이 성 건 교수지도

석사학위청구논문

향상된 배깅 알고리즘에 관한 연구

2011

성신여자대학교 대학원

통 계 학 과

유 정 윤

향상된 배깅 알고리즘에 관한 연구

이 성 건 교수지도

이 논문을 석사학위논문으로 제출함

2010년 11월

성신여자대학교 대학원

통 계 학 과

유 정 윤

인 준 서

유정윤의 석사학위 논문으로 인준함.

심사위원 _____ 인

심사위원 _____ 인

심사위원 _____ 인

성신여자대학교 대학원

논 문 개 요

모든 분류기법의 목적은 주어진 데이터를 이용하여 목표변수를 가장 잘 예측할 수 있는 분류자를 형성하는 것이다. 예측력을 평가하는데 있어서 단일 분류자(single classifier)보다 앙상블(ensemble)방법을 이용하여 결합한 다중 분류자(multiple classifiers)가 더 우수한 수행능력을 보여주고 있다는 것은 경험적으로 증명되어 왔다. 그 중 배깅(bagging)은 단일분류자보다 예측력과 안정성이 뛰어나고 다른 다중 분류자에 비해 그 이론을 직관적으로 이해하기 쉬우며, 모형을 생성하는 과정이 쉽기 때문에 널리 사용되고 있다.

앙상블 기법을 이용하여 예측 알고리즘을 사용하는 가장 큰 이유는 변동성이 작으며 예측력이 뛰어난 알고리즘을 구축하기 위함이다. 그러나 대표적인 배깅 알고리즘들은 여전히 자료의 변동에 따라 예측의 정확도가 떨어지고 변동성이 커 다양한 형태의 자료에 대한 분류를 수행하는데 한계가 있다.

본 논문에서는 단일 분류방법론들을 결합하는 새로운 방법을 제안함으로써 예측의 정확성을 높이고 변동성을 줄이는 향상된 배깅 방법에 대한 연구를 하고자 한다. 기존의 배깅 알고리즘과 본 논문에서 제안한 방법을 실제 자료에 적용시켜 향상된 정도를 파악함으로써 배깅 알고리즘이 보다 개선되었는지 여부를 비교 연구한다.

목 차

제 1장 서 론

분류와 예측기법에서 주어진 데이터를 이용하여 목표변수를 가장 잘 예측할 수 있는 분류자(classifier)를 형성하는 것은 중요하다. 예측오류를 감소시키기 위하여 단일 분류자를 이용하는 것보다 앙상블(ensemble) 기법에 의해 얻어진 다중 분류자를 이용할 때 더 정확하고 일관적인 예측 결과가 나온다는 것이 증명된 바 있다 (Breiman, 1996).

배깅(bagging) 방법론은 앙상블 기법을 이용하여 분류자를 형성하는 대표적인 방법론 중 하나이다. 이는 분석용 데이터(training data)로부터 붓스트랩 표본(bootstrap sample) 기법으로 얻어진 여러 개의 데이터를 이용하여 각각의 분류자를 형성한 후, 이들 분류자의 분류규칙을 결합하여 최종의 분류자를 형성하는 방법이다.

배깅 방법론은 의사결정나무(decision tree)와 같이 안정적이지 않은(unstable) 분류자에서 예측결과의 정확도를 높이는데 탁월한 성능을 발휘한다. 하지만 SVM(support vector machines)과 같이 비교적 안정적인 분류자에서는 예측능력이 향상되지 않거나 단일 분류자를 사용할 때보다 예측력이 더 낮아지는 경우가 발생한다. 이와 같은 단점을 보완하기 위하여 브래깅(bragging), 나이스 배깅(nice bagging), 절사 배깅(trimmed bagging) 등의 다양한 배깅 기법이 논의되어왔다. 새롭게 제안된 배깅 방법들은 기존의 배깅 보다 향상된 결과를 가져왔지만 기본 분류자(base classifier)에 따라서는 여전히 예측 정확도가 떨어지는 문제점을 가지고 있다. 본 논문에서는 기존에 논의되었던 배깅 방법들보다 예측력이 뛰어나며, 안정적인 분류자를 이용하여 결합하는 경우에도 예측력이 떨어지지 않는 배깅 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 의사결정나무, 판별분석, 로지스

틱 회귀(logistic regression), SVM과 같은 여러 가지 단일 분류 방법론과 앙상블 기법을 이용한 배깅 방법론의 이론을 소개하고, 각각의 방법론에 수반되는 장, 단점을 살펴본다. 3장에서는 본 논문에서 제안하는 가중 배깅 방법론을 소개한 후, 4장에서 기존의 배깅 방법론과 본 논문에서 제안한 배깅 방법론을 실제자료에 적용 및 모의실험 하여 결과를 살펴본다. 특히 각각의 배깅 기법을 적용하여 얻어진 예측력이 단일 분류자를 적용하여 얻어진 예측력에 비해 어느 정도 향상됐는지 그 정도를 파악함으로써 각 기법의 수행능력을 비교 해 본다. 5장에서는 본 연구를 정리하고 향후 연구방향을 제시한다.

제2장 분류방법론

목표변수를 분류 및 예측하는 대표적인 알고리즘으로 의사결정나무와 선형판별분석, SVM, 로지스틱 회귀분석 등이 있다. 앙상블 기법을 이용하여 분류모형을 생성하는 대표적인 알고리즘으로는 배깅 알고리즘이 있다. 전자를 단일 분류 방법론이라 하고, 후자를 결합 분류 방법론이라 한다. 본 장에서는 각각의 단일분류방법론과 배깅 방법론의 개념을 파악하고 각 분류방법론의 장, 단점을 살펴보는데 초점을 맞추고자 한다.

2.1.단일 분류방법론

2.1.1 의사결정나무

의사결정나무는 의사결정규칙(decision rule)을 나무구조로 도표화하여 관심대상이 되는 집단을 몇 개의 소집단으로 분류(classification)하거나 예측(prediction)을 수행하는 분석방법이다. 의사결정나무는 종속변인에 가장 큰 영향을 주는 독립변인의 특정값을 기준으로 표본 집단에 대한 최초 분리가 이루어지며 순차적으로 더 이상의 분리가 이루어지지 않을 때까지 분리를 수행한다. 이는 하나의 나무구조를 이루고 있으며, 나무구조가 시작되는 뿌리마디(root node), 하나의 마디로부터 분리되어 나간 두 개 이상의 마디들인 자식마디(child node), 자식마디의 상위마디인 부모마디(parent node), 각 나무줄기의 끝에 위치하고 있는 끝마디(terminal node) 등 여러 가지의 마디

라고 불리는 구성요소들로 이루어져 있다.

의사결정나무의 형성 단계에서 부모마디로부터 자식마디들이 형성될 때 입력변수(input variable)의 선택과 범주(category)의 병합이 이루어질 분리기준을 설정해야 한다. 이때 분리 기준과 그에 따르는 알고리즘으로는 지니지수(Gini index)에 의해 분리여부를 결정하는 CART(Classification And Regression Tree) 알고리즘, 엔트로피지수(Entropy index)에 의해 분리되는 C4.5 알고리즘, ANOVA F-검정 또는 Levene, Pearson 카이제곱 검정을 사용하여 가장 작은 유의확률에 대응되는 변수를 분리변인으로 선택하는 QUEST 알고리즘 등이 있다.

의사결정나무는 분류 또는 예측의 과정이 나무구조에 의한 추론규칙(induction rule)에 의해서 표현되기 때문에 연구자가 그 과정을 쉽게 이해하고 설명할 수 있다는 장점을 가지고 있으며, 변인들 간의 상호작용효과(interaction effects)를 파악할 수 있다. 또한 모수적인 방법과는 달리 선형성(linearity)이나 정규성(nomality) 또는 등분산성(equal variance) 등의 가정을 필요로 하지 않으며, 순서형 또는 연속형 변수에 대해 단지 순위(rank)만이 분석에 영향을 주기 때문에 이상치(outlier)에도 민감하지 않고 변수에 대한 결측값도 하나의 범주로 간주하여 이를 모형화 과정에서 처리할 수 있다는 장점을 가지고 있다. 반면에 의사결정나무는 연속형 변수(continuous variable)를 비연속적인 값으로 취급하기 때문에 분리의 경계점 부근에서는 오류가 발생할 확률이 높다 (강현철 외, 2006).

