

# 지역 기반 분류기의 앙상블 학습

최 성 하<sup>†</sup> · 이 병 우<sup>†</sup> · 양 지 훈<sup>\*\*</sup>

## 요 약

기계학습에서 분류기들의 집합으로 구성된 앙상블 분류기는 단일 분류기에 비해 정확도가 높다는 것이 입증되어왔다. 본 논문에서는 새로운 앙상블 학습으로서 데이터의 지역 기반 분류기들의 앙상블 학습을 제시하여 기존의 앙상블 학습과의 비교를 통해 성능을 검증하고자 한다. 지역 기반 분류기의 앙상블 학습은 데이터의 분포가 지역에 따라 다르다는 점에 착안하여 학습 데이터를 분할하여 해당하는 지역에 기반을 둔 분류기들을 만들어 나간다. 이렇게 만들어진 분류기들로부터 지역에 따라 가중치를 둔 투표를 적용하여 앙상블 방법을 이끌어낸다. 본 논문에서 제시한 앙상블 분류기의 성능평가를 위해 단일 분류기와 기존의 앙상블 분류기인 배깅과 부스팅 등을 UCI Machine Learning Repository에 있는 11개의 데이터 셋으로 정확도 비교를 하였다. 그 결과 새로운 앙상블 방법이 기본 분류기로 나이브 베이즈와 SVM을 사용했을 때 다른 방법보다 좋은 성능을 보이는 것을 알 수 있었다.

키워드 : 기계학습, 앙상블, 배깅, 부스팅

## Ensemble Learning of Region Based Classifiers

Sungha Choi<sup>†</sup> · Byungwoo Lee<sup>†</sup> · Jihoon Yang<sup>\*\*</sup>

### ABSTRACT

In machine learning, the ensemble classifier that is a set of classifiers have been introduced for higher accuracy than individual classifiers. We propose a new ensemble learning method that employs a set of region based classifiers. To show the performance of the proposed method, we compared its performance with that of bagging and boosting, which are existing ensemble methods. Since the distribution of data can be different in different regions in the feature space, we split the data and generate classifiers based on each region and apply a weighted voting among the classifiers. We used 11 data sets from the UCI Machine Learning Repository to compare the performance of our new ensemble method with that of individual classifiers as well as existing ensemble methods such as bagging and boosting. As a result, we found that our method produced improved performance, particularly when the base learner is Naive Bayes or SVM.

Key Words : Machine Learning, Ensemble, Bagging, Boosting

### 1. 서 론

기계학습 분야 중에 하나인 지도학습에서 투표를 하는 앙상블 학습은 주어진 학습데이터를 가장 잘 설명하는 하나의 가설을 찾는 것이 아니라 가설들의 집합을 만들고 새로운 데이터의 라벨을 가설들의 집합들로 투표하여 예측하는 것이다[6]. 좀 더 형식적으로 표현하면, 가설들의 집합  $\{h_1, \dots, h_k\}$  을 만들고 가중치의 집합  $\{w_1, \dots, w_k\}$  을 설정하여  $H(X) = w_1 h_1(X) + \dots + w_k h_k(X)$  인 분류기를 만드는 것을 투표를 하는 앙상블 학습이라고 한다. 이러한 앙상블 분류기는

단일 분류기보다 더 좋은 성능을 보인다는 것이 실험을 통해 증명되었다[1,5,13].

앙상블 분류기가 그것을 구성하는 각각의 단일 분류기보다 정확도가 더 높기 위한 조건은 각 분류기가 임의 추측보다 정확하며 입력 공간의 다른 부분에서 에러를 가져야한다는 것이다[11]. 이는 각각 분류기의 정확성과 다양성을 나타낸다.

기존의 앙상블 학습에서 가장 잘 알려진 방법으로는 배깅(Bagging)[4]과 부스팅(Boosting)[8]이 있다. 이러한 기존의 앙상블 방법에 대해 다음과 같은 두 가지 문제점을 제시할 수 있다.

첫째, 기본 학습 알고리즘이 분류하기 어려운 데이터일 경우 데이터를 재추출하더라도 역시 어려운 데이터일 수 있다. 이런 경우에는 특히 배깅 방법을 사용할 경우 에러를 갖는 지역이 집중되어 있으며 이는 좋은 앙상블의 조건인 분류기의 다양성에 위배된다.

둘째, 각 지역에 따라 분포가 틀리지만 어떤 분류기가 그 지역에 대해 높은 정확도를 보이는지 알지 못한다. 예를 들

\* 본 연구는 한국과학재단 특정기초연구(Ro1-2004-000-10689-0) 지원으로 수행되었음.

† 준회원 : 서강대학교 대학원 컴퓨터학과 석사과정

† 준회원 : 서강대학교 대학원 컴퓨터학과 석사과정

\*\* 정회원 : 서강대학교 컴퓨터학과 부교수

논문접수 : 2007년 1월 5일, 심사완료 : 2007년 6월 20일

어, 부스팅의 경우 선행되는 분류기가 잘못된 분류를 하는 인스턴스에 대해 가중치를 두기 때문에 선행되는 분류기가 분류하지 못한 부분을 잘 분류하도록 학습한다. 하지만 실제로 분류할 때 그 부분이 어디지는 알지 못한다. 만약, 각 분류기 별로 잘 분류하는 인스턴스가 어떠한 것인지 안다면 보다 정확하게 분류할 수 있을 것이라 기대된다.

