의 E = *0*	False의 E = *0*
전문가2	전문가4
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 0	False의 빈도수 : 0
False의 E = *0*	False의 E = *0*

그림 6 초기 전문가 모델

전문가 1	전문가3
True의 빈도수 : 1	True의 빈도수 : 1
True의 E = $\sqrt{1}+0.08$	True의 E = $\sqrt{1}+0.08$
False의 빈도수 : 0	False의 빈도수 : 0
False의 E = *0*	False의 E = *0*
전문가2	전문가4
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 0	False의 빈도수 : 0
False의 E = *0*	False의 E = *0*

그림 7 AND 연산 학습 데이터의 1번 학습

전문가 1	전문가3
True의 빈도수 : 1	True의 빈도수 : 1
True의 E = $\sqrt{1}+0.08$	True의 E = $\sqrt{1}+0.08$
False의 빈도수 : 1	False의 빈도수 : 0
False의 E = $\sqrt{1}+0.08$	False의 E = *0*
전문가2	전문가4
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 0	False의 빈도수 : 1
False의 E = *0*	False의 E = $\sqrt{1}+0.08$

그림 8 AND 연산 학습 데이터의 2번 학습

$$\sqrt{1}+0.08+0<\sqrt{1}+0.08+\sqrt{2}+0.08$$

(True의 가능성) < (False의 가능성)

전문가 1	전문가3
True의 빈도수 : 1	True의 빈도수 : 1
True의 E = $\sqrt{1}+0.08$	True의 E = $\sqrt{1}+0.08$
False의 빈도수 : 1	False의 빈도수 : 1
False의 E = $\sqrt{1}+0.08$	False의 E = $\sqrt{1}+0.08$
전문가2	전문가4
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 1	False의 빈도수 : 1
False의 E = $\sqrt{1}+0.08$	False의 E = $\sqrt{1}+0.08$

그림 9 AND 연산 학습 데이터의 3번 학습

전문가 1	전문가3
True의 빈도수 : 1	True의 빈도수 : 1
True의 E = $\sqrt{1}+0.08$	True의 E = $\sqrt{1}+0.08$
False의 빈도수 : 1	False의 빈도수 : 1
False의 E = $\sqrt{1}+0.08$	False의 E = $\sqrt{1}+0.08$
전문가2	전문가4
True의 빈도수 : 0	True의 빈도수 : 0
True의 E = *0*	True의 E = *0*
False의 빈도수 : 2	False의 빈도수 : 2
False의 E = $\sqrt{2}+0.08$	False의 E = $\sqrt{2}+0.08$

그림 10 AND 연산 학습 데이터의 4번 학습

4. 실험 및 결과

4.1 실험 방법

본 논문에서 제시한 학습 알고리즘의 성능을 비교하기 위해서 나이브 베이즈 분류기, C4.5를 통한 의사결정 나무 기법을 함께 비교하였다. UCI Machine Learning Repository에서 제공하는 실험 데이터를 사용하였으며, 임의로 10개의 데이터 셋을 뽑아내어 각 알고리즘을 적용하여 수행하고 비교 검증하였다. 데이터의 속성 개수와 목적 속성의 개수는 표 1에 표현하였다. 다양한 속성의 개수와 클래스 개수에 따라 성능의 차이가 일반 알고리즘과 성능 상 어느 정도 차이가 나는지 확인하기 위하여, 나이브베이지 분류기와 C.4.5 의사결정 나무 분류기법을 사용하였다. 실험을 위해 나이브베이즈 분류기 및 C4.5 알고리즘은 공개 데이터마이닝 소프트웨어인 Weka을 사용하여 실험하였고, 제안한 알고리즘은 자체 구현하였다. 그림 6은 본 연구의 실험 과정을 도식화한 것이다. 즉, 데이터 집합에 알고리즘을 적용하기 위하여 데이터 공통적인 전처리 과정을 거친 후, 다른 알고리즘과 비교를 하기 위하여 C4.5 의사결정나무와 나이브베이지안 분류기와 비교 실험을 하였다. 이때 성능을 비교 평가하고 결론을 도출하였다. 그림 11은 연구 과정에 대해 도식화한 것이다.

