JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 10, pp. 2879-2887, Oct. 2024

Check for updates

에이다부스트에서 약한 학습기로 사용하는 결정트리 알고리즘들의 비교: CART, LDA

이 종 찬*
청운대학교 컴퓨터공학과 교수

Comparison of Decision Tree Algorithms as Weak Learners in AdaBoost: CART and LDA

Jong Chan Lee*
Professor, Department of Computer Engineering, ChungWoon University, Incheon 22100, Korea

[요 약]

본 논문은 앙상블을 기반으로 하는 에이다부스트(Adaboost) 알고리즘이 약한 분류기에서 강한 분류기로 점진적으로 학습되는 과정에 대해 살펴본다. 이 과정에서 가장 중요한 부분은 약한 분류기를 구성하는 부분이다. 이에 대해 본 논문은 일반적으로 널리 사용되는 CART 알고리즘과 함께, 분류와 회귀 분야에서 오랫동안 사용되어왔던 LDA 알고리즘을 가지고 결정트리를 구성하는 방법을 알아본다. 그리고 이들 2가지 결정트리를 에이다부스트의 모델에 적용하고 학습 데이터를 이 모델에 입력하여 결과를 알아보았다. 그리고 LDA의 성능이 기존의 CART 접근 방식보다 다소 개선된 것을 확인하였다. 이러한 결과는 학습 과정에서 결정 트리의 각 노드는 분류면을 나타내는데, CART는 속성 축에 직교하는 분류면을 형성하는 반면 LDA는 속성 축과 관련 없이 임의의 각도를 가지는 분류면을 형성한다는 점, 즉 노드 구조의 차이 때문으로 해석된다.

[Abstract]

This paper explores the process by which the AdaBoost algorithm, an ensemble-based method, incrementally trains weak classifiers into strong classifiers. The most critical aspect of this process lies in the construction of weak classifiers. To this end, this study investigated the method of constructing decision trees, using both the widely employed classification and regression tree (CART) and linear discriminant analysis (LDA) algorithms, which have long been used in classification and regression tasks. These two types of decision trees were applied to the AdaBoost model, and training data were input to assess the results, thereby observing that the performance of LDA slightly improved compared to the traditional CART approach. These results were attributed to the difference in node structure, with each node of the decision tree representing a classification plane in the learning process; CART forms classification planes orthogonal to the attribute axes, whereas LDA forms planes with arbitrary angles unrelated to the attribute axes.

색인어 : 앙상블 알고리즘, 에이다부스트, 결정트리, CART(분류 및 회귀 트리), 선형 판별 분석

Keyword : Ensemble Algorithm, AdaBoost, Decision Tree, Classification and Regression Tree (CART), Linear Discriminant Analysis (LDA)

http://dx.doi.org/10.9728/dcs.2024.25.10.2879

Received 29 August 2024; **Revised** 11 October 2024
Accepted 16 October 2024

***Corresponding Author; Jong Chan Lee**
Tel: ※ 개인정보 표시제한
E-mail: jclee@chungwoon.ac.kr

Ⅰ. 서 론

앙상블(ensemble) 학습은 여러 분류기의 결과를 투표(vote)하여 최종결정하는 것으로 배깅과 부스팅 기법으로 크게 나누어진다. 배깅의 경우 학습 데이터에 샘플링을 다르게 하여 여러 개의 학습 집합으로 나눈 다음, 각각의 학습 집합을 별도의 분류기 모델에 할당하고 학습한다. 그리고 테스트 데이터를 이 모델들에 입력하면 여러 개의 예측 결과가 나오는데, 이들에 투표기법의 다수결을 이용해 최종 결과를 얻는다. 따라서 이 방법은 과적합(overfit)으로 인해 낮은 편기(bias)와 높은 분산(variance)의 특징을 가지며, 각 모델에서 샘플링과 학습을 하는 과정은 모델마다 서로 독립이므로 병렬로 수행할 수 있다. 반면 부스팅은[1],[2] 학습 데이터에 약한 학습기로 학습하여 예측이 틀리는 사건에 대해 샘플링 가중치를 높임으로써 다음 학습기에 학습 데이터로 선택될 확률을 높이는 과정을 여러 번 반복하여 사건들을 정확하게 분류하도록 기회를 주는 것이다. 최종 결과는 학습 결과에 가중치를 반영하는 투표를 통해 결정한다. 따라서 이 방법은 분류기의 성능이 0.5보다 약간 좋은 약한 분류기를 사용하는 과소적합(underfit)으로 인해 높은 편기와 낮은 분산의 특징을 가진다. 또한 다수의 분류가 순차적으로 샘플링과 학습을 반복하며 약한 학습기에서 강한 학습기로 변화해 간다.

결정트리는 수치와 범주형 데이터 모두에 적용하여 데이터 특성을 결정 규칙으로 분류하며 트리 형식으로 나타낸다. 이는 데이터를 시각화해 주어 해석을 쉽게 할 수 있는 장점이 있어 분류와 회귀 분야에 전통적으로 사용되어 오고 있다. 여러 결정트리는 근사 최적의 분류 방법에 따라 구분되며, 정보이득(IG) 함수로 이를 표현하게 된다. 기존의 알고리즘들의 IG 함수들은 엔트로피, 지니 인덱스, 고유 벡터, 고유값 등을 조합하여 목적을 달성하려 했다. 따라서 결정트리에서 가장 중요한 부분은 IG 함수를 고안하는 것이고, 이 IG 함수의 목표는 근사 최적의 분류면을 구하여 트리의 노드에 표현하는 작업을 반복하여 되도록 작은 트리로 전체 규칙 정보를 대표할 수 있도록 하는 것이다[1],[2].

