

지역 전문가의 앙상블 학습

(Ensemble Learning of Regional Experts)

이 병 우 ^{*} 양 지 훈 ^{**}
(Byung-woo Lee) (Yang jihoon)

김 선 호 ^{***}
(Seonho Kim)

요약 본 논문에서는 지역 전문가를 이용한 새로운 앙상블 방법을 제시하고자 한다. 이 앙상블 방법에서는 학습 데이터를 분할하여 속성 공간의 서로 다른 지역을 이용하여 전문가를 학습시킨다. 새로운 데이터를 분류할 때에는 그 데이터가 속한 지역을 담당하는 전문가들로 가중치 투표표를 한다. UCI 기계 학습 데이터 저장소에 있는 10개의 데이터를 이용하여 단일 분류기, Bagging, Adaboost와 정확도를 비교하였다. 학습 알고리즘으로는 SVM, Naive Bayes, C4.5를 사용하였다. 그 결과 지역 전문가의 앙상블 학습 방법이 C4.5를 학습 알고리즘으로 사용한 Bagging, Adaboost와는 비슷한 성능을 보였으며 나머지 분류기보다는 좋은 성능을 보였다.

키워드 : 앙상블 학습, 부스팅, 배깅, 지역 전문가

Abstract We present a new ensemble learning method that employs the set of region experts, each of which learns to handle a subset of the training data. We split the training data and generate experts for different regions in the feature space. When classifying a data, we apply a weighted voting among the experts that include the data in their region. We used ten datasets to compare

· 이 연구는 서강대학교 특별연구비 지원에 의한 연구임
· 이 논문은 2008 한국컴퓨터종합학술대회에서 '지역 전문가의 앙상블 학습'의 제목으로 발표된 논문을 확장한 것임

^{*} 학생회원 : 서강대학교 컴퓨터공학과
elval212@naver.com

^{**} 종신회원 : 서강대학교 컴퓨터공학과 교수
yangjh@sogang.ac.kr

^{***} 비회원 : 서강대학교 컴퓨터공학과
shkim@lex.yonsei.ac.kr

논문접수 : 2008년 9월 3일

심사완료 : 2008년 11월 16일

Copyright©2009 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제2호(2009.2)

the performance of our new ensemble method with that of single classifiers as well as other ensemble methods such as Bagging and Adaboost. We used SMO, Naive Bayes and C4.5 as base learning algorithms. As a result, we found that the performance of our method is comparable to that of Adaboost and Bagging when the base learner is C4.5. In the remaining cases, our method outperformed the benchmark methods.

Key words : Ensemble Learning, Boosting, Bagging, Region Experts

1. 서론

기계 학습분야에서 분류기의 정확도를 높이기 위해 다양한 방법들이 연구되어 왔다. 앙상블 학습은 이러한 방법들 중 하나로써, 학습 데이터를 정확하게 표현할 수 있는 하나의 가설을 선택하는 것이 아니라 가설들의 집합을 구성한다. 새로운 데이터를 예측할 때에는 그 가설들의 예측을 결합하여 최종 결정을 내린다[1].

앙상블 학습을 하는 이유는 다양한 분류기의 결과를 결합함으로써 좀 더 신뢰성이 높은 예측값을 얻을 수 있으며 복잡한 문제를 풀기 쉽도록 여러 개의 하위 문제로 나눔으로써 보다 효율적으로 해결할 수 있기 때문이다. 또한 하나의 분류기가 모든 패턴에 대해서 좋은 성능을 발휘할 수 없기 때문에 다양한 분류기들을 사용하는 것이다. 다양한 논문을 통해서 앙상블 방법이 단일 분류기의 성능을 향상시킬 수 있다고 실험적으로 증명되었다[2-4].

