
동적 중요도 결정 방법을 이용한 새로운 앙상블 시스템

서동훈* · 이원돈**

A New Ensemble System using Dynamic Weighting Method

DongHun Seo* · Won Don Lee**

요 약

본 논문에서는 분류자들 속에 중요도 정보를 삽입하여 동적 중요도 결정이 가능한 앙상블 시스템을 제안하였다. 그동안 앙상블 시스템에서 중요도는 훈련이 끝나고 결정된 중요도를 사용하였다. 한 번 결정된 중요도는 테스트 데이터에 상관없이 정적으로 사용되었다. 이 문제를 푸는 방법으로 관문 네트워크에서 구조적으로 계층을 두는 프로세스를 추가하여 동적 중요도 결정이 가능하게 하는 방법이 있지만 프로세스가 추가된다는 단점이 있다. 본 논문에서는 이런 추가적인 프로세스 없이 간단하게 동적 중요도 결정이 가능한 방법을 보여주고 구조적 변경 없이 기존의 시스템에 쉽게 적용할 수 있으며 AdaBoost보다 나은 성능을 보여주는 알고리즘을 제안한다.

ABSTRACT

In this paper, a new ensemble system using dynamic weighting method with added weight information into classifiers is proposed. The weights used in the traditional ensemble system are those after the training phase. Once extracted, the weights in the traditional ensemble system remain fixed regardless of the test data set. One way to circumvent this problem in the gating networks is to update the weights dynamically by adding processes making architectural hierarchies, but it has the drawback of added processes. A simple method to update weights dynamically, without added processes, is proposed, which can be applied to the already established ensemble system without much of the architectural modification. Experiment shows that this method performs better than AdaBoost.

키워드

앙상블 시스템, 데이터 마이닝, 부스팅, 관문 네트워크

Key word

Ensemble system, Data mining, Boosting, Gating network

* 정회원 : 충남대학교 대학원 박사과정 (sm1835dh@hanmail.net)

접수일자 : 2011. 04. 27

** 정회원 : 충남대학교 전기정보통신공학부 교수

심사완료일자 : 2011. 05. 31

(교신저자, wdlee@cnu.ac.kr)

I. 서 론

앙상블 시스템이란 두 개 이상의 분류자들을 사용하여 결과를 예측하는 데이터 마이닝 기술 중 하나이다. Hansen 과 Salamon은 1990년에 신경 회로망들로 배열된 시스템을 통해 한 개의 신경 회로망을 사용하는 것보다 더 나은 성능을 보여주었다.[1] 이 논문에서 다수의 분류자들을 사용하는 것이 일반적인 성능을 개선할 수 있다는 것을 보여주었고 앙상블 시스템의 가능성을 보여주었다. Schapire는 부스팅을 통해서 약한 분류자들을 조합하여 강한 분류자에 근접한 성능을 보여줄 수 있다는 것을 증명하였다.[2] Freund와 Schapire는 [2]에서 제안한 알고리즘을 체계적으로 정리하여 AdaBoost 알고리즘을 개발하였다.[3][4] 앙상블 시스템 분야에서 AdaBoost는 매우 중요한 역할을 하였다. 첫째로 앙상블 시스템의 기본 모형을 제공하였다. 둘째로 강한 분류자를 만들기 어려운 분야에서 약한 분류자들의 조합만으로 성능을 개선할 수 있다는 장점을 보여주었다. 이런 장점들은 앙상블 시스템의 학문적 성장에 도움을 주었고 많은 연구자들은 다양한 이름의 앙상블 시스템을 제안하였다. 이런 연구들로는 동적 분류자 선택[5], 분류자 통합[6], 분류자들의 투표합[7], 다양한 분류자들의 조합[8]-[11], 분류자 앙상블들[12],[13], 전문가들의 혼합[14],[15] 등이 있다.

앙상블 시스템은 크게 2가지 부분으로 나눌 수 있다. 첫번째는 부스팅(Boosting)으로 약한 분류자를 통해 훈련 데이터에서 분류를 잘못하는 부분을 강조하여 새로운 훈련 데이터를 만들어 내는 것이다. 이렇게 만들어진 훈련 데이터는 다음 번 약한 분류자를 생성하는데 사용되어진다. 새롭게 생성된 분류자는 이전 분류자보다 강조한 부분을 더 잘 분류할 수 있다.[16] 두번째는 Breiman이 제안한 배깅(Bootstrap aggregating의 약자)으로 분류자들의 결과를 투표하여 결론을 도출하는 방법으로 앙상블 시스템의 기본이 되는 기술이다.[17] 앙상블 시스템에서 분류자들을 조합하는 방법은 성능에 직접적으로 큰 영향을 주는 부분이다. 분류자들을 조합하는 방법은 부류(Class)가 값들(labels)로 이루어진 경우와 부류가 연속적인 값을 가지는 경우로 나눌 수 있다.

