

# 다중 분류기 시스템을 이용한 자동 문서 분류

김 인 철\*

요 약

단일 분류기에 비해 높은 분류성능을 얻기 위해 다수의 분류기들을 결합하여 사용하는 방법은 폭넓게 이용되어 온 기술이다. 하나의 다중 분류기 시스템을 구성하는 일은 다음 두 가지 문제들을 가지고 있다. 첫째는 어떻게 기반 분류기들을 생성하느냐 하는 것이고 둘째는 이들의 예측 결과를 어떻게 결합하느냐 하는 것이다. 본 논문에서는 Bagging, Boosting, Stacking 등 기존의 대표적인 다중 분류기 시스템들의 특징을 살펴보고, 문서 분류를 위한 새로운 다중 분류기 시스템들인 Stacked Bagging, Stacked Boosting, Bagged Stacking, Boosted Stacking들을 제안한다. 이들은 Bagging, Boosting, Stacking과 같은 기존 다중 분류기 시스템들의 장점들을 결합한 일종의 혼합형 다중 분류기 시스템들이다. 본 논문에서는 제안된 다중 분류기 시스템들의 성능을 평가하기 위해 MEDLINE, 유즈넷 뉴스, 웹 문서 등의 문서집합을 이용한 문서 분류 실험들을 전개하였다. 그리고 이러한 실험결과를 통해 제안한 혼합형 다중 분류기 시스템들은 전반적으로 기존 시스템들보다 우수한 성능을 보이는 것으로 나타났다.

## Automatic Document Classification Using Multiple Classifier Systems

In-Cheol Kim\*

ABSTRACT

Combining multiple classifiers to obtain improved performance over the individual classifier has been a widely used technique. The task of constructing a multiple classifier system(MCS) contains two different issues : how to generate a diverse set of base-level classifiers and how to combine their predictions. In this paper, we review the characteristics of existing multiple classifier systems : *Bagging*, *Boosting*, and *Stacking*. For document classification, we propose new MCSs such as Stacked Bagging, Stacked Boosting, Bagged Stacking, Boosted Stacking. These MCSs are a sort of hybrid MCSs that combine advantages of existing MCSs such as *Bagging*, *Boosting*, and *Stacking*. We conducted some experiments of document classification to evaluate the performances of the proposed schemes on MEDLINE, Usenet news, and Web document collections. The result of experiments demonstrate the superiority of our hybrid MCSs over the existing ones.

**키워드 :** 문서 분류(Document Classification), 다중 분류기 시스템(Multiple Classifier System), 메타 분류기(Meta Classifier)

### 1. 서 론

웹문서, 디지털 라이브러리 등, 전자문서의 증가로 인하여 정보 요구를 충족하기 위한 정보접근(information access)의 문제점이 가중되고 있다[1]. 이에 따라 문서의 자동 분류(classification), 군집화(clustering), 여과(filtering) 등과 같이 비정형 텍스트 데이터 조직을 통해 정보 접근 문제를 해결하려는 노력이 정보검색, 인공지능 분야에서 다양하게 수행되고 있다[2]. 본 논문에서는 다중 분류기 시스템(multiple classifier system)을 이용하여 문서 분류의 성능과 신뢰도를 향상시킬 수 있는 연구와 실험 결과를 기술하였다.

훈련 데이터 집합을 바탕으로 데이터의 구조적 패턴을 명시적 지식으로 학습한 뒤, 신규 데이터에 대하여 분류 클래스를 제시하는 귀납적 교사학습(inductive supervised learning) 기법은 문서 분류에도 높은 정확도를 보인다[3, 4]. 일반적으로 분류기(classifier) 또는 학습된 모델(learned model)은 훈련 데이터에 특정 학습 알고리즘(learning algorithm)을 적용한 결과로 유도된 구조적 패턴으로서, 임의의 미분류 데이터에 하나의 클래스를 매핑하는 함수기능을 갖는다. 다중 분류기 시스템은 분류기들의 앙상블(Ensemble of Classifiers)이라고도 불리는데, 훈련 데이터 집합과 학습 알고리즘으로부터 다수의 분류기들을 유도해내고, 이들의 예측을 결합하여 최종 분류 클래스를 결정하는 모든 방식을 말한다. 상호보완적인 다수의 분류기들을 이용하는 이

\* 종신회원 : 경기대학교 정보과학부 교수  
논문접수 : 2004년 5월 15일, 심사완료 : 2004년 7월 23일

들 다중 분류기 시스템들은 단일 분류기들에 비해 부가적인 비용을 필요로 하기는 하나 높은 분류 성능과 신뢰도를 얻을 수 있는 것으로 알려져 있다. 다중 분류기 시스템은 Breiman의 Bagging[5]을 비롯하여, Boosting[6], Stacking [7] 등 대표적인 다중 분류기 시스템들 외에 현재 다양한 파생 기법들이 연구되고 있다[8].

본 연구에서는 다중 분류기 시스템의 일반적인 개념을 살펴본 뒤, 분류기들의 생성방식과 이들을 결합하는 방식을 중심으로 기존 다중 분류기 시스템들인 Bagging, Boosting, Stacking 방식의 특징과 장점을 분석해본다. 또 본 연구에서는 기존의 순수 Bagging, Boosting, Stacking 방식에서는 포함하고 있지 않은 분류기 생성과정의 편향성(bias)이나 분류기 결합방식을 도입하여 새로운 다중 분류기 시스템들인 Stacked Bagging과 Stacked Boosting, 그리고 Bagged Stacking과 Boosted Stacking을 제안한다. 그리고 이러한 새로운 다중 분류기 시스템들을 문서 분류에 적용하여 문서 분류의 정확도를 높여 보려고 한다. 본 연구에서는 MEDLINE 학술 기사, 유즈넷 뉴스(Usenet news) 기사, 그리고 웹 문서 등의 문서집합을 이용한 문서 분류 실험을 통하여 기존 다중 분류기 시스템들과 본 연구에서 제안하는 새로운 다중 분류기 시스템들의 분류 성능을 비교해본다. 또 이 밖에도 실험을 통해 메타 학습 알고리즘과 문서 모델이 다중 분류기 시스템들의 분류 성능에 어떤 영향을 주는지도 함께 분석해본다.