Bagging[5]과 Boosting[6]의 경우 새로운 데이터를 분류할 때, 앙상블을 구성하는 기본 학습기(Base Learner)가 그 데이터를 학습했는지를 고려하지 않는다. 즉, 새로운 데이터를 학습하지 않은 기본 학습기도 투표에 참여시킨다. 만약 기본 학습기가 잘 분류할 수 있는 데이터를 알 수 있다면 보다 향상된 앙상블을 구성할 수 있을 것이다. 따라서 우리는 지역 전문가를 이용한 새로운 앙상블 방법을 제시하고자 한다. 이 앙상블 방법에서는 학습 데이터를 분할하여 속성 공간의 서로 다른 지역을 이용하여 전문가를 학습시킨다. 새로운 데이터를 분류할 때에는 그 데이터가 속한 지역을 담당하는 전문가들로 가중치 투표표를 한다.

본 논문은 총 5절로 구성되어 있다. 2절에서는 좋은 앙상블의 조건을 소개하고, 3절에서는 본 논문에서 제시하는 지역 전문가의 앙상블 학습 방법에 대하여 기술한다. 4절에서는 실험 결과를 기술하고 분석하여 마지막 절에서는 결론을 맺고 향후 과제에 대하여 언급하겠다.

2. 좋은 앙상블의 조건

Hansen과 Salamon은 앙상블이 단일 분류기의 정확

도를 향상시키기 위한 필요·충분 조건은 각 분류기들이 임의 추측(Random Guess)보다 정확해야 하고 다양해야(Diverse) 한다고 하였다[7]. 여기서 임의 추측보다 정확하다는 것은 정확도가 50% 이상임을 의미하며 다양하다는 것은 오분류하는 패턴이 서로 다르다는 것을 의미한다. 앙상블을 구성하는 분류기들이 다양하지 않을 경우, 한 분류기가 분류하고자 하는 인스턴스를 오분류한다면 나머지 분류기들도 오분류하게 되므로 앙상블을 하여도 최종 예측값은 잘못된 수밖에 없다. 그러나 만약 분류기들이 다양하다면 한 분류기가 오분류하더라도 다른 분류기에 의해서 옳게 분류될 수 있기 때문에 앙상블을 통해 옳게 분류될 수 있다.

학습 데이터의 작은 변화에도 불구하고 생성되는 분류기뿐만 아니라 정확도에서도 상대적으로 큰 변화가 생길 때 그 학습 알고리즘을 불안정(Unstable)하다고 한다[8]. 결정트리를 생성시킬 때 사용되는 C4.5와 같은 알고리즘이 대표적인 불안정한 학습 알고리즘이다. 반면에 불안정 알고리즘과 달리 학습 데이터가 변화더라도 생성되는 분류기는 크게 변하지 않는 알고리즘을 안정(Stable)하다고 한다. SVM, Naive Bayes가 대표적인 안정적인 학습 알고리즘이다.

Breiman은 Bagging이 이러한 불안정한 학습 알고리즘으로 생성한 분류기들의 성능을 향상시킨다고 하였다 [4]. 반면에 Bagging과 Boosting 같은 앙상블 학습이 안정적인 학습 알고리즘도 정확도를 상승시킬 수 있다는 연구도 있지만 불안정한 학습 알고리즘에 비하면 극히 일부의 데이터에 대해서만 성능 향상을 보였다. 심지어 SVM을 이용한 앙상블이 단일 분류기보다 성능이 떨어진다는 연구 결과도 있었다[9-11]. 이처럼 불안정한 학습 알고리즘을 사용할 경우 정확도가 크게 상승하는 이유는 다양성을 확보할 수 있기 때문이다. Bagging과 Boosting은 샘플링을 통해 학습 데이터에 변화를 줄 수 있다. 불안정한 학습 알고리즘을 사용했을 경우 이러한 학습 데이터의 변화는 생성되는 분류기에도 큰 변화를 주게 된다. 이로 인해 분류기들의 다양성을 어느 정도 확보할 수 있게 되는 것이다.