에이다부스트는 목적지에 빠르게 수렴하고 가설(모델)의 개수인 라운드 수를 제외하고 별도의 매개변수가 필요 없다는 점, 다양한 분류 알고리즘을 약한 분류기로 사용할 수 있다는 장점이 있다. 이에 따라 본 논문은 부스팅 기법의 일종인 에이다부스트에 대해 이론적인 배경과 함께 알고리즘의 발전 과정을 알아본다. 그리고 에이다부스트에서 핵심인 결정트리를 바탕으로 약한 분류기를 구성하는 두 가지 방법인 선형 판별 분석(Linear Discriminant Analysis, LDA) 알고리즘[3],[4]과 CART(Classification And Regression Tree) 알고리즘[5]-[8]을 소개하고 서로의 성능을 비교하여 각각의 차이점에 대해 알아본다. LDA는 Fisher의 선형판별함수를 기반으로 데이터의 특성을 가장 잘 표현하는 속성의 선형조합을 구한다는 점에서 주성분 분석(Principal Component Analysis, PCA)과 유사한 점이 있으며, 부류 간의 관계를 고려하느냐에 따라 차이점을 구분한다. 이러한 LDA는 차원 축소 방법을 사용하며 여러 응용에 사용되어 좋은 결과를 보여왔다. 그리고 CART 방법은 최근에 소개된 앙상블 알고리즘에서 가장 널리 사용되고 있다는 점에서 비교 대상으로 선택하였다.

2장에서는 에이다부스트의 핵심 요소인 약한 분류기를 구성하는 CART와 LDA로 결정트리의 학습 알고리즘을 설명한다. 3장에서는 2장에서 소개된 2가지 약한 분류기를 적용하는 에이다부스트를 소개하고 이론적인 배경을 소개한다. 그리고 이진 분류기에서 다중 분류기로 발전하는 과정을 정리한다. 4장은 이들 2가지 약한 분류기를 에이다부스트 알고리즘에 적용한 결과를 여러 학습 데이터의 실험을 통해 비교해 본다.

Ⅱ. 배 경

결정트리는 계층적 구조, 분할, 학습, 그리고 예측 기준에 따라 종류를 분류하며, 시각적인 해석을 쉽게 한다는 특징을 가진다. 또한 결정트리는 과적합의 가능성 때문에 절단(pruning)이나 앙상블 기법과 결합하여 사용한다.

이 장에서는 결정트리를 구성하는 CART와 LDA의 배경이론을 설명하여 앙상블의 약한 분류기로 사용되는 과정으로 연결되도록 한다.

2-1 CART 결정트리

CART[5]는 Random Forest[9], GBM[10], XGBoost[8], LGBM[11]과 같은 앙상블 알고리즘에서 결정트리를 위한 기본 모델로 사용된다는 점에서 중요도를 가진다. 그리고 CART는 명목 범주형(nominal categorical) 데이터 보다는 순서가 있는 연속 수치형(continuous numerical) 데이터를 처리하기에 적합한 것으로 알려져 있다. 이는 명목 범주형 데이터가 레이블링(labeling) 과정에서 원-핫 인코딩(one hot encoding)을 사용함으로 인해, 결정트리를 깊은 편향(skewed) 이진 트리 형태로 만들어, 절단 과정을 거치고 나면 성능 저하로 이어지게 하는 경향이 있기 때문이다. 특히 앙상블 모델에서는 약한 분류기를 사용하는데 이때 전체 성능에 영향을 미치게 된다.

CART 알고리즘은 혼란(impurity)의 정도를 정의하면서 출발하는데 혼란 정도를 측정하기 위하여 식 (1)과 같은 C4.5에서의 엔트로피(entropy)와 CART에서의 지니 인덱스(gini index)가 가장 널리 사용된다. 여기서 c는 부류 수, m은 임의의 노드에서 전체 사건의 수를 의미한다. 엔트로피와 지니 인덱스를 설명하기 위해 그림 1과 같은 예로 식 (1)의 계산 과정을 설명한다.

$$\text{Entropy } H(m) = -\sum_{i=1}^{c} p(i|m) \cdot \log_2 p(i|m) \quad (1)$$

$$\text{Gini Index } G(m) = 1 - \sum_{i=1}^{c} p(i|m)^2$$

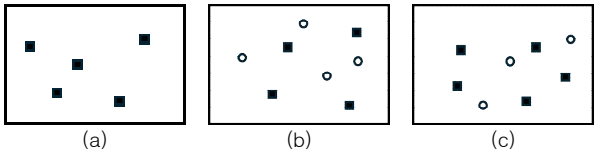


그림 1. 임의의 노드에서 사건들의 예
Fig. 1. Examples of events at random nodes

그림 1의 (a)에서는 하나의 부류만으로 구성되어 가장 낮은 혼란도(0)를 가지며 엔트로피와 지니 인덱스 모두가 0의 값을 가진다.

$$H(t) = -\left(\frac{5}{5}\log_2\frac{5}{5} + \frac{0}{5}\log_2\frac{0}{5}\right) = 0,$$

$$G(t) = 1 - \left(\left(\frac{5}{5}\right)^2 + \left(\frac{0}{5}\right)^2\right) = 0$$

