*Journal of Korean*
*Data & Information Science Society*
*2005, Vol. 16, No. 4, pp. 999~1012*

# Comparison of Boosting and SVM[1)]

Yongdai Kim[2)] · Kyoung Hee Kim[3)] · Seuck Heun Song[4)]

**Abstract**

We compare two popular algorithms in current machine learning and statistical learning areas, boosting method represented by AdaBoost and kernel based SVM (Support Vector Machine) using 13 real data sets. This comparative study shows that boosting method has smaller prediction error in data with heavy noise, whereas SVM has smaller prediction error in the data with little noise.

***Keywords*** : boosting, classification, prediction, SVM

## 1. 머리말

최근의 기계학습 분야와 통계학습 분야에서 크게 각광을 받고 있는 두개의 알고리즘은 AdaBoost로 대표되는 boosting 알고리즘과 kernel 기법을 사용한 SVM (Support Vector Machine)이다. Boosting알고리즘의 효시인 AdaBoost는 Freund 와 Schapire (1997)에 의하여 제안 되었으며, 그 기본 아이디어는 약한 학습기 (weak learner) 여러 개를 결합하여 아주 예측력이 뛰어난 학습기 (learner혹은 classifier)를 만드는 것이다. AdaBoost의 높은 예측력으로 인하여 발표된 즉시 많은 관심을 끌었으며, 다양한 종류의 부스팅 알고리즘들 - Confidence rated boosting (Singer & Schapire 1999), Margin Boost (Mason, Baxter, Bartlett & Frean 2000), Logit Boost (Friedman 2001)이 개발되었다.

SVM은 Vapnik교수에 의하여 처음으로 제안되었으며 (Vapnik 1998), 고차원의 입력변수를 갖는 자료를 kernel 기법을 이용하여 아주 쉽게 분석할 수 있다는 장점으

---

1) 이 논문은 2004년도 서울대학교 신임교수 연구정착금, 과학기술부/한국과학재단 우수연구센터 육성사업 (R11-2000-073-00000) 및 한국과학재단 특정기초연구 (R01-2003-000-10589) 지원으로 수행되었음.

2) 제1저자 : 서울시 관악구 신림동 서울대학교 통계학과 조교수
E-mail : ydkim@stats.snu.ac.kr
3) 서울시 관악구 신림동 서울대학교 복잡계통계연구소 연구원
4) 서울시 성북구 안암동 고려대학교 통계학과 부교수

로 인하여, 이미지 분석이나 마이크로어레이 자료의 분석 등에 널리 쓰이고 있다. SVM의 가장 중요한 장점 중의 하나는 추정이 필요한 변수의 개수가 입력변수의 차원에 의존하지 않고 자료의 수에 의존한다는 것이다. 따라서, 입력변수의 차원이 자료의 수보다 아주 큰 경우 (예: 이미지 분석, 마이크로어레이 자료)에 아주 유용하게 사용되어지고 있으며 높은 예측력으로 크게 각광을 받고 있다. SVM의 또 다른 장점으로는 예측모형이 sparse하다는 것인데, 즉 추정이 필요한 변수 중 소수개의 변수만이 0아닌 계수 값을 갖고 나머지 변수들의 계수는 모두 0이 된다는 것이다. 이러한 sparse성질은 이론적인 장점과 더불어 자료의 압축 등에 유용하게 사용되어지고 있다.

Boosting과 SVM은 비슷한 시기에 개발되어서 많은 관심을 끌고 있고 또 많은 실제 문제에 적용되고 있는 것에 비하여, 이 두 알고리즘에 대한 심층적 비교연구는 매우 미미한 실정이다. Schapire et al. (1998)등은 SVM에 사용된 개념인 margin을 이용하여 부스팅의 이론적 성질을 연구하였다. SVM에 대한 boosting의 장점의 하나는 분류기의 생성뿐 아니라 확률추정이 가능하다는 것이다. 또한. 약한 분류기를 사용하므로써 입력변수의 차원이 아주 크지 않은 경우에는 계산속도가 빠르다. 또한, 이론적으로는 regularized boosting 알고리즘은 점근적으로 일치하며 수렴속도가 최적이 됨이 증명되었다 (Jiang 2002). 이에 비하여, SVM의 가장 중요한 장점은 계산량이 입력변수의 차원에 의존하지 않으므로 입력변수가 자료의 수 보다 아주 큰 경우에 효율적으로 작동하며, 또한, 많은 논문에서 실제 자료에서 매우 예측력이 높은 것으로 보고되고 있다.