3. 지역 전문가의 앙상블 학습

본 논문에서 제시하는 지역 전문가의 앙상블 학습은 결정 트리와 비슷하게 데이터를 분할하면서 학습을 한다. 그러나 결정 트리과 달리 말단 노드(Leaf Node)뿐만 아니라 모든 노드에 분류기가 존재하며 데이터를 분할하기 위한 속성(Attribute)을 선택하는 방법도 결정 트리와 다르다. 또한 Bagging이나 Boosting이 기본 학습기를 생성할 때 모두 같은 학습 알고리즘을 사용하는 것과는 달리 서로 다른 기본 학습 알고리즘을 사용하

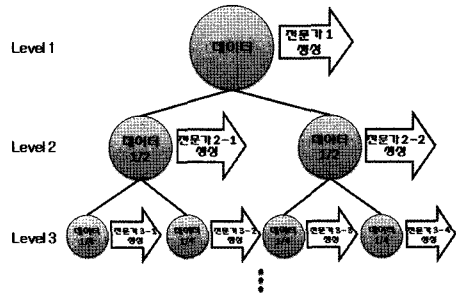


그림 1 지역 전문가의 앙상블 학습 개괄도

수 있다. 앞으로 이 앙상블 학습을 ELRE (Ensemble Learning of Region Experts)라 지칭한다. 그림 1은 ELRE의 개괄도를 보여준다.

3.1 학습 과정

ELRE는 학습 데이터를 분할하며 다양한 분류기를 생성한다. 분류기를 생성할 때 주어진 학습 알고리즘 중 그 지역에서 가장 좋은 성능을 보이는 학습 알고리즘을 이용한다. 이를 위해 주어진 학습 알고리즘 중 10겹 교차검증(10-fold Cross-Validation) 정확도가 가장 높은 알고리즘을 선택한다. 만약 10겹 교차검증 정확도가 동일할 경우 학습 정확도(Training Accuracy)가 높은 알고리즘을 선택한다. 두 정확도가 모두 동일할 경우, 만약 부모 노드(Parent Node)가 있다면 뿌리 노드(Root Node)부터 부모 노드까지의 경로(Path)에 존재하는 분류기에서 가장 많이 선택된 학습 알고리즘을 선택한다. 만약 부모 노드가 없다면, 즉 뿌리 노드라면 미리 정해놓은 학습 알고리즘의 우선순위에 따라 학습 알고리즘을 선택한다.

선택된 학습 알고리즘으로 그 지역의 학습 데이터를 학습하여 분류기를 생성한다. 이 분류기를 그 지역의 전문가(Expert)라 하자. 전문가를 생성한 후 학습 데이터를 분할한다. 학습 데이터 S를 이등분하기 위해 각 속성에 대하여 이분할 값을 구한다. 정확히 이등분이 되지 않을 경우 최대한 이등분이 되는 값을 구한다. 본 논문에서는 이를 위해 속성값(Attribute Value)이 연속적인 값(Continuous Value)인 데이터만을 고려한다. 식 (1) [12], (2)를 이용하여 각 속성의 Score를 구하고 Score가 가장 큰 속성을 선택한다.

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

$$Score(S, A) = Entropy(S) - \frac{||S_1| - |S_2||}{|S|} \sum_{v=1}^2 \frac{|S_v|}{|S|} \cdot Entropy(S_v) \tag{2}$$

c는 클래스 개수, p_i는 S 중에서 클래스 i에 속하는 데이터의 비율, A는 데이터의 속성, S₁과 S₂는 A의 분할 값에 의하여 분할된 데이터이다.

