하이브리드 다중모델 학습기법을 이용한 자동 문서 분류

Automatic Text Categorization Using Hybrid Multiple Model Schemes

명순희(Soon-Hee Myoung)*, 김인철(In-Cheol Kim)**

초 록

본 논문에서는 다중 모델 기계학습 기법을 이용하여 자동 문서 분류의 성능과 신뢰도를 향상시킬 수 있는 연구와 실험 결과를 기술하였다. 기존의 다중 모델 기계 학습법들이 훈련 데이터 또는 학습 알고리즘의 편향에 의한 오류를 극복하고자 한 것인데 비해 본 논문에서 제안한 메타 학습을 이용한 하이브리드 다중 모델 방식은 이 두 가지의 오류 원인을 동시에 해소하고자 하였다. 다양한 문서 집합에 대한 실험 결과, 본 논문에서 제안한 하이브리드 다중 모델 학습법이 전반적으로 기존의 일반 다중모델 학습법들에 비해 높은 성능을 보였으며, 다중 모델의 결합 방식으로서 메타 학습이 투표 방식에 비해 효율적인 것으로 나타났다.

ABSTRACT

Inductive learning and classification techniques have been employed in various research and applications that organize textual data to solve the problem of information access. In this study, we develop hybrid model combination methods which incorporate the concepts and techniques for multiple modeling algorithms to improve the accuracy of text classification, and conduct experiments to evaluate the performances of proposed schemes. Boosted stacking, one of the extended stacking schemes proposed in this study yields higher accuracy relative to the conventional model combination methods and single classifiers.

키워드: 문서분류, 기계학습, 다중모델 학습, hybrid multiple model, text classification, multiple modeling algorithm

* 용인송담대학인터넷경영정보과 조교수(shmyoung@ysc.ac.kr)

** 경기대학교전자계산학과 부교수(kic@kyonggi.ac.kr)

■ 논문 접수일 : 2002. 10. 30

■ 게재 확정일 : 2002. 11. 30

1 서 론

웹문서, 디지털 라이브러리 등, 전자문서의 증가로 인하여 정보 요구를 충족하기 위한 정보접근(information access)의 문제점이 가중되고 있다. 정보 접근 문제를 문서의 자동 분류, 범주화, 군집화, 여과 등과 같이 비정형 텍스트 데이터 조직을 통해 해결하려는 노력이 정보검색, 인공지능 분야에서 다양하게 수행되고 있다(Mladeni'c 1998). 본 논문에서는 다중 모델 기계 학습 기법(multiple model scheme)을 이용하여 자동 텍스트 분류의 성능과 신뢰도를 향상시킬 수 있는 연구와 실험 결과를 기술하였다.

사전 분류된 훈련예를 바탕으로 데이터의 구조적 패턴을 명시적 지식으로 학습하여 신규 사례에 대하여 분류항목을 제시하는 귀납적 교사학습 기법은 문서 분류에도 높은 정확도를 보인다(Chen 1995, Hong 1999). 분류기, 또는 학습된 모델은 훈련 데이터에 대한 학습알고리즘의 적용결과로 유도된 구조적 패턴으로 데이터 구조를 이용하여 미분류 데이터에 클라스를 매핑하는 기능을 갖는다. 다중 모델 기계학습 기법은 동일한 훈련예 집합에서 유도된 다수 모델의 예측를 통합하여 안정성 있는 클라스 예측치를 제시하는 전략이다. 다중 모델 학습법은 Breiman의 Bagging 알고리즘을 비롯하여(Breiman 1996), Boosting(Schaphire 1999), Stacking 알고리즘(Wolpert 1992) 등 대표적 다중 모델 학습법 외에 다양한 파생 기법이 연구되고 있다(Bauer 1999).

본 연구에서는 다중 모델 학습법의 개념과 종류 별 특징을 기술하고 기존의 다중모델 학습법의 개념을 응용한 하이브리드 형태의 메타 학습 기법을 제안하고, 문서집합에 대한 분류 실험을 통해 하이브리드 다중모델 기법으로 유도된 다중모델 분류기와 일반 다중 모델 분류기, 그리고 단일 모델을 이용한 분류기의 성능을 비교하였다. 실험 데이터로는 MEDLINE 학술 기사 데이터, 웹문서, 그리고 유즈넷 기사를 이용하였는데 서로 다른 유형의 텍스트 데이터 집합에 대한 비교실험을 통해 텍스트 모델, 특징 집합 크기 등의 도메인 특성과 다중 모델학습법 및 메타 학습법의 효과를 분석하였다.

2 관련 연구

2.1 문서 모델

기계학습 알고리즘을 적용하기 위해 문서는 특징과 특징 값의 벡터로 표현되어야 하며, 문서 내용을 대표할 수 있는 주요 키워드를 문서의 특징으로 사용한다. 문서는 텍스트의 태그 제거, 불용어 처리, 어미 변화 처리 등의 텍스트 전처리를 거쳐 문서에 포함된 고유한 어휘의 모음으로 변환된다. 문서 모델을 생성하기 위해서는 특징 추출(feature extraction)을

거쳐 문서의 내용을 대표하는 특징의 부분 집합의 벡터만으로 표현하게 된다. 벡터는 특징 값에 따라 이진벡터(binary vector) 또는 가중치 벡터(weight vector)로 표현된다(Salton 1983).