2.1.2 SVM

SVM(Support Vector Machine)은 기본적으로 두 범주를 갖는 관측값들을 분류하는 방법이다. SVM의 목적은 주어진 많은 데이터들을 가능한 멀리

두 개의 집단으로 분리시키는 최적의 초평면을 찾는 것이다. SVM이 다른 알고리즘과 차별화되는 특징은 단지 점들을 분리하는 초평면을 찾는 것으로 끝나는 것이 아니라 점들을 분리할 수 있는 수많은 후보평면들 가운데 마진(margin)이 최대가 되는(maximum-margin) 제약조건을 둔다는 것이다. 여기에서, 마진이란 초평면으로부터 각 점들에 이르는 거리의 최소값을 의미하는데, 이러한 제약조건을 둠으로써 SVM의 초평면이 유일하게 정해지도록 한다. 마진을 최대로 하면서 점들을 두 집단으로 분류하려면 결국 한 집단에 속하는 점들과의 거리 중 최소값과 다른 한 집단에 속하는 점들과의 거리 중 최소값이 같도록 초평면이 위치해야 하며, 이러한 초평면을 최대마진초평면(maximum margin hyperplane)이라고 한다.

입력 공간에서의 데이터들이 선형적으로 분리가 되지 않는 경우, 입력공간에서의 데이터를 비선형변환을 통해 보다 높은 차원의 공간에서 데이터를 선형적으로 분리가능하게 만들 수 있다. 그러나 특징공간은 매우 높은 차원의 공간이므로 내적 커널(inner-kernel)을 이용하여 특징공간에서 직접 연산을 수행하지 않고 입력공간에서 데이터를 처리할 수 있다. 일반적으로 커널함수의 타입에 따라 SVM은 다항식분류기(polynomial classifier), 방사기저함수(radial basis: RBF) 분류기 등으로 나눌 수 있다.

SVM은 실제 우리 주위에 존재하는 비선형분류라는 현실적인 문제들에서는 효과적인 성능을 낼 수 없다는 한계가 존재했다. 하지만 데이터가 선형분리가 불가능한 경우 커널함수를 사용함으로써 SVM이 비선형문제들에 대해서도 효력을 발휘하기 시작하였고 다양한 분야에서 적극적으로 사용되게 되었다. 커널을 사용하는 것은 우리가 실제로 데이터를 배치하는 입력공간(input space)에서는 잘 나누어지기 힘든 비선형문제를 고차원의 공간으로 이동시켜서 이 새로운 공간에서 SVM의 선형판별을 수행함으로써 마치 처음의 입력공간에서 매우 복잡한 비선형 판별문제를 해결한 것과 같은 효과를 얻는 것을 가리킨다. SVM 분류자는 최적의 분류평면을 형성하는데 있

어서 알고리즘이 간단하다. 또한 입력공간의 비선형적인 높은 차수를 특정 공간에 선형적으로 투영하여 해석할 수 있도록 한다. 그러나 이러한 SVM은 결측치가 존재하는 경우 이의 대체에 대한 방법이 미흡하다는 단점이 존재한다.

2.1.3 선형판별분석

선형판별분석(Linear Discriminant Analysis: LDA)은 데이터 분류와 차원 축소를 위하여 널리 알려진 기술 중의 하나이다. 선형판별분석은 집단 간 데이터 분산을 나타내는 행렬(between-class scatter matrix)과 집단 내 데이터 분산을 나타내는 행렬(within-classs scatter matrix)의 비율을 최대화하는 초평면(hyperplane)을 찾는 알고리즘이다. 선형판별분석은 다음과 같은 과정을 통하여 이루어진다.

n차원 벡터 공간의 열벡터 X_{ij}를 $m(m < n)$차원 특징 공간으로 매핑(mapping)하는 선형변환에 의해 생성된 새로운 특징 벡터 Y_{ij}는 식(2.1)과 같이 정의된다.

$$Y_{ij} = W^T X_{ij} \tag{2.1}$$

여기서 X_{ij}와 Y_{ij}는 W에 의한 선형 변환 이전 및 이후의 i번째 집단에 속한 j번째 열벡터를 나타낸다. 집단 간의 분산행렬을 나타내는 식은 식(2.2)와 같다.

$$S_b = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T \tag{2.2}$$

집단 내의 분산행렬을 나타내는 식은 식(2.3)와 같다.

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{N_i} (X_{ij} - \mu_i)(X_{ij} - \mu_i)^T \tag{2.3}$$

여기서 μ_i는 i번째 집단의 평균을, μ는 전체평균을 C는 클래스의 수를, N_i는 i번째 데이터의 수를 나타낸다. 집단 간 분산 행렬의 결정과 집단 내의 분산 행렬의 비를 최대화 하는 최적의 선형변환행렬은 식(2.4)를 통하여 계산하게 된다.

$$W_{opt} = \frac{\| W^T S_W W \|}{\| W^T S_b W \|} \tag{2.4}$$

새로운 데이터를 입력하였을 때, 각 집단에서의 평균과의 거리를 계산한 후 거리가 가장 작은 집단에 속하게 된다.

LDA는 집단간의 분리를 최대화시켜 주므로 집단 간의 특징 벡터들을 비교적 정확하게 분류하는 장점이 있다. 또한 선형함수의 형태로 표현되므로 그 결과를 이해하고 실제 문제에 적용하는 것이 매우 편리하다. 그러나 선형판별함수는 부분모집단의 공분산행렬들이 같다는 가정 하에 유도된 것이므로 이러한 가정이 적절하지 않은 자료에 대해서는 상당히 왜곡된 결과가 도출될 수 있다. 또한 LDA는 집단의 중심(평균)이 그 집단의 대부분을 가지고 있다는 가정에서 출발하고 있으므로 비선형적으로 이루어진 데이터와 같이 동일한 평균을 가진 데이터에는 부적합한 한계를 지니고 있다. 따라서 분류문제보다는 차원축소의 한 접근법으로 많이 사용된다.

2.1.4 로지스틱 회귀 분석

목표변수 Y가 2개의 가능한 값(0, 1)을 갖는 이항 반응이고 설면변수 X가 있을 때 목표변수 Y의 평균은 Y=1의 값을 가질 확률과 같다. 성공 또는 불량 등의 확률 P_x에 대해 선형 확률모형(linear probability model)은 $P=\alpha+\beta x$과 같이 나타낼 수 있다. 그런데 β와 x가 취할 수 있는 값에 대하여 제한이 없으므로 선형 확률모형 역시 $-\infty$에서 $+\infty$ 사이의 어떠한 값도 취할 수 있다. 하지만 이러한 모형은 확률의 값이 0과 1사이의 값을 가져야 하므로 구조적으로 결함을 갖고 있다. 따라서 이러한 식 대신 관심의 확률 Px에 대해 설명변수 X와 비선형 식(2.5)를 사용한다.

$$P_x = \frac{\exp(\beta_0+\beta x)}{1+\exp(\beta_0+\beta x)} \tag{2.5}$$

k개의 설명변수가 있는 경우에 관심확률 P_x는 식(2.6)과 같이 모형화되며 로짓모형은 식 (2.7)의 식과 같다.

$$P_x = \frac{\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k)}{1+\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k)} \tag{2.6}$$

$$logit(p_x) = \ln\frac{P_x}{1-P_x} = \beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k \tag{2.7}$$

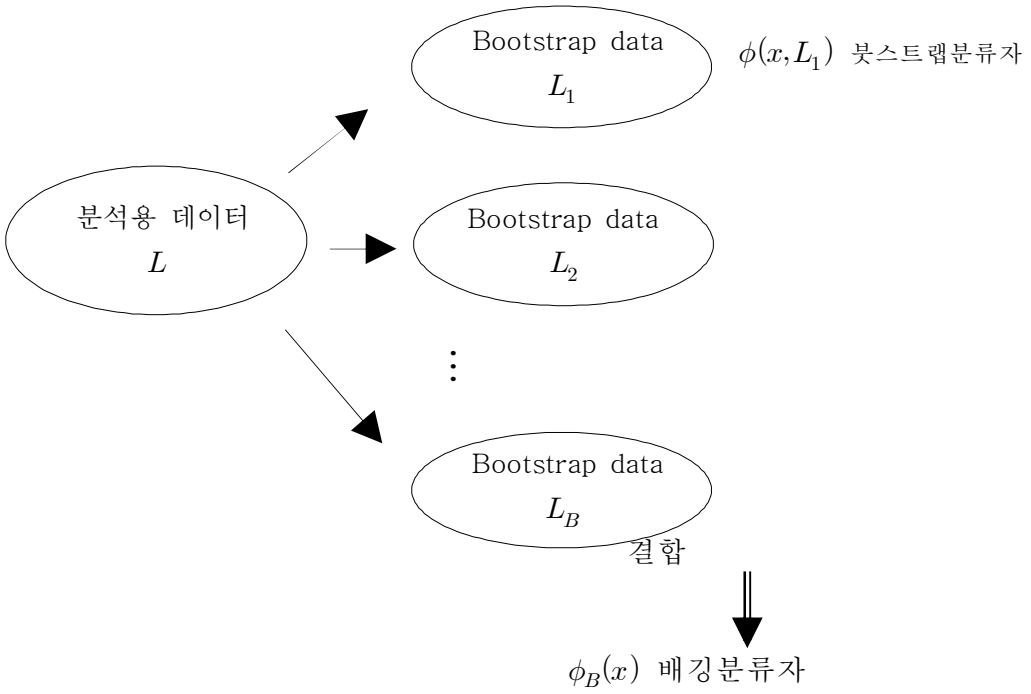
즉, 목표변수가 범주형일 때, 선형회귀모형의 단점을 극복하기 위하여 확률에 대한 로짓변환(logit transformation)을 고려하여 모형 식의 좌변과 우변이 모두 실수 상의 값을 가지도록 한 후 분석하는 것이다.