## 2. 문서 분류

문서분류(document classification) 혹은 텍스트범주화(text categorization)는 미리 정의해둔 일정한 수의 클래스(class) 혹은 범주(category)들에 문서들을 분류하는 작업을 말한다. 그리고 특히 본 연구에서는 각 문서  $d$ 는 오직 하나의 클래스에만 할당되는 것으로 가정한다. 따라서 일반적인 학습기반의 문서분류작업은 다음과 같이 정의 할 수 있다. 먼저 클래스들의 집합  $C$ 와 문서들의 집합  $D$ 에 대해, 임의의 문서를 하나의 클래스에 대응시키는 하나의 목적함수(target function)  $T: D \rightarrow C$ 가 존재한다고 가정한다. 그리고 훈련문서집합  $D' \subset D$ 에 속한 모든 문서  $d$ 들에 대해, 이들이 속한 클래스인  $T(d)$ 는 이미 알고 있는 것으로 가정한다. 이때 우리는 교사학습(supervised learning)을 통해, 훈련집합에 포함된 정보로부터 목적함수  $T$ 에 근사한 하나의 모델(model) 혹은 가설함수(hypothesis function)  $H: D \rightarrow C$ 를 찾는다. 그리고 새로운 문서  $d'$ 에 대해, 이러한 근사함수  $H$ 를 적용함으로써 새로운 문서  $d'$ 이 속한 클래스  $H(d')$ 를 구할 수 있다.

### 2.1 문서 모델

하나의 긴 문자열(character string)로 볼 수 있는 문서들에 대해 분류작업을 수행하기 위해서는 먼저 하나의 문서를 학습 알고리즘과 분류작업에 적합한 형태로 변환하여야 한다. 그 동안의 정보검색 연구를 통해, 단어(word)들을 문서 표현의 기본단위로 사용하는 것이 좋은 결과를 내며, 한 문서내에서 그들의 순서는 대부분의 분류작업에 그다지 큰 영향을 미치지 않는다는 사실을 알게 되었다. 따라서 문서 분류작업을 위해서는 대부분 문서를 단순히 단어들의 집합(bag of words)으로만 표현한다.

이와같이 문서를 단어들의 집합으로 표현하는 것은 기계 학습(machine learning)에 사용되는 보편적인 훈련 데이터 형태인 속성과 속성값(attribute-value)의 벡터로 표현하는 것과 동일하다. 즉 문서를 표현하는 각 단어는 하나의 속성에 대응되며, 이때 그 단어가 해당 문서내에 등장하는 빈도수(frequency)를 속성값으로 가지게 된다. 일반적으로 문서 집합에 등장하는 모든 단어들을 문서 표현을 위한 속성들로 사용하기 어려우므로는 주로 문서 내용을 대표할 수 있는 주요 키워드들만을 선별하여 특징어(feature)로 사용한다. 하나의 문서를 특징벡터로 표현하는 대표적인 방법으로는 다음과 같이 각 단어의 등장 유무만을 이진값으로 나타내는 이진벡터(binary vector) 표현법과 단어의 실제적인 빈도수를 값으로 갖는 가중치 벡터(weight vector) 표현법, 그리고 단어의 빈도수와 역문서빈도수에 기초한 TFIDF 표현법 등이 있다[9].

특징어 선택(feature selection)은 각 문서를 표현하는 특징벡터의 차수(dimentionality)를 줄임으로써 분류작업의 효율성을 높이기 위한 처리과정을 말한다. 특징어 선택 방법으로는 일정한 척도(measure)를 기준으로 각 단어의 중요도(importance)나 관련성(relevance)을 개별적으로 평가하여 필요한 수만큼의 특징어를 선택하는 여과방법(filtering)과 내포된 특징 분류기의 성능을 높일 수 있는 단어들의 부분집합을 점진적으로 구해가는 포장방법(wrapper) 등이 있다[10]. 일반적으로 문서분류를 포함한 정보검색분야에서는 여과 방식이 많이 사용되며, 주요 여과 방법으로는 TFIDF, 정보 획득(information gain),  $\chi^2$  테스트, LSI(Latent Semantic Indexing) 등이 있다[11]. 특히 이 가운데 정보 획득(IG)은 식 (1)과 같이 계산된다.

$$\begin{aligned} \text{InforGain}(w) = & P(w) \sum_i P(ci | w) \log \frac{P(ci | w)}{P(ci)} \\ & + P(\bar{w}) \sum_i P(\bar{ci} | \bar{w}) \log \frac{P(\bar{ci} | \bar{w})}{P(\bar{ci})} \end{aligned} \quad (1)$$

### 2.2 문서 분류기

훈련문서집합에 특정 기계학습(machine learning) 방법을

적용함으로써 얻어지는 문서분류를 위한 일반 모델을 우리는 문서 분류기(document classifier)라 부른다. 이러한 문서 분류기는 다양한 학습방법을 통해 얻어 질 수 있는데, 문서 분류에 많이 이용되는 기존의 대표적인 학습방법으로는 Naive Bayesian 학습법, k-NN(k Nearest Neighbor) 학습법, 결정 트리(Decision Tree) 학습법 등을 들 수 있다. 각 학습방법은 자신만의 고유한 분류기 표현법과 학습 알고리즘을 가지고 있다. 예컨대 결정트리 학습법은 이름 그대로 분류기가 하나의 결정트리로 표현되며, 이러한 결정트리를 생성하는 학습 알고리즘으로는 ID3, C4.5 등이 있다.

### 2.2.1 Naive Bayesian 학습법

Naive Bayesian 학습법은 확률에 기반한 문서 분류방법을 제공하며, 비교적 문서의 양이 많은 경우에도 높은 분류 성능을 보이고, 데이터의 잡음(noise)에도 강한 장점이 있다[12]. Naive Bayesian 분류 방식은 식 (2)과 같이 분류 대상 문서  $d$ 가 각 클래스  $c_i$ 에 속할 조건부 확률  $P(c_i|d)$ 을 계산하고, 이를 바탕으로 가장 높은 확률값을 갖는 클래스  $c_{MAP}$ 로 분류한다.

$$c_{MAP} = \arg \max_{c_i \in C} P(c_i|d) \quad (2)$$

그리고 이때 조건부 확률  $P(c_i|d)$ 는 Bayes 정리에 기초하여 식 (3)와 같이 계산한다.

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

특히 Naive Bayesian 분류학습법은 문서를 표현하는 각 특징어들간에 조건부 독립성(conditional independence)이 있다고 가정함으로써 분류를 위한 확률계산을 용이하게 한다. 비록 이러한 가정은 실제 문서에서는 만족하기 어렵지만, 그럼에도 불구하고 이러한 가정에 기초를 둔 Naive Bayesian 분류기는 문서분류에 높은 성능을 보여주는 것으로 알려져 있다[13].