본 논문에서는 이 두개의 알고리즘을 이용하여 실제 자료를 분석하는데 생기는 장/단점과 문제점, 그리고 실제자료에 대한 예측력 등을 비교하여, 대용량자료를 분석하는 사용자에게 여러 가지 정보를 제공하고자 한다. UCI machine learning repository로부터 13개의 실제 자료를 선정하고 이 자료들을 boosting과 SVM으로 분석한 후 그 결과를 비교하였다. 본 논문의 결과는 주어진 자료에서 boosting과 SVM 중 어떤 방법이 더 적합한지에 대한 지침서로 사용되어질 수 있을 것이라 사료된다.

본 논문은 다음과 같이 구성되어진다. 2절과 3절에서는 각각 boosting과 SVM 알고리즘에 대하여 설명을 하며, 4장에서는 비교에 사용되어진 13개의 자료에 대한 간략한 설명 및 비교방법에 대하여 설명하고 분석결과를 정리 및 해석한다. 5장은 맺음말로 자료분석을 통한 boosting 과 SVM알고리즘의 비교를 통하여 발견된 사실을 정리하고 가능한 연구주제에 대하여 간략하게 토의한다.

## 2. Boosting

부스팅(Boosting)은 약한 학습기(weak learner)들을 결합시켜 예측모형을 구축하는 알고리즘이다. 이것의 목적은  강력한 예측모형을 여러 개의 약한 학습기들의 선형결합으로 구축하는 것이다. 약한 학습기란 그것의 오분류율이 임의 추측보다 근소한 차이라도 더 작은 경우를 뜻한다.

2.1절에서는 Freund 와 Shapire(1997)에 의해 제안된 부스팅 알고리즘인

"Adaboost"을 중점적으로 설명할 것이고, 2.2절에서는 AdaBoost의 확장된 알고리즘 중 하나인 Gradient boosting machine (Friedman 2001)을 소개한다. Gradient boosting machine알고리즘은 R 시스템에 함수로 되어 있으며, 본 논문의 자료 분석에 사용된 알고리즘이다.

## 2.1 AdaBoost

학습자료 $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$이 관찰되었다고 하자. 여기에서 $x_i$는 입력 공간 $\chi$에서의 원소이고 $y_i$는 집단을 구분하는 변수로서 1 혹은 -1의 값을 갖는다. 그리고 약한 학습기의 집합을 $\mathbb{G} = \{G_m(x) : \chi \rightarrow \{-1, 1\} \mid m = 1, \ldots, M\}$이라 표기하자. 여기에서 각각의 학습기는 주어진 입력 자료에 대해 자료의 소속집단을 나타내는 분류기(classifier)이다.

AdaBoost 알고리즘의 작동원리는 반복적으로 수정된 자료에 축차적으로 약한 학습기 $G_m(x), m = 1, 2, \ldots, M$의 수열(sequence)을 구성하는 것이다. 이러한 학습기들은 각각 다른 가중치를 부여받게 되고 다음과 같이 최종적인 예측을 하게 된다.

$$G(x) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right) \tag{2.1}$$

식 (2.1)의 $\alpha_1, \alpha_2, \ldots, \alpha_M$은 AdaBoost 알고리즘을 통해서 계산되고, 각각 $G_m(x)$에 기여하는 정도를 나타낸다. 즉, 이러한 계수들은 더 정확한 학습기가 더 높은 영향을 미치게 하는 것이다. 집단의 수가 2개인 경우, $\alpha_m$은 전형적으로 다음과 같이 놓는다.

$$\alpha_m = ln\left(\frac{1 - \epsilon_m}{\epsilon_m}\right),\ m = 1, \ldots, M \tag{2.2}$$

$\epsilon_m$은 $m$단계에서의 가중 오분류율이고 식 (2.2)는 Freund와 Shapire에 의해 제안되었다.

아래의 AdaBoost 알고리즘을 간략히 설명하면 다음과 같다. 우선 첫 단계에서는 모든 가중치를 $w_i = 1/N$으로 놓아 단순히 자료를 일반적인 방법으로 분류하게 된다. 오분류율이 계산된 이후 업데이트된 가중치가 계산되고, 각 부스팅 단계에서 입력 개체 $(x_i, y_i), i = 1, 2, \ldots, N$ 각각에 가중치인 $w_1, w_2, \ldots, w_N$을 적용하여 자료가 수정된다. 이러한 가중치는 각각의 반복 $m = 2, 3, \ldots, M$에서 개체에 적용된 개별적인 관측치가 된다. 그러나 $m$ 단계에서는, 전 단계의 분류기인 $G_{m-1}(x)$에 의해 잘못 분류된 관측치는 더 높은 가중치를 갖게 되고 정확하게 분류된 관측치는 가중치가 낮아진다. 그러므로 반복횟수가 늘어날수록, 정확히 분류하기 어려운 관측치는 계속 가중치가

늘어나게 된다. 즉, 각각의 분류기는 전 단계에서 잘못 분류된 관측치가 더 잘 분류되도록 한다.