로지스틱 회귀분석은 종속변인과 독립변인들의 관계를 비선형적으로 표현하기 때문에 신용평가 문제에서와 같은 현상들의 예측이나 분석에 적합한 분석방법이다. 또한 독립변인들의 연속성 여부에 상관없이 종속변수가 질적 변수인 경우에 사용될 수 있다는 장점을 가지고 있다.

2.2 결합 분류방법론

2.2.1 배깅 방법론

데이터가 불안정한 경우 즉, 데이터가 조금이라도 바뀐 상태에서 분류자의 변동성이 큰 경우에 이의 변동성을 감소시키고자 앙상블 기법을 이용하여 분류자를 형성할 수 있다. 배깅(bagging : **B**ootstrap **Agg**regat**ing**) 알고리즘은 붓스트랩방법을 이용한 앙상블 기법으로 Breiman(1996)에 의해 처음 소개되었다.



[그림 2.1] 배깅 알고리즘

y가 이산형이거나 연속형인 모집단에서 추출한 분석용 데이터를 L=

$\{(y_n, x_n), n=1, \ldots, N\}$이라고 하자. 이 분석용 데이터 L에서 복원단순임의추출에 의해 관측값이 N개인 B개의 붓스트랩 분석용 데이터$\{L_k,\ k=1,\ldots,B\}$를 생성한다. 분석용 데이터 집합에서 각각의 단일 분류자를 형성하여 단일 분류자 집합 $\{\varphi(x, L_k)\}$을 얻는다. 앞에서 얻어진 단일 분류자를 결합하는 방법으로 목표변수가 연속형일 때 평균, 범주형일 때는 다중 투표(majority vote)를 사용한다. 이렇게 결합되어 형성된 분류자를 배깅 분류자라 하고 식(2.12)와 같이 나타낸다 (Breiman,1996).

$$\varphi_B = \underset{1<b<B}{average}\ \varphi(x, L_b) \tag{2.8}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자

목표변수가 연속형일 때와 범주형일 때의 알고리즘은 다음과 같다.

① 연속형 자료의 목표변수

목표변수 y가 연속형일 때, 단일 분류자 φ를 모든 L에 대해서 평균한 예측 오차(average prediction error)는 식(2.9)과 같이 표현 될 수 있다.

$$e = E_L E_{Y,X}(Y - \varphi(X, L))^2 \tag{2.9}$$

결합 분류자 φ_A의 오차는 식(2.10)와 같이 표현될 수 있다.

$$e_A = E_{Y,X}(Y-\varphi_A(X,P))^2 \tag{2.10}$$

여기에서 $\varphi_A(X,P) = E_L\varphi(X,L)$ 이다.

식(2.9)을 전개하면 식(2.11)과 같고, 식(2.10)를 전개하면 식(2.12)와 같다.

$$e = E_L(E_{Y,X}Y^2) - 2E_{Y,X}(E_L Y\varphi(X,L)) + E_{Y,X}(E_L\varphi^2(X,L)) \tag{2.11}$$

$$e_A = E_{Y,X}Y^2 - 2E_{Y,X}Y\varphi_A(X,P) + E_{Y,X}\varphi_A^2(X,P) \tag{2.12}$$

검증용 데이터 (X,Y)와 분석용 데이터 L 은 서로 독립이므로 식(2.11)의 $E_L(E_{Y,X}Y^2)$과 식(2.12)의 $E_{Y,X}Y^2$은 서로 같고 , 식(2.11)의 $E_{Y,X}(E_L Y\varphi(X,L))$과 식(2.12)의 $E_{Y,X}Y\varphi_A(X,P)$도 서로 같다. 따라서 식(2.11)과 식(2.12)를 Jensen의 부등식 $(E(Z))^2 \le E(Z^2)$ 을 이용하여 정리하면 식(2.11)의 단일 분류자를 모든 L에 대해서 평균한 예측오차는 식(2.12)의 결합분류자의 오차보다 크다는 것을 알 수 있다. 그리고 주어진 x에 대해서 식(2.11)와 식(2.12)를 비교하면 식(2.13)와 같이 나타낼 수 있다.

$$E_L\varphi^2(x,L) \ge [E_L\varphi(x,L)]^2 \tag{2.13}$$

배깅 과정을 통해서 식(2.13)의 차이만큼 변동성이 감소했음을 알 수 있으며 단일 분류자를 모든 L에 대해서 평균한 예측오차는 결합분류자의 오차보다 낮은 오차를 가짐을 알 수 있다. 만약, 단일 분류자 $\varphi(x,L)$ 가 많이 변하지 않으면 식(2.13)의 양변은 거의 같을 것이고, 단일 분류자 $\varphi(x,L)$ 가 더 크게 변할수록 더 향상된 결합 분류자가 만들어지므로 결합 분류자 φ_A는

항상 단일 분류자 φ보다 좋은 분류자를 형성한다.

② 범주형 자료의 목표변수

분석용 데이터에서 목표 변수 y가 범주일 때, 단일 분류자 $\varphi(x,L)$는 범주 $y\in\{1,...,J\}$을 갖는다 하자. 분석용 데이터 L이 고정되어 있는 상태에서 정분류율(correct classification rate)은 식(2.14)과 같이 나타낼 수 있다.

$$\begin{aligned} r(L) &= P[Y=\varphi(X,L)] \\ &= \sum_j P[\varphi(X,L)=j \mid Y=j]\, P[Y=j] \end{aligned} \tag{2.14}$$

식(2.14)을 모든 L에 대해서 평균한 정분류율은 식(2.15)로 표현할 수 있다.

$$\begin{aligned} r &= \int_L r(L)dP_L \\ &= \sum_j \int_L P_L[\varphi(X,L)=j \mid Y=j]\, P[Y=j]dP_L \end{aligned} \tag{2.15}$$

모든 L에 대해서 각 분류자와 범주가 같을 확률을 $Q(j|x)$ 라 정의하고, 즉 $Q(j|x)=P_L[\varphi(x,L)=j]$ L 과 j 와 독립이므로 정분류율은 식(2.16)과 같이 표현 될 수 있다.

$$r=\sum_j E[Q(j|X) \mid Y=j]P[Y=j]$$

$$= \sum_j \int Q(j|x) P(j|x) P_X(dx) \tag{2.16}$$

총계를 통해서 얻어진 정분류율은 $\text{argmax}_i Q(i|x)$와 j 가 같은 범주에 대해서만 식(2.16)에 적용하면 식(2.17)으로 표현될 수 있다.

$$r_A = \sum_j \int I(\arg\max_i Q(i|x) = j) P(j|x) P_X(dx) \tag{2.17}$$

여기에서 $I(\bullet)$ 는 지시(indicator) 함수이다.

$$C = \{x | \text{argmax}_j P(j|x) = \text{argmax}_j Q(j|x)\} \tag{2.18}$$

식(2.18)을 만족하는 x들에 대해서 식(2.17)의 부분은 식(2.19)과 같이 나타낼 수 있다.

$$\sum_j I(\text{argmax}_i Q(i|x) = j) P(j|x) = \max_j P(j|x) \tag{2.19}$$

총계를 통해 얻어진 정분류율은 식(2.19)을 식(2.17)에 적용하고, $x \in C$ 가 아닌 x 에 대해서 C' 이라 하면 식(2.20)와 같이 나타낼 수 있다.

$$r_A = \sum_j \int I(\arg\max_i Q(i|x) = j) P(j|x) P_X(dx)$$

$$= \int_{x \in C} \sum_j I(\arg\max_i Q(i|x) = j) P(j|x) P_X(dx)$$

$$+ \int_{x \in C'} \sum_j I(\arg\max_i Q(i|x) = j) P(j|x) P_X(dx)$$

$$= \int_{x \in C} \max_j P(j|x) P_X(dx) + \int_{x \in C'} \sum I(\varphi_A(x) = j) P(j|x) P_X(dx) \qquad (2.20)$$

만약 $x \in C$ 이면, $\sum_j (i|x) P(j|x)$ 은 $\max_j P(j|x)$보다 작을 수 있다. 따라서 $P_X(C) \simeq 1$ 일 때, 단일 분류자 φ는 최적과 거리가 멀지만 결합 분류자 φ_A는 거의 최적이다. 배깅 과정을 통해서 얻어진 결합 분류자는 식(2.21)와 같이 배깅 분류자로 표현될 수 있다.

$$\varphi_B(x) = \varphi_A(x, P_L) \qquad (2.21)$$

여기에서 P_L은 각 관측값 $(y_n, x_n) \in L$가 $1/N$ 의 확률을 가지는 분포를 하며 확률분포 P 에 붓스트랩 근사한다.