### 2. 앙상블과 기본 학습알고리즘과의 관계

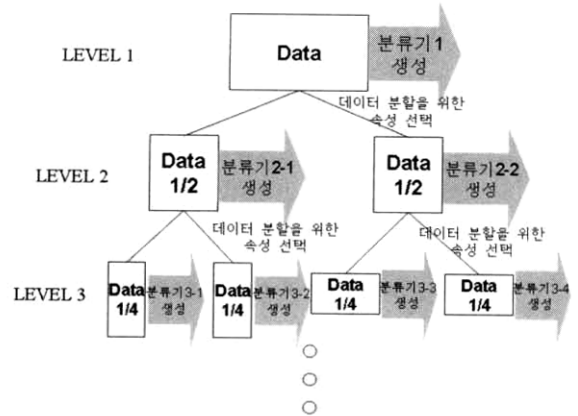
학습 데이터의 작은 변화에 의해서 학습알고리즘으로부터 생성되는 분류기가 크게 변할 때 그 알고리즘을 불안정(Unstable)하다고 한다[4]. 예를 들면 결정트리를 생성시킬 때 사용되는 C4.5와 같은 알고리즘은 욕심쟁이 알고리즘(Greedy Algorithm)으로 학습데이터의 작은 변화에도 트리의 큰 변화를 일으킬 수 있다. 이처럼 학습데이터의 변화에 따라 결과도 크게 변하는 불안정한 알고리즘들은 높은 분산을 가지고 있다고도 한다[3].

일반적으로 배깅과 부스팅과 같은 앙상블 학습은 기본 학습알고리즘에 독립적이지만 위와 같은 불안정한 학습알고리즘에서 잘 동작한다. Breiman은 배깅이 이러한 불안정한 분류기들의 성능을 향상시킨다고 하였다[4]. 반면에 배깅과 부스팅이 나이브 베이즈나 선형 판별식 분석을 사용하는 안정적인(Stable) 알고리즘에서는 큰 정확도 상승은 눈에 띄지 않는다고 밝혔다[3]. 또한 Dietterich도 결정트리를 포함하여 이보다 덜 표현적인 가설을 생성하는 알고리즘에서 부스팅이 추천된다고 하였다[7]. 그래서 대부분의 앙상블 학습과 관련된 논문들이 결정트리들의 앙상블에 초점을 맞추고 있다[1,5,10,17].

이처럼 불안정한 학습알고리즘에서 잘 동작하는 이유는 좋은 앙상블의 조건인 각 분류기들이 입력 공간의 다른 부분에서 에러를 가져야한다는 다양성과 관련이 있다[11]. 이러한 다양성은 분류기들의 앙상블에 있어서 매우 중요한 성질이고 앙상블의 효율성과의 관계에 대해서 많은 연구가 이루어져왔다[6,11,12]. 예를 들어 만약 5개의 분류기들이 정확도가 모두 90%이면서 각 분류기의 에러가 독립적이고 다수결로 분류를 한다고 하자. 그럴 경우에 10%의 오분류 확률로 3개 이상의 분류기에서 그러한 사건이 발생해야 오분류된다. 이러한 확률을 계산해보면 약 0.9%로 단일 분류기의 10%보다 10분의 1로 줄었다. 더불어, 오분류 확률은 분류기가 늘어남에 따라 더욱더 줄어들 것이다. 하지만 실제로는 분류기들 사이에 같은 분포에서 샘플링된 데이터로 하기 때문에 완벽한 다양성을 유지할 수 없다. 그렇기 때문에 실제로는 위와 같은 성능 증가는 나타나지 않는다. 그럼에도 불구하고 불안정한 알고리즘을 사용한 앙상블 방법은 다양성을 어느 정도 갖는 분류기들을 생성하기 때문에 배깅과 부스팅을 적용하여 분류기들의 성능을 향상시킬 수 있다.

### 3. 지역 기반 분류기의 앙상블 학습

(그림 1)은 지역 기반 분류기의 앙상블 학습을 트리 형식



(그림 1) 지역 기반 분류기의 앙상블 학습에 대한 개괄도

으로 나타낸 것이다. 트리의 각 노드에 있는 분류기들을 지역 기반 분류기라 하고 특히 말단 노드에 있는 분류기들만을 사용하는 것을 RBC로 지칭한다. RBC는 입력 인스턴스가 속하는 말단 노드의 분류기 하나만이 분류에 사용된다. 그리고 이러한 모든 지역 기반 분류기들로 지역에 따라 가중치를 두어 여러 개의 분류기가 투표를 하는 앙상블 분류기를 지역 기반 분류기의 앙상블이라 하며 앞으로 RBE라 지칭할 것이다. 앞으로 RBC, RBE와 함께 (그림 1)에 있는 트리를 기반으로 용어를 사용한다.

#### 3.1 지역 기반 분류기의 앙상블 학습 알고리즘

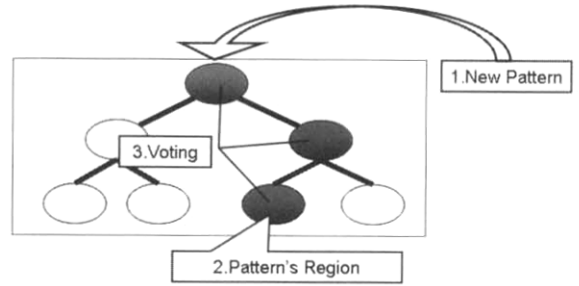
RBE의 학습은 입력으로 들어온 학습데이터로부터 학습을 통한 분류기 생성과 학습데이터의 분할을 반복하며 분류기를 생성시켜 나간다. 이러한 방식으로 여러 개의 분류기를 생성시킬 수 있다. RBE의 학습의 구체적인 절차를 (그림 2)에 나타내었고 앙상블 분류기를 통한 분류는 (그림 3)에 나타내었다.

(그림 2)에서 보는 바와 같이 우선 전체 학습 데이터  $D_1$ 을 선택되어진 학습 알고리즘으로 학습하여 분류기  $C_1$ 을 생성시킨다. 그 후에 학습 데이터를 이등분한다. 여기서 이등분 하는 이유는 RBE의 목적이 앙상블 분류기를 생성하는 것이므로 한쪽에만 학습데이터가 몰려서 불균형한 분류기를 생성하는 것을 방지하기 위해서이다. 본 논문에서는 연속적인 값을 속성으로 하는 데이터만 고려하였다. 데이터를 이등분하는 방법은 다음과 같다. 전체 데이터  $S$ 가 있을 때 각 속성에 대하여  $S$ 의 개수를 이등분 할 수 있는 값을 구한다. 이등분을 할 수 없다면 최대한 가까운 이분할 값을 구한다. 이보다 작은 값을 갖는 데이터들의 집합을  $S_1$ , 큰 값을 갖는 데이터들의 집합을  $S_2$ 라 한다. 여기에 식(1)과 식(2)를 이용하여 정보획득(Information Gain)[15]을 구한다.