다음으로, 그림 1의 (b)는 가장 높은 혼란도를 가진다. 여기서, 2개의 부류일 때 가장 높은 엔트로피인 1을, 지니 인덱스에서 가장 높은 값인 0.5를 가지게 된다.

$$H(t) = -\left(\frac{4}{8}\log_2\frac{4}{8} + \frac{4}{8}\log_2\frac{4}{8}\right) = 1.0$$

$$G(t) = 1 - \left(\left(\frac{4}{8}\right)^2 + \left(\frac{4}{8}\right)^2\right) = 0.5$$

마지막으로, 그림 1의 (c)는 다음과 같이 혼란도를 계산할 수 있다.

$$H(t) = -\left(\frac{5}{8}\log_2\frac{5}{8} + \frac{3}{8}\log_2\frac{3}{8}\right) = 0.95$$

$$G(t) = 1 - \left(\left(\frac{5}{8}\right)^2 + \left(\frac{3}{8}\right)^2\right) = 0.47$$

CART에서는 이와 같은 혼란도를 이용해 식 (2)와 같은 정보 이득(Information Gain, IG) 함수를 구하게 되는데, 이는 루트 노드가 왼/오른쪽 노드로 분리될 때 혼란도의 변화량을 의미한다. CART 결정트리 알고리즘은 가장 커다란 IG로 트리를 구성해 나가는 과정이다. 이와 관련해서 그림 2와 같은 예를 들 수 있다. 여기서 P: 부모, L: 왼쪽, R:오른쪽이다.

$$IG = G(P) - \frac{m_L}{m} \times G(L) - \frac{m_R}{m} \times G(R) \quad (2)$$

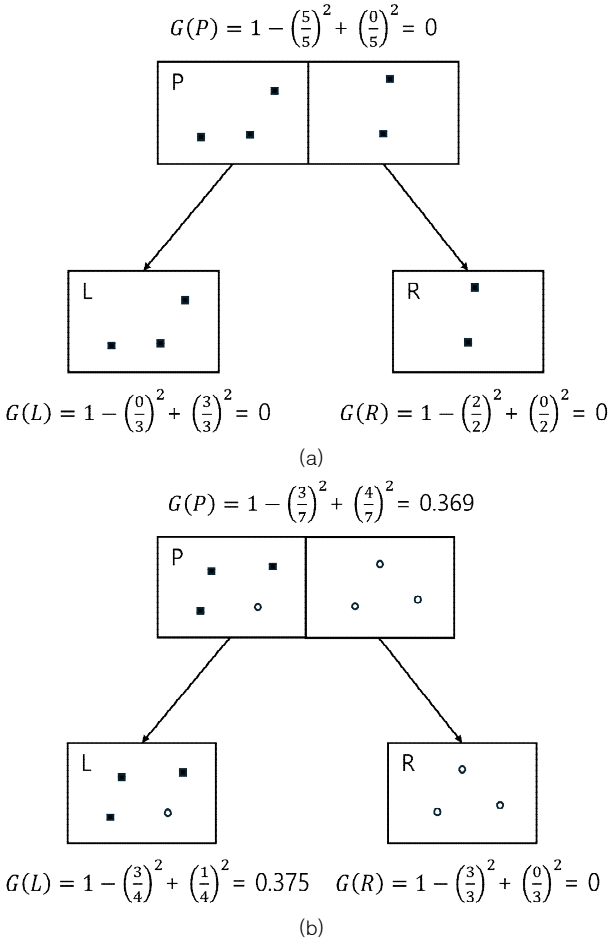


그림 2. 결정트리에서 정보 이득(IG)의 예
Fig. 2. Example of information gain (IG) in a decision tree

그림 2의 (a)를 식 (2)에 적용하면, 정보 이득 함수는 $IG = 0 - \frac{3}{5}\times 0 - \frac{2}{5}\times 0 = 0$이고, (b)에서 정보 이득 함수는 $IG = 0.369 - \frac{4}{7}\times 0.375 - \frac{3}{7}\times 0 = 0.155$이다. 따라서 ((a)의 IG < (b)의 IG))이므로 CART의 임의 노드에서는 (b)를 선택하여 결정트리를 구성하게 된다.

2-2 LDA 결정트리

선형 판별 분석(LDA)은 차원 축소 기법을 기본으로 하여 데이터 분류 알고리즘을 구성한다. LDA와 함께 주성분 분석(PCA) 기법도 차원 축소를 사용하는데[3], LDA는 부류 간의 관련 정보를 이용하는 반면(지도 학습), PCA는 부류는 상관없이 전체 데이터의 분산을 최대화하는 방향을 찾는다(비지도 학습). PCA는 데이터 집합을 구성하는 속성들이 이루는 공분산 행렬의 고유벡터(eigenvector)와 고유값(eigenvalue)을 계산하여 주요 성분을 찾는다. 그리고 주로 데이터의 구조를 이해하고 시각화하거나 노이즈를 제거하기 위해 사용된다. 두 기법의 차이점은 LDA는 부류 간/내 분산을 최대화/최소화

하고, PCA는 전체 데이터의 분산을 최대화한다[4]. 그리고 LDA는 최대 부류수−1개의 판별 축(W)을 가지며, PCA는 데이터 차원 수만큼의 판별 축을 가진다.