식 (2)를 이용할 경우, 분할 속성을 선택할 때 결정 트

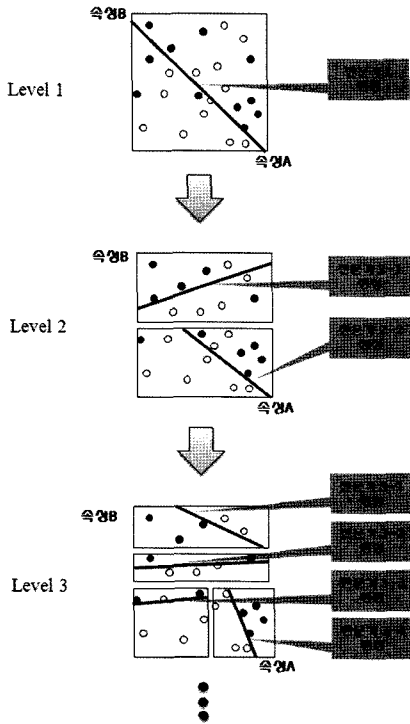


그림 3 ELRE의 학습 과정

리처럼 단순히 정보획득(Information Gain)[13]만을 보는 것이 아니라 학습 데이터를 얼마나 균등하게 나눌 수 있는지를 반영하게 된다. 학습 데이터를 균등하게 나누는 이유는 한쪽으로 학습 데이터가 몰릴 경우 불필요하게 트리의 크기가 커지기 때문이다. 또한 학습 데이터가 부족한 다른 쪽은 학습 데이터 부족으로 인하여 충분한 학습을 하지 못하기 때문이다. 학습 데이터가 충분하지 않을 경우 학습 알고리즘이 표현할 수 있는 가설 공간에 학습 데이터에 대해서 정확도가 같은 여러 개의 가설들이 존재할 수 있으므로 이중에서 선택한 가설이 실제 함수(Target Function)를 정확하게 표현하지 못할 수 있다[1].

정보획득이 큰 속성을 선택하는 이유는 정보획득이 큰 속성을 선택함으로써 클래스가 한쪽으로 치우치게 되며 이는 결과적으로 데이터를 분할해 감에 따라 분류가 쉬울 것이라는 기대를 할 수 있기 때문이다.

그림 2와 같이 이등분하여 생성한 데이터에 위의 과정을 반복하여 전문가를 생성해 나간다. 다음의 분할 종료 조건을 만족할 시 분할을 종료하거나 미리 정해놓은 레벨까지 확장 후 종료한다.

- 학습 데이터의 개수가 10개 미만일 경우
- Entropy가 0일 경우

3.2 분류 과정

새로운 패턴을 분류할 때에는 그 패턴이 포함된 지역

을 담당하는 전문가만을 이용한다. 그림 3은 ELRE의 분류 과정을 보여준다.

그림 3과 같이 전체 학습 데이터를 학습한 뿌리 노드부터 분류하고자 하는 패턴이 속한 지역만을 학습한 말단 노드까지의 경로에 있는 전문가들이 투표를 한다. 학습 과정에서 이 전문가들만이 이 패턴에 대해서 학습을 했기 때문에 나머지 전문가들은 투표에 참여하지 않는다. 결정 트리와 같이 말단 노드에 의해서만 분류를 하지 않는 이유는 말단 노드로 갈수록 과잉적합(Overfitting)이 발생할 여지가 있기 때문이다. 조상 노드(Ancessor Node)들의 예측값을 이용함으로써 과잉적합의 문제를 완화할 수 있으며 앙상블 학습의 효과도 얻을 수 있게 되는 것이다.

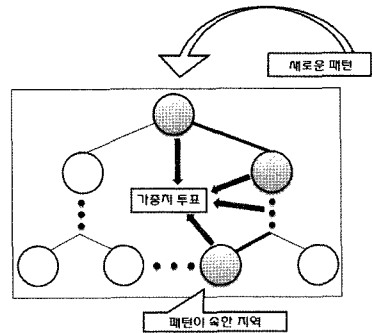


그림 3 ELRE의 분류 과정

3.1장에서 언급했듯이 데이터를 분할하는 이유는 분할해 감에 따라 분류가 쉬울 것이라 기대했기 때문이다. 그러나 실제 학습 정확도를 측정해보았을 때 그렇지 않은 경우도 존재하였다. 따라서 말단으로 갈수록 높은 가중치를 주기보다는 학습 정확도에 따라 서로 다른 가중치를 주어 투표를 한다. 식 (3)은 전문가의 가중치를 구하는 식이다.

$$\text{Expert_Weight}(N) = \frac{|D|}{|PD|} \cdot PW \quad (3)$$

N 은 가중치를 구하고자 하는 전문가, D 는 N 에 할당된 학습 데이터 중 N 이 정확하게 예측한 데이터, PD 는 N 에 할당된 학습 데이터 중 부모 노드의 전문가가 정확하게 예측한 데이터, PW 는 N 의 부모 노드에 있는 전문가의 가중치이다.