전문가들의 혼합 분야에서 연구된 방법들은 대부분 관문 네트워크(gating network)를 사용하여 구현되어 있

다. [14],[15] 훈련 데이터로 분류자를 훈련하는 동안 분류자의 중요도를 결정하는 기능을 만드는데 이것을 관문 네트워크라고 한다. 이 때 생성된 관문 네트워크는 테스트 데이터를 통해 각각 전문가들의 중요도를 결정하는데 사용된다. 입력된 데이터에 따라 중요도가 매번 바뀌기 때문에 동적으로 중요도를 결정하는 점이 한번 결정된 중요도를 계속 사용하는 기존의 앙상블 시스템과 다른 점이다. 구조적인 관점에서 관문 네트워크는 기존의 앙상블 시스템에 한 개의 층을 더 쌓은 모양을 가지고 있고 신경망 모양과 흡사한 모양을 가지고 있다. 관문 네트워크는 훈련 데이터를 통해 EM(expectation maximization) 알고리즘을 사용하여 훈련한다.

본 논문에서는 전문가들의 혼합 분야에서 분류자 안에 훈련 데이터의 정보를 삽입하고 구조적인 변경 없이 입력된 데이터에 따라 분류자의 중요도를 스스로 결정하여 사용하는 새로운 앙상블 시스템을 제안한다. 분류자 안에 삽입하는 정보는 훈련 데이터에서 중요도를 계산하여 훈련 과정에 만들어진다. 의사 결정 트리를 분류자로 사용하여 실험을 하였고, 잘 알려진 앙상블 시스템 중의 하나인 AdaBoost와 성능을 비교하였다. 본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로 AdaBoost에 대해서 소개하고자 한다. 제 3장에서는 제안된 방법을 설명하고자 한다. 제 4장에서는 기계 학습 분야에서 사용되는 UCI Machine Learning Repository을 사용하여 실험하였다.[18] UCI Machine Learning Repository에는 알고리즘들의 비교를 위해 무료로 데이터들을 공개하고 있으며 이 중의 7개의 데이터들을 가지고 AdaBoost와 제안된 알고리즘을 비교 분석하여 연구 결과를 얻고 향후 연구방향을 제시하고자 한다.

II. 관련 연구

2.1 AdaBoost

Freund와 Schapire가 제안한 AdaBoost 알고리즘은 앙상블 시스템의 대표적인 알고리즘 중 하나이다. AdaBoost의 약한 분류자들을 조합하여 강한 분류자를 만들 수 있다는 개념은 강한 분류자에만 집중되어 있던 학문적 관심을 넓혀주는 계기가 되었다. 실제 환경에서는 강한 분류자를 생성하기 어렵고 약한 분류자는 생성

할 수 있는 경우가 많기 때문에 AdaBoost는 데이터 마이닝 분야에 중요한 업적으로 평가 되고 있다.