본 논문에서 사용하는 LDA 분류 알고리즘의 아이디어는 주어진 데이터 집합(X)에 대해 부류 간의 분리도를 최대로 하는 직선(2차원) 또는 초평면(3차원 이상)을 식 (3)과 같은 Fisher의 선형 판별 함수를 이용해 찾는다. 그리고 이 직선(초평면) P에 데이터 집합의 각 사건들을 투영(W^TX)할 때 차원 축소가 이뤄지고 부류 간을 분류도 이루어진다.

$$\frac{W^TBW}{W^TVW}=\frac{W^T[\sum_i(\overline{X_i}-\overline{X})(\overline{X_i}-\overline{X})^T]W}{W^T[\sum_i\sum_j(X_{ij}-\overline{X_i})(X_{ij}-\overline{X_i})^T]W} \quad (3)$$

$$B=\sum_i(\overline{X_i}-\overline{X})(\overline{X_i}-\overline{X})^T \quad (4)$$

$$V=\sum_i\sum_j(X_{ij}-\overline{X_i})(X_{ij}-\overline{X_i})^T \quad (5)$$

식 (4)는 각 부류 간의 거리로 이루어지는 공분산 행렬이고, 식 (5)는 각 부류 안에 속하는 사건들의 거리를 의미하는 공분산 행렬이다. 따라서 분류에 가장 유리한 판별 축은, 식 (4)는 최대가 되고 식 (5)는 최소가 되는 즉, B/V가 최대가 되면 된다. 이를 위해 B/V를 식 (3)과 같이 변형하고 Cauchy-Schwartz 부등식을 이용하여 B/V의 최대 벡터를 구한다. 이때 가장 큰 고유값을 가지는 판별 축이 W가 되며 이 축에 학습 데이터의 사건들을 투영한 다음, 식 (1)의 엔트로피 함수를 이용하여 분류 임계값(Th)을 구함으로써 직선(초평면), 즉 임의 노드에서 부류를 분류하는 분류면을 구한다. 각 노드는 분류면을 나타내는 가중치 벡터(W)와 임계값으로 구성된다.

LDA 분류 알고리즘은 각 영역에 하나의 부류만 남을 때까지 각 노드에서 입력된 사건들을 (6)식과 같이 P를 기준으로 왼쪽과 오른쪽(PL, PR) 노드로 분할하는 과정을 반복하며 이진 결정트리를 구성해 나간다.

$$\text{분류면}(\overline{P}) = \{X \mid X\in R^n \text{ and } W^TX = Th\} \quad (6)$$
$$P^R = \{ X \mid X\in H^n \text{ and } W^TX \geq Th \}$$
$$P^L = \{ X \mid X\in H^n \text{ and } W^TX < Th \}$$

LDA는 그동안 얼굴 인식, 문서 분류, 의료 데이터 분석 등 다양한 응용 분야에서 사용되어 좋은 결과를 보여왔다.

Ⅲ. CART와 LDA 결정트리를 약한 학습기로 사용하는 에이다부스트

3-1 에이다부스트

Yoav Freund[1]는 약한 학습모델을 점진적으로 강하게 하여 강한 학습기로 만들어 가는 에이다부스트 알고리즘을 제안하였다. 이는 학습모델에서 오 분류된 사건의 가중치를 높임으로써 이 사건이 다음 모델의 학습 데이터에 포함되도록 샘플링 확률을 높인다. 그리고 정해진 횟수만큼 이 과정을 직렬방식으로 반복하여 정확도가 증가하도록 한다. 입력 사건에 대한 부류는 마지막 추정 과정에서 여러 모델의 결과들을 결합하여 결정한다. 이러한 학습과 추정 과정을 자세히 살펴보면 다음과 같다. 여기서 학습 데이터는 m개(i=1,⋯,m)이며 속성 벡터는 k개($x_1,...,x_k$)이고 부류 값은 y={+1, −1}로 정한다.

먼저 일양 분포에 따르는 가중치(D) 확률과 복원 추출 방법으로 학습 사건들에서 정해진 수만큼(m개) 샘플링 한다.

$$D_1(i)=1/m$$

결정트리, SVM, LDA 등의 분류 알고리즘을 절단 등의 방법으로 정확도가 0.5보다 약간 우수한 정도의 약한 학습기를 구성하고 이를 모델로 만든 후, 샘플된 데이터를 이 모델(h)로 학습하여 각각 추정치($\hat{y}=h_1(x_i)$)를 구한다.

분류된 결과($\hat{y}$)에 따라 식 (7), (8), (9)의 순서대로, 다음 모델에 입력 데이터를 선택하기 위한 가중치를 구한다.

$$\varepsilon_1=\sum_{i=1}^{m}D_1(i)I(y_i\neq\hat{y_i}) \quad (7)$$

$$\alpha_1=\frac{1}{2}\ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right) \quad (8)$$

$$D_2(i)=\frac{1}{z}D_1(i)\exp(-\alpha_1y_1\hat{y_1}) \quad (9)$$

여기서 $I(A\neq B)$는 $If(A\neq B)$ 1, $Else$ 0인 함수로 식 (7)은 오 분류된 사건만의 가중치를 모두 더한 것이고, 식 (8)은 그림 3과 같이 에러가 0.5보다 약간 좋은 약한 분류기($\varepsilon<0.5$)에서 $\alpha>0$이 되도록 한다. 또한 식 (9)는 다음 모델을 위한 샘플 가중치를 구하는 것으로 오 분류된 사건에 해당하는 가중치를 높여 다음 학습에 선택될 확률을 높이고, 올바르게 분류된 사건에는 가중치를 내리기 위한 정책을 반영한 식이다. 그리고 z는 일반화 상수로 각 사건의 가중치를 모두 더한 값이며 가중치(D)가 확률이므로 [0,1] 범위에서 결정되도록 하는 역할을 한다.