전문가로 새로운 패턴을 예측할 때 예측값을 구하는 것이 아니라 각 클래스별로 클래스 확률(Class Probability)을 구하고 이 클래스 확률에 그 전문가의 가중치를 곱하여 각 클래스별로 투표값을 구한다. 각 전문가에서 구한 투표값을 클래스별로 합하고 그 중 투표값이 가장 큰 클래스를 ELRE의 예측값으로 정한다. 식 (4)는 클래스의 투표값을 구하는 식이고 식 (5)는 최종 예

측값을 구하는 식이다.

$$Voting(x, P) =$$

$$\sum_{v=1}^L Expert_Weight(N_v) \cdot Class_Prob(x, P, N_v) \quad (4)$$

$$Classification(P) = \text{argmax}_{i \in C} Voting(i, P) \quad (5)$$

x 는 클래스, P 는 분류하고자 하는 패턴, L 은 투표에 참여하는 전문가의 개수, N_v 는 투표에 참여하는 전문가, $Class_Prob(x, P, N_v)$ 는 분류기 N_v 가 패턴 P 를 클래스 x 라 예측할 확률, C 는 클래스의 집합이다.

4. 실험

4.1 실험 방법

각 노드의 전문가를 생성하기 위해 기본 학습 알고리즘으로서 Naive Bayes, SVM을 구현한 SMO[14], C4.5를 사용하였다. 본 연구에서 SMO는 선형커널을 사용하였다. 3.1장에서 언급한 학습 알고리즘의 우선순위는 SMO, Naive Bayes, C4.5 순으로 하였다. 기본 학습 알고리즘은 공개 데이터마이닝 소프트웨어인 Weka 3.5.7[15]을 사용하였다. UCI Machine Learning Repository[16]에 있는 10개의 데이터 세트를 이용하여 실험하였으며 10겹 교차검증으로 정확도를 측정하였다. 표 1은 실험에 사용한 데이터 세트에 대한 정보이다.

표 1 실험에 사용한 데이터 세트

데이터 세트	데이터 개수	클래스 개수	속성 개수
Vehicle	846	4	18
Glass	213	6	9
Abalone	4177	28	8
Hypothyroid	215	3	5
Ionosphere	351	2	34
Ecoli	336	8	7
SPECTF	267	2	44
Segment	2310	7	19
Page-Blocks	5473	5	10
Spambase	4601	2	57

ELRE의 성능을 검증하기 위해 기본 학습 알고리즘의 단일 분류기, Boosting의 한 종류인 Adaboost[17]의 앙상블 분류기, Bagging의 앙상블 분류기를 사용하였다. ELRE는 레벨 3까지 확장하여 총 7개의 지역 전문가를 생성하였다. 총 7개의 전문가를 생성하였지만 각 전문가를 생성하기 위해서는 3개의 분류기를 생성하기 때문에 실제 생성한 분류기의 개수는 21개이다. 따라서 공정한 비교를 위해 Adaboost와 Bagging도 21개의 기본 분류기를 생성하였다.

4.2 실험 결과

표 2는 각 분류기의 정확도를 비교한 표이다. 각 데이터 세트별로 정확도가 가장 높은 분류기의 정확도를 굵은 글씨로 표시하였다.

이 실험 결과를 통해서 알 수 있듯이, ELRE는 모든 단일 분류기보다 정확도가 높을 뿐만 아니라 SMO, Naive Bayes를 기본 학습 알고리즘으로 사용한 Bagging이나 Adaboost보다 정확도가 높았다. SMO와 Naive Bayes와 같이 안정한 알고리즘을 사용할 경우 학습 데이터가 변하더라도 분류기는 크게 변하지 않기 때문에 Adaboost와 Bagging은 크게 성능 향상을 하지 못한다. 반면 ELRE는 데이터 세트뿐만 아니라 학습 알고리즘도 분류기들 간에 서로 다를 수 있다. 따라서 단일 분류기 중에서 SMO나 Naive Bayes가 성능이 가장 좋은 데이터에서도 ELRE는 서로 다른 학습 알고리즘으로 생성된 전문가를 이용하여 다양성을 확보할 수 있기 때문에 Adaboost, Bagging보다 좋은 정확도를 보인다고 생각된다. 또한 데이터를 분할함으로써 데이터를 분류하기 쉽도록 만들어 주었으며 분류하고자 하는 패턴을 학습한 전문가들만이 투표에 참여하였기 때문에 앙상블을 구성하는 분류기들의 정확도 측면에서도 Bagging과 Adaboost 보다 앞설 수 있다.