그림. 1은 [4]에서 제안한 AdaBoost의 전체적인 프로세스를 보여준다. D_1 은 m 개의 event들을 가진 훈련 데이터가 있을 때 $1/m$ 의 동일한 확률 분포를 가진 데이터를 나타낸다. S_1 은 앞에서 설명한 D_1 에서 이벤트 각각의 확률 분포를 통해 중복을 허용한 무작위로 추출한 훈련 데이터를 의미한다. 새롭게 만들어진 훈련 데이터에는 중복된 이벤트들이 존재할 수 있다. 분류자는 훈련 데이터를 통해 훈련을 하게 되며 이 때, 약한 분류자를 생성하도록 한다. 생성된 분류자 C_1 를 통해 S_1 을 테스트 데이터로 사용하여 예측하면 약한 분류자이기 때문에 잘못 예측되는 이벤트들이 있다. 잘못 예측된 이벤트들의 확률 분포 값들은 부스팅 과정을 통해 증가시킨다. 이 때, 확률 분포를 증가시키는 기준은 에러율을 가지고 결정한다. 에러율이란 잘못 예측된 이벤트들의 개수와 m 개의 이벤트들의 비율로 결정되며 그림. 1에서는 α_1 으로 정의하였다. 잘못 예측된 이벤트들의 확률 분포 값들은 증가하고 상대적으로 잘 예측된 이벤트들의 확률 분포 값들은 감소하도록 조정이 한다. α 값을 통해 조정된 확률 분포는 정규화 과정을 통해 확률 분포의 합을 1로 유지한다. α 값은 분류자의 중요도를 의미하며 α 값이 $1/2$ 값을 넘는 경우, 분류자가 제대로된 기능을 못한 것으로 간주하여 해당 분류자를 버리고 그동안 만들어진 분류자들을 가지고 배깅 단계로 넘어간다. 조정된 확률 분포를 갖는 D_2 를 이전 과정과 같이 중복을 허용한 무작위 추출하여 훈련 데이터 S_2 를 만들고 S_2 를 이용하여 약한 분류자 C_2 를 만든다. 반복된 과정을 t 번 반복하여 약한 분류자 t 개를 만든다. 분류자들로부터 최종적으로 예측을 하기 위해서 배깅 과정을 거친다. 이 과정에서는 입력된 이벤트를 만들어진 분류자들(C_1, \dots, C_k)에 입력하여 나온 결과들을 각각의 부류별로 합하여 가장 큰 값을 가진 부류가 결과로 도출된다. 예를 들어 부류1이 0.4이고 부류2가 0.3 부류3가 0.9이면 부류3가 최종 결과로 결정되며 부류3가 예측되었다고 나타낸다. 그림 1에서는 부류가 긍정과 부정 2개만 나타내고 있기 때문에 배깅 과정에서 부류의 합이 양수이면 긍정, 음수이면 부정의 결과를 도출한다.

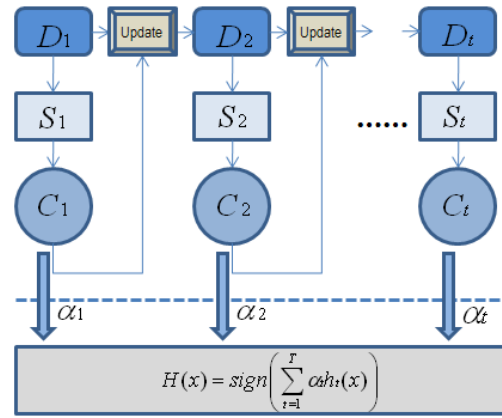


그림. 1 앙상블 시스템 (AdaBoost)
Fig. 1 AdaBoost

2.2. 관문 네트워크

앙상블 시스템의 연구 분야 중 하나인 전문가들의 혼합 분야에서 연구된 방법들은 대부분 관문 네트워크 (gating network)를 사용하여 구현되었다. 관문 네트워크란 앙상블 시스템을 구성하는 다수의 분류자들을 분류를 위하여 조합할 때, 각각의 분류자들의 weight 값을 입력 값에 의존하여 결정하는 네트워크를 말한다. 일반적인 앙상블 시스템들은 훈련 중 각각의 분류자들에게 중요도들을 할당한다. 중요도는 훈련 중 만들어진 분류자의 성능에 따라 결정되기 때문에 분류자들의 중요도들은 일반적으로 서로 다른 값들을 가지게 된다. 하지만 이 방법은 분류자의 중요도가 한번 결정되고 나면 입력 값에 상관없이 중요도가 분류자를 대표하게 되는 문제점을 가진다. 전형적인 앙상블 시스템에선 부스팅 과정을 통해 분류자들을 얻기 때문에 분류가 어려운 훈련 데이터를 부스팅을 통해 생성한다면 결과적으로 분류자의 에러율이 증가하게 된다. 분류가 어려운 훈련 데이터를 입력으로 사용하여 만들어진 분류자는 에러율이 높기 때문에 다른 분류자와 비교하여 중요도가 낮게 설정될 가능성이 높아진다. 앙상블 시스템에서 중요도가 다른 분류자보다 낮게 설정된다는 것은 분류자의 의견이 전체 결과에 적은 영향을 미친다는 의미이다. 앙상블 시스템에 있는 다수의 분류자 중 분류가 어려운 이벤트들에 집중해서 만들어진 분류자가 다른 분류자보다 특정 이벤트에 대해서 더 잘 분류할 수 있다는 것이 관문 네트워크의 관점이다. 관문 네트워크에서는 입력 값에 따른 계