여러 개(T개)의 가설 모델들로부터 정보를 모아 강한 분류

기가 되기 위한 결과를 산출하는 과정으로 다음과 같다.

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (10)$$

식 (10)에서 $h_t(x)$는 t번째 모델의 추정치이며 α와는 반비례 관계가 있다. 즉 추정치에 오류가 많으면 α는 작고, 반대로 오류가 적으면 α는 크게 되어 이를 전부 합하면 올바르게 판단된 추정치가 더 큰 역할을 하도록 한다는 것이다.

3-2 샘플링을 위한 가중치의 결정

식 (7)의 오 분류된 가중치의 합(ε)을 가지고 식 (8)에서 모델 가중치(α)를 구하고 식 (9)의 샘플 가중치가 구해지는 과정을 Raul Rojas[6]는 다음과 같이 설명하였다. 식 (10)의 출력으로부터 식 (11)과 같은 t개 약한 분류기의 선형 결합이 유도된다. 여기서 $H_t(x_i)$는 $\hat{y_i}$를 의미한다. 부류가 $y_i = \{+1, -1\}$일 때 전체 오류의 기대치는 식 (12)로 나타내진다.

$$H_{t-1}(x_i) = \alpha_1 h_1(x_i) + \alpha_2 h_2(x_i) + \ldots + \alpha_{t-1} h_{t-1}(x_i) \quad (11)$$
$$H_t(x_i) = H_{t-1}(x_i) + \alpha_t h_t(x_i)$$

$$E = \sum_{i=1}^{m} \exp(-y_i \hat{y_i}) \quad (12)$$

식 (12)는 $y_i \hat{y_i} > 0$이면 즉 $y_i = \hat{y_i}$이면 E는 감소하고, $y_i \hat{y_i} < 0$이면 즉 $y_i \neq \hat{y_i}$이면 E는 증가해야 한다는 성질을 반영한 것이다. 식 (12)에서 $\hat{y_i}$에 식 (11)을 대입하면 다음과 같이 정리된다.

$$E = \sum_{i=1}^{m} \exp\{-y_i (H_{t-1}(x_i) + \alpha_t h_t(x_i))\} \quad (13)$$

$$E = \sum_{i=1}^{m} D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\},\ D_i^{(t)} = \exp\{-y_i H_{t-1}(x_i)\}$$

$$E = \sum_{y_i = h_t(x_i)} D_i^{(t)} e^{-\alpha_t} + \sum_{y_i \neq h_t(x_i)} D_i^{(t)} e^{\alpha_t},\ \because\ y_i = \{+1, -1\}$$

식 (13)의 마지막 식에서 앞의 항은 $y_i h_t > 0$일 때 즉, 모델에서 올바른 부류를 산출할 때($y_i = \hat{y_i}$)를 의미하고, 뒤의 항은 반대로 $y_i h_t < 0$일 때 즉, 모델이 틀린 결과를 산출한 때($y_i \neq \hat{y_i}$)를 의미한다. 예를 들어 앞항이 $D_1^{(t)} e^{-\alpha_t} + D_3^{(t)} e^{-\alpha_t} + \ldots$이고, 뒤항이 $D_2^{(t)} e^{\alpha_t} + D_4^{(t)} e^{\alpha_t} + \ldots$라는 예를 들 수 있다.

E를 최소로 하는 가중치 α^t는 식 (14)와 같이 미분을 이용해 구한다.

$$\left(\underset{\alpha^t}{argmin}\ E \right) = D_c e^{-\alpha_t} + D_e e^{\alpha_t} \quad (14)$$

여기서 $D_c = D_1^{(t)} + D_3^{(t)+\cdots},\ D_e = D_2^{(t)} + D_4^{(t)+\cdots}$

$$\frac{dE}{d\alpha_t} = -D_c e^{-\alpha_t} + D_e e^{\alpha_t} = 0$$

$$D_e e^{\alpha_t} = D_c e^{-\alpha_t} = D_c \frac{1}{e^{\alpha_t}}$$

$$D_e (e^{\alpha_t})^2 = D_c$$

$$e^{\alpha_t} = \sqrt{\frac{D_c}{D_e}} \quad \because \text{항상 } e^{\alpha t} > 0$$

이제 양변에 $\log_e$를 취하면, $\alpha_t = \ln\left(\sqrt{\frac{D_c}{D_e}}\right) = \ln\left(\frac{D_c}{D_e}\right)^{1/2}$

$$\therefore\ \alpha_t = \frac{1}{2}\ln\left(\frac{D_c}{D_e}\right)$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{D - D_e}{D_e}\right),\ \because\ D = D_c + D_e$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$$

이때 $\varepsilon_t = \frac{D_e}{D}$인데 항상 D=1이므로 $D_e = \sum_{y_i \neq h_t} D_i^{(t)}$이다. 따라서 $\varepsilon_t = \sum_{i=1}^{m} D_i^{(t)} I(y_i \neq \hat{y_i})$로 나타내 진다.