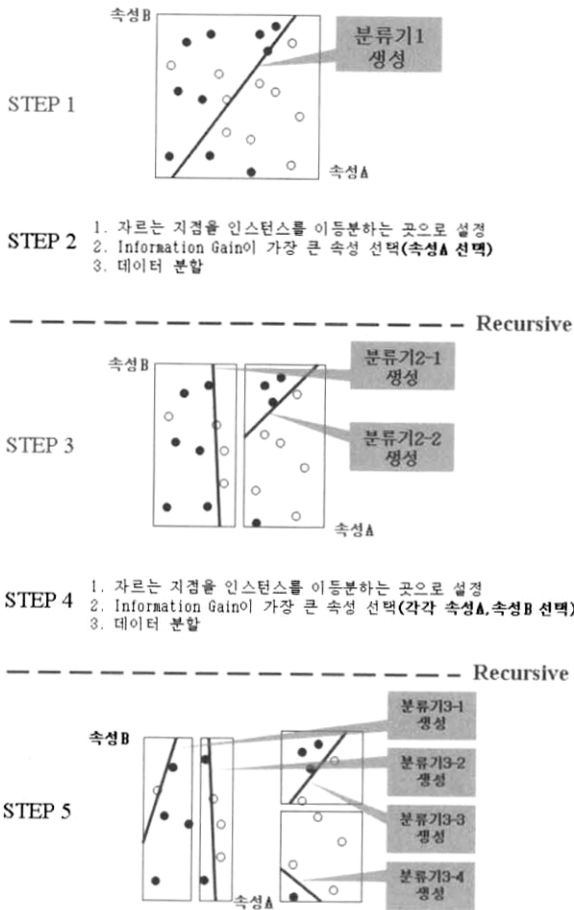
$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i \tag{1}$$

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^2 \frac{|S_i|}{|S|} Entropy(S_i) \tag{2}$$

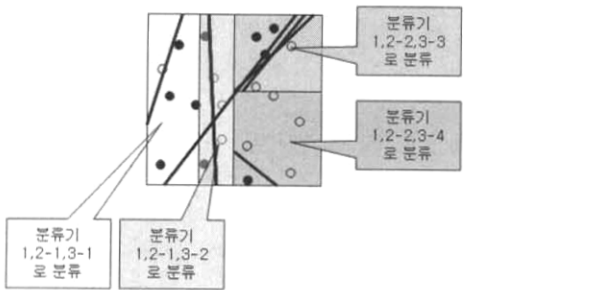
위 식에서  $c$ 는 클래스 개수,  $p_i$ 는  $S$ 중에서 클래스  $i$ 에 속하는 데이터의 비율,  $A$ 는 데이터의 속성,  $S_1$ 과  $S_2$ 는  $A$ 에 대하여 이등분한  $S$ 의 부분 집합이다. 모든 속성들 중에서 가장 정보획득이 큰 속성을 선택하여 데이터를 이등분한다. 이로 인해 나누어진 데이터는 클래스가 한쪽으로 치우쳐지게 되며 이는 결과적으로 데이터가 분할해 감에 따라 좀 더 분류하기 쉬울 것이라는 기대를 할 수 있다. 이렇게 이등분하여 나온 데이터에 위의 과정을 반복하여 분류기  $C_{21}, C_{22}$ 를 생



(그림 4) 지역 기반 분류기의 앙상블 투표



(그림 2) RBE의 학습에 대한 구체적인 과정



1. 4개의 지역으로 분할되고 각 지역당 3개의 분류기가 영향을 준다.
2. 분류할 패턴은 위의 4개의 지역 중 오직 하나에만 속한다

(그림 3) RBE를 통한 분류

성해 나간다. 이는 재귀(Recursive) 호출에 의해 구현될 수 있다. 본 논문에서는 지역 분할의 단계를 3회로 설정하여 총 7개의 분류기( $C_1, C_{21}, \dots, C_{33}, C_{34}$ )를 생성하였다. 이렇게 생성된 분류기로 새로운 패턴이 들어올 때의 클래스를 결정하기 위하여 그 패턴이 속한 지역에 따라 분류기들을 선택해 투표한다. (그림 4)에서 보는 바와 같이 분류해야 할 패턴이 속하는 말단 노드와 그 직속 상위 노드들로부터만 투표를 한다. 이러한 노드들을 제외한 나머지 노드들은 투표에 참여하지 않는다. 만약 투표에 참여하는 노드들이 짝수이어서 다수결로 동률이 되었을 때는 학습 데이터에 대해 더 높은 정확도를 갖는 말단노드의 결정에 따른다.

### 3.2 RBE의 분석

RBE에서 데이터를 분할해 나가는 것에 대하여 다음 두 가지 측면에서 살펴보자. 첫째, 데이터를 분할해 나갈수록 분류기의 성능이 더 좋아진다. 둘째, 분할하기 전의 데이터로부터 학습한 분류기가 분할한 후의 데이터로부터 학습한 분류기들을 보완해준다. 이는 데이터를 분할해나가면서 얻는 이점이 되고 앙상블을 하는 근거가 된다. 우선 첫 번째 이슈에 대해 논해보자.

RBE의 학습에 있어서 데이터를 분할해 나가는 것은 데이터의 클래스 분포를 단순화 시켜 나가는 것이기 때문에 하단 노드로 갈수록 분류하기가 쉬워질 것이라고 생각할 수 있다. 그러므로 자식 노드의 분류기들의 집합이 부모 노드의 분류기보다 학습데이터에 대해서 충분히 적은 에러를 보일 것이라고 예상된다. 이를 **Theorem 1**으로 제시하고 증명해보자.