### 2.2.2 k-NN(Nearest Neighbor) 학습법

대표적인 개체 기반 학습법(Instance based learning)인 k-NN 학습법은 훈련문서집합으로부터 미리 일반화된 모델을 학습하지 않고 단지 훈련집합을 저장해두기만 하였다가 분류 대상 문서가 주어지면 이 문서와 가장 가까운 k개의 이웃 훈련문서들의 클래스를 참고하여 문서를 분류한다[14]. 두 문서  $X$ 와  $Y$ 간의 거리(distance)는 식 (4)과 같이 계산한다.

$$X = (w_{x1}, w_{x2}, \dots, w_{xn})$$

$$dist(X, Y) = \sqrt{(w_{x1} - w_{y1})^2 + \dots + (w_{xn} - w_{yn})^2} \quad (4)$$

k-NN 학습법은 매번 새로운 문서를 분류할 때마다 그 문서와 각 훈련문서들과의 거리를 모두 계산하여야 하기 때문에, 일반적으로 많은 수의 특징어들로 각 문서들이 표현되는 경우에 분류에 많은 시간이 소요된다.

### 2.2.3 결정트리(Decision Tree) 학습법

결정트리 학습법은 훈련집합으로부터 트리 모양의 분류모델을 생성해낸다. 이러한 결정트리는 문서분류를 위한 지식을 표현하는 것으로서, 각 비 단말 노드(non-terminal node)는 하나의 특징어(feature)를 나타내고 이 노드에서 분기하는 각 가지는 하나의 특징 값(value)을 표시한다. 그리고 각 단말 노드(terminal node)는 하나의 분류 클래스를 나타낸다. 결정트리 생성과정에 가장 중요한 선택은 각 노드를 위한 특징어를 결정하는 일이며, 이것을 위해 훈련문서집합으로부터 각 특징어의 정보 획득량을 계산한다[14]. 결정트리 학습을 위한 알고리즘으로는 ID3와 이것을 개선시킨 C4.5 등이 있다. 복잡하고 많은 수의 문서집합에 대해 결정트리 학습법을 적용할 경우, 자칫 결정트리의 크기가 매우 커질 가능성이 있으며, 훈련집합에 포함될 수 있는 잡음에 매우 민감한 약점을 가지고 있다.

## 3. 다중 분류기 시스템

### 3.1 분류기들의 앙상블

최근들어 훈련집합으로부터 다수의 분류기들을 학습해내고 이들의 예측을 결합함으로써 단일 분류기보다 높은 분류 성능을 얻으려는 다중 분류기 시스템(MCS, Multiple Classifier System)에 관한 연구가 활발하다. 다중 분류기 시스템은 분류기들의 앙상블(Ensemble of Classifiers)이라고도 불리는데, 이 방법은 단일 분류기에 의존하는 방법에 비해서 서로 다른 다수의 분류기들의 예측결과를 상호보완적으로 이용할 수 있다는 점에서 더 높은 예측 정확도(accuracy)와 신뢰도(reliability)를 얻을 수 있다. 하나의 다중 분류기 시스템을 설계하기 위해서는 다음과 같은 두 가지 문제들을 해결해야 한다. 첫째는 어떻게 서로 다른 개별 분류기들을 생성할 것이냐는 것이며, 둘째는 이 분류기들을 어떤 방식으로 결합할 것이냐는 문제이다.

서로 다른 개별 분류기를 생성할 수 있는 방법은 몇 가지가 있다. 그 중 한 가지는 동일한 하나의 데이터집합에 서로 다른 학습 알고리즘들을 적용함으로써 신경망과 결정트리와 같이 서로 다른 표현의 분류기들을 유도하는 방법이다. 이 방법은 각기 다른 학습 알고리즘들이 갖는 분류기 표현법과 탐색 방법상의 고유한 편향성(bias)을 이용하여 서로 다른 분류기들을 만드는 방법으로 볼 수 있다. 다른 한 가지 방법은 동일한 학습 알고리즘을 서로 다른 훈련 데이터 집합에 적용함으로써 서로 다른 분류기들을 생성해내는 방법

이다. 이때 서로 다른 훈련 데이터 집합은 데이터들의 전체 집합에 대한 표본 추출(sampling)과정을 통해 얻을 수 있다. 분류기 생성에 이용할 수 있는 대표적인 표본 추출방법들로는 복위없는 임의 표본 추출법(random sampling without replacement), 복위있는 임의 표본 추출법(random sampling with replacement) - 부트스트랩 표본 추출법(bootstrap sampling)이라고도 불림 -, 가중치가 적용된 복위 임의 표본 추출법(weighted random sampling with replacement) 등이 있다[8]. 이 방법은 전자와는 달리 훈련 데이터 집합상의 편향성을 이용하여 서로 다른 분류기들을 만드는 방법으로 볼 수 있다. 서로 다른 분류기들을 생성할 수 있는 또 다른 방법으로는 동일한 훈련 데이터 집합에 동일한 학습 알고리즘을 반복 적용하되, 매번 학습 인자들(learning parameters)의 값을 바꾸어 주는 것이다. 이 방법은 훈련 데이터 집합과 학습 알고리즘은 동일하지만 학습을 위한 탐색 시작점(starting point)이나 탐색 제어(control)에 변화를 줌으로써 서로 다른 분류기들을 만드는 방법으로 볼 수 있다.

앞서 언급한 바와 같이 다중 분류기 시스템 설계에 중요한 두 번째 문제는 분류기들을 결합하는 방식에 관한 것이다. 대표적인 결합방법들로는 분류기들중에 하나만을 골라 그 분류기의 예측결과를 따르는 최고 분류기(best classifier)방법, 모든 분류기들의 예측결과를 동등하게 취합하는 다수결 투표(majority voting) 방법, 분류기들의 예측결과를 분류기 자신의 예상 분류성능에 따라 차등적으로 취합하는 가중치 투표(weighted voting)방법, 분류기들의 분류패턴을 학습하는 메타 학습기(meta learner)를 이용하는 방법 등이 있다[15].

(그림 1) Bagging

### 3.2 Bagging과 Boosting

서로 다른 훈련 데이터 집합들로부터 다수의 분류기들을 생성하는 대표적인 다중 분류기 시스템들로는 Bagging과 Boosting이 잘 알려져 있다. 기본적으로 Bagging과 Boosting은 복위있는 임의 표본 추출법(random sampling with replace-

ment), 즉 부트스트랩 표본 추출법(bootstrap sampling)을 사용하여 서로 다른 훈련 데이터 집합들을 만들어 낸다. 부트스트랩 표본 추출법은 아래와 같이 정의할 수 있다. 훈련 데이터 집합  $L$ 이 다음과 같이 주어졌을때,  $L = (x_n, y_n)$ ,  $n = 1, \dots, N'$ , 여기서  $x_n$ 은  $n$ 번째 데이터를 나타내는 속성 값 벡터,  $y_n$ 은 그 데이터의 클래스값(class value)을 표시한다. 부트스트랩 표본 추출은 집합  $L$ 로부터  $N'$ 보다 작은  $N$ 의 크기를 갖는  $K$ 개의 부분집합들을 복위(replacement)를 허용해서 임의 표본 추출하는 것을 말한다. (그림 1)은 Bagging 방법을 도식화한 것이다. Bagging은 부트스트랩 표본 추출법을 통해 얻어진 서로 다른 훈련 데이터 집합들에 동일한 학습 알고리즘을 적용하여 다수의 분류기들을 생성한 다음, 새로운 데이터에 대한 분류 작업시 다수결 투표(majority voting)방식으로 이들 분류기들의 예측결과들을 취합하여 최종 결론을 내린다.