배깅은 개념적으로 간단하고 직관적이며 몇몇 단일 방법론을 결합하는 경우에 성공적인 면모를 보인다 (Lemmens and Croux,2006). 분석용 데이터가 불안정(unstable)하면 배깅 분류자는 결합을 통하여 향상되어진다. 그러나 분석용 데이터가 안정적이면 배깅 분류자는 분석용 데이터에서 얻어진 단일 분류자와 비슷하다 (Breiman, 1996). 실제로 배깅은 편의(bias)가 큰 경우를 제외하고 분류자의 분산을 줄여준다 (Bunlmann and Yu, 2002, Buja and Stuetzle, 2006). 일반적으로 불안정한 분류자는 낮은 편의와 높은 분산을 가

지고 있으므로 배깅에 적합하다 (Breiman, 1998). 하지만 SVM과 같은 안정적인 분류자에서는 정확한 예측을 하는 것에 악영향을 미친다 (Dietterich, 2000).

2.2.2 수정된 배깅 방법론

앙상블 기법을 이용하여 생성된 배깅 분류자는 단일 분류자에 비해 예측력이 향상되고 안정적인 결과를 보인다. 하지만 데이터나 기본 분류자로 쓰이는 단일 알고리즘이 안정적인 경우에는 오히려 예측력이 떨어지는 단점을 가지고 있다는 것이 밝혀진 바 있다. 이러한 단점을 보완하기 위한 수정된 배깅 방법론들이 계속 연구되고 있다. 본 절에서는 예측력 향상을 위한 수정된 배깅 방법론들에 대하여 간략히 살펴보도록 한다.

1) 브래깅 알고리즘

브래깅(bragging : **B**ootstrap **r**obust **agg**regat**ing**)은 데이터에 따라서 배깅 분류자가 여전히 안정적이지 않은 것을 보완하기 위한 수정된 배깅 방법론 중 하나이다(Buhlmann, 2003). 이는 붓스랩으로 얻어진 각각의 단일 분류자를 결합하는 과정에서 평균 대신에 그보다 더 민감하지 않은 추정량인 메디안(median)을 사용한다. 로버스트(robust)한 대표적 통계량에는 후버와 햄펠의 M-추정량(Huber's estimator and Hampel's redescending M-estimator) 등이 있지만 이들을 이용함으로써 향상되는 정도는 미미하다는 연구결과가 있다 (Buhlmann, P., 2003). 브래깅 분류자는 식(2.22)과 같이 나타낼 수 있다.

$$\varphi_{BR} = \underset{1<b<B}{median} \ \varphi(x, L_b) \tag{2.22}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자

브래깅은 분류자의 결합 방법으로 로버스트한 결합방법인 메디안을 사용한다. 따라서 극단적인 분류자, 혹은 예측값 으로 부터 받는 영향을 줄여 결합 분류자의 변동성을 줄여 예측력을 높일 수 있다.

2) 나이스 배깅 알고리즘

나이스 배깅(Nice-bagging)은 붓스트랩 데이터로부터 얻어진 단일 분류자 가운데 성능이 좋은 분류자만을 골라 평균의 방법으로 결합 분류자를 생성하는 배깅 방법이다(Skurichina, M et al. 1998). 나이스 배깅은 붓스트랩을 하지 않은 분석용 데이터로부터 단일 분류자를 생성하여 오류율(error rate)을 구하고, 붓스트랩으로 얻어진 데이터로부터 생성한 각각의 분류자들의 오류율과 비교한다. 이 때 전체 분석용 데이터로부터 얻어진 분류자의 오류율보다 낮은 붓스트랩 분류자만을 추출하여 이를 평균 내어 결합 분류자를 만든다. 나이스 배깅은 식 (2.24) 같이 나타낼 수 있으며, 여기서 L_{tr}은 전체 분석용 데이터를, L_b는 b번째 붓스트랩 된 데이터를, B'는 식(2.23)의 조건을 만족하는 b 중에서 가장 큰 값을 나타낸다.

$$ER(g(\bullet, L_b^*)) < ER(g(\bullet, L_{tr}^*)) \tag{2.23}$$

$$\varphi_{NB} = \underset{1<b<B'}{average} \ \varphi(x, L_b) \tag{2.24}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자
- B' : 식(2.23)의 조건을 만족하는 b 의 최대값

나이스 배깅은 전체 분석 데이터로부터 얻어진 단일 분류자 보다 예측력이 떨어지는 붓스트랩 분류자를 제거 한 후 평균하여 결합 분류자를 생성하는 방법이다. 수행능력이 좋지 않은 분류자의 정보를 배제하고 나머지의 붓스트랩 분류자를 사용하여 결합하기 때문에 모든 분류자를 결합한 배깅 방법보다 예측력이 우수하다.

3) 절사 배깅 알고리즘

나이스 배깅은 전체 분석 데이터로부터 생성한 분류자의 오류율 보다 높은 오류율을 가진 붓스트랩 분류자를 제거한 후 남은 분류자를 결합하는 방법이다. 하지만 분석 데이터와 검증용 데이터의 분류에 따라서 전체 분석용 데이터로부터 생성된 분류자의 오류율 또한 크게 달라질 수 있다. 결과적으로 오류율이 높은 붓스트랩 분류자를 포함시키거나 오류율이 낮은 붓스트랩 분류자를 제거한 후 배깅 분류자를 생성할 위험성이 존재한다. 절사 배깅 (trimmed bagging)은 이와 같은 잘못된 정보를 포함하거나 유용한 정보가 손실되는 것을 막기 위해 일정 절사비율을 정하여 절사한 후 평균하는 α-절사평균의 개념을 이용한다 (Christophe Croux et al. 2007). 즉, 오류율이 높은 α%의 붓스트랩 분류자를 제외한 나머지 붓스트랩 분류자들을 평균하

여 결합하는 것이다. 이때 적절한 절사비율 α를 정해야 하는데, 오류율이 높은 붓스트랩 분류자는 충분히 제거되고, 동시에 되도록 많은 붓스트랩 분류자를 결합 할 수 있도록 α를 설정하는 것이 좋다. α를 너무 작게 잡을 경우 오류율이 높은 붓스트랩 분류자가 제거되지 않아 이를 결합모형에 반영하게 되며, 너무 크게 설정했을 경우 너무 많은 양의 붓스트랩 분류자를 제거하게 되므로 적은 양의 붓스트랩 분류자만을 결합하게 되어 위험할 수 있다. 따라서 본 논문에서는 적당한 비율인 0.25로 α를 선택한다. 절사 배깅은 식(2.25)와 같이 나타낸다.

$$\varphi_{TB} = \underset{1 < b < \lfloor (1-\alpha)B \rfloor}{average} \varphi(x, L_b) \tag{2.25}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자

제3장 가중평균을 이용한 배깅 방법론

본 장에서는 기존 제시되었던 배깅 방법론을 수정하여 새로운 배깅 방법론을 제안하고자 한다. 배깅 방법론을 비롯한 수정된 배깅 방법론이 분류, 예측 알고리즘으로 사용되고 있으나 여전히 데이터나 분류자의 종류에 따라 예측력이 떨어진다는 한계점을 가지고 있다. 본 연구에서는 배깅 방법론을 구축하는 과정 중, 단일 분류자를 결합하는 방법에 가중평균의 개념을 적용한 새로운 배깅 방법론을 제안한다. 가중평균의 가중치로는 예측 정확도(accuracy : 정분류율)을 이용한다. 첫 번째 방법으로 모든 붓스트랩 단일분류자를 가중평균 하여 결합 분류자를 생성해 보고, 두 번째로는 절사 배깅에서와 같이 오류율이 높은 α%의 붓스트랩 단일 분류자를 절사한 후 남은 분류자를 가중평균 하여 결합 분류자를 생성한다.