Binary vector model : V =. (1,0,0, , 1)
Weight vector model : V = (2,0,1, ,5)

특징 추출은 문서 표현 벡터의 차수(dimensionality)를 줄임으로써 분류 성능의 희생 없이 계산의 양을 축소하고자 하는 것이다. 특징 추출의 방법으로는 일정한 척도(measure)로 특징들을 개별적으로 평가하여 필요한 만큼의 특징을 선택하는 여과방법(filtering)과 내포된 특정 분류기의 성능을 높일 수 있는 특징들의 부분집합을 점진적으로 구해가는 포장방법(wrapper), 두 가지 접근 방식이 있는데 일반적으로 텍스트 연구에서는 여과방식이 많이 사용된다.(Han,2001) 주요 여과 방법으로는 키워드의 문헌 내 빈도수와 역 문헌 빈도수에 기반하여 용어 가중치를 부여하는 TFIDF, 특정 특징이 분류된 데이터의 순도에 기여하는 정도를 측정한 정보 획득(information gain), 특징의 클라스 분포에 기초한 상관성을 나타내는 χ^2 test, 키워드 간의 유사도와 상관성에 의거 추출하는 LSI(Latent Semantic Indexing) 등이 있다(Schankar 2000). 이들 방식 가운데 문서 특징 추출에 우수한 성능을 보이는 정보 획득(IG)은 특정한 특징을 중심으로 분류 전의 엔트로피와 분류 후의 엔트로피의 차를 수치로 나타냄으로서 데이터의 주제 구분 역량을 측정한다.

클라스가 C, 문서집합이 S일 때 엔트로피는 수식(1)과 같이, 정보획득(IG)은 수식(2)와 같이 계산된다.

$$Entropy(S) = \sum_{i=1}^{C} - p_i \, log_2 p_i \quad (1)$$

$$InfoGain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

2.2 분류학습법

기계학습법 가운데 문서 분류에 많이 적용되어 온 학습법으로 Naive Bayesian, k-NN(k Nearest Neighbor), 결정 트리(Decision Tree)를 들 수 있는데 이들은 텍스트 분류에 적용되어 그 효과가 검증된 것으로 그 특징과 방법에 관하여 간략히 기술한다.

Naive Bayesian 학습법은 확률에 기반한 추론 방법을 제공하므로 결정 근거의 가독성(readability)이 높고, 데이터의 노이즈에 강한 장점이 있다. 문서의 주제 정보량의 측정을 특징의 분포에 의존하는 텍스트 처리 분야에서는 최적의 결정 규칙을 유도할 수 있는 학습 알고리즘으로 텍스트 처리 관련 연구에 광범위하게 이

용되어 왔다(Lewis 1994).

Naive Bayesian 분류 방식은 확보된 훈련예로 부터 특징이 각 클라스에 속할 조건부 확률을 계산하고, 분류 대상 문서가 각 클라스 별로 속할 조건부 확률이 가장 높은 클라스로 분류한다. 조건부 확률(conditional probability)은 Bayes 정리에 기초하여 수식(3)와 같이 계산한다.

$$P(c_i|d)=\frac{P(d|c_i)P(c_i)}{P(d)} \quad (3)$$

단, $D=d_1, d_2, \ldots, d_{|D|}$: 훈련문서

$C=c_1, c_2, \ldots, c_{|C|}$: 분류클래스

이를 위해 확보된 훈련문서로부터 각 특징의 사전 확률을 계산하는데 Naive Bayesian 학습법에서는 변수들 간의 조건부 독립성을 가정하여 계산을 용이하게 한다. 비현실적인 독립성 가정에도 불구하고 성능 저하는 미미한 것으로 나타나 실용성이 높은 방식으로 이용된다.(Lewis, 1994) 문서 분류 시 수식(4)의 방식으로 문서의 특징이 속한 클래스 별로 조건부 확률이 가장 높은 클래스로 분류하게 된다.

$$c^* = \underset{c_i \in C}{\operatorname{argmax}} P(c_i|d) \quad (4)$$

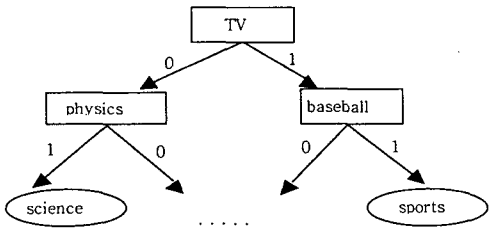
k-NN(Nearest Neighbor) 학습법은 대표적인 개체 기반 기계학습법(Instance based learning)으로 k-NN 방식은 훈련예를 사전 학습하지 않고 신규문서 Y가 주어지면 각 훈련문서와의 유클리디안 거리를 수식(5)를 이용하여 계산한 후 가장 가까운 k개의 문서가 속한 클라스 X로 분류한다.

$$X=(w_{x1}, w_{x2}, \ldots, w_{xn})$$
$$Y=(w_{y1}, w_{y2}, \ldots, w_{yn})$$
$$dist(X, Y)=$$
$$\sqrt{(w_{x1}-w_{y1})^2+\ldots+(W_{xn}-w_{yn})^2} \quad (5)$$

개선안으로는 k개의 이웃에 대하여 분류대상 문서 Y에 가까운 정도에 따라 이웃문서의 클라스 X에 가중치를 적용하는 거리가중(distance-weighted) k-NN 등이 이용된다(Mitchell 1997).

ID3, C4.5 알고리즘 등의 결정 트리(Decision Tree) 학습법은 귀납적 추론의 결과를 트리 모양의 구조로 나타낸다. 문서 분류에 적용 시 결정트리는 문서분류를 위한 지식을 표현하는데 비 단말 노드는 하나의 문서특징을 나타내고 한 노드에서 분기하는 각 가지는 하나의 특징값을 표시하며, 단말 노드는 하나의 분류 클라스를 나타낸다. 분기의 정점이 되는 각 노드의 선택은 매 분기 시 문서집합의

엔트로피를 낮추는데 각 특징이 기여하는 정도를 수식(1), (2)와 같이 계산한 정보 획득량(Information Gain, IG)에 따라 결정한다(Mitchell 1997).