(그림 2) Boosting

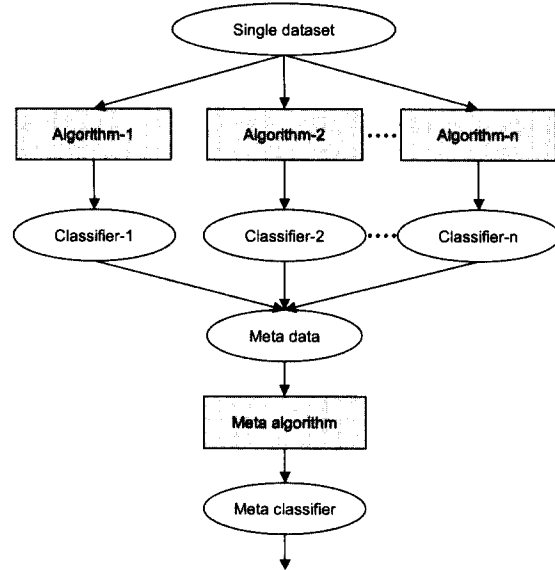
이에 반해, Boosting은 좀더 복잡한 방식으로 다중 분류기 시스템을 만든다. (그림 2)와 같이 Boosting 역시 부트스트랩 표본 추출방법을 사용하나 선택 가능한 각 데이터들이 모두 같은 확률을 가지는 것이 아니라 서로 다른 가중치(weight)를 가지도록 설계되었다. 또 이와 같은 방법으로 생성된 분류기들의 예측결과를 결합하는 방법도 Bagging과는 달리 각 분류기마다 서로 다른 가중치를 부여하고 그 가중치에 따라 분류기의 예측결과를 차등 반영하는 가중치 투표방식(weighted voting)을 사용한다. 즉 Boosting은 훈련 데이터 집합의 선택과정뿐만 아니라 분류기들의 예측결과를 결합하는 과정에도 각각 차등화된 가중치를 적용한다는 점에서 Bagging과 다르다. Boosting에서 분류기들을 생성하고 그들을 결합하는 방식을 좀더 자세히 살펴보면 다음과 같다. 먼저 원래 데이터 집합에 특정 학습 알고리즘을 적용하여 하나의 분류기를 생성해낸다. 그리고 이 분류기로 잘못 분류되는 훈련 데이터들만을 골라 이들의 가중치를 증가시켜 준

다. 그런 다음 이렇게 가중치가 조정된 데이터 집합으로부터 부트스트랩 표본 추출을 적용하여 동일한 크기의 훈련 데이터 집합을 확보한다. 그리고 이 새로운 훈련 데이터 집합으로부터 이전과 동일한 학습 알고리즘을 적용함으로써 또 다른 분류기를 유도해낸다. 이와 같은 과정은 훈련 데이터들에 대한 분류기의 분류 오류율(error)  $e$ 가 0이 되거나 오히려 0.5 이상이 될 때까지 계속된다. 그리고 이렇게 생성된 분류기들을 이용하여 새로운 데이터에 대한 분류작업을 할 때, 각 분류기의 분류 오류율(error)을  $e$ 라고 하면 그 분류기의 예측결과에  $-\log(e/(1-e))$ 만큼의 가중치를 적용하여 최종결과에 반영한다.

Boosting은 분류기 생성시에 가중치 부트스트랩 표본 추출 방법을 적용함으로써 분류기별로 서로 분리된(disjoint) 훈련 데이터 집합을 이용하도록 유도하고, 또한 생성되는 각 분류기도 자신에게 할당된 특정 훈련 집합에만 높은 성능을 보일 수 있도록 특화되는 경향을 보인다. 또 분류작업시 최종 결과를 도출할 때에도 각 분류기의 분류성능에 따라 가중치를 적용함으로써 분류성능이 높은 분류기의 예측결과가 더 많이 반영되는 경향을 보인다. 서로 다른 훈련 데이터 집합들을 이용하는 다중 분류기 시스템들인 Bagging과 Boosting 모두 다양한 영역의 데이터 집합들에서 높은 성능을 보이는 것으로 알려져 있다.

### 3.3 Stacking

Stacking은 (그림 3)과 같이 동일 훈련 데이터 집합에 대해 서로 다른 학습 알고리즘들을 적용하여 기반 분류기(base classifier)들을 생성하고, 이들을 결합하기 위해 다시 메타 학습 알고리즘(meta learner)을 적용하는 대표적인 다중 분류기 시스템이다. Bagging과 Boosting이 서로 다른 훈련 데이터 집합들에 기초하여 분류기들을 생성하고 이들의 예측결과를 단순히 투표 방식으로 결합하는데 반해, Stacking은 서로 다른 학습 알고리즘들에 기초하여 분류기들을 생성하고 이들의 예측결과를 토대로 상위 단계에서 또 한번 학습하는 메타 학습기를 두어 최종 결과를 도출한다. 따라서 Stacking 방식의 다중 분류기 시스템은 하위의 기반 단계(base level, level-0)와 상위의 메타 단계(meta level, level-1) 등 두 단계의 분류기들로 구성된다. Stacking에서 기반 분류기(base classifier)들은 각 훈련 데이터에 대한 클래스 확률 분포를 계산해줌으로써 상위의 메타 단계에서 메타 분류기 학습에 사용할 메타 데이터(meta data)를 생성한다. 한편 상위 단계의 메타 분류기는 이러한 메타 데이터를 바탕으로 어떤 기반 분류기의 예측이 신뢰할만지를 판단하는 역할을 수행한다. Stacking시스템에 분류를 위한 새로운 데이터가 주어지면 기반 단계와 상위 메타 단계, 두 단계의 분류기를 거치면서 최종 예측 클래스를 판단해낸다.