약한 분류기의 가정에 따라 $\varepsilon_t \leq 0.5$, 즉 오류가 0.5보다 작아야 한다. 식 (8)을 그래프로 나타낸 그림 3에서 지금까지 살펴보았던 α와 ε의 관계를 정리할 수 있는데 $\varepsilon_t = 0.5$에서 $\alpha = 0$이고, $\varepsilon_t = 0$에서 $\alpha = \infty$을 가지고, $\varepsilon_t \leq 0.5$ 영역에서 α와 ε는 반비례 관계가 있다. 이는 t번째 모델의 오류(ε)가 크면 결과에 영향을 줄이기 위해 α는 작은 값을 가지게 되고, 반대로 오류가 적으면 α는 큰 값을 가진다는 것이다.

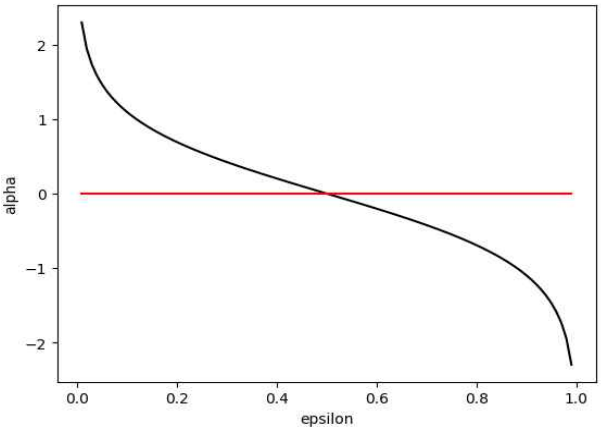


그림 3. α(alpha)와 ε(epsilon)의 관계
Fig. 3. Relationship between α(alpha) and ε(epsilon)

3-3 다중 분류 알고리즘

Freund와 Schapire[2]는 이중 부류에서 다중 부류를 가지는 분류기로 쉽게 확장될 수 있도록 y={-1, 1}를 y={0, 1}로 테스트 출력값을 변형하였다. 이 과정의 핵심은 가설 모델들로부터 정보를 모아 결과를 산출하는 식 (10)을 식 (15)로 변환하는 것이다. 식 (15)에서는 각 부류에 대해 가중치(α^t)의 합을 구하고, 이 값이 가장 큰 부류를 입력 x에 대한 추정값(H(x), $\hat{y_i}$)으로 할당하고 있다.

$$H(x) = \underset{c}{\arg max}\left(\sum_{t=1}^{T}\alpha^t I(h^t(x)=c)\right),\ \text{c=1,…,부류 수} \quad (15)$$

식 (15)로 부터 y={0, 1}를 가지는 이진 분류 알고리즘은 다중 분류 알고리즘으로 쉽게 확장할 수 있는 기반이 되었는데, 이와 관련해 Ji Zhe 등[7]이 다음과 같은 다중 분류 알고리즘인 SAMME를 제안하였다.

SAMME(Stagewise Additive Modeling using a Multi-class Exponential loss Function) 알고리즘

- y={0,1} 문제를 y={0,…,c-1} 문제로 확대
- Freund 모델의 err(t)<0.5 조건이 err(t)<1-1/c로 변경. 만일 c=3이라면 err(t) < 2/3(0.67)

1. 초기화 : 가중치 $D_i = 1/m,\ i = 1,...,m$

2. For t=1 to T
   가중치 확률(D_i)에 따라 훈련데이터(x_i)를 선택하고 분류기($H^{(t)}(x_i)$)에 입력한다.

$$err^{(t)} = \frac{\sum_{i=1}^{m} D_i I(c_i \neq h^{(t)}(x_i))}{\sum_{i=1}^{m} D_i}$$

$$\alpha^{(t)} = \ln\frac{1-err^{(t)}}{err^{(t)}} + \ln(c-1) \quad (16)$$

$$D_i = D_i \cdot \exp(\alpha^{(t)} \cdot I(c_i \neq h^{(t)}(x_i)))$$

   D_i의 합이 1이 되도록 일반화한다.

3. 출력

$$H(x) = \underset{c}{\arg max}\left(\sum_{t=1}^{T}\alpha^t I(h^t(x)=c)\right)$$

식 (16)은 Freund 모델에서 식 (8)이 변경된 부분인데 이를 이용해 $\alpha^{(t)} > 0$에서 $err^{(t)} < 1-\frac{1}{c}$임을 식 (17)과 같이 증명할 수 있다.

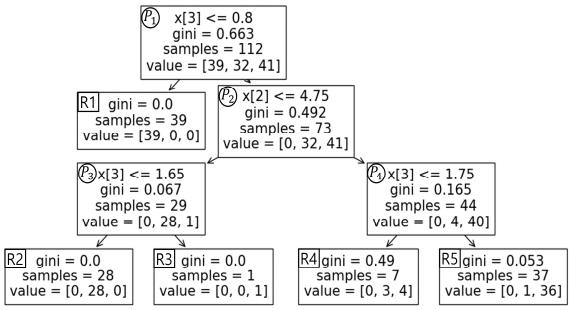
$$\alpha^{(t)} = \ln\frac{1-err^{(t)}}{err^{(t)}} + \ln(c-1) > 0 \quad (17)$$

$$\frac{1-err^{(t)}}{err^{(t)}} > \frac{1}{c-1},\ \frac{1}{err^{(t)}} > \frac{c}{c-1},\ \therefore\ err^{(t)} < 1-\frac{1}{c}$$