〈그림 1〉 문서분류를 위한 결정트리

3 다중모델 학습법

3.1 개념적 분류

다중 모델 학습 전략은 학습예의 경미한 변화에도 학습된 분류기가 민감한 성능의 변화를 보이는 불안정성(unstability)을 해소하거나 문제 도메인에 적용된 학습알고리즘 간의 성능 편차를 상쇄하려는 것이다. 이론적으로는 훈련예의 분산(variance)효과와 기계 학습알고리즘 고유의 편향(bias)에서 비롯되는 오차를 축소하려는 것이다. 분산은 훈련예가 실세계의 분포를 반영하기 어려운 한계에서 오는 오차의 원인이며, 편향은 기계학습 고유의 한계에 기인하는 오차율이다. 편향은 학습 알고리즘 고유의 지식 및 개념 표현 방식(concept description language bias), 탐색 편향(search bias)이 주요 원인이다. 또한 결정 트리와 같이 분류기가 과도하게 훈련예의 구조를 반영함으로써 일반 사례의 분류 성능이 저하되는 것을 방지하고자 분류기의 구조를 단순화(pruning)하는 데서 비롯되는 과적합 방지 편향(overfitting-avoidance bias)도 오차의 원인을 제공한다.

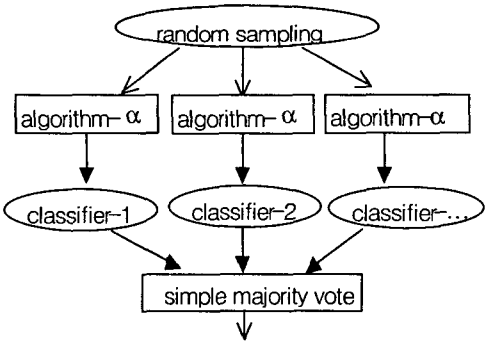
대표적인 다중모델 기법으로 Bagging과 Boosting은 훈련예의 분산에 기인한 오차를 축소하려는 전략이며, Stacking은 학습 알고리즘의 편향 효과를 상쇄하는 방식이다. 이들 다중학습 기법은 첫째, 모델 생성 과정과 둘째, 다중 모델의 분류 예측치를 취합하여 결론을 내는 방식에서 근본적인 차이를 보인다. 다중 모델은 동일 학습 알고리즘에 훈련예를 달리하여 생성한 동질적(homogeneous) 모델과, 다수의 알고리즘에 의한 이질적(heterogeneous) 모델로 구분할 수 있는데, Bagging과 Boosting이 동질적 모델을 다수결로 취합하며 Stacking은 이질적 모델의 분류 예측을 재학습하여 지능적으로 통합한다.

3.2 Bagging 및 Boosting 알고리즘

Bagging은 Boosting과 함께 동질적 모델의 분류 예측에 대해 다수결로 결론을 제시하는 메커니즘 때문에 '앙상블' 또는 '위원회' 방식의 투표알고리즘으로 불린다. 이들 방식은 순차적으로 모델을 생성하기 위한 훈련예를 확보하는데 동일한

데이터집합에 치환을 허용하는 bootstrap 샘플링을 통해 변화시킴으로써 실세계 데이터와의 유사성을 높여 제한된 데이터를 바탕으로 학습하는 데서 오는 오류를 완화하려는 것이다. 동일한 데이터라도 데이터 집합의 변화가 학습결과의 차이를 초래하기 때문에 데이터의 중복을 허용한다.

이와 같이 Bagging과 Boosting은 기본 방식은 유사하나 훈련예 샘플링 방식과 최종 분류결정 과정이 상이하다. Bagging은 임의의 횟수만큼 중복을 허용하며 무작위로 부분집합을 추출하고, 다중 모델의 분류 결과를 단순 다수결로 취합하여 클라스를 결정한다. 〈그림 2〉는 Bagging의 개념을 보여준다.



〈그림 2〉 Bagging의 개념도

Bagging 방식의 분류학습 원리는 학습 데이터집합 $L=(x_n, y_n), n=1,\ldots,N$에서 ($y_n$: 클라스 라벨) 클라스 예측치를 유도하는데 각 n 개의 데이터 크기의 학습데이터 집합 L'을 bootstrap 샘플링하여 일련의 분류기 모델을 생성한 다음, 각 분류기의 분류 결과를 최다 득표 방식으로 통합함으로써 하나의 학습 집합만을 사용하여 생성된 예측치 보다 정확하고 안정적인 분류 클라스 $j\in 1,\ldots,J$를 얻는 것이다. 다음은 Bagging 알고리즘을 요약한 것이다.