데이터 전처리 과정을 위해, 연속형 데이터를 구간형으로 이산화(discretization)을 하였으며, 이때 사용된 방법으로는 MDL[11]을 사용하여 supervised 기법을 사용하였으며, 전체 데이터 셋의 수가 부족하여 이산화과 한

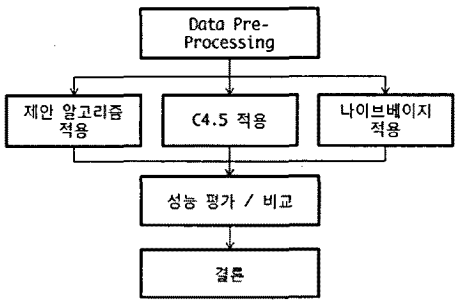

그림 11 연구 실험 방법

쪽으로 치우친 경우, unsupervised 기법으로 10개 단위 구간으로 구간화를 하였다. 또한 제안 알고리즘의 β 값을 0.08으로 하여 테스트를 하였다.

4.2 성능 평가 및 비교

표 2는 데이터 속성 및 목적 속성의 설명을 나타낸 표이다. 속성은 최소 5개부터 최대 279개까지의 데이터에 대해 학습을 비교해보도록 하겠다. 표 2의 데이터를 이용하여 학습한 결과는 표 3에서 볼 수 있다.

표 3에서 보는 바와 같이 나이브베이지안과 의사결정나무에 비해, 성능상은 크게 떨어지지 않고 비슷한 성능을 내는 것을 실험을 통하여 확인할 수 있었다. 이에 대

표 2 실험 데이터 설명

No	데이터 이름	속성수	목적 속성수
1	iris	5	3
2	labor relation	16	2
3	University	17	2
4	ZOO	17	7
5	Congressional Voting Records	17	2
6	Mushroom	22	2
7	Thyroid Disease	29	2
8	Dermatology	33	6
9	annealing	39	6
10	Arrhythmia	279	16

표 3 알고리즘 성능 비교

No	제안 알고리즘	C4.5	나이브 베이지안
1	95	96	94
2	99	92	94
3	98	98	98
4	93	99	100
5	96	96.9	91
6	92	100	95
7	96	98	97
8	91	98	98
9	97	99	96
10	85	75	92

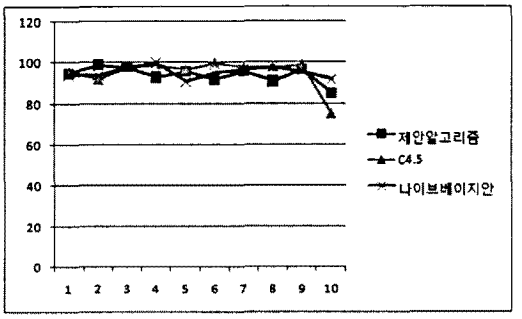

그림 12 C4.5와 나이브베이지안 및 제안 알고리즘 성능 비교

한 알고리즘 성능을 그래프로 비교하면 그림 12와 같다. x축은 표 2 데이터 집합의 순번을 나타낸 것이며, y축은 예측 정확도를 나타낸 그래프이다.

제안한 알고리즘은 의사결정나무와 나이브 베이지언 알고리즘에 비해 속성값 발생 빈도를 기반한 단순한 알고리즘 구성과 복잡한 수리모형 대신 단순한 수리모형이 사용되었음에도 충분한 성능을 발휘한 것이라고 할 수 있다. 실제 데이터 환경에서 대용량의 실시간 데이터를 처리하기 위해서 기존의 학습 방식은 많은 연산을 함으로써 발생되는 비용이 높지만, 제안한 학습 방식은 단순하지만 어느 정도의 정확도 수준을 유지하면서도 비용이 덜 드는 것이다.

결론적으로 제안 알고리즘은 기존 알고리즘에 비해 큰 차이를 보이지 않음을 확인할 수가 있다. 또한 실시간 데이터를 학습한다고 하여도 기존의 방법들처럼, 알고리즘 모형을 변경하여 적용을 하고나, 일정한 크기의 데이터를 나누어 알고리즘 적용을 할 필요 없이, 한번 생성된 속성 값 기반의 모형은 추가적인 변형 없이 계속적인 데이터 인스턴스의 생성에도 적용이 가능하다.