(그림 3) Stacking

앞서 설명한 Stacking 방식의 다중 분류기 시스템을 다시 정리하면 다음과 같다.  $I$ 개의 서로 다른 클래스들과  $N'$ 개의 데이터들로 구성된 훈련 데이터 집합  $L = (x_n, y_n), n = 1, \dots, N'$ 으로부터  $K$ 개의 서로 다른 기반 분류기들을 생성하였다고 가정하자. 그리고 훈련 데이터 각각에 대해 기반 분류기별로 그 데이터가 속할 클래스들의 확률분포를 다음과 같이 구해준다고 하자.  $P_{kn} = (p_{k1}(x_n), \dots, p_{ki}(x_n), \dots, p_{kl}(x_n))$ , 이때  $P_{kn}$ 는  $k$ 번째 기반 분류기가  $n$ 번째 훈련 데이터에 대해 계산한 확률벡터이며,  $p_{ki}(x_n)$ 는  $n$ 번째 훈련 데이터가  $i$ 번째 클래스에 속할 확률을 의미한다. 그러면  $N'$ 개의 훈련 데이터들을 통해  $K$ 개의 서로 다른 기반 분류기들로부터 얻어진 이와 같은 확률벡터들을 다음과 같은 집합  $L'$ 으로 결합한다.  $L' = (P_{1n}, \dots, P_{kn}, \dots, P_{Kn}, y_n), n = 1, \dots, N'$ . 이것을 메타 데이터들의 집합이라고 부른다. 그리고 이러한 메타 데이터들에 하나의 메타 학습 알고리즘을 적용함으로써 최종 클래스를 예측할 수 있는 메타 분류기  $M'$ 을 유도한다. 새로운 데이터를 분류하기 위해서는 각 기반 분류기  $M_k$ 가 이 데이터에 대해 예측하는 클래스 확률벡터  $(p_{11}, \dots, p_{1I}, \dots, p_{k1}, \dots, p_{kI}, \dots, p_{K1}, \dots, p_{KI})$ 를 구해 주게 되고, 이것은 메타 분류기  $M'$ 의 입력으로 전달이 되어 최종 분류결과인  $M'$ 의 출력을 얻을 수 있다. Stacking은 Bagging이나 Boosting과 같은 다중 분류기 시스템들보다 앞서 소개가 되었으나 이들에 비해 아직은 실제 적용 사례가 많지 않다. 그 이유는 Bagging이나 Boosting 방식에 비해 복잡도가 높다는 점과 메타 학습을 위해 어떤 알고리즘을 사용하는 것이 바람직한가 등 아직 연구가 미흡한 부분들이 남아 있기 때문인 것으로 생각된다.

#### 4. 확장형 다중 분류기 시스템

앞에서 살펴본 바와 같이 대표적인 다중 분류기 시스템들인 Bagging, Boosting, 그리고 Stacking은 각각 기반 분류기들을 생성하는 방법과 그들을 결합하는 방법에 있어 차이가 있다. 기반 분류기 생성 관점에서 보면, Bagging과 Boosting은 선택된 훈련 데이터 집합들의 편향성(bias)을 이용하는 반면, Stacking은 적용하는 학습 알고리즘들의 편향성을 이용한다는 점에서 차이가 있다. 또 기반 분류기들을 결합하는 방식을 비교해보면, Bagging과 Boosting은 단순히 기반 분류기들을 투표방식으로 결합하는데 비해 Stacking은 한 단계 높은 메타 학습층을 두어 기반 분류기들의 예측패턴으로부터 최종결과를 예측해낼 수 있도록 한다. 기존의 실험연구들에 의하면 다양한 실험 데이터 집합들에서 메타 학습기를 이용한 결합방식이 투표를 이용한 결합방식들에 비해 대체로 더 우수한 분류 성능을 보여주는 것으로 알려져 있다. 본 연구에서는 기존의 순수 Bagging, Boosting, Stacking 방식에서는 포함하고 있지 않은 분류기 생성과정의 편향성이나 새로운 분류기 결합방식을 도입하여 새로운 다중 분류기 시스템들인 Stacked Bagging과 Stacked Boosting, 그리고 Bagged Stacking과 Boosted Stacking을 제안한다. 그리고 이러한 새로운 다중 분류기 시스템들을 문서 분류에 적용하여 문서 분류의 정확도를 높여 보려고 한다.

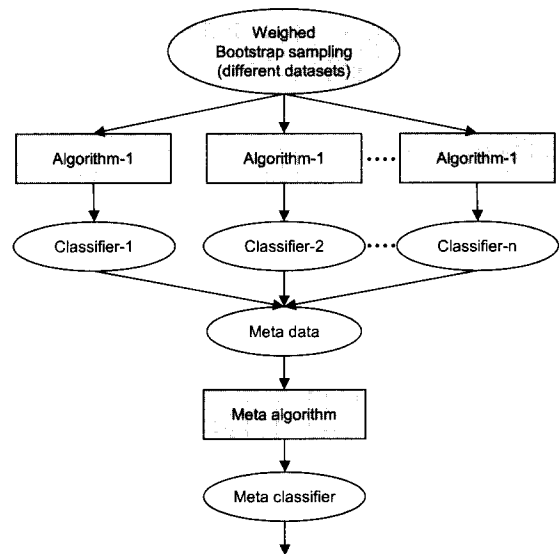
##### 4.1 Stacked Bagging과 Stacked Boosting

기존의 순수 Bagging과 Boosting은 기반 분류기들을 비교적 단순한 다수결 투표와 가중치 투표방법으로 결합한다. 본 연구에서 새로이 제안하는 다중 분류기 방식인 Stacked Bagging과 Stacked Boosting은 기존 Bagging과 Boosting에서 사용하는 투표기반의 분류기 결합방식을 유연성이 큰 Stacking의 메타 학습기 방식으로 교체, 확장한 것들이다. 즉 새로운 Stacked Bagging과 Stacked Boosting은 기반 분류기들을 생성할 때는 종래의 방식대로 각각 순수 부트스트랩 표본 추출법과 가중치 부트스트랩 표본 추출법을 적용하여 서로 다른 훈련 데이터 집합들을 유도하는 대신, 기반 분류기들을 결합할 때는 기반 분류기들이 생성한 메타 데이터에 기초하여 하나의 메타 분류기를 생성하고 이것을 이용한다. 이렇게 함으로써 훈련 데이터 집합상의 편향성을 이용하는 기존의 순수 Bagging과 Boosting의 장점에 기반 분류기들의 결합을 위해 메타 학습을 이용하는 Stacking의 장점을 더할 수 있다.

문서 분류에 적용할 Stacked Bagging 방식을 다시 정리하면 다음과 같다. 먼저 부트스트랩 표본 추출법(bootstrap sampling)을 통하여 훈련 문서 집합의 구성을 달리 하면서 일정 수의 기반 분류기들을 생성한 다음, 일정한 수의 문서

집합에 대한 각 기반 분류기의 예측 결과인 클래스 확률 분포를 생성한다. 그리고 이 클래스 확률분포를 메타 데이터로 삼아 상위 단계의 메타 분류기를 생성한다. 새로운 문서를 분류할 때는 먼저 이 문서를 기반단계 분류기들의 입력으로 제공하여 예측결과들을 출력해내고, 다시 이것을 상위단계 메타 분류기의 입력으로 제공하여 최종 분류 클래스를 도출해낸다.