입력: training set L, inducer I, integer T
(number of boostrap samples)
for i=1 to T

L'=bootstrap sample from L

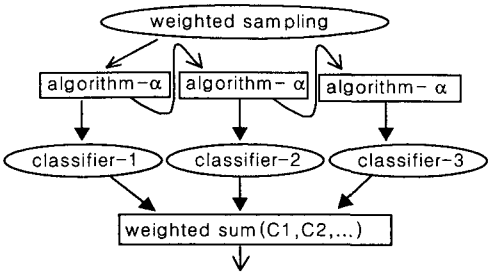
출력: classifier c^*

$$c_i = I(S') \quad (6)$$

최종 결정:

$$c^*(x) = \underset{y\in Y}{\operatorname{argmax}} \sum_{i:\, c_i(x)=y} 1 \quad (7)$$

Boosting은 모델 생성 과정에서 매회의 분류 결과에 따라 오분류된 데이터의 가중치 분포를 달리하면서 훈련 데이터를 추출, 순차적으로 모델을 생성하는 방식이다. 신규 사례는 각 모델의 가중치를 적용한 다수결(weighted sum)에 의해 최종 클라스를 결정한다. 학습 모델 생성 시 무작위로 훈련 예를 선택하는 Bagging과 달리 Boosting알고리즘은 일련의 분류 모델을 생성하는데 전 회차의 분류 오차에 근거하여 훈련데이터의 가중치를 갱신한 다음, 가중치에 비례하여 샘플링 함으로써 훈련예의 분포를 변화해 나가는 적응형(adaptive) 투표 알고리즘이다. 다음 〈그림 3〉은 Boosting,의 개념을 나타낸다.



〈그림 3〉 Boosting의 개념도

분류에 널리 이용되는 AdaBoost.M1 알고리즘의 경우 분류 모델 생성 횟수 (S=1 to T)마다 매회 분류 결과에 대하여 수식(8)과 같이 오차 ε_S를 계산하여 훈련예의 가중치에 수식(8)을 적용한 다음, 수식(10)과 같이 정규화하여 훈련예의 가중치 분포를 갱신한다.

$$E_s = \sum_{i : h_s(x_i) \neq y_i} D_S(i) \tag{8}$$

$$\alpha_S = \varepsilon_S/(1-\varepsilon_t) \tag{9}$$

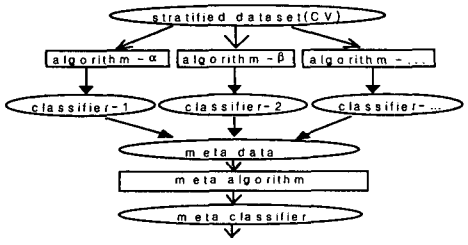
$$D_{s+1}(i) = \frac{D_S(i)}{Z_S} \times \begin{cases} \alpha_S & \text{if } h_S(x_i) = y_i \\ 1 & otherwise \end{cases} \tag{10}$$

매 회차의 분류모델과 오차는 함께 저장되며 이 과정은 오차 ε_S가 0 또는 0.5에 이를 때까지 계속된다. 테스트 데이터 또는 신규 사례의 클라스는 저장된 모든 모델이 분류한 예측치에 각 모델의 오차를 적용, 집계하여 최고 득점한 분류 클래스로 최종 결정된다. 수식(11)의 계산식과 같이 낮은 오차의 모델이 제시한 예측치에 높은 가중치를 적용한다.

$$h_{fin}(d) = \underset{y \in Y}{\operatorname{argmax}} \sum_{S : h_S(d) = y} \log \frac{1}{\alpha_S} \tag{11}$$

3.3 Stacking 알고리즘

Bagging과 Boosting이 단일 학습 알고리즘의 적용 결과를 다수결로 결정하는 투표 알고리즘 방식인데 반하여 Stacking은 다수 알고리즘의 예측 결과를 토대로 상위 단계에서 최종 학습하는 대표적인 메타 학습 알고리즘이다. Stacking은 이질적인 다수 모델을 병합하는 전략으로서 분류알고리즘의 성능을 측정하는데 이용되는 표준 방식인 교차검증(cross validation)보다 유연하고 정교한 기법이다. Stacking은 하위 기반 단계에서 각 데이터에 대해 클라스 확률 분포를 계산하고 이들 예측치를 결합하여 메타 학습기 생성을 위한 메타 데이터로 사용한다. 메타 데이터에 상위 메타 학습 알고리즘을 적용하여 메타 분류기를 유도, 생성한다. Stacking 방식은 신규 사례에 대하여 기반 단계와 상위 메타 단계, 두 단계의 분류기를 거쳐 최종 예측치를 내는데 그 개념과 알고리즘은 다음과 같다.



〈그림 4〉 Stacking의 개념도

데이터 집합 $L=(x_n, y_n), n=1,\ldots,N$을 벡터로 표현한다. 기반 단계 학습에서 데이터를 f개의 동일한 크기의 부분집합으로 분할, $L_1,\ldots,L_f$를 생성하고 L_f를 테스트 데이터로, $L^{(-f)}$를 훈련 데이터로 사용한다. K 개의 기반 학습 알고리즘 가운데 kth 알고리즘으로 훈련예 $L^{(-f)}$을 학습하여 기반 모델 M_k를 생성한다. $(k=1,\ldots,K)$ L_j의 모든 x 에 대하여 j번째 교차검증 결과 x 에 대한 모델의 예측치 $z_{kn}=v_{k(-j)}(x_n)$가 생성된다. 교차검증 회차 마다 K 개 모델의 예측치인 클래스 확률 분포가 생성되는데 각 분류기가 낸 확률 분포를 원래의 클래스 값과 결합하여 메타 학습기 생성을 위한 메타 데이터

$$L_{CV}=$$
$$(y_n, z_{1n},\ldots,z_{Kn}), n=1,\ldots,N$$

로 변환한다.