[Theorem 1] 주어진 학습 데이터  $D$ 가 있을 때 그로부터 학습 알고리즘  $L$ 을 통해 얻어진 가설을  $h_i$ 라 하자. 데이터  $D_0$ 가 주어졌을 때 데이터의 어떤 한 속성을 기준으로 데이터를 이분할 하여 얻어진 데이터를  $D_1, D_2$ 라 하면 각각  $L$ 을 통해  $h_0, h_1, h_2$ 가 생성된다. 이때  $h_0, h_1, h_2$ 의 학습 데이터에 대한 에러를 각각  $e_0, e_1, e_2$ 라 하면  $e_0 \geq e_1 + e_2$ 가 항상 성립한다.

[Condition] 주어진 학습알고리즘  $L$ 은  $L$ 의 결과 모델로서 표현 가능한 가설 공간에서 주어진 데이터에 대해 가장 오류가 적은 가설을 선택한다.

[Proof]  $h_0$ 가  $D_1$ 에서 발생하는 에러와  $D_2$ 에서 발생하는

에러를 각각  $e_{01}$  과  $e_{02}$  라 한다면  $e_0 = e_{01} + e_{02}$  임을 알 수 있다.  $e_0 \geq e_1 + e_2$  이 성립함을 보이기 위해서  $e_{01} \geq e_1 \wedge e_{02} \geq e_2$  를 보이는 것으로 충분하다. 이제 이를 보이도록 하겠다.

주어진 학습알고리즘  $L$ 이 표현할 수 있는 가설 공간에서  $h_0$ 의  $D_1$ 에 대한 결과와 같은 값을 내는 가설들의 집합을  $H_{01}$ 이라 하자. 그러면  $h_0 \in H_{01}$  임을 알 수 있다. 왜냐하면  $H_{01}$ 에 속하는 가설들은  $h_0$ 가  $D_1$ 에 대해서 내는 결과와 같기만 하면  $D_2$ 에 대해서는 어떠한 결과 값을 가져도 되는 가설들이기 때문이다. 즉,  $H_{01}$ 에 속하는 모든 가설들은  $D_1$ 에 대해서  $h_0$ 와 같은 결과 값을 갖기 때문에  $e_{01}$ 의 에러를 갖는 것을 알 수 있다.

여기서  $e_{01} < e_1$  이라고 가정하여 보자. 그러면 가정에 의해 학습알고리즘  $L$ 이 표현할 수 있는 가설공간에는  $D_1$ 이 주어졌을 때  $D_1$ 에 대해  $e_{01}$ 의 에러를 갖는  $H_{01}$ 은 속하지 않는다. 만약 속한다면 주어진 Condition에 의해  $e_{01} < e_1$ 인 에러  $e_1$ 을 갖는 가설을 선택하지 않고 그보다 성능이 좋은  $e_{01}$ 의 에러를 갖는  $H_{01}$ 의 가설을 선택할 것이기 때문이다. 그러므로  $H_{01}$ 에 속하는 가설보다 좋지 않은  $e_1$ 의 에러를 갖는 가설을 선택한 것이다.

이로써,  $L$ 이 표현할 수 있는 가설공간에는  $H_{01}$ 에 속하는 가설들은 존재하지 않음을 알 수 있다. 그러나  $L$ 은  $D_0$ 가 주어졌을 때  $H_{01}$ 에 속하는  $h_0$ 을 찾아냈으므로 이는 오류이다. 마찬가지로  $e_{02} < e_2$ 에 대해서도 보일 수 있다. 이로써  $e_{01} < e_1$  와  $e_{02} < e_2$ 은 모두 거짓임을 알 수 있다. 그러므로  $e_{01} \geq e_1$  and  $e_{02} \geq e_2$ 임을 알 수 있고 증명되었다.

<표 1, 2>는 Theorem 1이 실제로 어떻게 적용되는지 알아보기 위한 실험의 결과이다. Theorem 1에 주어진 조건이 강하긴 하지만 일반적으로 학습 알고리즘이 그와 같은 경향을 따르기 때문에 실제적으로 <표 1, 2>에서 보는 바와 같이 대부분의 데이터 셋에서 하위 단계로 내려 갈수록 학습 데이터에 대한 정확도가 높아지는 양상을 보인다. 예외적인 경우는 붉은체로 표시하였다. 이 실험은 기본 학습 알고리즘으로 나이브 베이즈와 SMO[14]를 선택하여 실험하였다. SVM은 Theorem 1에 주어진 조건을 만족한다고 알려져 있지만 SVM은 매우 많은 시간을 소비하기 때문에 실제로는 이를 근사화하여 구현한 SMO를 많이 사용한다.

이제 두 번째 이슈에 대해 살펴보자. RBE에서는 데이터를 분할해 나감에 따라 데이터의 클래스 분포가 단순화되기 때문에 하단 노드로 갈수록 분류하기 쉽게 된다. 그러므로 Theorem 1에서 증명한 것처럼 말단 노드의 분류기들이 다른 상위노드의 분류기들보다 학습데이터에 대해서 적은 에러를 보이는 것을 알 수 있고 실제의 실험으로도 대부분 그러한 결과가 나오는 것을 살펴보았다. 이러한 결과에 의하면 분류할 패턴이 특정 지역에 속할 때 해당하는 말단 노드의 분류기만으로 클래스를 결정을 하는 것이 더 높은 성능을 보일 것이라고 기대할 수 있다. 그럼에도 불구하고 상위노드의 분류기들이 하위노드의 분류기들을 보완해 준다는 두 번째 이슈를 다시 두 가지 측면에서 보이겠다.