---

**AdaBoost 알고리즘**

---

$x_i \in X,\ y_i \in Y = \{-1, +1\}$일 때 $(x_1, y_1), \ldots, (x_N, y_N)$이 주어지면,

1. 개체에 대한 가중치를 다음과 같이 초기화한다.

$$w_i = 1/N,\ i = 1, 2, \ldots, N$$

2. $m = 1$부터 $M$까지 다음 단계를 반복한다.

(a) 가중치 $w_i$를 사용하여 훈련 자료에 분류기 $G_m(x)$를 적합시킨다.

(b) 다음을 계산한다.

$$\epsilon_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$

(c) $\alpha_m = log((1-\epsilon_m)/\epsilon_m)$으로 놓는다.

(d) $w_i$를 업데이트 한다

$$w_i \leftarrow w_i \cdot exp\left[\alpha_m \cdot I(y_i \neq G_m(x_i))\right],\ i = 1, 2, \ldots, N$$

3. 최종 분류기 $G(x) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$를 구한다.

---

Adaboost 알고리즘의 가장 기본적인 이론적 특성은 그것이 훈련자료에서의 오분류율을 낮춘다는 것이다. 특히, 매 단계에서의 $\epsilon_m$이 0.5보다 작으면, 학습자료의 오분류율은 0으로 지수적으로 빠르게 감소한다는 것이다. 이러한 성질로 인하여, AdaBoost가 나왔을 때는 자료의 압축이나, 노이즈가 없는 자료의 분류 등에 사용되었다.

## 2.2 Gradient boosting machine

AdaBoost가 제안된 후로, 많은 연구자에 의해서 알고리즘의 작동원리에 대한 규명이 시도되었다. 그리고, 이러한 노력의 결실로 다양한 종류의 boosting 알고리즘들이 소개되었는데, 예로는 Confidence rated boosting (Singer & Schapire 1999), Margin Boost (Mason, Baxter, Bartlett & Frean 2000), Gradient boosting machine (Friedman 2001)등이 있다. 본 절에서는 이 중 gradient boosting machine에 대하여

간략히 소개하려고 한다.

먼저 AdaBoost 알고리즘이 지수적 손실(Exponential loss)을 갖는 최대경사법(steepest descent method)으로 이해할 수 있다(Schapir & Singer, 1999)는 것을 설명할 것이다. 그 후에 손실함수로 음이항 로그 가능도를 사용하고 경사도(gradient)를 이용하여 이를 확장한 gradient boosting 알고리즘을 설명할 것이다.

우선, Adaboost 알고리즘은 약한 학습기 집합 $\mathbb{G}$ 의 선형결합 공간에서 지수적 손실에 대해 경험적 위험을 최소화 하는 분류기를 찾아내는 알고리즘으로 이해해 볼 수 있다. 우선 다음과 같은 앙상블 모형을 고려하자.

$$G_m = \sum_{k=1}^{m} c_k f_k \tag{2.3}$$

현재의 추정치인 $G_m(x)$가 주어진 상태에서 개선된 추정치 $G_m(x) + cf(x)$를 구하는 문제를 고려하면, Adaboost는 다음과 같이 $(c_m, f_m)$를 찾는 방법이 된다.

$$(c_m, f_m) = argmin_{c \in [0,\infty], f \in \mathbb{G}} \sum_{i=1}^{n} exp(-y_i(G_m(x_i) + cf(x_i))) \tag{2.4}$$

즉, Adaboost는 다음과 같은 두 단계로 이루어져 있게 된다. 우선, $c$를 1로 하고 $w_i = exp(-y_i G_m(x_i))$라 할 때 $\sum_{i=1}^{n} w_i exp(-y_i f(x_i))$를 최소화 하는 $f_m$을 찾는다. 후에 $\sum_{i=1}^{n} w_i exp(-y_i c f_m(x_i))$를 최소화 하는 $c_m$찾는데 이것이 결국 Adaboost에서와 같은 값이 나오게 되는 것이다.

$$c_m = log(\frac{1-\epsilon_m}{\epsilon_m}) \tag{2.5}$$

이에 근거를 두고 Shapire와 Singer(1999)는 Adaboost 알고리즘을 약한 학습기가 실수함수인 경우에 대한 알고리즘으로 확장하였다.