메타 단계에서는 메타 데이터에 기반 단계 분류 알고리즘를 적용하여 메타 단계의 분류 모델 $\mathcal{M}$을 생성하고,메타 분류기 생성을 마치면 전체데이터 L을 이용하여 새로 기반 모델 M_k를 생성하여 신규 사례를 분류하는데 사용된다. 신규 데이터의 분류에는 모델 M_k와 $\mathcal{M}$을 함께 이용하는데 신규 데이터가 입력되면 모델 M_k는 벡터 $(z_1,\ldots,z_K)$를 생성하고, 이 벡터를 메타 모델에 적용, 최종 클라스를 제시한다

4 하이브리드 다중모델 기법

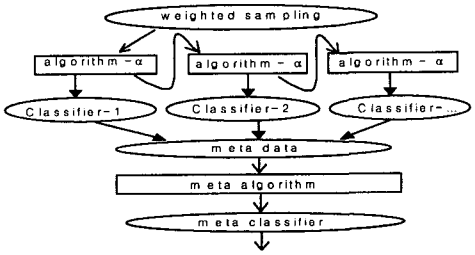
위의 기법과 이론에 기반한 여러 가지 파생적 다중모델 학습 알고리즘이 연구되고 있는데 일반적으로 투표 알고리즘의 파생기법은 훈련예 분포의 변화를 통해 분산효과를, Stacking의 파생 기법은 편향을 해소하려는 알고리즘 의존적 접근법이다. 본 연구에서는 확장형 투표알고리즘으로 Stacked Bagging 및 Stacked Boosting과 확장형 Stacking인 Bagged Stacking 및 Boosted Stacking을 제안하였다.

4.1 확장형 Bagging과 Boosting

확장형 투표 알고리즘은 기반 모델 생성에 투표 알고리즘을 이용하여 생성한 동질적 다중 모델의 통합에 Stacking과 같은 메타 학습을 사용함으로써 결론 제시에 보다 지능적으로 접근하였다. 이 기

법은 훈련예의 분산 효과를 해소함과 동시에 다중모델의 분류 패턴을 학습함으로써 새로운 사례에 대해 보다 정확도가 높은 통합 결론을 낼 것으로 기대하였다.

구현 방식은 Stacked Bagging의 경우 bootstrap 샘플링을 통하여 훈련예 집합의 구성을 달리 하면서 일정 수의 모델을 생성한 다음, 훈련에 사용되지 않은 미지의 문서집합을 적용하여 각 모델의 예측 결과인 클라스 확률 분포를 생성하고, 이들을 통합하여 다수결을 내는 대신 클라스 확률을 연결하여 메타 데이터를 생성하는 것이다. Stacked Boosting 역시 가중치를 적용한 bootstrap 샘플에서 유도된 모델로 미지의 문서 집합을 분류하여 메타 데이터를 확보한다.



〈그림 5〉 Stacked Boosting의 개념도

〈그림 5〉의 개념도와 같이 Stacked Boosting의 경우 기반 단계의 다중 모델의 수는 모델의 전체 오차가 0 또는 0.5에 이를 때까지 생성되므로 가변적이다. 확장형 투표 방식 모두 확보된 메타 데이터에서 역시 동일 알고리즘으로 메타 분류기를 생성한다. 본 연구에서는 동일 알고리즘만을 적용함으로써 알고리즘의 편향 효과에 관계없이 새로운 다중모델 통합 방식의 효과를 측정하고자 하였다. 신규 사례는 기반 단계의 다중 분류기를 거쳐 메타 데이터가 생성되고, 그것이 최종 메타 분류기에 적용된 결과로 클라스가 제시된다. 본 연구에서는 이와 같이 오차 해소의 방법으로 데이터의 구성 변화에만 의존하는 일반 투표 알고리즘에서 일차 분류 결과를 재학습하여 각 모델의 분류 패턴을 지식으로 학습, 최종 적용하는, 새로운 결론 통합 방식이 전체 분류 성능을 제고할 것으로 기대하였다.

4.2 확장형 Stacking

본 연구에서 제안한 Stacking의 확장형인 Bagged Stacking과 Boosted Stacking 역시 Bagging과 Boosting 방식으로 모델을 생성하되 다수 알고리즘 분류기의 예측 결과를 학습하고 메타 분류기를 생성하여 최종 클라스를 결정하는 것이다. 이 방식은 이질적인 학습알고리즘의 편향을 상쇄하는 동시에 훈련예의 변화에 따른 분산의 교정 효과가 있을 것으로 기대하였다.

Bagged Stacking은 적용하고자 하는 학습 알고리즘 마다 일정 개수의 다중 모델을 생성하여 미지의 사례를 분류하고, 모델 마다 제시한 클라스 확률을 합계, 저장한 다음, 각 사례의 클라스 확률을 연결하여 메타 데이터를 생성하고 여기서 메타 분류기를 유도한다. Boosted Stacking 역시 각 알고리즘 별로 모델의 오차에 따라

〈표 1〉 Boosted Stacking 알고리즘

Model Generation :

For each of k algorithms:
  Assign equal weight to each training instance
  For each of t iterations
    Apply the learning algorithm to weighted dataset and store resulting model
    Compute error *e* of model on weighted dataset and store error
    If *e* equals to zero, or e greater or equal to 0.5
      Terminate model generation
    For each instance in dataset
      If instance classified correctly by model
        Multiply weight of instance by e/(1-e)
        Normalize weight of all instances
  For each of t (or less) models
      Assign weight of zero to all classes
      Add -log(*e*/(1-*e*)) to weight of class predicted by model
      Sum and store class probability distribution

Meta Data Generation :

For each instance in dataset
    Generate meta data by combining class probability distribution

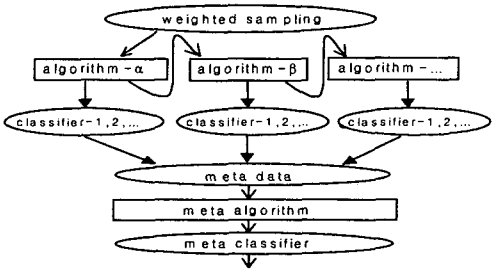
Meta-Level Model Generation :

Apply learning algorithm to meta data

Classification :

Apply a new instance to base-level models and the meta-level model

가변적으로 생성되는 이질적 기반 모델과 오차율을 저장하고, 미지의 사례를 적용하여 메타 데이터를 확보한 다음, 이 메타 데이터에서 메타 분류기를 유도한다. 신규 사례는 각 알고리즘 별 다중 모델의 예측을 거쳐 최종적으로 메타 분류기에 적용된 결과로 최종 클라스가 제시된다. 본 연구에서 제안한 Boosted Stacking의 개념과 알고리즘은 다음과 같다.