5. 결론 및 향후 과제

본 연구에서는 기존 학습 알고리즘과는 완전히 다르면서 대용량의 스트림 데이터를 안정적으로 처리하면서도 변화에 민감하게 대처할 수 있는 새로운 학습 알고리즘에 대해 제안을 하였다. 기존의 학습알고리즘이 단순 속성에 기반한 방식이라면 제안하는 방식은 속성 값 빈도에 기반한 방식으로, 새로운 데이터가 발생할 때마다 새로운 모형을 만드는 것이 아니라 모형의 파라미터만을 변경시킴으로 계산 속도를 획기적으로 줄일 수 있으므로 특히 데이터 량이 많은 스트림 마이닝에 적합하며, 특히 기존 일반적인 학습 알고리즘인 C4.5와 나이브베이지언과 비교를 해보아도 정확도에서 큰 차이가 나지 않음을 알 수 있었다. 이는 속성 값 기반의 전문가

분류기를 생성하여 전문가 집단을 구성한 뒤의 다수의 분류 결정을 이용하여 가능한 것으로 판단된다.

향후에는 보다 좋은 성능 향상을 위해, 목적 속성과 속성 값들 간의 상관 관계를 계산하고, 이를 이용하여 보다 높은 가중치를 분별하여, 제안 알고리즘을 이용하면 보다 성능이 향상된 분류기 개발이 가능할 것으로 기대된다. 또한 실제 현업 데이터의 적용을 통해 실 생활에 적용가능한지 여부를 확인해 볼 수 있을 것이다.

참 고 문 헌

[1] G. Widmer and M. Kubat, Learning in the Presence of Concept Drift and Hidden Contexts, Machine Learning, vol.23, no.1, pp.69-101, 1996.

[2] C. Aggarwal, A Framework for Diagnosing Changes in Evolving Data Streams. *Proceedings of the ACM SIGKDD Conference,* 2003.

[3] J.F. Gantz et al,. The Expanding Digital Universe: A Forecast of Worldwide Information Growth through 2010, IDC Whitepaper, March 2007.

[4] A. Tsymbal, The problem of concept drift: definitions and related work, Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Ireland, 2004.

[5] C. Aggarwal, Data Streams: Models and Algorithms, Springer, p.354, 2007.

[6] M. Last, "Online Classification of Nonstationary Data Streams," *Intelligent Data Analysis*, vol.6, no.2, pp.129-147, 2002.

[7] L. Cohen, M. Last, G. Avrahami, "Incremental Info-Fuzzy Algorithm for Real Time Data Mining of Non-Stationary Data Streams," *TDM Workshop*, Brighton UK, 2004.

[8] G. Hulten, L, Spencer, and P. Domingos, "Mining Time-Changing Data Streams," *Proc. of KDD 2001*, ACM Press, pp.97-106, 2001.

[9] P. Domingos and G. Hulten, "Mining high-speed data streams," In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.71-80, Boston, MA, 2000. ACM Press.

[10] J. W. Kim, J. W. Song, J. H. Lee, "Data Streams classification using Local Concept-adapted IOLIN System," *Proc. of the KIISE Korea Computer Congress 2008*, vol.13, no.1(C), pp.37-44, 2008. (in Korean)

[11] Fayyad, Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, France, 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 교수. 관심분야는 기계학습, 마이닝, 생물정보학 등

정 인 철

2002년 2월 시립인천대학교 인문대 졸업(학사). 2005년 2월 동국대학교 산업시스템공학과 졸업(석사). 2006년 9월~현재 동국대학교 산업시스템 공학과 박사과정. 관심분야는 기계학습, 데이터 마이닝, 에이전트, 지능형 정보시스템 등

권 영 식

1978년 2월 서울대학교 산업공학 졸업(학사). 1981년 2월 한국과학기술원 산업공학 졸업(석사). 1996년 8월 한국 과학기술원 졸업(박사). 1981년~현재 동국대학교 산업시스템 공학 교수. 관심분야는 데이터 마이닝, 지능형 정보 시스템