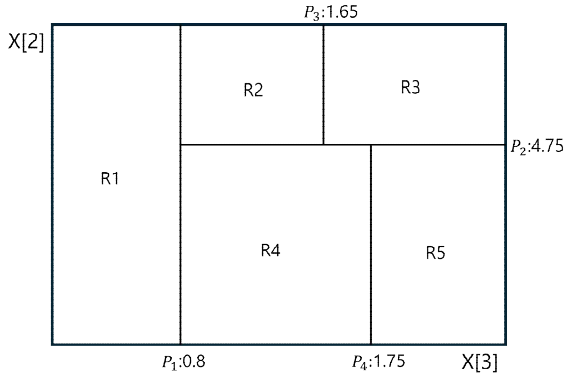
Ⅳ. 실 험

본 논문의 에이다부스트에서 약한 분류기로 사용하는 CART와 LDA에 대해서는 2장에서 서로의 특징을 설명하였다. 이에 대한 실험 결과를 비교하기 위해 학습 과정을 먼저 살펴본다. 우선 지니 계수를 기반으로 하는 결정트리는 이진 트리구조를 가지며, 그림 4에서 IRIS 데이터를 약한 분류기에 입력하여 학습하는 과정을 예로 설명한다. 이 데이터는 4개의 속성값을 가지나 간단히 2, 3번째 속성(X[2], X[3])만으로 결정트리를 구성하는 노드의 예로, 그림 4의 (a)는 깊이를 3으로 하는 경우를 나타낸다. 각 노드 중에 루트 노드의 첫째 줄 “X[3] <= 0.8”은 3번째(4개중) 속성이 0.8보다 작거나 같을 때를 의미하는 것으로 왼/오른쪽의 결정트리로 분리하기 위한 최적의 분리점을 의미한다. 둘째 줄은 이때의 지니 계수를 의미하며, 세 번째는 이 노드에서의 사건 수를 의미한다. 마지막으로 “value=[39,42,41]”은 IRIS 데이터의 부류 중 부류 1의 사건 수가 39, 부류 2, 3의 사건 수가 각각 42, 41을 의미한다. 마지막 말단 노드에서 “value=[0,3,4]”와 같이 부류 2와 부류 3이 같이 존재하여 임의의 노드에서 부류가 완전하게 분류되지 않은 것을 볼 수 있는데 이는 트리의 깊이를 3으로 제한하여 학습 중에 절단이 일어났기 때문이다. 그리고 그림 4의 (b)에서는 (a)의 결정트리로 분류된 데이터 영역을 의미하는 것으로 루트 노드로부터 말단 노드에 이르기까지 분리면에 따라 영역이 분류된 것을 볼 수 있다. 여기서 분류면이 각 축에 따라 직각으로 분류하고 있음을 볼 수 있다.

반면 그림 5의 (a)는 임의의 데이터를 LDA로 학습한 경우의 결정트리를 낸다. CART의 결정트리와 같이 이진 트리 형식을 가지며 임의의 노드는 식 (2)에서 B/V를 최대로 하는 공분산 행렬 W와 식 (5)와 같이 투영(W^TX)된 포인트 들 중에서 최적의 분류면을 결정하는 스칼라값인 Th_i로 구성된다. 각 데이터는 식 (6)과 같이 왼/오른쪽으로 분류하며 결정트리를 구성해 간다. 그림 5에서 (b)는 (a)의 결정트리에 따라 분류된 영역을 나타내는데 그림 4 (b)의 영역과 비교하면 분류면이 속성축과 관계없이 임의의 각을 가지는 특성을 가진다. 이는 분류면의 각은 각 노드에서 W와 T_i에 의해 결정되는 LDA 결정트리의 특징으로 그림 4의 결정트리와 비교하여 성능에 영향을 미친다. 이에 대해서는 실험 데이터로 결과를 비교한다.



(a) CART를 사용한 결정트리
(a) Decision tree using CART



(b) CART로 영역 분할
(b) Region segmentation using CART

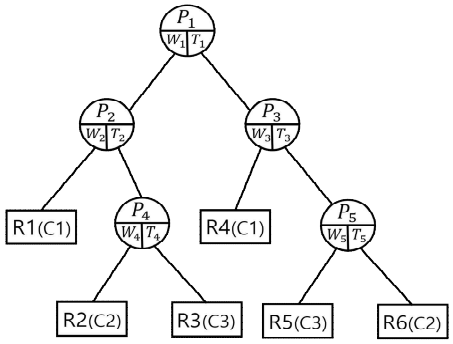
그림 4. CART 결정트리의 학습
Fig. 4. Learning of CART decision tree

실험을 위해 사용한 데이터는 표 1에 표시된 바와 같이 UCI 기계저장소[12]의 데이터(bs, car_num)와 인간의 수면 상태를 11개의 센서로 수집한 후 양자화한 데이터(sleep)을 사용하였다. 이들 데이터의 실험 결과를 위해 10겹 교차검증 방법을 사용하였고 그 결과가 표 2에 나타나 있다. 그리고 각 데이터에 따라 결과들의 비교를 위한 그래프가 그림 6에 나타나 있다.

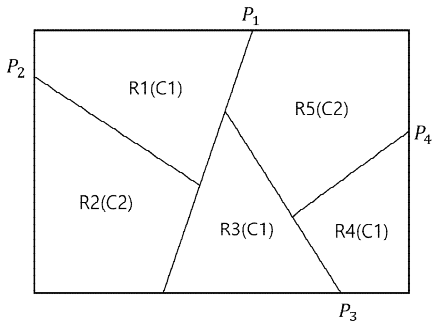
실험 결과 정도의 차이는 있으나 실험한 모든 데이터에서 LDA의 결과가 우수한 것을 볼 수 있다. 이는 CART와 유사한 방식인 속성축과 직교로 분류하는 C4.5를 이용한 전 연구[13],[14]에서도 확인할 수 있었는데, 분류면을 속성축에 고정하는 것 보다는 속성값에 따라 자유롭게 각을 가지도록 하여 근사 최적을 분류면을 구한 결과로 해석한다.