3.1 가중 배깅 방법론

일반적으로 다수의 변량의 평균값을 구할 때 중요도나 영향도를 반영 해주기 위한 방법으로 각 수치에 중요도 만큼의 가중치를 각각 곱하여 평균을 구하는 가중평균을 이용한다. 여기서 관찰값들을 $x_1, x_2, \cdots, x_N$으로, 가중치를 $w_1, w_2, \cdots, w_N$으로 나타내면 가중평균(weighted average)은 식(3.1)과 같이 나타낼 수 있다.

$$\text{가중평균} = \frac{w_1 x_1 + w_2 x_2 + \ldots + w_N x_N}{w_1 + w_2 + \ldots + w_N} \tag{3.1}$$

배깅 방법론은 붓스트랩 된 분석용 데이터에서 얻어진 단일 분류자를 평균을 이용하여 결합한다. 변동성이 큰 데이터를 사용하여 모형을 구축하는 경우에 붓스트랩 된 데이터에 따라 각 붓스트랩 단일 분류자의 오류율이 달라질 수 있다. 이 때 모든 붓스트랩 단일 분류자를 단순평균 내면 오류율이 가장 높은 붓스트랩 분류자와 오류율이 가장 낮은 붓스트랩 분류자가 결합 분류자를 생성하는데 동일한 기여를 하게 된다. 붓스트랩 분류자를 결합할 때 오류율이 높은 분류자의 기여도를 낮추고 오류율이 낮은 분류자의 기여도를 높인다면 단순평균으로 분류자를 결합했을 때 보다 향상된 예측력을 가질 수 있을 것이다.

가중 배깅은 붓스트랩 단일 분류자의 정분류율을 가중치로 하여 각각의 붓스트랩 분류자에 반영한 후 이를 평균내어 결합 분류자를 형성한다. 이때 정분류율은 검증용 데이터 중 정분류된 관측값의 비율로 계산하며 가중된 배깅 알고리즘은 식(3.2)와 같다.

$$\varphi_{WB} = \underset{1<b<B}{average} \; w_b \varphi(x, L_b) \tag{3.2}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자
- w_b : b번째 붓스트랩 분류자의 정분류율

3.2 가중 절사 배깅 방법론

붓스트랩된 각 데이터에서 생성된 단일 분류자를 결합할 때, 붓스트랩을 통하여 생성된 모든 단일 분류자를 평균하는 것 보다 오류율이 높은 α%를 절

사한 나머지 붓스트랩 단일분류자를 평균하는 경우를 살펴보자. 결과적으로 결합분류자의 예측력이 높아지고 이는 이미 수정된 배깅 방법인 절사 배깅의 아이디어가 되기도 하였다. 오류율이 높은 $\alpha\%$의 붓스트랩 단일 분류자를 제거한 후 나머지 분류자들을 사용한다 하더라도 그 중에서 오류율이 가장 높은 분류자와 오류율이 가장 낮은 분류자의 기여도에 차이를 반영하지 않는다면 결합 분류자를 생성하는데 있어서 잘못된 정보를 포함하게 될 가능성이 높아지게 될 것이다. 따라서 본 논문에서는 절사 배깅에 가중평균의 개념을 더한 가중 절사 배깅 방법론을 제시하고자 한다. 전체 붓스트랩된 데이터에서 붓스트랩 분류자를 생성한 뒤 오류율이 가장 높은 $\alpha\%$의 분류자를 제거하고 나머지 $(1-\alpha)\%$의 붓스트랩 단일 분류자를 가중평균 하여 결합하며 이는 식(3.3)과 같다.

$$\varphi_{WTB} = \underset{1<b<\lfloor (1-\alpha)B \rfloor}{average} \; w_b \varphi(x, L_b) \tag{3.3}$$

- L_b : b번째 붓스트랩 분석용 데이터
- $\varphi(x, L_b)$: b번째 붓스트랩 분석용 데이터로부터 생성된 분류자
- w_b : b번째 붓스트랩 분류자의 정분류율

제4장 실제자료의 적용

본 장에서는 앞에서 소개된 배깅 방법론들과 본 논문에서 제안한 배깅 방법론을 사용하였을 경우에 단일 분류 방법론을 사용하였을 때 보다 예측력이 어느 정도 향상되는지 알아본다. 이를 위해 실제 자료에 분류 방법론을 적용시켜 단일 분류 방법론의 오분류와 배깅 방법론의 오분류를 상대적으로 비교해볼 수 있는 상대 향상도를 계산하여 본다.

4.1 자료 및 평가방법 소개

단일 분류자를 사용하였을 때에 비하여 각 배깅 방법론을 사용했을 때의 예측력이 향상되었는지 여부와 그 정도를 알아보기 위하여 검증용 데이터에 각각의 분류자를 적용시켜 그 결과를 살펴본다. 이를 위해 주어진 데어터를 임의로 분석용 데이터와 검증용 데이터로 나눈다. 이때 각각의 데이터는 80%대 20%의 비율로 나눈다. 나눠지는 형태에 따라 결과가 바뀌는 것을 방지하기위해 데이터를 나누는 작업을 10번 실시한 후, 그 결과를 평균 내어 사용한다. 검증용 데이터에 분류자를 적용시켰을 때 전체 검증용 데이터 중 오분류된 관측값의 비율을 오류율(error rate) 이라고 한다. 평가결과를 통해서 배깅, 혹은 수정된 배깅 분류자의 수행 능력(performance)이 단일 분류자를 사용했을 경우에 비해 얼마나 증가 혹은 감소하였는지 알아본다. 이를 평가하기 위한 방법으로 식(4.1)과 같이 상대적 향상도(relative improvement)를 계산한다.

$$Relative\ improvement = \frac{ER_{base\ classifier} - ER_{bagging}}{ER_{base\ classifier}} \qquad (4.1)$$

$ER_{bace\ classifier}$와 $ER_{bagging}$은 각각 단일 분류자와 배깅 분류자를 검증용 데이터에 적용하여 얻어진 오류율이다. 10번의 반복을 통하여 얻어진 상대적 향상도의 평균값을 결과로 제시하고, 또한 이 결과를 평균 내어 얻어진 값이 0으로부터 음의방향 또는 양의방향으로 어느 정도 떨어져 있는지 살펴본다. 이것이 유의한 결과인지를 알아보기 위하여 유의수준 95%에서 T-검정(t-test)을 실시한다.

각 배깅 방법의 수행능력(performance)를 평가하기 위하여 우리는 6개의 잘 알려진 UCI(University of California Irvine)의 기계학습 분야의 데이터(Machine Learning Repository data set)를 사용하였다. 모든 데이터의 반응변수는 2개의 범주형 이며 데이터의 표본 수와 변수의 수는 [표 4.1]로 정리하였다.

[표 4.1] 사례적용에 이용된 표본크기(n)와 변수의 개수(p)

데이터	n	p
Iono	351	33
Spambase	4601	57
Tictacto	958	30
Wdbc	569	30
Spect	267	23
Spectf	369	45

모든 모형 설계와 적용은 R 소프트웨어를 이용하여 분석하였다. 기본분류

자로는 의사결정나무, SVM, 선형판별분석, 로지스틱회귀를 사용하였다. 의사결정나무 분류자의 모형설계와 결과예측은 R소프트웨어의 “tree” 패키지를 사용하였으며 SVM 분류자는 “e1701” 패키지를, 선형판별분석은 ”MASS” 패키지를 사용하였다. 각 배깅 방법론 별로 결합 방법 이외의 조건인 붓스트랩 횟수는 모두 B=250번으로 설정하였다.