층을 하나 더 더듬으로써 이 문제를 해결하였다. 관문 네트워크에서는 훈련 데이터로 분류자를 훈련하는 동안 분류자의 중요도를 결정하는 모듈을 만드는데 이것을 관문 네트워크라고 한다. 이 때 생성된 관문 네트워크는 테스트 데이터를 통해 각각 전문가들의 중요도를 결정하는데 사용된다. 기존의 앙상블 시스템과 다른 점은 입력된 데이터에 따라 중요도가 매번 바뀌기 때문에 동적으로 중요도를 결정하는 점이 한 번 결정된 중요도를 계속 사용하는 기존의 방식과 차별화된 차이이다. 구조적인 관점에서 관문 네트워크는 기존의 앙상블 시스템에 한 개의 층을 더 쌓은 모양을 가지고 있고 신경망 모양과 흡사한 모양을 가지고 있다. 관문 네트워크는 훈련 데이터를 통해 EM(Expectation maximization) 알고리즘을 사용하여 훈련한다.

III. 연구 설계

3.1 동적 중요도 결정 방법을 이용한 새로운 앙상블 시스템

관문 네트워크 알고리즘을 제안하기 전까지 앙상블 시스템에선 훈련을 통해 한 번 만들어진 분류자들의 중요도는 더 이상 변하지 않도록 되어 있다. 관문 네트워크에서는 테스트 이벤트마다 분류자들의 중요도를 구조적인 방법으로 층을 쌓아서 해결하였다. 하지만 이 방법은 모든 데이터에 대해서 훈련을 해야 하며 앙상블 시스템을 구성하기 위해서 계층을 따로 만들어야 하는 문제점을 가진다. 또한, 모든 데이터를 가지고 있어야 관문 네트워크를 훈련할 수 있기 때문에 점진적으로 분류자를 추가해 나가는 점진 학습(incremental learning) 분야에서는 관문 네트워크 알고리즘을 사용할 수 없는 단점을 가지고 있다. 본 논문에서는 중요도를 이벤트마다 결정할 수 있는 방법을 제안함으로써 앙상블 시스템이 구조적으로 계층을 두지 않고도 동적으로 중요도를 구할 수 있다는 것을 보여준다.

그림 2는 본 논문에서 제안하는 방법으로 ExVote라 명명하였다. 앙상블 시스템에서 부스팅을 통해 분류자들을 만드는 경우, 각각의 분류자는 훈련 데이터에 종속적인 성향을 가지고 있다. 그러므로 부스팅은 특정한 데이터에 집중적으로 훈련된 분류자를 생성하는 것을 의미한다. 이렇게 생성된 분류자는 현실 세계에서 전문가

와 같은 의미를 가진다. 전문가에게 자기 분야가 아닌 일반적인 상식을 질문한다면 전문가들도 잘못된 예측을 할 수 있다. 입력된 이벤트가 있었을 때, 이 이벤트가 자신의 분야인지 알 수 있는 방법은 훈련 데이터의 확률 분포를 통해서 추론할 수 있다. ExVote에서는 $p(xi)$ 가 분류자가 어떤 이벤트들에 집중되어 있는지를 나타내주는 척도로 사용하였다. $p(xi)$ 를 구하기 위해서 $count(Dt)$ 와 $h'(xi)$ 라 정의한 2개의 함수를 사용하였다. Dt 는 훈련 데이터가 가지는 확률 분포를 의미한다. 모든 데이터는 초기 값으로 균일 분포 값을 가지고 있다. m 개의 이벤트들로 구성된 훈련 데이터에서 한 개의 이벤트가 가지는 중요도, 즉 확률 값은 $1/m$ 이 된다. $count(Dt)$ 는 훈련 데이터에서 중요도가 $1/m$ 을 넘는 경우, 부스팅 과정에서 이전 분류자가 제대로 분류하지 못한 이벤트들을 다음 분류자가 집중적으로 다룰 수 있도록 중요도를 증가한 이벤트들의 개수를 의미한다. 이 값들은 각각의 분류자의 중요도를 결정할 때, 상대적 비교를 위해 구하였다. $h'(xi)$ 는 xi 란 이벤트가 입력 되었을 때, 최종적으로 결정된 공간을 가르기 위해 사용된 훈련 데이터 중 중요도가 $1/m$ 보다 큰 이벤트들의 개수를 의미한다. 의사 결정 트리를 예로 설명하면 최종 노드(leaf node)에 포함되는 훈련 데이터 중 중요도가 $1/m$ 보다 큰 이벤트들의 개수이다. 분류자에 테스트 이벤트 xi 를 집어넣어 결정된 공간 안에 있는 이벤트들에서 중요도가 $1/m$ 보다 큰 이벤트들이 많다는 것은 분류자는 이 공간을 중요하게 보고 결정했다는 것을 의미한다. 테스트 이벤트에 따라 분류자가 결정하는 공간이 바뀌게 되므로 분류자의 중요도는 어떤 훈련 데이터가 결정된 공간을 나누는데 영향을 미쳤는가에 초점을 맞추어야 한다.