<표 1> 각 분할 단계에 따른 학습 데이터에 대한 정확도(Naive Bayes)

Naive Bayes Data Set	Level 1	Level 2	Level3
XOR	60.00	72.00	94.00
Balance-Scale	90.88	<b>87.36</b>	90.88
Glass	55.61	64.95	77.57
Ionosphere	82.91	91.74	92.59
Iris	96.00	96.67	96.67
Pima	76.27	<b>75.23</b>	77.71
SatImage	79.73	<b>79.28</b>	<b>78.40</b>
Segment	81.67	89.27	89.87
Spambase	79.53	<b>78.66</b>	80.42
Vehicle	46.34	66.31	72.46
Waveform	80.22	80.48	81.44
Wdbc	93.85	93.85	95.08

<표 2> 각 분할 단계에 따른 학습 데이터에 대한 정확도 (SMO with Linear Kernel)

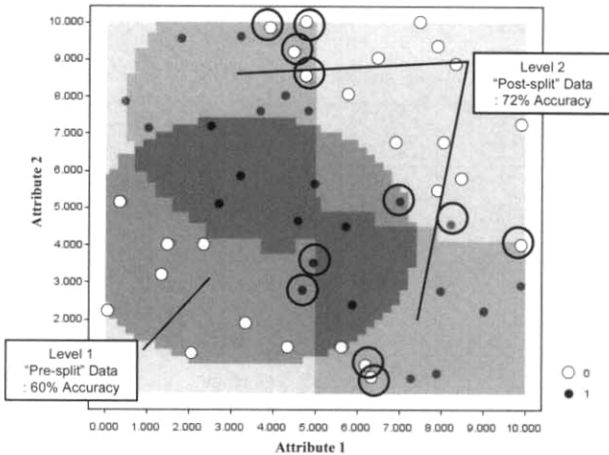
SMO Data Set	Level 1	Level 2	Level3
XOR	54.00	70.00	72.00
Balance-Scale	87.68	89.76	<b>89.44</b>
Glass	60.75	68.22	69.16
Ionosphere	91.45	92.88	<b>92.31</b>
Iris	96.67	97.33	98.67
Pima	77.44	77.71	<b>75.49</b>
SatImage	87.49	87.64	88.82
Segment	92.47	93.27	94.47
Spambase	90.76	91.28	92.46
Vehicle	76.24	79.55	84.16
Waveform	87.68	87.80	87.90
Wdbc	98.24	<b>97.89</b>	<b>97.72</b>

첫 번째로 하위노드의 분류기들이 상위노드의 분류기보다 학습 데이터에 대해 에러가 적더라도 상위노드의 분류기가 잘 분류하는 것을 하위노드의 분류기들이 전부 잘 분류할 수 있는 것은 아니다. 이는 각 분류기들이 입력공간의 다른 곳에서 에러를 갖는다는 다양성과 관련이 있다. 이를 형식적으로 나타내고 이를 반례로써 증명하면 Theorem 2와 같다.

[Theorem 2] 주어진 학습 데이터  $D_j$ 가 있을 때 그로부터 학습 알고리즘  $L$ 을 통해 얻어진 가설을  $h_j$ 라 하자. 데이터  $D_0 = \{ \dots (X_j, y_j), \dots \} (j=1, \dots, m)$ 가 주어졌을 때 데이터의 어떤 한 속성을 기준으로 데이터를 이분할 하여 얻어진 데이터를  $D_1, D_2$ 라 하면 각각  $L$ 을 통해  $h_0, h_1, h_2$ 가 생성된다고 하자.

이때,  $X_j \in D_0, h_0(X_j) = y_j (j=1, \dots, m)$ 인 모든  $X_j$ 가  $X_j \in D_1$  이라면  $h_1(X_j) = y_j$ 을 만족하고  $X_j \in D_2$  이라면  $h_2(X_j) = y_j$ 을 만족하는 것은 아니다.

[Proof] 반례로서 (그림 5)는 Theorem 1을 만족한다.(<표 1>의 XOR 형태를 따르는 데이터) 이를 보면 분할하기 전의 데이터를 가지고 학습한 분류기가 잘 분류하는 인스턴스 중에서 분할하고 난 데이터를 가지고 학습한 분류기들이 분류하지 못하는 인스턴스가 있음을 알 수 있다. (인스턴스에 O표시)



(그림 5) 상위 노드와 하위 노드의 분류기 결정선

두 번째로 말단 노드의 분류기들만 가지고 분류를 했을 때 학습데이터에 대해서는 높은 적응성(fitting)을 갖지만 새로운 데이터 패턴이 들어왔을 때 잘한다는 보장을 할 수 없다. 왜냐하면 학습 데이터의 부족으로 인해 분류기를 생성하는 데 있어서 어려움이 있을 수 있고 또한 학습데이터에 과잉적응 되는 현상이 발생할 수 있기 때문이다. 과잉적응(Over Fitting)은 기계학습 분야에서 자주 제기되는 문제이며 이는 학습데이터에 과도하게 맞추어져 실제 성능이 떨어지는 현상을 말한다. 전체데이터의 분포를 보지 않고 지역적으로 분할된 부분만 분류기가 보기 때문에 발생할 수 있다.

RBE에서 이를 방지해주는 역할을 해주는 것이 상위노드의 분류기들이다. 데이터의 일부분에 특화된 분류기만 클래스의 결정에 참여하는 것이 아니라 전체적인 데이터 분포를 보고 생성된 분류기들도 클래스의 결정에 참여하기 때문에 과잉적응 현상의 위험을 낮출 수 있다.

지금까지 RBE의 데이터 분할에 대한 의미에 대해서 살펴 보았다. 이제 이러한 데이터 분할에 의한 분류기 생성이 앙상블 학습의 측면에서 볼 때 어떠한 의미가 있는지 살펴보자.

2장에서 좋은 앙상블의 조건으로서 기본 학습알고리즘으로부터 생성되는 분류기들이 서로 다양성을 가져야하고 정확도가 높아야한다고 하였다[11]. 이러한 측면에서 볼 때 RBE의 학습은 각 분류기들이 다양성을 보장한다. 이는 앞에서 논의한 상위 노드의 분류기들이 하위 노드의 분류기들을 보완해준다는 두 번째 이슈와 연관이 있다. RBE는 정보 획득에 의해서 데이터를 분할하여 그중에 일부만 보기 때문에 각 분류기 별로 학습하는 데이터가 매우 다르다. 그렇기 때문에 학습 알고리즘이 분산이 낮은 안정적인 분류기를 생성하더라도 학습 데이터가 크게 다르게 만들기 때문에 다른 앙상블 방법에 비해 상대적으로 분산이 큰 분류기를 생성시킬 수 있다. 이로 인해 분류기의 다양성을 보장할 수 있다.