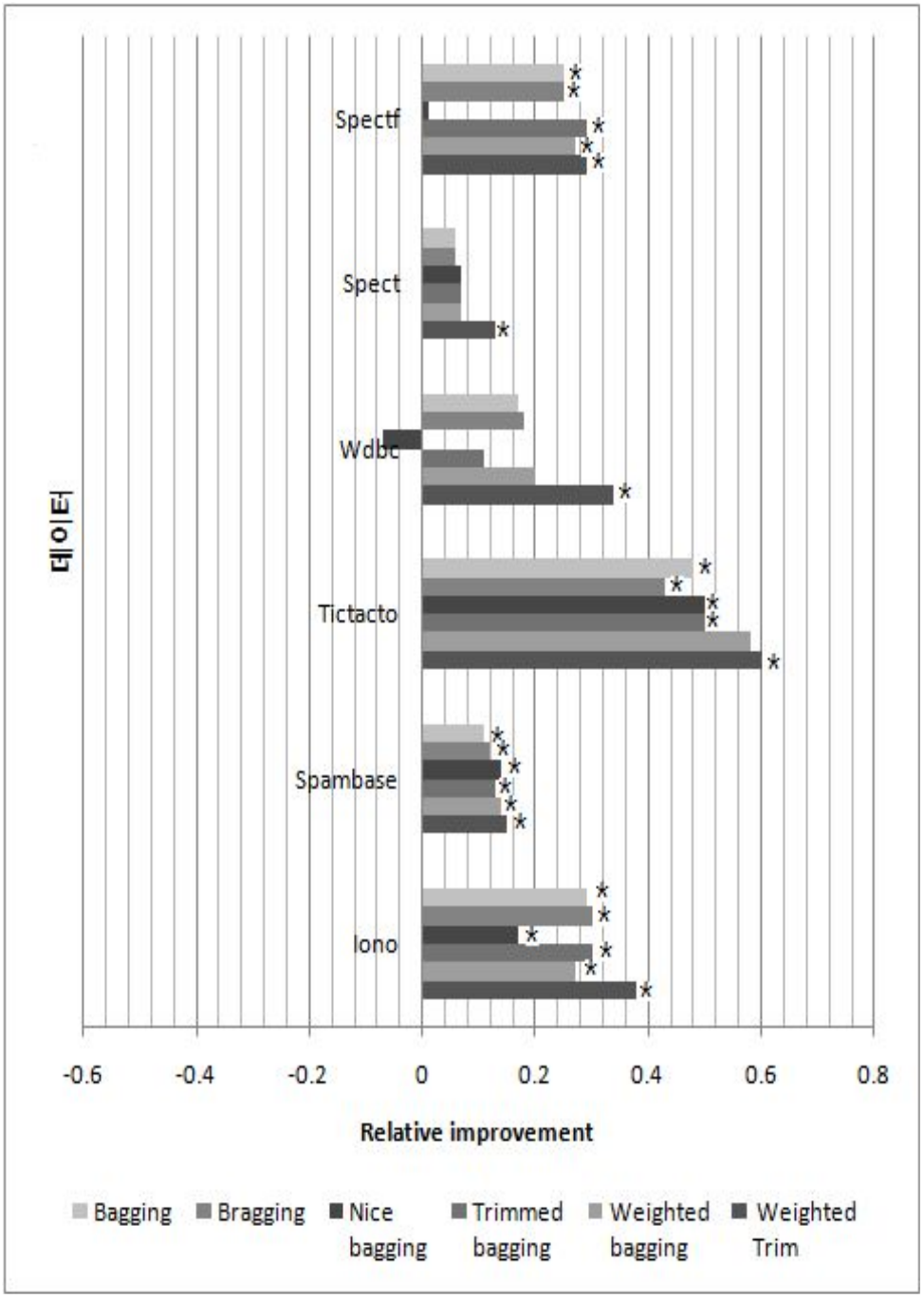
4.2 배깅 방법론의 적용 결과

안정적이거나 불안정적인 기본 분류자 모두에서 가중 배깅과 절사 가중 배깅의 수행 능력을 알아보기 위하여 의사결정나무, SVM, 선형판별분석, 로지스틱 회귀를 기본분류자로 사용하여 분석을 수행하였다. 기본 분류자를 결합하여 다양한 배깅 분류자를 형성한 후, 각 배깅 분류자의 예측력이 기본 분류자로만 예측 했을 때 보다 어느 정도 향상 되었는지를 살펴보았다. 상대적 향상도의 평균과 유의여부를 기본 분류자 별로 [표4.2]에서 [표 4.5]까지에 정리하였다. 표의 마지막 두 줄에는 유의한 상대적 향상도의 개수를 명시하였고 각각의 유의여부를 상대적 향상도 수치 옆에 *로 표시하였다. [표 4.2]에서 [표 4.5]까지의 결과를 개괄적으로 살펴보면 기본 분류자 별로 배깅과 브래깅의 상대적 향상도는 거의 같은 값을 보였다. 따라서 두 분류 방법은 매우 비슷한 예측력을 가지고 있다고 볼 수 있다. 이제 기본 분류자에 따른 각 배깅 방법론의 상대적 향상도를 자세히 살펴보고 그 특징을 알아보도록 한다.

[표 4.2] 의사결정나무를 기본 분류자로 사용했을 때 각 배깅 방법론의 상대적 향상도 평균

데이터	Bagging	Bragging	Nice	Trimmed	Weighted	Trimmed Weighted
Iono	0.29*	0.30*	0.17*	0.30*	0.27*	0.38*
Spambase	0.11*	0.12*	0.14*	0.13*	0.14*	0.15*
Tictacto	0.48*	0.43	0.50*	0.50*	0.58*	0.60*
Wdbc	0.17	0.18	-0.07	0.11	0.20	0.34*
Spect	0.06	0.06	0.07	0.07	0.07	0.13*
Spectf	0.25*	0.25*	0.01	0.29*	0.27*	0.29*
Positive	4	4	3	4	4	6
Negative	0	0	0	0	0	0

불안정적인 분류자 중 하나인 의사결정나무를 기본 분류자로 하여 배깅 분류자를 생성 한 결과가 [표 4.2]와 같다. 과반수의 데이터에서 상대적 향상도가 유의적으로 증가함을 알 수 있다. 이는 의사결정나무에서 배깅 분류자를 사용하였을 때의 오류율이 기본 분류자에 비하여 감소하였음을 나타낸다. 이 결과는 이미 여러 연구에서 밝혀진 내용과 부합한다. 배깅 분류자 뿐만 아니라 수정된 배깅 분류자 모두의 예측력이 기본 분류자보다 높아짐 또한 알 수 있다. 특히 가중된 절사 배깅 방법론은 모든 데이터에서 유의한 오류율의 감소를 보여 기존의 배깅 방법론 보다 월등하게 좋은 성능을 보이고 있다. [표 4.2]의 내용을 그래프로 나타낸 것이 [그림4.1]이다.

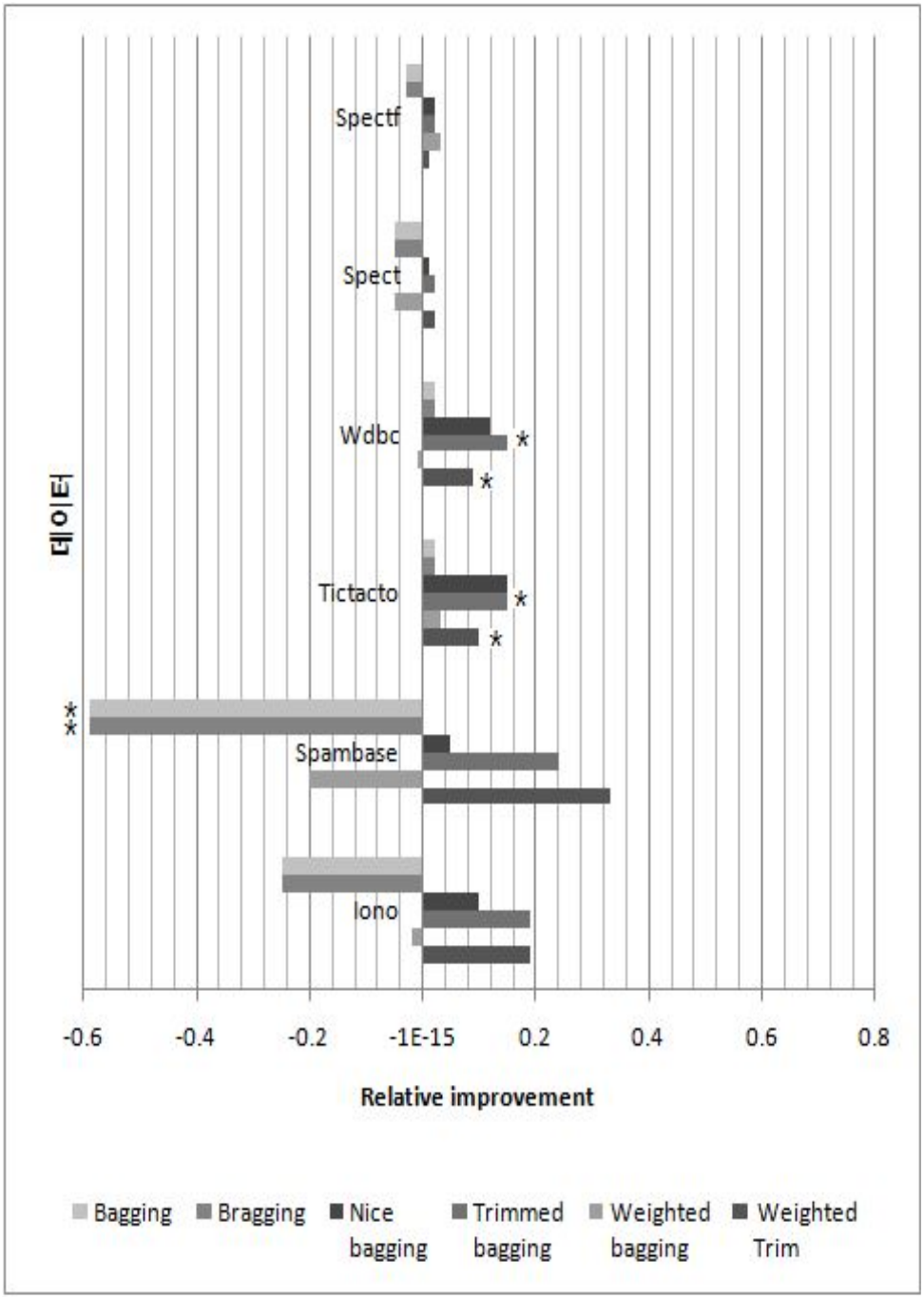


[그림 4.1] 의사결정나무에서 배깅 방법론의 상대적 향상도

[표 4.3] SVM를 기본 분류자로 사용했을 때 각 배깅 방법론의 상대적 향상도 평균

데이터	Bagging	Bragging	Nice	Trimmed	Weighted	Trimmed Weighted
Iono	-0.25*	-0.25*	0.10	0.19	-0.02	0.19
Spambae	-0.59*	-0.59*	0.05	0.24	-0.20	0.33*
Tictacto	0.02	0.02	0.15	0.15*	0.03	0.10*
Wdbc	0.02	0.02	0.12	0.15*	-0.01	0.09*
Spect	-0.05	-0.05	0.01	0.02	-0.05	0.02
Spectf	-0.03	-0.03	0.02	0.02	0.03	0.01
Positive	0	0	0	2	0	3
Negative	2	2	0	0	0	0