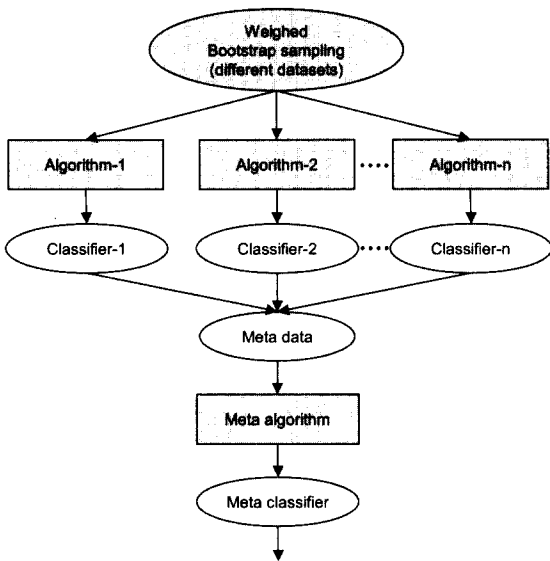
이번에는 문서 분류에 적용할 Stacked Boosting 방식을 다시 정리하면 다음과 같다. Stacked Boosting은 (그림 4)와 같이 가중치를 적용한 부트스트랩 표본 추출법(weighted bootstrap sampling)을 통하여 훈련 문서 집합의 구성을 달리 하면서 기반 분류기들을 생성한 다음, 일정한 수의 문서 집합에 대한 각 기반 분류기의 예측 결과인 클래스 확률 분포를 생성한다. 그리고 이 클래스 확률분포를 메타 데이터로 삼아 상위 단계의 메타 분류기를 생성한다. Stacked Boosting의 경우, 생성되는 기반 분류기의 오류율(error)이 0 또는 0.5에 이를 때까지 기반 분류기들을 생성하므로 생성되는 기반 분류기의 수는 Stacked Boosting의 경우처럼 일정하지 않고 가변적이다.



(그림 4) Stacked Boosting

##### 4.2 Bagged Stacking과 Boosted Stacking

기존의 순수 Stacking은 기반 분류기들의 생성을 위해 동일한 하나의 훈련 데이터 집합에 단지 학습 알고리즘들만을 바꾸어 적용한다. 본 연구에서 제안하는 또 다른 형태의 다중 분류기 방식인 Bagged Stacking과 Boosted Stacking은 기존 Stacking처럼 동일한 훈련 데이터 집합을 사용하지 않고 Bagging과 Boosting처럼 서로 다른 훈련 데이터 집합들을 이용하여 기반 분류기들을 생성한다. 즉 새로운 Bagged Stacking과 Boosted Stacking은 (그림 5)와 같이



(그림 5) Boosted Stacking

<표 1> Boosted Stacking 알고리즘

<p><b>Base ClassifierGeneration :</b>          Assign equal weight to each training instance          For each of k iterations              Apply the k-th base learning algorithm to weighted dataset and store the resulting base classifier              Compute error <math>e</math> of model on weighted dataset and store error              If <math>e</math> equals to zero, or <math>e</math> greater or equal to 0.5                  Terminate classifier generation          For each instance in dataset              If instance classified correctly by classifier                  Multiply weight of instance by <math>e/(1-e)</math>          Normalize weight of all instances</p> <p><b>Meta Data Generation :</b>          For each instance in dataset          For each of k iterations              Apply the k-th base classifier to predict the class probability of the instance          Generate meta data by combining class probabilities from the base classifiers</p> <p><b>Meta ClassifierGeneration :</b>          Apply the meta learning algorithm to the meta dataset and store the resulting meta classifier</p> <p><b>Classification :</b>          For each of k iterations              Apply the k-th base classifier to predict the class probability of the new instance          Generate a test data for the meta classifier by combining k predictions of the base classifiers          Return the class predicted by the meta classifier</p>
---

기본 분류기들을 생성할 때는 각각 순수 부트스트랩 표본 추출법(bootstrap sampling)과 가중치 부트스트랩 표본 추출법(weighted bootstrap sampling)을 적용하여 서로 다른 훈련 데이터 집합들을 유도하고 여기에 서로 다른 학습 알고리즘들을 적용한다. 하지만 기본 분류기들을 결합할 때는

기본 방식대로 메타 학습을 통해 하나의 메타 분류기를 생성하고 이것을 이용한다. 이렇게 함으로써 기본 분류기들의 결합을 위해 메타 학습을 이용하는 기존 Stacking의 장점에 훈련 데이터 집합상의 편향성을 이용하는 Bagging과 Boosting의 장점을 더할 수 있다. <표 1>은 요약된 Boosted Stacking의 알고리즘을 나타내고 있다.

## 5. 실험

### 5.1 실험 설계

본 연구에서는 문서 분류 실험을 통하여 Bagging, Boosting, Stacking과 같은 기존 다중 분류기 시스템들과 앞서 제안한 Stacked Bagging, Stacked Boosting, Bagged Stacking, Boosted Stacking 등 새로운 다중 분류기 시스템들의 분류 성능을 비교해 보려고 하였다. 또 이 밖에도 메타 학습 알고리즘이 다중 분류기 시스템들의 분류 성능에 어떤 영향을 주는 지도 함께 분석해보려고 하였다. 이와 같은 실험목적을 달성하기 위해 본 연구에서는 MEDLINE 학술 기사, 유즈넷 뉴스(Usenet news) 기사, 그리고 웹 문서 등을 실험용 문서 집합들로 사용하였다. OSHUMED는 MEDLINE 학술데이터베이스에서 확보한 문서집합으로서 정보검색 연구의 표준데이터로 이용되고 있으나 주제 분류 항목이 결여되어 있어 자동 분류의 훈련 데이터로 적합치 않았다. 따라서 본 실험에서는 MeSH(Medical Subject Heading) 주제가 가운데 'Mental Disorder'의 하위 토픽을 MEDLINE 데이터베이스에 직접 질의하여 OSHUMED와 동일한 형식의 훈련 문서 집합을 확보하였다. 유즈넷 뉴스 문서들은 5개의 컴퓨터 관련 주제영역으로부터 문서 데이터들을 확보하였다. 웹 문서의 경우는 몇 개의 미국 대학 전산학과 홈페이지에서 주제별로 확보한 문서집합을 이용하였다. 다음은 본 실험에 사용한 훈련 문서 집합에 대한 상세 내역이다.