결론적으로 $p(xi)$ 는 하나의 분류자에게 테스트 이벤트 xi 를 집어 넣었을 때, 분류자가 결정된 공간에 있는 중요도가 높은 훈련 데이터의 이벤트 개수와 전체 훈련 데이터에서 중요도가 높은 훈련 데이터의 이벤트 개수를 통해 구할 수 있다. 그렇기 때문에 $p(xi)$ 는 테스트 데이터에 따라 동적으로 중요도가 바뀌게 된다.

앙상블 시스템에서 $p(xi)$ 를 적용하는 방법은 그림 2의 맨 마지막 수식으로 표현하였다. 부스팅을 통해 구해진 분류자들은 테스트 이벤트 xi 가 입력되면 동적으로 결정된 각각의 $p(xi)$ 를 갖게 되고 이 값들은 지수함수를 통해 중요도가 높은 분류자의 결정이 최종 결과에 크게 작용할 수 있도록 하였다.

알고리즘 ExVote

입력 :

$$S = [(x_1, y_1), \dots, (x_m, y_m)],$$

$$x_i \in X, y_i \in Y, Y = \{c_1, \dots, c_k\}$$

초기값

$$D_1(i) = \frac{1}{m}$$

For t=1...T:

- 확률분포 D_t 를 사용하여 약한 분류자 훈련
- 약한 분류자의 에러를 구함

$$\epsilon_t = \sum_{i=1}^m D_t(i) [h_t(x_i) \neq y_i].$$

If $\epsilon_t > \frac{1}{2}$ 종료

- 확률 분포 갱신을 위한 변수

$$\alpha_t = \left(\frac{\epsilon}{1 - \epsilon} \right), \alpha_t \text{는 } 0 \leq \alpha_t \leq 1 \text{를 만족}$$

- 확률 분포 갱신

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} \alpha_t, & \text{만약 } h_t(x_i) = y_i \\ 1, & \text{만약 } h_t(x_i) \neq y_i \end{cases}$$

- Z_t 는 정규화 상수, 확률 분포를 적절하게 유지하기 위하여 사용

최종 결과

- $count(D_t)$: 훈련 데이터에서 중요도가 $\frac{1}{m}$ 보다 큰 값들의 개수

- $h'_t(x_i)$: 입력 데이터가 포함된 공간 속의 훈련 데이터 중 중요도가 $\frac{1}{m}$ 보다 큰 값들의 개수

- $p(x_i) = \frac{h'_t(x_i)}{count(D_t)}$: 입력 데이터가 포함된 공간의 중요도

- 각각의 부류에 따른 중요도들을 투표

$$H_j(x_i) = \sum_{t: h_t(x_i)=c_j} e^{p_t(x_i)}, j = 1, \dots, k$$

- 최대 투표 값을 가지는 부류를 최종 결과로 선정

그림 2. 동적 중요도 결정 방법을 이용한 새로운 앙상블 시스템 (ExVote)

Fig. 2 A new ensemble system using dynamic weighting method(ExVote)

3.2 실험설계

실험에 사용한 데이터는 UCI Machine Repository에서 제공하는 7개의 데이터를 사용하였다.[18] 표. 1에서는 7개의 데이터 각각에 대한 정보를 표시하고 있다. 이벤트

의 개수가 가장 많은 데이터는 car 로 1728개의 데이터를 가지고 있으며 가장 적은 이벤트의 개수를 가지는 데이터는 iris이다. 속성의 개수는 breast가 11개로 가장 많은 속성을 가지고 있다. 하지만 첫 번째 속성은 이벤트의 번호로 매겨져 있기 때문에 실제로는 10개의 속성을 사용하였다. 부류의 개수는 bacteria가 가장 많은 10개를 가지고 있으면 liver, diabetes, breast는 부류가 2개로 가장 적은 부류를 가진 데이터들이다. Iris 데이터는 data mining 분야에서 가장 많이 사용되는 데이터 중 하나로 UCI Machine Repository에서는 인용 빈도가 가장 높다. 이벤트들의 개수가 150개로 적으며 대체적으로 잘 분류되는 데이터이며 숫자로 된 4개의 속성을 갖는다. 부류는 3 종류로 구성되어 있다.