[표 4.3]은 안정적인 분류자 중 하나인 의사결정나무를 기본 분류자로 하여 배깅 분류자를 생성 한 결과이다. 의사결정 나무의 결과와 비교해 상대적 향상도가 음의 값을 가지는 경우가 많다. 특히 배깅과 브래깅의 경우 음의 값을 가지는 상대적 향상도가 2개이고 유의한 증가를 보인 경우는 단 한건도 존재하지 않았다. 이는 SVM을 기본 분류자로 사용한 경우 배깅 분류자의 오류율이 기본 분류자에 비하여 증가된다는 것을 보여준다. 다만, 절사 배깅과 가중된 절사 배깅 방법론의 상대적 향상도만이 유의적으로 증가해 안정적인 분류자 에서도 좋은 예측력을 보임을 알 수 있다. 그밖에 나이스 배깅과 가중 배깅은 유의한 향상도가 한건도 존재하지 않아 기본 분류자와 비슷한 예측력을 갖는다는 것을 알 수 있다. [표 4.3]의 내용을 그래프로 나타낸 것이 [그림4.2]이다.

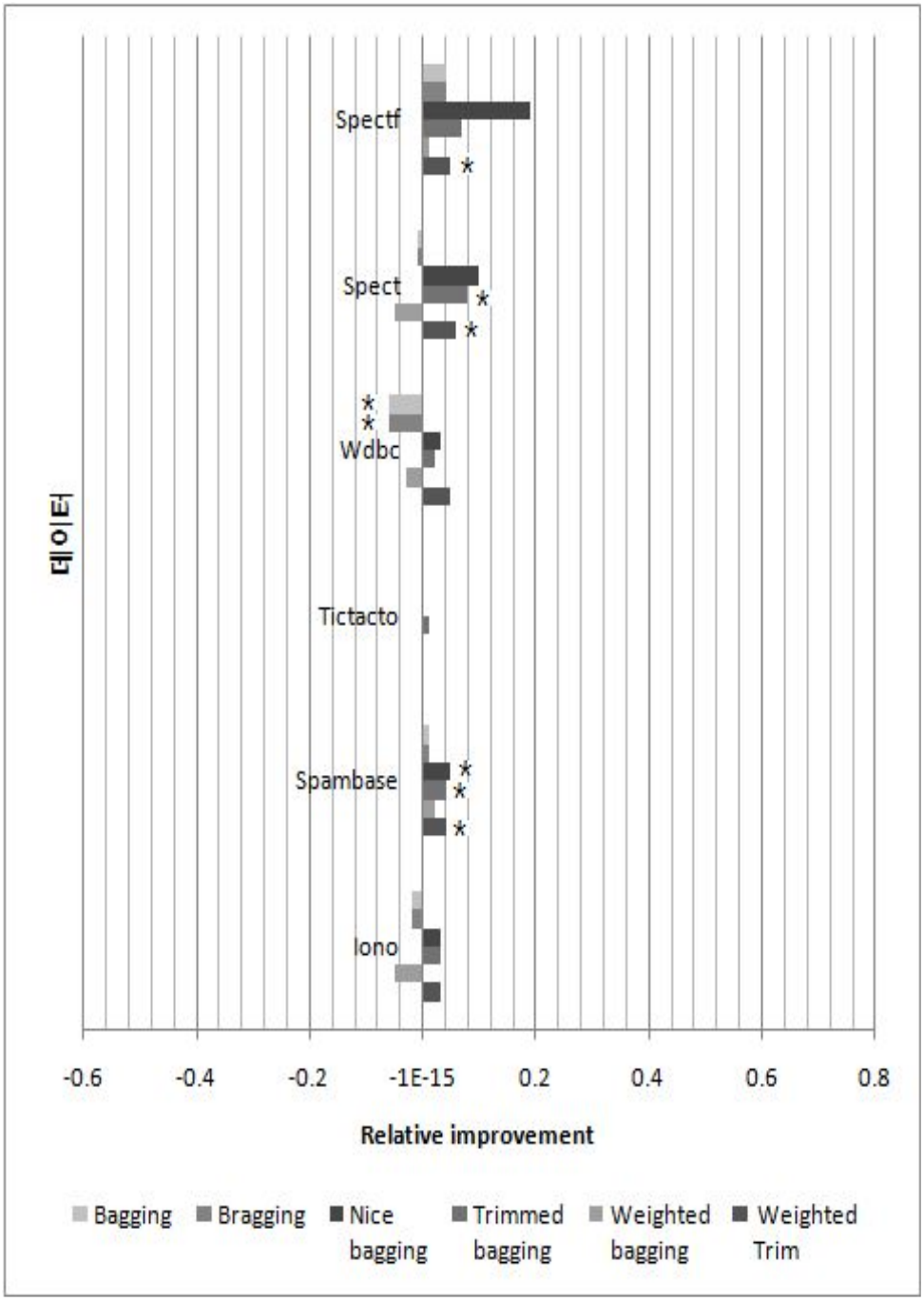


[그림 4.2] SVM에서 배깅 방법론의 상대적 향상도

[표 4.4] 선형 판별 분석에서 각 배깅 방법론의 상대적 향상도의 평균

데이터	Bagging	Bragging	Nice	Trimmed	Weighted	Trimmed Weighted
Iono	-0.02	-0.02	0.03	0.03	-0.05	0.03
Spambase	0.01	0.01	0.05*	0.04*	0.02	0.04*
Tictacto	0.00	0.00	0.00	0.01	0.00	0.00
Wdbc	-0.06*	-0.06*	0.03	0.02	-0.03	0.05
Spect	-0.01	-0.01	0.10	0.08*	-0.05	0.06*
Spectf	0.04	0.04	0.19	0.07	0.01	0.05*
Positive	0	0	1	2	0	3
Negative	1	1	0	0	0	0