- MEDLINE 문서집합
  - 클래스 개수 = 6
  - {alzheimer, anxiety, dementia, depression, memory, schizophrenia}
  - 클래스 당 문서 수 = 1,000
  - 총 문서 수 = 6,000
  - 훈련 문서 수 =  $0.66 \times 6,000$
  - 테스트 문서 수 =  $0.34 \times 6,000$
- 유즈넷 뉴스 문서집합
  - 클래스 개수 = 5
  - {comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x}
  - 클래스 당 문서 수 = 1,000
  - 총 문서 수 = 5,000
  - 훈련 문서 수 =  $0.66 \times 5,000$
  - 테스트 문서 수 =  $0.34 \times 5,000$
- 웹문서 집합
  - 클래스 개수 = 6
  - {student, faculty, staff, department, course, project}

클래스 당 문서 수  
 student : 1,641, faculty : 1,124, staff : 137, department : 182,  
 course : 930, project : 504  
 총 문서 수 = 45,181  
 훈련 문서 수 =  $0.66 \times 4518$   
 테스트 문서 수 =  $0.34 \times 4518$

문서 모델을 생성하기 위해 문서 집합들로부터 각 단어의 정보 획득량(IG, Information Gain)을 기준으로 특징어들을 1,000개씩 선택하여 가중치 벡터 모델을 구축하였다. 실험에 사용한 학습 알고리즘들은 다음과 같다. 단일 분류기 생성에는 k-NN, C4.5, Nave Bayesian(NB)등을 각각 적용하였고, Bagging과 Boosting, 그리고 Stacking 등 다중 분류기시스템의 기반 학습에 C4.5와 Nave Bayesian(NB)을 적용하고, 메타 학습에도 C4.5, Nave Bayesian(NB) 등을 적용하여 보았다. 실험은 Linux 운영체제를 갖춘 Intel Pentium IV processor, 256MB 하드웨어 환경에서 수행되었다. 실험을 위한 각 다중 분류기 시스템들은 WEKA 라이브러리[15]를 기초로 Java 프로그래밍 언어로 구현하였다.

5.2 실험 결과

실험에 사용된 세 가지 종류의 문서 집합 각각에 대한 분류기들의 분류 정확도를 백분율로 나타낸 것이 <표 2>와 같다. 표에서 각 다중 분류기시스템에 적용된 학습 알고리즘들을 나타내기 위해 기반 학습알고리즘(base learner)은 B로, 메타 학습알고리즘(meta learner)은 M으로 표시하였다. 실험 결과를 살펴보면, 먼저 분류기들의 성능이 문서 집합에 따라 큰 편차를 보였는데 MEDLINE과 웹 문서집합의 경우는 분류 정확도가 약 70~80% 대에 머무는 반면에 유즈넷 뉴스 문서집합은 모든 분류기가 97%가 넘는 높은 분류 정확도를 보였다. 이것은 유즈넷 뉴스 문서들의 경우 다른 훈련 문서집합들에 비해 각 뉴스기사들이 주제별로 잘 정리되어 있어 분류가 용이했던 때문으로 추측된다. 단일 분류기들 중에서는 확률에 기초한 Naive Bayesian 분류기가 세 가지 문서 집합들 모두에서 가장 좋은 성능을 보였다. 전체적으로 단일 분류기들에 비해 기존의 다중 분류기시스템들이, 또 기존의 다중 분류기시스템들에 비해 본 논문에서 제안한 혼합형 분류기시스템들이 더 높은 성능을 보였다. 이와 같은 사실은 <표 3>을 통해서도 확인할 수 있는데, <표 3>은 적용된 기반 학습 알고리즘과 메타 학습 알고리즘을 구분하지 않고 분류기 유형별로 평균 분류 성능을 구하여 정리한 것이다. 이러한 결과는 훈련 데이터 또는 학습 알고리즘의 편향성을 이용하여 기반 분류기들을 생성하는 방식과 별도의 메타 학습기를 이용하여 기반 분류기들을 결합하는 방식이 하나의 다중 분류기 시스템 속에서 통합될 때 문서 분류에 더욱 효과가 있음을 보여준 것이다.

<표 2> 문서 집합별 분류기들의 성능(정확도 %)

분류기	Generalizers	Document collections		
		MEDLINE	UsenetNews	Web
단일 분류기	k-NN	63.77	97.21	75.73
	C4.5	72.98	97.12	79.31
	NB	73.57	97.29	81.78
기존의 다중분류기 (voting)	Bagging(B : C4.5)	74.01	97.12	81.85
	Bagging(B : NB)	74.17	97.34	82.57
	Boosting(B : C4.5)	74.48	98.06	83.02
	Boosting(B : NB)	74.66	98.06	83.19
기존의 다중분류기 (stacking)	Stacking(M : C4.5)	74.99	97.88	83.19
	Stacking(M : NB)	74.21	97.76	82.98
혼합형 다중분류기 (extended bagging/boosting)	StackedBagging (B : C4.5, M : C4.5)	75.18	98.16	83.26
	StackedBagging (B : C4.5, M : NB)	74.66	98.18	82.85
	StackedBagging (B : NB, M : C4.5)	74.87	98.16	84.39
	StackedBagging (B : NB, M : NB)	74.07	97.48	83.24
	StackedBoosting (B : C4.5, M : C4.5)	76.12	98.49	83.94
	StackedBoosting (B : C4.5, M : NB)	75.85	98.36	83.21
	StackedBoosting (B : NB, M : C4.5)	76.23	98.79	85.05
	StackedBoosting (B : NB, M : NB)	75.93	98.09	83.19
	혼합형 다중분류기 (extended stacking)	BaggedStacking (M : C4.5)	77.03	98.56
BaggedStacking (M : NB)		76.29	98.58	85.22
BoostedStacking (M : C4.5)		79.07	98.62	87.02
BoostedStacking (M : NB)		78.29	98.62	86.61

<표 3> 분류기 유형별 평균성능(정확도 %)

분류기	MEDLINE	Usenet	Web	
단일 분류기 (k-NN, C4.5, NB의 평균)	70.11	97.21	78.94	
기존의 다중 분류기	Bagging	74.09	97.23	82.21
	Boosting	74.57	98.06	83.11
	Stacking	74.60	97.82	83.09
혼합형 다중 분류기	StackedBagging	74.70	98.00	83.44
	StackedBoosting	76.03	98.43	83.85
	BaggedStacking	76.66	98.57	86.08
	BoostedStacking	78.68	98.62	86.82