Car 데이터는 자동차의 가격, 유지비, 문의 개수, 승차 정원, 짐칸의 크기, 안정성의 특성 값들을 통해 자동차 모델을 평가하는 데이터이다. 총 이벤트들의 개수가 1728개이며 속성들은 위에 열거한 6개로 구성되어 있다. Balance 데이터는 심리학 실험 데이터로 총 625개의 이벤트들로 구성되어 있으며 4개의 속성들을 가진다. 각각의 속성들은 { 1, 2, 3, 4, 5 }의 총 5개의 값들로 구성된다. 부류는 3 종류를 가지는 데이터이다. Liver 데이터는 간 질환 데이터로 345개의 이벤트들을 가지있다. Diabetes 데이터는 피마족의 당뇨병 발병에 관련된 데이터이다. 이 데이터는 1990년에 UCI에 오픈된 데이터이며 피마 인디언의 당뇨병 발병과 관련하여 768개의 이벤트들로 구성된 데이터이다. Breast 데이터는 유방암 진단을 위해 유방에 생긴 덩어리들의 디지털라이즈된 이미지들로부터 속성들을 추출하여 유방암을 진단하려고 만든 데이터이다.

표. 1 UCI 데이터.

Table. 1 UCI Machine Repository.

데이터	#이벤트	#속성	#부류
iris	150	4	3
car	1728	6	4
balance	625	4	3
liver	345	7	2
diabetes	768	8	2
bacteria	1484	9	10
breast	699	11	2

IV. 실험

실험은 AdaBoost, ExVote1 그리고 ExVote2로 나누어 비교하였다. ExVote1은 분류자가 입력 이벤트에 대해서 최종 결정을 한 공간안에 있는 이벤트들을 사용한 알고리즘이다. ExVote2는 최종 결정을 하기 직전의 공간 안에 있는 이벤트들을 사용한 알고리즘이다. 분류자가 의사 결정 트리인 경우, 최종 노드 직전의 부모 노드 공간 안에 있는 이벤트들의 중요도를 계산하여 분류자의 중요도를 결정하였다. 중요도를 계산하는 방법은 3.1에서 설명한 방법과 동일하다.

실험에서 신뢰할 만한 결과를 위해 10-Cross validation을 사용하였다. 10-Cross validation은 데이터의 개수가 적은 경우, 신뢰할 만한 실험 결과를 얻기 위한 방법이다. 데이터를 10개의 셋으로 균등하게 나눈 뒤, 9개는 훈련 데이터로 나머지 한 개는 테스트 데이터로 사용한다. 훈련 데이터를 통해 생성된 분류자로 테스트 데이터를 테스트하고 이런 과정을 10번을 하여 평균을 내는 것을 10-Cross validation이라 한다. 본 논문에서는 한 데이터에 10-Cross validation을 200번하여 평균 에러율을 구하여 충분히 신뢰할 만한 결과를 산출하였다. 표.2에서 6.63이란 6.63%의 에러를 가진다는 의미로 에러율이란 테스트 데이터에서 틀린 데이터의 개수를 테스트 데이터의 개수로 나눈 것을 말한다. 분류자로는 의사 결정 트리를 사용하였다. 실험에서 약한 분류자를 만들기 위해 프루닝 비율을 바꾸어 가면서 실험을 하였다. 의사 결정 트리에서 프루닝 비율은 최종 노드를 결정하는 기준이다. 트리를 생성하는 중, 현재 노드가 최종 노드의 조건에 만족하면 노드를 더 이상 갈라나가지 않고 최종 노드로 결정한다. 최종 노드의 조건은 일반적으로 현재 노드에 할당된 데이터 중에 한 부류가 프루닝 비율을 넘는 경우, 더 이상 갈라나가지 않는다. 프루닝 비율은 75%, 80%, 85%, 90%, 95%로 5개의 단계로 실험하였다. 프루닝 비율이 75%란 최종 노드에 한 부류가 75% 이상 존재하면 더 이상 노드를 나누지 않는 것을 의미한다. 마찬가지로 80%는 최종 노드에 한 부류가 80% 이상 존재하는 것을 의미한다. 이렇게 얻어진 실험 결과는 표.2에 나와 있다. 표시된 숫자들은 테스트 데이터를 수행하여 나온 에러율의 평균 값을 의미한다. 진하게 강조된 에러율들은 각각의 데이터에서 가장 좋은 결과를 갖는 알고리즘을 보여주고 있다. iris의 경우 프루닝 비율이 75%인

경우 ExVote1이 가장 좋은 성능을 보여주는 것을 알 수 있다. 표. 2의 결과를 보면 전체적으로 ExVote2가 가장 좋은 성능을 나타내는 것을 알 수 있다.