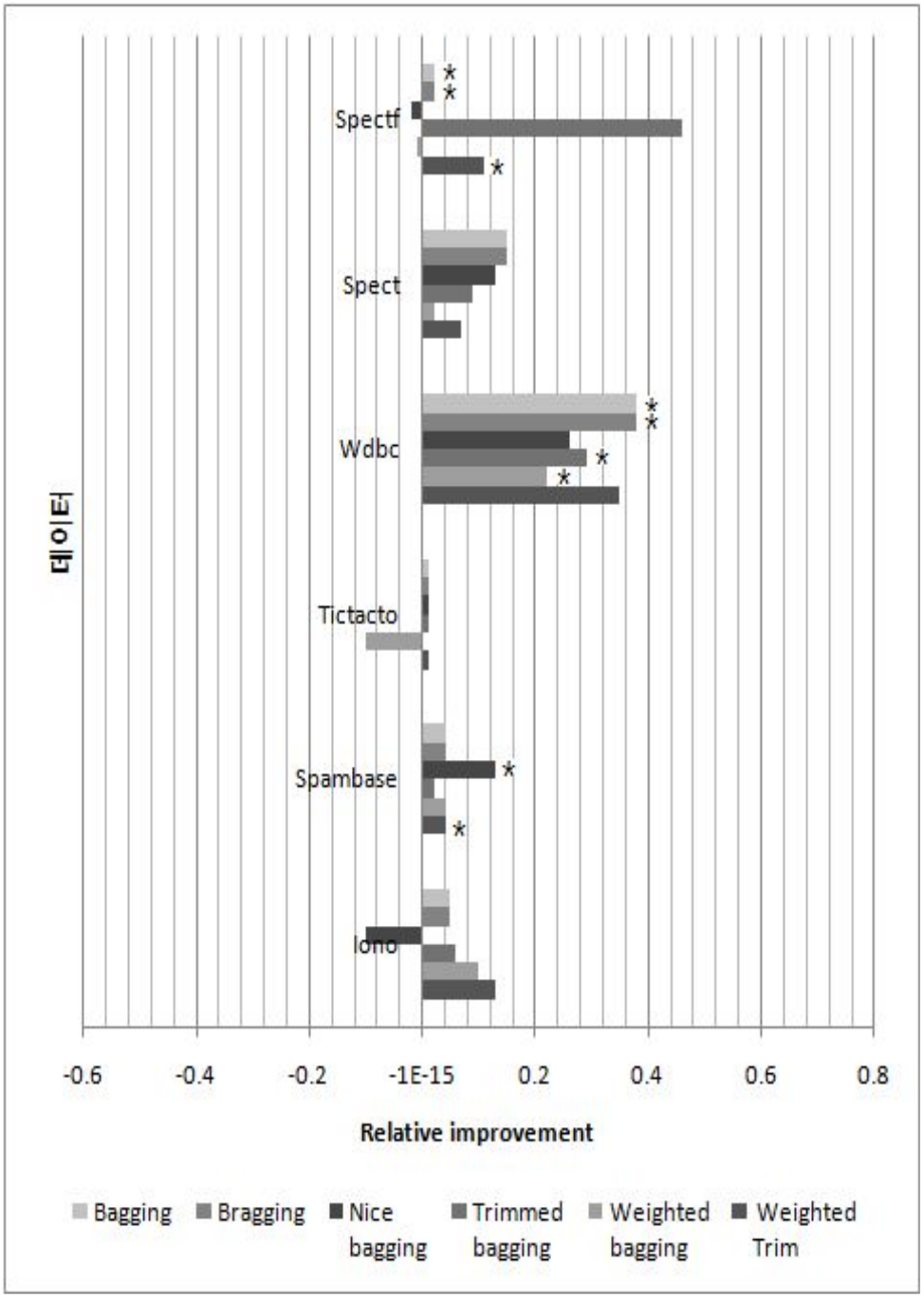
[표 4.4]와 [표 4.5]는 선형 분류자인 선형판별분석과 로지스틱 회귀를 기본 분류자로 하여 배깅 분류자를 생성 한 결과이다. [표 4.4]에서 배깅과 브래깅의 경우 유의적으로 음의 값을 가지는 상대적 향상도가 한 개 씩 존재한다. 이 밖의 수정된 배깅 방법론에서는 유의하게 양의 값을 가지는 경우가 존재한다. 선형판별분석에서도 역시 가중된 절사 배깅을 이용하는 경우에 가장 좋은 결과를 보였다. 로지스틱 회귀의 결과인 <표 4.5>의 결과 또한 이와 비슷함을 알 수 있다. [표 4.4]와 [표 4.5]의 내용을 그래프로 나타낸 것이 [그림4.3]과 [그림4.4]이다.



[그림 4.3] 선형판별분석에서 배깅 방법론의 상대적 향상도

[표 4.5] 로지스틱 회귀에서 각 배깅 방법론의 상대적 향상도의 평균

데이터	Bagging	Bragging	Nice	Trimmed	Weighted	Trimmed Weighted
Iono	0.05	0.05	-0.10	0.06	0.1	0.13
Spambase	0.04	0.04	0.13*	0.02	0.04	0.04*
Tictacto	0.01	0.01	0.01	0.01	-0.10	0.01
Wdbc	0.38*	0.38*	0.26	0.29*	0.22*	0.35
Spect	0.15	0.15	0.13	0.09	0.02	0.07
Spectf	0.02*	0.02*	-0.02	0.46	-0.01	0.11*
Positive	2	2	1	1	1	2
Negative	0	0	0	0	0	0



[그림 4.4] 로지스틱회귀에서 배깅 방법론의 상대적 향상도

제 5장 결론 및 향후연구과제

본 논문에서는 기존의 배깅 방법론과 수정된 배깅 방법론을 학습하였고, 나아가 이전의 배깅 방법론보다 향상된 배깅 방법론을 제안하였다. 배깅 방법론이 의사결정나무와 같이 불안정한 분류자에서 성공적인 예측력을 보여준다고 알려졌지만 이는 곧 SVM과 같이 안정적인 분류자 에서는 예측력이 떨어진다는 단점을 가지고 있다. 이를 해결하기 위하여 여러 가지 수정된 배깅 기법이 연구되어왔으며, 본 논문 또한 배깅의 예측력을 높일 수 있는 결합방법으로 가중평균을 이용한 방법과 α-절사평균과 가중평균을 동시에 이용한 방법을 제안하였다. 본 논문에서 제안한 수정된 배깅 방법론이 기존의 방법에 비해 예측력 측면에서 어느 정도의 향상이 이루어졌는지를 알아보기 위하여 실제 자료에 배깅을 비롯하여 수정된 배깅 방법들을 적용시켜 보았다. 그 결과 가중평균만을 이용한 분류 방법은 배깅 방법 보다는 예측력이 좋아졌지만 다른 수정된 배깅 방법론에 비하여 뚜렷한 향상정도를 보이지 못했다. 반면에 가중된 절사 배깅 방법은 모든 분류자에서 상대 향상도가 유의하게 양의 값을 가졌으며 특히 기본 분류자와 비교했을 때의 상대 향상도가 유의적으로 음의 값을 갖는 경우는 단 한건도 존재하지 않았다. 이는 기존 배깅 방법론의 예측력이 안정적인 기본 분류자를 사용했을 때 떨어진다는 단점을 보완할 수 있다.

본 논문은 변동성이 큰 분류자에서도 예측력을 높일 수 있는 결합 방법을 제안하였다는 데에 그 의의가 있다. 본 논문에서 제시한 방법은 반응변수가 2개의 범주형인 경우만으로 국한시켜 향상도를 측정하였기 때문에 그 결과를 일반화시키기에는 다소 무리가 따른다. 또한 α-절사평균과 가중평균을 이용한 배깅 방법을 이용하는 경우에 적절한 α의 값을 정할 수 있는 기준

이 명확하지 않았다. 이와 같은 내용을 보안하여 연구 해 본다면 보다 향상된 배깅 방법론을 구현할 수 있을 것이다.

참 고 문 헌

[1] 강현철, 한상태, 최종후, 이성건, 김은석, 엄익현, 김미경 (2006). 고객관계관리(CRM)를 위한 데이터마이닝 방법론, 자유아카데미, 서울.

[2] 오현정 (2002). 데이터 마이닝에서 배깅과 부스팅의 비교연구, 동국대학교, 통계학과 석사학위 논문, 서울.

[3] 이영섭 오현정 김미경 (2005). 데이터 마이닝에서 배깅, 부스팅, svm 알고리즘 비교 분석, *응용통계연구.* 제 18권 2호, 343-354.

[4] Lemmens, A., Croux, C. (2006). Bagging and Boosting Classification Trees to Predict Churn, *Journal of Marketing Research.* 43, 276 - 286.

[6] Breiman, L. (1996). Bagging predictors. *Machine Learning.* 24, 123 - 140.

[7] Breiman, L.(1998). Arcing classifiers. *Ann. Statist.* 26, 201 - 849.

[8] Buhlmann, P. (2003). Bagging, subbagging and bragging for improving some prediction algorithms. *Recent Advances and Trends in Nonparametric Statistics. Elsevier, Amsterdam.* 9 - 34.

[9] Buja, A., Stuetzle, W. (2006). Observations on bagging. *Statist. Sinica* 16, 323 - 351.

[10] Croux, C., Kristel Joossens, K., Aurelie Lemmens, A. (2007). Trimmed bagging, *Computational Statistics and Data Analysis,* 52, 362 - 368.

[11] Dietterich, T.G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and

randomization. *Machine Learning.* 40, 139 - 158.

[12] Friedman, J.H. (2001). The Elements of Statistical Learning(Data Mining, Inference, and Prediction), Springer.

[13] Skurichina, M., Duin, B. (1998). Bagging for linear classifiers. *Pattern Recognition* 31, 909 - 930.

ABSTRACT

A Study on Advanced Bagging Algorithm

Jeong-Yoon Yoo
Department of Statistics
The Graduate School
Sungshin Women's University

Bagging is one of the most effective procedure to improve the performance on unstable estimators or classifiers. It draws bootstrap sample, and then averages the resulting prediction rule. When using unstable classifier, bagging reduce the variance of a classifier.

However, there is no guarantee that bagging will improve the performance of any base classifier. When using stable base classifiers, like support vector machines, it may even yield a deterioration of predictive accuracy.

In this paper, we propose new bagging algorithm to improve the performance of any classifiers. The first idea is to aggregate the bootstrapped classification rules that weight to reflect the predict accuracy. The second idea is similar previous one. Instead of averaging over all bootstrapped classifiers, we trim away those bootstrapped

classifiers that result in the highest error rate. Afterward, we perform the same work previous procedure.

On the basis of real data experiments, we conclude that idea proposed in this paper is performing well comparably to standard bagging when applied to unstable classifiers as decision trees. Moreover, it yields better results when applied to stable base classifiers, like support vector machines.