〈그림 6〉 Boosted Stacking

5 실 험

5.1 실험 환경

문서 분류 실험을 통하여 단일모델과 다중모델, 하이브리드 다중 모델기법의 분류기 성능을 비교하고자 하였다. 특징 선택은 문서 분류에 성능이 좋은 것으로 연구된 정보 획득(IG) 방식을 채택하고 특징 집합의 크기에 따른 성능 차이를 검토하고자 하였다(Yang 1997). 문서 모델은 이진벡터와 가중치 벡터 두 가지로 표현하여 특징 값 표현 방식의 효과를 분석하고자 하였다. 학습 알고리즘은 단일 모델에 결정트리(C4.5), Nave Bayesian, k-NN을, 투표 알고리즘에 C4.5를, Stacking에는 기반 학습에 C4.5와 Naive Bayesian을 공통으로 사용하고, 메타 학습에 Decision Stump, C4.5, Naive Bayesian을 적용하였다. 실험은 Linux 운영체제를 갖춘 Intel Pentium IV processor, 256MB 하드웨어 환경에서 수행되었으며 텍스트 처리도구는 Rainbow와 WEKA를 이용하였다. 프로그램 언어는 JAVA, C를 사용하였다.

실험데이터는 MEDLINE 학술 기사, 텍스트 분류용으로 공개된 유즈넷 뉴스와 웹문서 집합을 사용하였다. MEDLINE 학술데이터베이스의 문서집합으로서 정보검색 연구의 표준데이터로 이용되는 OSHUMED는 주제 분류항목이 결여되어 있어 자동 분류의 훈련데이터로 적합치 않았으므로 MeSH(Medical Subject Heading) 주제어 가운데 'Mental Disorder'의 하위 토픽을 데이터베이스에 질의하여 검색된 결과로 OSHUMED와 동일한 형식의 훈련예를 확보하였다. 〈표 2〉는 실험에 사용한 훈련데이터의 내역이다.

5.2 실험 결과 및 평가

기계학습 분야에서 일반적으로 사용되는 중요한 성능 평가 방법은 분류기 유도를 위한 훈련예로 사용한 적이 없으며 클

〈표 2〉 훈련문서 집합

	MEDLINE 문서집합	유즈넷 뉴스	웹 문서집합
클라스 개수	6*	5**	6***
총 문서 수	6000	5000	4518
클라스 당 훈련예	1000	1000	137~1741
전체 훈련예	0.66 * 6000	0.66 * 5000	0.66 * 4518
테스트 문서 수	0.34 * 6000	0.34 * 5000	0.34 * 4518

*{alzheimer, anxiety, dementia, depression, memory, schizophrenia}
**{comp.graphics,comp.os.ms-windows.misc,comp.sys.ibm.pc.hardware,comp.sys.mac.hardware,comp.windows.x}
***{student, faculty,staff, department,course,project}

〈표 3〉 문서 집합별 분류 정확도

단위: %

종 류	학습기법	50-bin	50-wt	100-bin	100-wt
단일모델	k-NN	74.30	75.80	74.62	75.21
	C4.5	77.03	77.55	77.88	79.05
	NB	76.97	89.88	81.33	69.49
다중모델(투표)	BagC4.5	79.90	79.77	82.56	83.41
	BoostC4.5	77.29	80.09	81.78	83.80
다중모델(Stacking)	StackDS	52.37	51.01	53.03	51.59
	StackC4.5	78.59	77.90	82.50	79.12
	StackNB	79.38	78.66	82.30	75.63
하이브리드다중(투표)	StackedBag	78.80	78.71	78.51	78.71
	StackedBoost	78.49	77.78	77.82	78.38
하이브리드 다중 (Stacking)	BaggedStack	80.15	77.29	83.98	80.06
	BoostedStack	80.61	78.64	82.63	80.13

(3-a) MEDLINE 문서 집합

종 류	학습기법	50-bin	50-wt	100-bin	100-wt
단일모델	k-NN	96.94	91.94	97.47	89.47
	C4.5	97.29.	96.59	96.88	86.70
	NB	97.65	66.65	97.06	70.00
다중모델(투표)	BagC4.5	97.35	97.18	97.48	97.00
	BoostC4.5	97.41	97.18	97.76	97.41
다중모델(Stacking)	StackDS	38.94	38.88	39.00	38.88
	StackC4.5	97.65	96.65	97.52	96.71
	StackNB	97.82	96.71	98.00	96.79
하이브리드다중(투표)	StackedBag	97.38	96.94	97.38	97.00
	StackedBoost	97.42	97.32	97.42	96.92
하이브리드다중 (Stacking)	BaggedStack	97.40	97.08	97.50	97.50
	BoostedStack	97.48	97.26	97.40	97.08

(3-b) 유즈넷뉴스 문서 집합

라스 라벨을 부여한 사례에 대해 분류 성과를 측정하는 것이다. 미분류 사례공간 V, 클라스 라벨 집합 Y, $X = V \times Y$는 클라스 라벨을 부여한 훈련예 공간일 때 사례 $\langle x, y \rangle \in X$ 대하여 분류한 결과의 정확도이다. 수식 (12)와 같이 정확도를