그러나 C4.5를 기본 학습 알고리즘으로 사용한 Bagging과 Adaboost와는 비슷한 성능을 보였다. ELRE는 실제로 총 7개의 분류기를 갖고 있지만 Adaboost는 21

표 2 ELRE와 각 방법들과의 정확도 비교

	SMO			Naive Bayes			C4.5			ELRE
	단일	Ada	Bagging	단일	Ada	Bagging	단일	Ada	Bagging	
Vehicle	74.55	74.43	74.90	46.28	46.28	46.52	75.02	76.92	74.32	77.27
Glass	53.94	58.14	55.22	47.45	47.92	50.76	69.39	77.42	76.56	74.18
Abalone	25.31	25.31	25.58	23.80	23.80	23.64	20.90	22.15	24.47	27.10
Hypothyroid	89.85	95.81	89.77	96.75	95.81	97.21	94.91	94.42	93.49	97.23
Ionosphere	88.03	87.78	88.04	82.90	91.76	82.62	88.05	93.18	91.20	92.33
Ecoli	84.22	86.30	83.63	86.29	86.23	86.27	70.30	82.43	84.50	86.62
SPECTF	79.42	76.40	79.40	67.86	68.54	70.41	74.54	80.15	79.78	81.30
Segment	92.90	93.20	92.90	79.65	79.65	79.96	96.80	98.27	97.45	97.45
Page-Blocks	92.91	92.91	93.42	90.15	90.15	90.02	97.06	97.19	97.37	97.17
Spambase	90.39	90.78	90.81	79.50	79.50	79.70	93.04	95.35	94.24	93.74

개의 분류기를 갖고 있으며, 분류할 때 ELRE는 3개의 분류기를 사용하지만 Adaboost와 Bagging은 21개의 분류기를 사용한다. 따라서 C4.5와 같이 불안정한 알고리즘을 사용할 경우 Adaboost와 Bagging이 ELRE보다 다양성 측면에서 앞설 수 있다. 또한 Adaboost와 Bagging은 샘플링을 통해서 기본 학습기를 학습시키기 때문에 기본 학습기의 학습 데이터가 크게 다를 수 있지만 ELRE에서 자식 노드에 할당된 데이터는 부모 노드에 반드시 포함되기 때문에 Adaboost와 Bagging만큼 서로 다른 학습 데이터를 만들지는 못한다. 이로 인해 C4.5를 사용할 경우 몇몇 데이터 세트에서 Adaboost가 ELRE 보다 좋은 정확도를 보였다고 생각된다.

5. 결론 및 향후 과제

본 논문에서 제시한 새로운 앙상블 방법인 ELRE는 할당된 학습 데이터에서 성능이 우수한 학습 알고리즘을 선택해주기 때문에 분류기를 학습시킬 때 어떤 학습 알고리즘을 사용할지 고민할 필요가 없다. 이로 인해 정확도가 높은 알고리즘으로 앙상블 분류기를 구성할 수 있으며 하나의 학습 알고리즘만으로는 표현할 수 없는 데이터를 다양한 학습 알고리즘을 이용하여 표현할 수 있다. 또한 투표에 참여하는 전문가들의 학습 알고리즘이 서로 다를 수 있기 때문에 전문가들 간의 다양성을 확보할 수 있다.