**Real 부스트 알고리즘 : Realboost**

$x_i \in X,\ y_i \in Y = \{-\infty, +\infty\}$일 때 $(x_1, y_1), \ldots, (x_N, y_N)$이 주어지면,

1. 다음과 같이 초기화한다.

$$G_0(x) = 0$$

2. $m = 1$부터 $M$까지 다음 단계를 반복한다.

(a) $f_m$을 찾는다.

$$f_m = argmin_{f \in \mathbb{G}} \sum_{i=1}^{n} exp(-y_i(G_{m-1}(x_i) + f(x_i)))$$

(b) 다음을 업데이트 한다.

$$G_m = G_{m-1} + f_m$$

3. 최종 분류기 $G(x) = sign\left(\sum_{m=1}^{M} G_m(x)\right)$를 구한다.

이제 손실함수를 음이항 로그 가능도로 고려해보자(Friedman, Hastie & Tibshirani, 2000).

$$\Psi(y, F) = log(1 + exp(-2yF)),\ y \in \{-1, 1\}. \tag{2.6}$$

식 (2.6)에서 $F(x) = \frac{1}{2}\log\left(\frac{Pr(y=1 \mid x)}{Pr(y=-1 \mid x)}\right)$이다. 음이항 로그 가능도 손실함수를 최소로 하는 탐색법은 다음과 같다.

$$\rho_m = argmin_{\rho} \sum_{i=1}^{N} log(1 + exp(-2y_i(F_{m-1}(x_i) + \rho h(x_i; a_m)))) \tag{2.7}$$

식 (2.7)에서 $h(\boldsymbol{x}; \boldsymbol{a})$는 모수 $\boldsymbol{a} = \{a_1, a_2, \ldots\}$로 특징지워지는 입력변수 $\boldsymbol{x}$에 대한 간단한 모수적 함수이다.

기저 학습자(base learner)를 회귀나무로 두면 그것에 의해 만들어지는 영역의 disjoint한 성격 때문에 각각의 종결 노드 $R_{lm}$에서의 개별적인 업데이트를 사용할 수 있다.

$$\gamma_{lm} = argmin_{\gamma} \sum_{x_i \in R_{lm}} log(1 + exp(-2y_i(F_{m-1}(x_i) + \gamma))) \tag{2.8}$$

식 (2.8)에 대하여 닫힌 형태의 해는 존재하지 않는다. Friedman은 단일 뉴튼 랩슨 단계에 의해 식 (2.8)를 근사하였으며 형태는 다음과 같다.

$$\gamma_{lm} = \frac{\sum_{x_i \in R_{lm}} \tilde{y}_i}{\sum_{x_i \in R_{lm}} |\tilde{y}_i| (2 - |\tilde{y}_i|)} \tag{2.9}$$

식 (2.9)에서 유사 반응(pseudo-response) $\tilde{y}_i$은 다음과 같다.

$$\tilde{y}_i = -\left[\frac{\partial \Psi(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x) = F_{m-1}(x)} = \frac{2y_i}{(1 + \exp(2y_i F_{m-1}(x_i)))} \tag{2.10}$$

식 (2.10)을 고려하면 의사결정나무를 이용한 gradient boosting 알고리즘을 구할 수 있다.

---

**가능도 경사도 부스트 알고리즘 : $L_2$ Tree Boost**

---

$x_i \in X,\ y_i \in Y = \{-1, +1\}$일 때 $(x_1, y_1), \ldots, (x_N, y_N)$이 주어지면,

1. 다음과 같이 초기화한다.
$$F_0(x) = \frac{1}{2}\log\frac{1+\bar{y}}{1-\bar{y}}$$

2. $m = 1$부터 $M$까지 다음 단계를 반복한다.

(a) 다음을 계산한다
$$\tilde{y}_i = 2y_i/(1 + exp(2y_i F_{m-1}(x_i))),\ i = 1, \ldots, N$$

(b) 의사결정나무의 집합을 고려한다.
$\{R_{lm}\}_1^L = L$번째 종결노드에서의 의사결정나무 $tree(\{\tilde{y}_i, x_i\}_1^N)$

(c) 뉴튼 랩슨으로 근사한다.
$$\gamma_{lm} = \sum_{x_i \in R_{lm}} \tilde{y}_i / \sum_{x_i \in R_{lm}} |\tilde{y}_i| (2 - |\tilde{y}_i|),\ l = 1, \ldots, L$$

(d) 업데이트 한다.
$$F_m(x) = F_{m-1}(x) + \sum_{l=1}^{L} \gamma_{lm} \mathbf{1}(x \in R_{lm})$$

---

## 3. Support Vector Machine(SVM)

본 절의 내용은 Cristianini & Shawe-Taylor(1999)를 참고한 것이다.