그러나 이렇게 학습한 분류기는 데이터의 일부만으로 학습을 하였기 때문에 전체 데이터에 대해서는 정확도가 상당히 낮아지게 된다. 하지만, RBE는 각 분류기가 잘 분류할 수 있는 지역에 대해서만 분류 결정에 참여시키기 때문에

분류기의 정확도 또한 보장할 수 있다. 이는 앞에서 논의한 데이터를 분할해 나갈수록 분류기의 정확도가 더 좋아진다는 첫 번째 이슈와 관련이 있다.

## 4. 실험 및 결과

### 4.1 실험 방법

본 논문에서 제시한 RBE가 학습 알고리즘에 독립적으로 작동하는 기법임을 보이기 위해 기본 학습 알고리즘으로서 나이브 베이즈 분류기뿐만 아니라, SVM을 구현한 선형커널을 가진 SMO[14]와 C4.5[16]를 통한 결정트리를 사용하였다. 실험에 쓰일 데이터는 UCI Machine Learning Repository[2]에 있는 실제 데이터를 가져왔으며 실험은 11개의 데이터 셋에서 수행하였다. 데이터 셋에 대한 설명은 <표 3>을 참고한다.

RBE의 성능을 검증하기 위한 비교 대상으로는 RBE를 구성하는 분류기들을 생성시키는 데 사용된 기본 학습 알고리즘의 단일 분류기, 부스팅을 구현한 알고리즘인 Adaboost.M1[9]으로 7개의 분류기를 생성시킨 앙상블 분류기, 배경 기법으로 역시 7개의 분류기를 생성시킨 앙상블 분류기 그리고 3장에서 논의된 바 있는 RBC를 비교 대상으로 하여 분류 정확도를 비교하였다. 본 논문에서 RBE로 학습하여 3레벨까지 데이터를 분할하여 7개의 분류기를 생성시켰기 때문에 공정한 비교를 위해 부스팅과 배경도 7개의 분류기를 생성하였다.

기본 학습 알고리즘은 공개 데이터마인닝 소프트웨어인 Weka3[18]을 사용하였고 기타 앙상블 기법에 사용된 프로그램은 자체 구현하였다.

<표 3> 실험에 사용된 데이터 셋

Data Set Name	# example		# class	# attribute	
	train	test		disc.	con.
BALANCE-SCALE	625	10-fold	3	none.	4
GLASS	214	10-fold	6	none.	9
IONOSPHERE	351	10-fold	2	none.	34
IRIS	150	10-fold	3	none.	4
PIMA	768	10-fold	2	none.	8
SAT IMAGE	4435	2000	7	none.	36
SEGMENT	1500	810	7	none.	19
SPAM BASE	4601	10-fold	2	none.	57
VEHICLE	846	10-fold	4	none.	18
WAVEFORM	5000	10-fold	3	none.	40
WDBC	569	10-fold	2	none.	30

### 4.2 실험 결과

<표 4>부터 <표 6>은 각 기본 학습알고리즘 별로 단일 분류기, 다른 앙상블 방법들(배경, 부스팅) 그리고 RBC 각 각에 대해 지역 기반 분류기의 앙상블(RBE)과 정확도 비교

〈표 4〉 RBE와 각 방법들과의 정확도 비교 (Naive Bayes)

Naive Bayes DATA SET	Single	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	<b>89.92</b>	88.63	X	<b>89.76</b>	88.63		<b>89.60</b>	88.63		88.00	<b>88.63</b>	
GLASS	47.45	<b>56.32</b>	O	47.92	<b>56.32</b>	O	48.85	<b>56.32</b>	O	<b>61.06</b>	56.32	X
IONOSPHERE	82.91	<b>89.75</b>	O	<b>91.76</b>	89.75	X	82.91	<b>89.75</b>	O	89.21	<b>89.75</b>	
IRIS	96.00	<b>96.67</b>		96.67	96.67		95.33	<b>96.67</b>	O	95.33	<b>96.67</b>	O
PIMA	75.62	<b>76.15</b>		75.88	<b>76.15</b>		75.75	<b>76.15</b>		72.37	<b>76.15</b>	O
SATIMAGE	79.60	<b>80.75</b>	O	79.85	<b>80.75</b>		79.70	<b>80.75</b>		78.30	<b>80.75</b>	O
SEGMENT	77.04	<b>89.63</b>	O	77.04	<b>89.63</b>	O	76.67	<b>89.63</b>	O	89.51	<b>89.63</b>	
SPAMBASE	79.50	<b>79.87</b>		79.50	<b>79.87</b>		<b>79.90</b>	79.87		79.39	<b>79.87</b>	
VEHICLE	46.28	<b>69.83</b>	O	46.28	<b>69.83</b>	O	46.52	<b>69.83</b>	O	67.21	<b>69.83</b>	O
WAVEFORM	80.02	<b>82.66</b>	O	80.02	<b>82.66</b>	O	80.06	<b>82.66</b>	O	80.68	<b>82.66</b>	O
WDBC	93.33	<b>93.50</b>		<b>95.96</b>	93.50	X	92.98	<b>93.50</b>		<b>94.03</b>	93.50	

〈표 5〉 RBE와 각 방법들과의 정확도 비교 (SMO with Linear Kernel)

SMO DATA SET	Single	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	87.52	<b>89.28</b>	O	87.52	<b>89.28</b>	O	87.84	<b>89.28</b>	O	88.16	<b>89.28</b>	O
GLASS	53.94	<b>61.51</b>	O	58.14	<b>61.51</b>	O	53.92	<b>61.51</b>	O	59.24	<b>61.51</b>	O
IONOSPHERE	88.03	<b>88.91</b>		87.78	<b>88.91</b>	O	87.47	<b>88.91</b>	O	87.77	<b>88.91</b>	O
IRIS	<b>96.67</b>	95.33	X	<b>98.00</b>	95.33	X	<b>96.00</b>	95.33		<b>96.00</b>	95.33	
PIMA	<b>77.19</b>	76.81		<b>77.19</b>	76.81		76.80	<b>76.81</b>		74.19	<b>76.81</b>	O
SATIMAGE	85.05	<b>86.40</b>	O	85.30	<b>86.40</b>	O	84.95	<b>86.40</b>	O	86.15	<b>86.40</b>	
SEGMENT	92.72	92.59		<b>92.72</b>	92.59		<b>93.09</b>	92.59		<b>93.46</b>	92.59	X
SPAMBASE	90.39	<b>91.87</b>	O	90.83	<b>91.87</b>		91.14	<b>91.87</b>		91.76	<b>91.87</b>	
VEHICLE	74.55	<b>77.16</b>	O	74.43	<b>77.16</b>	O	73.95	<b>77.16</b>	O	<b>77.40</b>	77.16	
WAVEFORM	86.60	86.56		<b>86.60</b>	86.56		86.34	<b>86.56</b>		85.74	<b>86.56</b>	
WDBC	97.89	97.54		<b>97.89</b>	97.54		<b>97.89</b>	97.54		95.97	<b>97.54</b>	O