종 류	학습기법	50-bin	50-wt	100-bin	100-wt
단일모델	k-NN	74.30	75.80	74.62	75.21
	C4.5	77.03	77.55	77.88	79.05
	NB	76.97	69.88	81.33	69.49
다중모델(투표)	BagC4.5	79.90	79.77	82.56	83.41
	BoostC4.5	77.29	80.09	81.78	83.80
다중모델 (Stacking)	StackDS	52.37	51.01	53.03	51.59
	StackC4.5	78.59	77.90	82.50	79.12
	StackNB	79.38	78.66	82.30	75.63
하이브리드다중 (투표)	StackedBag	78.80	78.71	78.51	78.71
	StackedBoost	78.49	77.78	77.82	78.38
하이브리드 다중(Stacking)	BaggedStack	80.15	77.29	83.98	80.06
	BoostedStack	80.61	78.84	82.63	80.13

(3-c) 웹 문서 집합

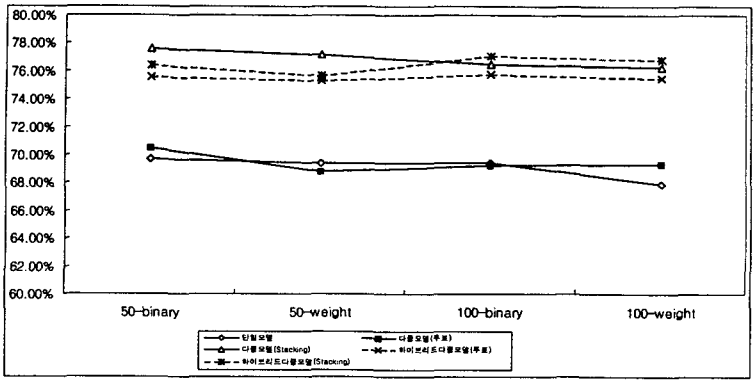
평가한다.

$$acc = \Pr(C(v) = y) \qquad (12)$$

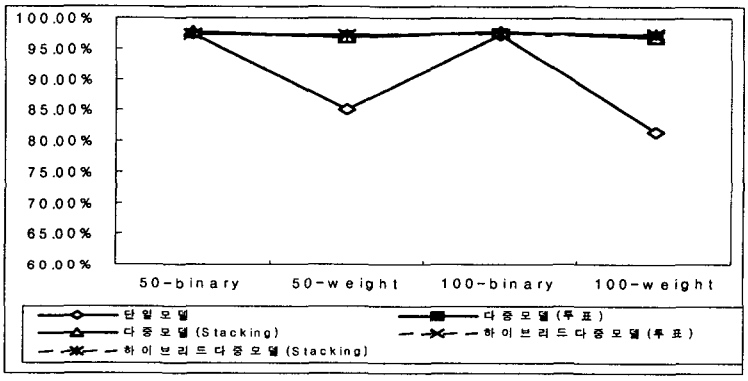
이 정확도는 훈련예가 속한 모집단 전체에 대하여 분류기를 적용했을 때, 즉 일반화했을 때 기대되는 성능으로 예측 정확도(predictive accuracy)인 것이다. 모델 생성 과정은 주어진 것 중 가능한 최상의 모델 찾기로서 오차율, 즉, 사례 공간 X 과 제시된 예측치 간에 정의된 손실 함수를 최적화하면 최상의 모델이다.

일반적으로 분류 알고리즘의 성능은 분류 모델의 교차 검증을 통하여 측정해 왔으나 본 실험에서는 단일 모델 분류기를 성능 측정의 최저기준으로 하여 여러 종류의 다중 모델 분류기를 통한 분류 성능을 실험하였다. 분류기 종류 별로 각 종류 문서 집합에 대해 10등분 교차 검증을 거친 분류 정확도는 표 3과 같다.

본 실험에서는 문서 모델의 이진 표현법이 전반적으로 가중치 표현법보다 우수한 성능을 보였다. 문서 특징 부분집합 50 내지 250개 사이에서 특징 집합의 크기에 따른 성능 차이는 미미하였다. 문서 집합 별 분류 정확도는 큰 편차를 보였는데 정확도가 높은 유즈넷 뉴스의 경우 주제 관련성이 높은 컴퓨터 전문용어가 많이 분포된 것이 원인으로 판단된다. 성능이 낮은 Decision Stump(StackDC)를 메타 학습기로 사용한 경우 정확도가 비현실적으로 낮은데 메타 학습기도 우수한 알고리즘을 이용하는 것이 분류효과에 유리함을 보여준다. 위와 같은 실험결과를 종류별로 평균하여(StackDS 제외) 정확



(7-a) MEDLINE 문서 집합



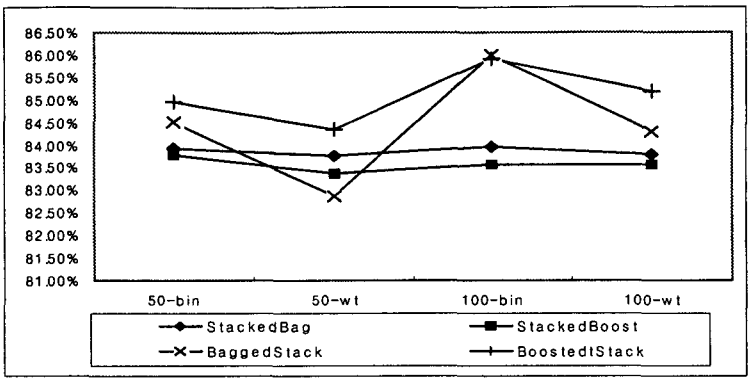
(7-b) 유즈넷뉴스 문서 집합

〈그림 7〉 전체 분류기의 분류 정확도 비교

도를 그라프로 나타내면 〈그림 7〉과 같다.