### 3.1 선형 분리 가능한 경우

본 논문에서는 $l$개의 변수를 갖는 $n$개의 관측치로 구성된 두 집단의 판별문제를 고려한다. 훈련표본은 $\{X_1, X_2, \ldots, X_k, \ldots, X_n\}$, $X_i \in R^l$ 이고 이는 판별함수를 구축하는데 사용된다. 또한 이러한 변수들의 레이블은 $\{y_1, y_2, \ldots, y_k, \ldots, y_n\}$로, 알려져 있으며 -1 또는 +1 을 갖게 된다. 입력변수들의 가중합으로 표현되는 결정함수는 선형 판별 함수라고 불리며 그 형태는 다음과 같다.

$$f(x) = < w \bullet x > + b = \sum_{i=1}^{l} w_i x_i + b \tag{3.1}$$

여기서 $\langle x \cdot y \rangle$는 $x$와 $y$의 내적을 나타내는 기호이다.

초평면인 $f(x) = 0$은 +1의 레이블을 갖는 표본과 -1의 레이블을 갖는 표본을 분리한다. 즉, 초평면을 경계로 모든 표본이 정확히 분리될 수 있는 경우를 고려한 것이다. 서포트 벡터 기계의 목표는 +1에 속하는 표본과 -1에 속하는 표본 사이의 가장 큰 마진(margin)을 형성하는 초평면을 찾는 것이다. 여기에서 마진은 초평면과 가장 가까이 위치하는 관측치의 초평면까지의 거리로 정의된다.

$x$점으로부터 초평면까지의 부호(signed) 거리는 다음과 같다.

$$d(w, b) = \frac{| < w \bullet x > + b |}{| | w | |} \tag{3.2}$$

그렇다면 최적화 문제는 다음과 같이 표현해 볼 수 있다.

$$max_w C$$
$$s.t. \quad \frac{(< w \cdot x_i > + b) y_i}{| | w | |} \geq C \ , \ i = 1, 2, \ldots, n \tag{3.3}$$

또는 동등하게 다음과 같이 표현할 수 있다.

$$(< w \cdot x_i > + b) y_i \ \geq \ C \ | | w | | \ , \ i = 1, 2, \ldots, n$$

이 식이 성립되는 어떠한 $w$에 대해서 $w$의 곱의 형태 또한 식을 만족하므로 우리는

임의로 $||w|| = \frac{1}{C}$로 놓을 수 있다. 그러므로 식 (3.3)은 다음과 같다.

$$\min ||w||^2 \ s.t. \ y_i\{< w \bullet x > + b\} \geq 1, \ i = 1, 2, \ldots, n \tag{3.4}$$

## 3.2 선형 분리 가능하지 않은 경우

3.1절에서는 입력 자료가 선형식으로 완전하게 분리 가능한 경우를 다루었다. 그러나 일반적으로 이는 가능하지 않으며 몇몇 개의 자료점들은 초평면으로 분리될 수 없는 경우가 존재한다. 이러한 경우 최적화 문제를 풀기 위하여 slack 변수를 도입한다. slack 변수는 분리 가능한 경우의 제약을 완화시켜주는 역할을 하게 된다. 그러나 모든 slack 변수들이 큰 값을 갖는 평범한(trivial) 해를 얻지 않기 위해서 이것들의 합에도 제약을 주게 된다. 그렇다면 최적화 문제는 다음과 같다.

$$\begin{gathered} \min ||w||^2 \ s.t. \ y_i\{< w \bullet x > + b\} \geq 1 - \xi_i \\ \xi_i \geq 0, \ i = 1, 2, \ldots, n, \ \sum_i \xi_i \leq C \end{gathered} \tag{3.5}$$

여기서 $C$는 regularization 모수로써 모형의 복잡도를 결정한다. 일반적으로 $C$는 시험 자료나 교차확인(cross-validation)을 통하여 결정된다.