〈표 6〉 RBE와 각 방법들과의 정확도 비교 (C4.5)

C4.5 DATA SET	Single	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	77.62	<b>79.37</b>	O	<b>79.54</b>	79.37		<b>82.26</b>	79.37	X	79.05	<b>79.37</b>	
GLASS	69.39	<b>70.82</b>	O	<b>74.70</b>	70.82	X	<b>74.70</b>	70.82	X	65.71	<b>70.82</b>	O
IONOSPHERE	88.05	<b>90.34</b>	O	<b>92.62</b>	90.34	X	<b>90.63</b>	90.34		88.06	<b>90.34</b>	O
IRIS	94.67	94.67		94.00	<b>94.67</b>		<b>95.33</b>	94.67		93.33	<b>94.67</b>	O
PIMA	73.53	<b>74.57</b>		73.14	<b>74.57</b>	O	<b>76.40</b>	74.57	X	72.09	<b>74.57</b>	O
SATIMAGE	85.20	<b>87.95</b>	O	<b>88.20</b>	87.95		<b>88.90</b>	87.95		86.05	<b>87.95</b>	O
SEGMENT	96.17	<b>97.04</b>		<b>97.90</b>	97.04		96.17	<b>97.04</b>		95.80	<b>97.04</b>	O
SPAMBASE	93.05	<b>94.31</b>	O	<b>95.26</b>	94.31		<b>94.35</b>	94.31		93.00	<b>94.31</b>	O
VEHICLE	<b>75.02</b>	74.79		<b>76.80</b>	74.79	X	<b>76.21</b>	74.79	X	73.37	<b>74.79</b>	O
WAVEFORM	75.36	<b>77.18</b>	O	<b>81.56</b>	77.18	X	<b>80.78</b>	77.18	X	75.10	<b>77.18</b>	O
WDBC	92.80	<b>94.03</b>		<b>96.49</b>	94.03	X	<b>94.56</b>	94.03		93.68	<b>94.03</b>	

를 한 표이다. 정확도가 더 높은 곳에 굵은 글씨로 표시하였고 그 차이가 두드러지게 RBE가 좋은 곳에는 O표시를 하였고 나쁜 곳에는 X표시를 하였다.

단일 분류기와의 정확도 비교에서는 기본 학습알고리즘에 상관없이 대체적으로 정확도가 상승했음을 알 수 있다. 이는 RBE가 기본 학습 알고리즘에 독립적으로 앙상블 분류기로서의 역할을 하고 있음을 나타내고 있다.

다른 앙상블 방법인 배깅과 부스팅과의 정확도 비교를 살펴보자. 표에서 보는 바와 같이 나이브 베이즈와 SVM을 기

본 학습알고리즘으로 사용한 앙상블들의 비교에서는 RBE가 대체적으로 정확도가 높음을 알 수 있다. 특히 일부 데이터에 대해서는 정확도의 차이가 두드러지게 성능이 좋았으며, 그 외에도 다른 앙상블 방법보다 크게 성능이 떨어지는 경우는 거의 없었다. 이제 C4.5를 기본 학습알고리즘으로 실험한 결과를 살펴보자. C4.5를 기본 학습알고리즘으로 선택하였을 때는 부스팅과 배깅의 앙상블 기법이 RBE보다 성능이 대체적으로 좋은 것으로 나타났다. 이는 단일 분류기와의 실험에서 살펴본 것처럼 RBE도 정확도가 상승했지만 부스

팅과 배깅이 정확도를 그보다 더 상승시켰기 때문에 이와 같은 결과가 나타난 것이다. 2장에서 논한 바와 같이 부스팅과 배깅이 결정 트리(ex. C4.5)와 같은 불안정한(Unstable)한 알고리즘에서 큰 정확도 상승을 보이는 것을 알 수 있다. 위의 실험 결과들을 종합하여 볼 때, 나이브 베이즈와 SVM과 같은 안정적인 알고리즘에서는 RBE가 성능이 더 좋았음을 알 수 있고 C4.5와 같은 불안정한 알고리즘에서는 기존의 앙상블 방법인 부스팅과 배깅이 더 좋다는 것을 알 수 있다.

이제 RBC와의 비교를 살펴보면 표에서 보는 바와 같이 기본 학습알고리즘과 상관없이 대부분의 데이터 셋에서 RBE가 정확도가 더 높았으며, 일부 데이터에 대해서는 주목할 만한 정확도 차이를 보였다. 이 결과는 3.2절에서 설명한 RBE를 사용하는 이점을 실험적으로 뒷받침해주고 있다.

## 5. 결론 및 향후과제

지금까지 새로운 앙상블 방법으로서 지역 기반 분류기의 앙상블 학습을 제시하였다. 이 방법은 기존의 앙상블 방법과 비교했을 때 안정적인 알고리즘(Stable Algorithm)을 기본 학습기로 사용한 경우에 더 뛰어난 정확도를 보였다.

이는 좋은 앙상블의 조건 중에 하나인 분류기간에 서로 독립적으로 입력공간의 다른 곳에서 예러를 가져야한다는 것과 관련하여 정보획득에 의해 속성을 선택하여 그 속성을 기준으로 데이터를 분할하여 데이터 사이의 분포를 최대한 다르고 쉽게 만든다. 이렇게 데이터를 분할해 나가면서 데이터의 일부만 보고 학습을 하기 때문에 분류기의 다양성 측면에서 다른 앙상블 방법보다 더 좋을 것이라고 기대된다.