실험 결과 대부분 문서집단에서 하이브리드 다중학습 모델의 분류 성능이 단일 모델과 기존의 다중 모델을 능가하였다. 전체적으로 단일 모델 78.36%, 다중 모델 83.55%, 하이브리드 다중 모델이 84.30%의 평균정확도를 보였으며 이러한 결과는 훈련예의 분산과 학습알고리즘의 편향 효과를 동시에 해소 내지 완화가 가능함을 보여준다.

특히 메타학습 기법의 다중 모델 통합 효과가 두드러졌는데 동질 모델과 이질 모델 모두의 통합효과가 높은 것으로 나타났다. 확장형 투표알고리즘과 같은 동질모델의 성능 향상은 통합방법의 효율성이 그 원인일 것으로 해석되며, 모델 통합 방식에서 투표 방법 보다는 각 모델의 분류 패턴을 학습하는 지능적 메타 학습이 우수한 것으로 보인다. 하이브리드 기법 가운데서는 〈그림 8〉에서 나타났듯이



〈그림 8〉 하이브리드 다중 모델 분류기의 평균 정확도

Stacking 종류의 이질 모델이 우수한 결과를 보였는데 이는 학습 알고리즘의 편향이 학습 데이터의 분산보다 오차율에 미치는 영향이 크기 때문인 것으로 판단된다. 또한 하이브리드 분류기의 결과는 매우 낮은 편차를 보이는데(단일 학습기 0.066, 다중 학습기 0.023, 하이브리드 0.01), 이는 하이브리드 분류기의 안정성을 보여준다.

6 결 론

본 논문에서는 기계학습법을 이용한 자동 분류 기법 가운데 다중 모델 기계 학습법의 개념과 특징을 고찰하였다. 나아가 분류 오차의 다양한 원인을 동시에 교정하고자 하는 하이브리드 다중 모델 학습법을 제안하고 텍스트 자동 분류에 적용, 실험하였다. 실험 결과 전반적으로 메타 학습기법을 이용한 하이브리드 다중모델기법이 기존의 다중 학습기법들 보다 높은 성능을 보임으로써 문서 분류 도메인에서 그 효과를 입증하였다. 하이브리드 다중모델 학습법은 분류 효율성 뿐만 아니라 그 안정성 면에서도 매우 신뢰할 만한 것으로 나타났다.

기계학습법 가운데서도 다중 모델 기법을 문서 분류에 적용한 본 연구 결과는 이들 기법이 학습 알고리즘이나 문제 도메인 독립적으로 사용될 수 있는 가능성을 제시하였다. 본 연구의 결과 다중 모델 학습법은 모델의 성능을 최적화할 수 있는 일반적인 기법으로 다양한 응용분야에 적용되어 분류 효율을 향상시킬 수 있을 것으로 보인다. 특히 본 연구에서 제안하고 실험한 하이브리드 다중 모델 학습은 일반 다중 모델 학습에 비해 높은 계산 비용을 요하지만 성능의 향상과 신뢰도의 확보가 중요시되는 응용분야에서는 그 비용이 정당화될 것으로 판단된다.

참 고 문 헌

Bauer, Eric, Kohavi, Ron 1999. "An Empirical comparison of voting classification algorithms: bagging, boosting, and variants." *Machine Learning*, 36: 105-142.

Breiman, Leo. 1996. "Bagging predictors." *Machine Learning*, 24: 49-64.

Chen, H., 1995. "Machine learning for information retrieval: neural networks, symbolic learning and genetic algorithms." *JASIS* , 46: 194-216.

Domingos, P., Pazzani, M. 1997. "On the optimality of the simple Bayesian classifier under zero one loss." *Machine Learning*, 29: 103-130.

Han, Jiawei, Micheline Kamber. 2001. *Data Mining: Concepts and Techniques*. New York: Morgan Kaufmann.

Hong, Se June, Weiss, Sholom M. 1999. "*Advances in Predictive Model Generation for Data Mining.*" IBM Research Report RC-21570. 〈http://citeseer.nj.nec.com〉.

Lewis, D. D., Ringuette, M. 1994. "A comparison of two learning algorithms in text categorization." In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, 1994.* 81-93.

Mitchell, Tom. 1997. *Machine Learning.* New York: McGraw-Hill.

Mladeni'c, D., Grobelnik, M. 1998. "Efficient text categorization. In *Text Mining workshop on the 10th European Conference on Machine Learning ECML98.*

Salton, Gerard. 1983. Introduction to Information Retrieval. New York: McGraw-Hill.

Schaphire, Robert E, 1999. "Theoretical views of boosting." *In Computational Learning Theory: 4th European Conference, EuroCOLT '99.*

Shankar, Shrikanth, Karypis, George, 2000. "A Feature weight adjustment algorithm for document categorization." In *SIGKDD'00 Workshop on Text Mining, Boston, MA, 2000.*

Witten, Ian H., Frank, Eibe. 2000 *Data Mining: Practical Machine*

Learning Tools and Techniques with Java Implementations. New York: Morgan Kaufman.

Wolpert, David H. 1992. "Stacked generalization." *Neural Networks*, 5: 241-259.

Yang, Yiming, Pedersen, Jan O. 1997. "A comparative study on feature selection in text categorization." In *Proceedings of the Fourteenth International Conference on Machine Learning.*