식 (3.5)는 선형 부등 제약식을 갖는 이차형태로 볼록함수의 최적 문제이다. 이 문제를 풀기 위한 라그랑지 함수는 다음과 같다.

$$\begin{aligned} L(w, b, \xi, \alpha, \gamma) = \frac{1}{2} < w \bullet w > + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i < w \bullet x > + b - 1 + \xi_i) \\ - \sum_{i=1}^{n}\gamma_i\xi_i \ \ s.t. \ \ \alpha_i \geq 0, \ \gamma_i \geq 0 \end{aligned} \tag{3.6}$$

식 (3.6)은 $\alpha, \gamma$ 에 대해서 최대여야 하며, $w, b, \xi$에 대해서는 최소여야 한다. 우선, $w, b, \xi$에 관해 미분한 것을 0으로 놓으면 다음과 같은 식을 얻을 수 있다.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.7}$$

$$0 = \sum_{i=1}^{n} \alpha_i y_i \tag{3.8}$$

$$\alpha_i = C - \gamma_i \tag{3.9}$$

식 (3.7), (3.8) 그리고 (3.9)를 식 (3.6)에 대입하면 다음과 같은 라그랑지 쌍대(dual) 목적함수를 얻는다.

$$\min_{\alpha}(-\sum_{i=1}^{n}\alpha_i+\frac{1}{2}\sum_{i=1}^{n}y_iy_j\alpha_i\alpha_j<x_i\bullet x_j>)\ s.t.\ \sum_{i=1}^{n}y_i\alpha_i=0\ ,\ 0\le\alpha_i\le C \quad (3.10)$$

또한 Karush-Kuhn-Tucker(KKT) 조건은 다음의 제약을 포함한다.

$$\alpha_i[\,y_i(\,x_i^Tw+b\,)-(1-\xi_i)\,]=0 \quad (3.11)$$

$$\gamma_i\xi_i=0 \quad (3.12)$$

$$y_i(\,x_i^Tw+b\,)-(1-\xi_i)\ge 0 \quad (3.13)$$

식 (3.11), (3.12) 그리고 (3.13)의 제약은 원(primal) 문제와 쌍대 문제에서의 해를 유일하게 결정하게 된다. 해는 다음과 같다.

$$\hat{w}=\sum_{i=1}^{n}\hat{\alpha}_iy_ix_i \quad (3.14)$$

이 때 $\alpha_i>0$인 표본 $x_i$는 서포트 벡터(Support Vector)라 불린다. 식 (3.13)를 결정함수 $G_w(x)$로 고려하면, 판별되어야 하는 자료와 서포트 벡터 사이의 내적으로 표현되는 (3.15)식을 얻을 수 있다.

$$G(x)=sign(\sum_{i=1}^{n}\alpha_iy_ix^Tx_i+b) \quad (3.15)$$

## 3.3 비선형 서포트 벡터 분류기

지금까지 서포트 벡터 분류기는 입력 공간 내에서 선형식을 찾는 것으로 기술되었다. 더 일반적인 경우를 고려하기 위해 커널을 사용하여 입력자료를 차원이 높은 비선형 입력 공간으로 변형할 것이다. 그 후에 선형식을 사용하여 분리할 수 있다. 일반적으로 차원이 넓혀진 공간에서의 선형 분리는 더 좋은 결과를 나타내며 원래의 공간에서는 비선형적인 경계선으로 다시 재표현이 가능하게 된다.

우리는 최적화 문제(3.6)과 그것의 해를 입력자료의 내적만을 포함하는 식으로 표현할 수 있다. 비선형 서포트 벡터 분류기에서의 라그랑지 쌍대 목적함수는 다음과 같은 형태를 갖는다.

$$\min_{\alpha}(-\sum_{i=1}^{n}\alpha_i+\frac{1}{2}\sum_{i=1}^{n}y_iy_j\alpha_i\alpha_j<h(x_i)\bullet h(x_j)>) \quad (3.16)$$

그리고 결정함수의 형태는 다음과 같다.

$$G(x)=sign\left(\sum_{i=1}^{n}\alpha_iy_i<h(x)\cdot h(x_i)>+b\right) \quad (3.17)$$

즉, 식 (3.16)과 (3.17)은 고차원 공간에서의 내적을 계산하면 바로 구할 수 있다. 이

러한 복잡한 계산은 양정치 커널 함수 $k$를 사용함으로써 쉽게 구할 수 있게 된다. 즉,

$$< h(x) \cdot h(x_i) >= k(x, x_i) \tag{3.18}$$

이제, 최적화 문제는 다음과 같다.

$$\min_{\alpha}(- \sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j)) \tag{3.19}$$

여기에서 $k(x, x')$은 입력 공간으로의 비선형 근사를 수행하는 커널 함수이고 제약식은 변하지 않는다.