ELRE는 데이터를 분할함으로써 복잡한 학습 데이터를 분류하기 쉽게 만들어준다. 그렇지 않을 경우에는 정확도가 높은 전문가의 가치치를 높여주므로 보다 정확한 예측 결과를 얻을 수 있다. 또한 분류기들이 서로 다른 학습 데이터로 학습할 수 있도록 해주어 전문가들 간의 다양성을 확보할 수 있다.

ELRE에서 지역 전문가는 특정 지역만을 학습하고 분류하고자 하는 데이터를 학습한 전문가들만을 이용하여 투표하므로 보다 정확한 예측값을 얻을 수 있다. 또한 데이터를 분할하면서 학습하기 때문에 말단 노드로 갈수록 학습 데이터의 개수가 줄어들게 되어 과잉적합의 문제점이 발생할 수 있으나 조상 노드를 투표에 참여시킴으로써 이를 완화시켜준다.

본 연구결과, 레벨이 증가할수록 ELRE의 정확도가 높아지는 것만은 아니다. 높아지는 양상을 보이다가 어떤 레벨 이상이 되면 정확도가 다시 떨어지는 양상을 보이기도 한다. 물론 배경이나 부스팅도 무조건 분류기의 개수가 많다고 해서 성능이 향상되는 것이 아니라 오히려 떨어질 수도 있다. 본 논문에서 ELRE의 분할 종료 조건을 정하였지만 이 조건만으로는 이러한 현상을 막을 수가 없다. 따라서 불필요한 분할을 막기 위해서 분할 종료 조건을 추가할 필요가 있다. 또는 결정트리의 가지

치기(Pruning)와 같이 불필요한 전문가를 없애는 방법을 사용할 수도 있다. 이는 향후과제로 남겨둔다.

참고 문헌

- [1] T. G. Dietterich, "Ensemble method in machine learning," LNCS, Vol.1857, pp. 1-15, 2000.
- [2] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithm: bagging, boosting, and variants," Machine Learning, Vol.36, No.1-2, pp. 105-142, 1999.
- [3] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," Machine Learning, Vol.40, No.2, pp. 139-157, 2000.
- [4] D. Optiz and R. Maclin, "Popular ensemble methods: an empirical study," Journal of AIR, Vol.11, pp. 169-198, 1999.
- [5] L. Breiman, "Bagging predictors," Machine Learning, Vol.24, No.2, pp. 123-140, 1996.
- [6] Y. Freund and O. Schapire, "Experiments with a new boosting algorithm," Proc. 13th International Conf. on Machine Learning, pp. 148-156, 1996.
- [7] L. Hansen and P. Salamon, "Neural network ensembles," IEEE Trans. PAMI, Vol.12, pp. 993-1001, 1990.
- [8] R. O. Duda, P. E. Hart and D. G. Stork, 2nd ed., Pattern Classification, Wiley-interscience, 2000.
- [9] G. Valentini, M. Muselli and F. Ruffino, "Bagged Ensembles of SVMs for Gene Expression Data Analysis," The IEEE-INNS-ENNS International Joint Conference on Neural Networks, pp. 1844-1849, 2003.
- [10] I. Buciu, C. Kotropoulos and I. Pitas, "Combining support vector machines for accuracy face detection," Proc. ICIP, pp. 1054-1057, 2001.
- [11] T. Evgeniou, L. Perez-Breva, M. Pontil and T. Poggio, "Bound on the generalization performance of kernel machine ensembles," Proc. ICML, pp. 271-278, 2000.
- [12] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
- [13] J. Quinlan, "Induction of Decision Tree," Machine Learning, Vol.1, No.1, pp. 81-106, 1986.
- [14] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in Advances in Kernel Methods, ed. Schölkopf, B., Burges, C., Smola, A., The MIT Press, pp. 185-208, 1999.
- [15] I. Witten and E. Frank, 2nd ed., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation, Morgan Kaufmann, San Francisco, 2005.
- [16] C. Blaque and C. Merz, UCI Repository of Machine Learning Database, <http://www.ics.uci.edu/~mlnlp/LRepository.html>, 1998.
- [17] Y. Freund and R. Schapire, "A decision theoretic generalization of online learning and an application to boosting," Journal of CSC, Vol.55, pp. 119-139, 1997.