표. 2 실험 결과들.

Table. 2 The experiment results.

데이터	알고리즘	75%	80%	85%	90%	95%
iris	AdaBoost	6.63	6.46	5.88	7.78	7.46
	ExVote1	5.82	5.57	5.55	5.76	7.25
	ExVote2	5.94	5.69	5.47	5.77	7.21
car	AdaBoost	4.27	4.4	4.57	4.62	4.88
	ExVote1	4.12	4.23	4.41	4.45	4.71
	ExVote2	4.11	4.17	4.34	4.42	4.68
balance	AdaBoost	23.44	23.67	23.71	23.63	23.73
	ExVote1	23.28	23.39	23.51	23.49	23.75
	ExVote2	22.82	23.07	23.25	23.29	23.6
liver	AdaBoost	29.99	30.35	30.42	30.62	38.62
	ExVote1	29.64	29.69	29.63	29.44	37.26
	ExVote2	30.04	29.92	29.92	29.6	37.28
diabetes	AdaBoost	26.25	25.94	26.34	26.57	27.59
	ExVote1	26.18	25.91	25.99	25.86	25.67
	ExVote2	26.07	25.81	25.89	25.76	25.56
bacteria	AdaBoost	42.02	42.42	42.65	43.21	43.86
	ExVote1	41.34	41.48	41.59	41.47	41.8
	ExVote2	41.29	41.43	41.45	41.44	41.78
breast	AdaBoost	22.17	22.11	21.95	21.79	21.54
	ExVote1	21.38	21.43	21.38	21.33	20.83
	ExVote2	21.28	21.33	21.35	21.29	20.82

그림 3부터 그림 6까지는 표. 2에서 의미있는 결과를 그래프로 표현하였다. 가로축은 프루닝 비율을 세로축은 에러율을 보여주고 있다. 에러율이 낮을 수록 분류자가 예측을 잘한다는 것을 의미한다. 그림. 3에서 보는 것처럼 프루닝 비율을 85% 정도의 약한 분류자로 앙상블 시스템을 만든 경우보다 프루닝 비율을 75% 정도의 분류자로 앙상블 시스템을 만드는 경우가 그림. 5을 비교해 보면 더 나은 성능을 가진다는 것을 알 수 있다. 프루닝 비율이 95%인 경우, 강한 분류자에 속하기 때문에 이런 분류자로 앙상블 시스템을 구성한 결과를 보면 프루닝 비율을 75%로 유지한 시스템보다 높은 에러율을 보여주고 있다.

표. 2를 보면 대부분의 데이터에서 AdaBoost의 성능이 프루닝 비율에 따라 변동되는 것과 마찬가지로 ExVote1과 ExVote2의 에러율도 동기화 되어 움직이는 것을 알 수 있다. 이것은 분류자의 성능이 전체 시스템에 큰 영향을 미친다는 것을 보여준다. ExVote2가 대체적으로 에러율이 낮다는 것은 최종 노드에서 중요도를 계산하는 것보다 부모 노드에서 중요도를 계산하는 것이 중요하다는 것을 보여준다.