$$\sum_{i=1}^{n} y_i \alpha_i = 0\ ,\ 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n \tag{3.20}$$

제약을 갖는 식 (3.19)는 라그랑지 승수를 결정하고, 입력 공간상의 초평면을 최적으로 분리하는 결정함수는 다음과 같이 주어진다.

$$G(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + b\right) \tag{3.21}$$

결론적으로, 적절한 커널함수를 사용하여 선형식일 경우 사용했던 모든 것은 비선형인 경우에 적용될 수 있게 된다. 다양한 커널 함수를 사용하면 서포트 벡터 알고리즘은 다양한 learning machine을 구축할 수 있게 된다.

SVM에 많이 쓰여지는 커널 함수로는 다음의 3가지가 있다.

**Polynomial kernel**

$$k(x,\ x') = (1 + < x \cdot x' >^{d})$$

**Radial Basis kernel**

$$k(x,\ x') = \left(-\frac{||x - x'||^2}{\sigma^2}\right)$$

**Sigmoid kernel**

$$k(x, x') = tanh(1 + < x, x' >)$$

# 4. 실험

## 4.1 자료 및 실험방법

본 절에서는 boosting과 SVM의 예측력을 비교하기 위하여 <표 4-1>의 13개의 자료를 분석하였다. 위의 모든 자료들은 ftp://ftp.ics.uci.edu/pub/machine-learning-databases에서 얻은 것이며 훈련 자료와 시험 자료로 구분되어 있는 경우에도 이를 합한 후에 분석하였다. Boosting 알고리즘에서는 약한 분류기의 복잡도 (즉, 의사결정나무의 크기)와 반복회수가 예측모형의 복잡도를 결정하는 regularization 모수이고, SVM에서는 slack 변수의 제약식 (3.5)에 사용된 $C$와 kernel 함수에 사용되는 모수가 regularization 모수이다. 본 실험에서는 radial basis kernel을 사용하였으며, 이 경우에는 $\sigma^2$가 regularization모수가 된다. 본 실험에서는 이러한 regularization 모수를 5-fold 교차확인을 통하여 선택하였다.

예측오차를 구하기 위하여 다음의 과정을 사용하였다. 먼저, 자료들을 훈련자료와 시험자료로 7 대 3의 비율로 하여 임의로 나누었다. 그리고, 각 훈련자료에 대하여 5-fold 교차확인을 통하여 boosting과 SVM의 regularization 모수를 선택하였으며, 시험자료를 통하여 예측오차를 계산하였다. 이러한 과정을 100번 반복하여 100개의 예측오차를 구한 후 이를 평균을 구하여 최종 예측오차를 구하였다.

<표 4-1> 자료

| ID | 자료 | 관측치 수 | 변수의 수 | 집단 |
|---|---|---|---|---|
| 1 | Sonar | 208 | 60 | 2 |
| 2 | Vowel | 990 | 12 | 11 |
| 3 | Breast Cancer | 699 | 9 | 2 |
| 4 | Glass | 214 | 9 | 7 |
| 5 | Segmentation | 2310 | 19 | 7 |
| 6 | Pima-Indian-Diabetes | 768 | 8 | 2 |
| 7 | Ionosphere | 351 | 34 | 2 |
| 8 | German | 1000 | 24 | 2 |
| 9 | Vehicle | 846 | 18 | 4 |
| 10 | cmc | 1473 | 9 | 3 |
| 11 | Boston housing | 506 | 13 | 3 |
| 12 | LED | 500 | 7 | 10 |
| 13 | Bupa | 345 | 6 | 2 |

## 4-2. 예측오차

<표 4-2>에 13개의 자료의 예측오차를 정리하였다. 13개의 자료 중 7개의 자료에서는 boosting의 예측력이 좋았으며, 나머지 6개의 자료에서는 SVM의 예측력이 좋았다. 특히, 예측오차가 작은 자료인 Vowel, Ionoshpere에서 SVM의 예측오차가 boosting의 예측오차에 비하여 50%이상이 작았으며, 예측오차가 큰 자료 Glass, LED 등에서는 boosting의 예측력이 많이 우수하였다. 결론적으로, SVM은 잡음이 작은 자료에서 잘 작동하며 boosting은 잡음이 큰 자료에서 잘 작동한다는 것을 알 수 있었다. 이미지 분석이나 음성인식같이 잡음이 작은 경우에는 SVM을 추천할 만 하며, medical 자료 등 잡음이 많은 자료에는 boosting이 더 좋은 학습방법이라고 결론지을 수 있다.