또한, 또 다른 좋은 앙상블의 조건인 각 분류기는 정확해야한다는 것과 관련하여 데이터를 분할하여 학습을 하면 그 지역을 제외한 다른 데이터에 대해서는 성능이 떨어지기 때문에 그 분류기가 정확하게 분류할 수 있는 지역에만 투표를 참여함으로써 분류를 결정하게 된다.

위와 같은 두 가지 특성으로 인해 지역 기반 분류기의 앙상블 학습이 SVM과 나이브 베이즈와 같은 안정적인 알고리즘에서 다른 앙상블 방법보다 더 높은 성능 증가를 가져온 것으로 생각된다. 이러한 방법을 가능하게 하는 것은 각 분류기가 어느 인스턴스에서 높은 성능을 갖는지 알기 때문에 데이터에 큰 변화를 줄 수 있는 것이다.

이처럼 지역 기반 분류기의 앙상블 학습은 기본 분류 알고리즘의 성질에 좀 더 독립적으로 작동할 수 있다는 점에서 그 의의가 있다고 할 수 있다.

본 논문에서는 RBE가 데이터 분할단계로 3단계까지만 진행하여 각 말단 노드의 깊이가 같은 3단계의 전 이진 트리(Complete Binary Tree)로 구성하여 총 7개의 분류기로 실험하였다. 전 이진 트리로 구성된 이유는 앙상블 분류기의 개수에 대한 파라메타로서 데이터를 몇 등분 할 것인지, 몇 단계로 분할할지를 지정해 놓았기 때문이다(그림 1 참조). 이를 좀 더 확장하여 데이터의 이등분이 아닌 다중 분

할, 그리고 각 지역별로 다른 분할 단계를 거쳐 전 이진 트리가 아닌 앙상블 분류기를 생성시킬 수도 있다. 이를 위해서는 이산 값을 갖는 속성에 대해서도 분할할 수 있는 기준이 필요하고 데이터의 다중 분할과 분할 단계의 최적화 기준을 마련해야한다. 이는 향후과제로 남겨둔다.

## 참고 문헌

- [1] Bauer, E. & Kohavi, R., "An Empirical Comparison of Voting Classification Algorithm: Bagging, Boosting, and Variants", *Machine Learning*, 36(1-2), pp. 105-142, 1999.
- [2] Blake, C. & Merz, C., UCI Repository of Machine Learning Database, <http://www.ics.uci.edu/~mllearn/MLRepository.html>, 1998.
- [3] Breiman, L., "Bias, Variance, and Arcing Classifiers", Technical Report TR, 460, UC Berkeley, 1996.
- [4] Breiman, L., "Bagging Predictors", *Machine Learning*, 24(2), pp. 123-140, 1996.
- [5] Dietterich, T., "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization", *Machine Learning*, 40(2), pp. 139-157, 2000.
- [6] Dietterich, T., "Ensemble method in Machine learning", In J. Kittler and F. Roli (Ed.) *First International Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, pp. 1-15, 2000.
- [7] Dietterich, T., "Ensemble Learning", In *The Handbook of Brain Theory and Neural Networks*, Second edition, The MIT Press, pp. 405-408, 2002.
- [8] Freund, Y. & Schapire, R., "Experiments with a new boosting algorithm", In *Proc. of the Thirteenth International Conference on Machine Learning*, pp. 148-156, 1996.
- [9] Freund, Y. & Schapire, R., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Science*, 55, pp. 119-139, 1997.
- [10] Friedman, J., Hastie, T. & Tibshirani, R., "Additive Logistic Regression: a Statistical View of Boosting", *Annals of Statistics*, 28(2), pp. 337-374, 2000.
- [11] Hansen, L. & Salamon, P., "Neural Network Ensembles", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12, pp. 993-1001, 1990.
- [12] L.I. Kuncheva and C.J. Whitaker. "Measures of diversity in classifier ensembles", *Machine Learning*, 51, pp. 181-207, 2003.
- [13] Opitz, D. & Maclin, R., "Popular Ensemble Methods: An Empirical Study", *Journal of Artificial Intelligence Research*, 11, pp. 169-198, 1999.

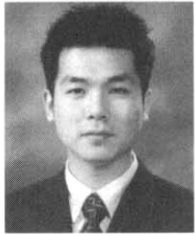
[14] Platt, J. Fast Training of Support Vector Machines using Sequential Minimal Optimization, chapter 12, pp. 185-208, The MIT Press, 1999.

[15] Quinlan, J., "Induction of Decision Tree", Machine Learning, 1(1), pp. 81-106, 1986.

[16] Quinlan, J., C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[17] Quinlan, J., "Bagging, Boosting, and C4.5.", In Proc. of the Thirteenth National Conference on Artificial Intelligence, pp. 725-730, 1996.

[18] Witten, I. & Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation, Second edition, Morgan Kaufmann, 2005.



**최 성 하**

e-mail : belle79@gmail.com  
 2005년 서강대학교 컴퓨터학과(학사)  
 2007년~현재 서강대학교 대학원  
 컴퓨터학과 석사과정  
 관심분야: 기계학습, 패턴인식 등



**이 병 우**

e-mail : elva1212@gmail.com  
 2006년 서강대학교 컴퓨터학과(학사)  
 2006년~현재 서강대학교 대학원 컴퓨터학과  
 재학  
 관심분야: 기계학습, 데이터마이닝 등



**양 지 훈**

e-mail : yangjh@sogang.ac.kr  
 1987년 서강대학교 전자계산학과(학사)  
 1989년 아이오와 주립대학교 대학원  
 컴퓨터학과(공학석사)  
 1999년 아이오와 주립대학교 대학원  
 컴퓨터학과(공학박사)

1999년~2000년 HRL Lab. LLC., Research Staff Member  
 2000년~2002년 SRA International, Inc., Professional Staff  
 Member

2002년~현재 서강대학교 컴퓨터학과 부교수  
 관심분야: 기계학습, 데이터마이닝, 인공지능, 생물정보학 등