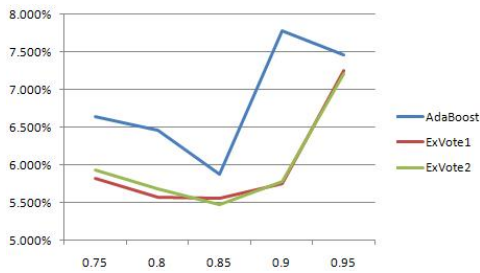


그림. 3 iris 데이터
Fig. 3 iris data

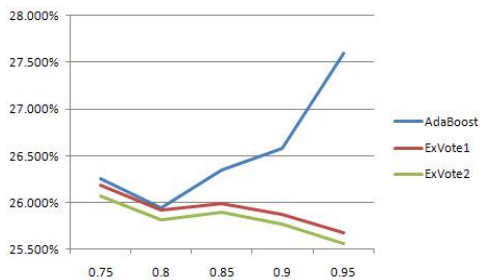


그림. 4 diabetes 데이터
Fig. 4 diabetes data

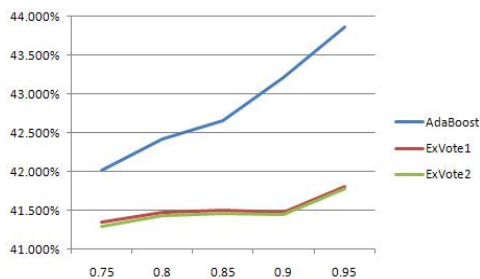


그림. 5 bacteria 데이터
Fig. 5 bacteria data

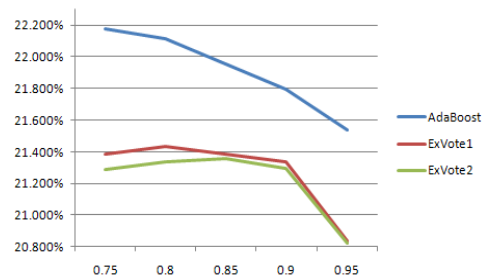


그림. 6 breast 데이터
Fig. 6 breast data

V. 결론

본 논문에서는 앙상블 시스템에서 분류자들 속에 중요도 정보를 삽입하여 동적 중요도 결정이 가능한 전문가들의 혼합을 제안하였다. 그동안 앙상블 시스템에서 중요도는 훈련이 끝나고 결정된 중요도를 사용하였다. 한 번 결정된 중요도는 테스트 데이터에 상관없이 정적으로 사용되었다. 이 문제는 관문 네트워크를 통해 구조적으로 계층을 두는 프로세스를 추가하여 동적 중요도 결정이 가능하게 하는 방법이 있지만 프로세스가 추가된다는 단점이 있다. 본 논문에서는 이런 추가적인 프로세스 없이 간단하게 동적 중요도 결정이 가능한 방법을 보여주고 구조적으로 큰 변경 없이 기존의 시스템에 쉽게 적용할 수 있으며 AdaBoost보다 나은 성능을 보여주었다는데 의미를 찾을 수 있다. 중요도의 변경 범위에 따른 결과를 조사하여 최적화된 결과를 도출할 수 있는 알고리즘 개발이 향후 연구 개발할 예정이다.

참고문헌

- [1] L.K Hansen and P.Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp993-1001, 1990.
- [2] R.E Schapire, "The strength of weak learnability," *Machine Learning*, vol.5, no. 2, pp. 197-227, 1990.
- [3] Yoav Freund and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an

- application to boosting," *Journal of Computer and System Sciences*, 55(1):119 - 139, August 1997.
- [4] Y. Freund and R. Schapire. "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence*, 14(5):771 - 780, 1999.
- [5] K. Woods, W.P.J. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405-410, 1997.
- [6] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 26, no. 1, pp. 52 - 67, 1996.
- [7] R. Battiti and A.M. Colla, "Democracy in neural nets: Voting schemes for classification," *Neural Networks*, vol. 7, no. 4, pp. 691 - 707, 1994.
- [8] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418 - 435, 1992.
- [9] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66 - 75, 1994.
- [10] G. Rogova, "Combining the results of several neural network classifiers," *Neural Networks*, vol. 7, no. 5, pp. 777 - 781, 1994.
- [11] L. Lam and C.Y. Suen, "Optimal combinations of pattern classifiers," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 945 - 954, 1995.
- [12] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensemble methods," *Neural Computation*, vol. 6, no. 6, pp. 1289 - 1301, 1994.
- [13] L.I. Kuncheva, "Classifier ensembles for changing environments," *5th Int. Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, F. Roli, J. Kittler, and T. Windeatt, Eds., vol. 3077, pp. 1 - 15, 2004.
- [14] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79 - 87, 1991.
- [15] M.J. Jordan and R.A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181 - 214, 1994.
- [16] R.E Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197-227, 1990
- [17] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996
- [18] C.J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. www.ics.uci.edu/~mlearn/MLRepository.html

저자소개

서동훈(DongHun Seo)



2003년 충남대학교(학사)
2005년 충남대학교 대학원(석사)
2005년~현재 충남대학교 컴퓨터
공학과 박사과정

※ 관심분야: 기계 학습, 데이터 마이닝

이원돈(Won Don Lee)



1979년 서울대학교(학사)
1982년 U. of Illinois 대학원(석사)
1986년 U. of Illinois 대학원(박사)
1987년~현재: 충남대학교
전기정보통신공학부 교수

※ 관심분야: 신경 회로망, 멀티미디어, 데이터마이닝