또 전반적으로 메타 학습기를 이용한 분류기들의 통합 효과도 확인할 수 있는데, 동일한 학습 알고리즘으로부터 얻어진 동질적인 분류기들(homogenous classifiers)의 통합에



도, 서로 다른 학습 알고리즘들로부터 얻어진 이질적인 분류기들(heterogenous classifiers)의 통합에도 모두 효과가 높은 것으로 나타났다. 예컨대 각각 서로 이질적인 학습 알고리즘인 k-NN, C4.5, Naive Bayesian을 적용하는 단일 분류기들에 비해 이들을 메타 학습으로 통합하는 Stacking이 더 높은 성능을 보일 뿐 아니라, 동일한 하나의 학습 알고리즘을 적용하는 Bagging과 Boosting에 비해 이들을 메타 학습으로 확장한 StackedBagging과 StackedBoosting이 각각 더 개선된 성능을 보였다. 본 실험에 이용된 메타 학습기들간의 성능을 비교해보면, Naive Bayesian에 비해 C4.5가 거의 모든 경우에 더 높은 성능을 보였다. 한 가지 흥미로운 사실은 <표 2>의 실험 결과에는 나타나지 않았으나 Decision Stump(StackDC)와 같은 낮은 성능의 학습 알고리즘을 메타 학습기로 이용하는 경우, 다중 분류기 시스템들은 단일 분류기들보다 더 낮은 성능을 보였다. 이것은 다중 분류기 시스템이 채용하는 메타 학습기에 따라 성능이 크게 달라질 수 있다는 것을 암시한다.

한편, 본 논문에서 제안한 혼합형 다중 분류기 시스템들간의 분류 성능을 비교해보면, Stacking을 확장한 Bagged Stacking과 Boosted Stacking이 Bagging과 Boosting을 확장한 Stacked Bagging과 Stacked Boosting 보다 성능이 높게 나타났다. 이 두 다중 분류기 유형간의 차이점은 적용되는 기반 학습알고리즘의 차이뿐이다. 즉, StackedBagging과 StackedBoosting은 모두 하나의 동일한 기반 학습 알고리즘만을 이용하는데 반해, BaggedStacking과 BoostedStacking은 서로 다른 기반 학습 알고리즘들을 이용한다. 따라서 기반 학습 알고리즘들의 편향성을 이용하는 혼합형 다중 분류기들이 그렇지 않은 다중 분류기들에 비해 더 우수한 결과를 보인 것으로 해석할 수 있다. 네 가지 혼합형 다중 분류기 시스템들 중에서는 훈련 데이터의 편향성, 학습 알고리즘의 편향성, 그리고 메타 학습기까지 이 모두를 이용하는 Boosted Stacking이 가장 안정적이고 높은 성능을 보여 주었다.

## 6. 결 론

본 논문에서는 문서 분류를 위한 새로운 다중 분류기 시스템들인 Stacked Bagging, Stacked Boosting, Bagged Stacking, Boosted Stacking들을 제안하였다 이들은 Bagging, Boosting, Stacking과 같은 기존 다중 분류기 시스템들의 장점들을 결합한 일종의 혼합형 다중 분류기 시스템들이다. 본 논문에서는 MEDLINE, 유즈넷 뉴스, 웹 문서 등의 문서집합을 이용한 문서 분류 실험들을 전개하여 새로운 다중 분류기 시스템들의 성능을 평가하였다. 그리고 이러한 실험 결과를 통해 새로이 제안한 혼합형 다중 분류기 시

스템들이 전반적으로 기존 시스템들보다 우수한 성능을 보인다는 것을 확인하였다. 혼합형 다중 분류기 시스템들은 높은 계산 비용을 요구하지만 성능 향상과 신뢰도 확보가 중요시 되는 응용 분야들에서는 그 비용이 정당화될 것으로 판단된다.

## 참 고 문 헌

- [1] Sahami, Mehran. "Using Machine Learning to Improve Information Access," a Dissertation, Stanford : Dept. of Computer Science, Stanford University, 1998.
- [2] Mladenić, D., Grobelnik, M., "Efficient Text Categorization," In *Text Mining workshop on the 10th European Conference on Machine Learning ECML98*, 1998.
- [3] Chen, H., "Machine Learning for Information Retrieval : Neural Networks, Symbolic Learning and Genetic Algorithms," *JASIS*, Vol.46, pp.194-216, 1995.
- [4] Hong, Se June and Sholom M. Weiss, "Advances in Predictive Model Generation for Data Mining," IBM Research Report RC-21570, 1999.
- [5] Breiman, Leo, "Bagging Predictors," *Machine Learning*, Vol.24, pp.49-6, 1996.
- [6] Schaphire, Robert E., "Theoretical Views of Boosting," In *Computational Learning Theory : 4th European Conference, EuroCOLT '99*, 1999.
- [7] Wolpert, David H., "Stacked Generalization," *Neural Networks*, Vol.5, pp.241-259. 1992.
- [8] Bauer, Eric and Ron Kohavi, "An Empirical Comparison of Voting Classification Algorithms : Bagging, Boosting, and Variants," *Machine Learning*, 36, pp.105-142, 1999.
- [9] Salton, Gerard, "Introduction to Information Retrieval," New York : McGraw-Hill, 1983.
- [10] Han, Jiawei and Micheline Kamber, "Data Mining : Concepts and Techniques," New York : Morgan Kaufmann, 2001.
- [11] Shankar, Shrikanth and Karypis, George, "A Feature Weight Adjustment Algorithm for Document Categorization," In *SIGKDD '00 Workshop on Text Mining, Boston, MA*, 2000.
- [12] Mitchell, Tom, "Machine Learning," New York : McGraw-Hill, 1997.
- [13] Lewis, D. D. and M. Ringuette, "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, pp.81-93, 1994.
- [14] Domingos, P. and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero One Loss," *Machine Learning*, Vol.29, pp.103-130, 1997.

- [15] Witten, Ian H. and Eibe Frank, "Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations," New York : Morgan Kaufman, 2000.
- [16] Wolpert, D. and W. Macready, "Combining Stacking with Bagging to Improve a Learning Algorithm," Technical report. Santa Fe : Santa Fe Institute, 1996.
- [17] Yang, Yiming and Jan O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," In *Proceedings of the Fourteenth International Conference on Machine Learning*, Vol.43, 1997.

### 김 인 철

e-mail : kic@kyonggi.ac.kr

1985년 서울대학교 수학과(학사)

1987년 서울대학교 계산통계학과(이학석사)

1995년 서울대학교 계산통계학과(이학박사)

1989년~1995년 경남대학교 전산통계학과  
전임강사, 조교수

1996년~현재 경기대학교 정보과학부 부교수

관심분야 : 지능형 정보검색, 기계학습, 데이터마이닝, 게임AI