<표 4-2> 예측오차

| ID | 자료 | Boosting<br>오분류율 평균(표준오차) | SVM<br>오분류율 평균 (표준오차) |
|---|---|---|---|
| 1 | Sonar | 0.1489(0.0437) | 0.1327(0.0388) |
| 2 | Vowel | 0.0975(0.0395) | 0.0113(0.0071) |
| 3 | Breast Cancer | 0.0359(0.0092) | 0.0364(0.0117) |
| 4 | Glass | 0.2518(0.0521) | 0.3062(0.0465) |
| 5 | Segmentation | 0.0478(0.0077) | 0.0591(0.0072) |
| 6 | Pima-Indian-Diabetes | 0.2363(0.0218) | 0.2287(0.0221) |
| 7 | Ionosphere | 0.0922(0.0258) | 0.0491(0.0180) |
| 8 | German | 0.2326(0.0182) | 0.2385(0.0176) |
| 9 | Vehicle | 0.2343(0.0225) | 0.1711(0.0210) |
| 10 | cmc | 0.4405(0.0188) | 0.4714(0.0211) |
| 11 | Boston housing | 0.2175(0.0284) | 0.2106(0.0294) |
| 12 | LED | 0.2630(0.0326) | 0.2930(0.0282) |
| 13 | Bupa | 0.2744(0.0353) | 0.2917(0.0349) |

잡음에 대한 학습방법의 성능의 차이는 매우 특이한 발견이다. 그 이유는 boosting과 SVM 모두 regularization 모수가 있어서 예측모형의 복잡도를 조절할 수 있기 때문이다. 본 논문의 실험결과는 regularization 모수만으로는 자료의 잡음에 대하여 완전하게 반응할 수 없으며, 학습방법이 잡음의 정도에 따라 바뀌어야 한다는 것을 보여준다. 이러한 결과는 새로운 것으로써 앞으로 좀더 심도 있는 연구가 필요한 것으로 사료된다.

## 5. 맺음말

본 논문에서는 boosting과 SVM의 예측력을 실제 자료를 통하여 비교하였다. 두 학습방법의 예측력은 자료에 따라 다르게 나왔는데, 특히 잡음이 많은 자료인 경우에는 boosting알고리즘이 예측력이 좋았으며, 잡음이 작은 경우에는 SVM의 예측력이 우수하였다. 잡음의 양 뿐 아니라 자료의 크기, 입력변수의 수 등을 이용하여 두 알고리즘을 비교하였으나, 본 논문의 실험결과로는 큰 차이를 발견하지 못하였다. 이러한 방향으로 좀더 심도 있는 연구가 필요하다.

계산속도면에서는 SVM이 boosting보다 좋았다. 특히, 변수의 개수가 많고 자료의 수가 적은 경우에 SVM의 계산속도가 빨랐는데, 이는 SVM 알고리즘의 계산량이 변수의 개수에는 의존하지 않으며 관측치의 수에만 의존하기 때문일 것으로 사료된다.

## 참고문헌

1. Cristianini, N. & Shawe-Taylor, J. (1999). *An Introduction to Support Vector Machines*, Cambridge University Press.
2. Freund, Y. & Schapire, E. R. (1997). A decision theoretic generalization of on-line learning and an application to Boosting, *Journal of Computer and System Sciences*, 55, 119-139.
3. Friedman, J. (2001). Greedy function approximation : a gradient boosting machine, *The annals of statistic,* 29(5), 1189 - 1232.
4. Friedman, J., Hastie, T. & Tibshirani, R. (2000). Additive logistic regression : A statistical view of boosting (with discussion and a rejoinder by the authors), *The annals of statistic,* 28(2), 337-407.
5. Jiang, W. (2002). On weak base hypotheses and their implications for boosting regression and classification, *The annals of statistic,* 30, 51-73.
6. Mason, L., Baxter, J., Bartlett, P. & Frean, M. (2000). Functional gradient techniques for combining hypotheses. In A.J. Smola, P.L. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, 221-247. MIT Press.
7. Schapire, R., Freund. Y., Bartlett. P. & Lee. W. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods, *The annals of statistic,* 26(5), 1651 - 1686.
8. Schapire, R. & Singer, Y. (1999). Improved boosting algorithms using confidence -rated predictions, *Machine Learning*, 37(3), 297-336.
9. Vapnik, V. (1998). *Statistical Learning Theory,* Wiley.

[ 2005년 10월 접수, 2005년 11월 채택 ]