표 1. 실험에 사용된 데이터와 이의 특성
Table 1. Data used in the experiment and its characteristics

Data	Attribute	Event	Class
Balance & Scale(bs)	4	625	3
Car Evolution(car_num)	6	1728	4
Sleep Stage Scoring(sleep)	11	799	11



(a) LDA를 사용한 결정트리
(a) Decision tree using LDA



(b) LDA로 영역 분할
(b) Region segmentation using LDA

그림 5. LDA의 결정트리 학습
Fig. 5. Decision tree learning of LDA

표 2. 에이다부스트에 약한 분류기로 2가지 결정트리를 적용하여 실험한 결과
Table 2. The results of the experiment using AdaBoost with two types of decision trees as weak classifiers

Data	bs	car_num	sleep
CART(DT)	87.50	92.94	87.24
LDA	94.24	93.63	89.49

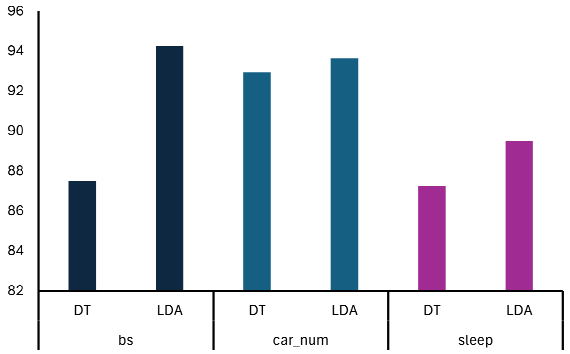


그림 6. 실험 데이터에 2가지 결정트리 알고리즘을 적용하여 10겹 교차로 성능을 비교하는 표 2를 나타내는 그래프
Fig. 6. A graph representing Table 2, which compares the performance of two decision tree algorithms on the experimental data using 10-fold cross-validation

V. 결 론

지금까지 에이다부스트 알고리즘의 약한 분류기에서 출발하여 오류를 포함하는 사건을 강화하는 방법을 반복하여 강한 분류기로 진화하는 과정을 살펴보았다. 요즘 분류 알고리즘의 성능 향상을 위해서 뛰어난 하나의 알고리즘보다는 여러 분류기의 힘을 합하여 하나의 결론에 이르게 하는 앙상블이 대세가 되고 있다.

본 논문에서는 에이다부스트의 약한 분류기를 위해 일반적으로 사용되는 CART와 CART 대신 Fisher의 선형판별함수로부터 유도된 LDA로 대체하는 2가지 결정트리 구성 방법과 이를 앙상블에 적용하는 이론적인 배경에 대해 알아보았다. 그리고 2가지 결정트리를 적용할 때의 결과에 대해서도 알아보았고 실험에 사용한 학습 데이터에서 LDA의 결과가 다소 우수한 것으로 나타났다. 이러한 결과는 C4.5, CART와 비교하여, LDA가 학습이 이루어짐에 따라 결정트리의 노드를 결정하는 과정에서 선형 분리면으로 공간을 분할하는 방법의 차이에서 오는 결과라고 해석한다.

최근의 앙상블 알고리즘 중에 이용도가 높고 따라서 미래의 대세가 될 Random Forest, GBM, XGBoost, LGBM 알고리즘과 결합하여 성능을 높이려는 연구가 다방면으로 이루어지고 있다[15],[16]. 이러한 연구의 하나로 앙상블 알고리즘들이 CART 결정트리를 기본으로 하고 있으나 이를 본 논문에서 소개한 LDA와 같은 결정트리도 적용해 볼 수 있을 것이다. 또한 전처리로 LDA를 이용하여 차원 축소를 한 후 딥러닝과 같은 다양한 학습을 수행하는 연구들도 앞으로의 연구 주제가 될 수 있을 것으로 본다.

참고문헌

[1] Y. Freund, “Boosting a Weak Learning Algorithm by Majority,” *Information and Computation,* Vol. 121, No. 2, pp. 256-285, September 1995. https://doi.org/10.1006/inco.1995.1136

[2] Y. Freund and R. E. Schapire, “A Short Introduction to Boosting,” *Journal of Japanese Society for Artificial Intelligence,* Vol. 14, No. 5, pp. 771-780, September 1999.

[3] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, “Linear Discriminant Analysis: A Detailed Tutorial,” *AI Communications,* Vol. 30, No. 2, pp. 169-190, 2017. https://doi.org/10.3233/AIC-170729

[4] W. Loh and N. Vanichsetakul, “Tree-structured Classification via Generalized Discriminant Analysis,” *Journal of the American Statistical Association*, Vol. 83, No. 403, pp. 715-728, 1988. https://doi.org/10.2307/2289295

[5] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, New York: Chapman and Hall/CRC, 1984. https://doi.org/10.1201/9781315139470

[6] R. Rojas, “AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting,” Freie University, Berlin, Tech Rep 1, Vol. 1, pp. 1-6, 2009. https://doi.org/10.36227/techrxiv.172107276.63524590/v1

[7] J. Zhu, H. Zou, S. Rosset, and T. Hostie, “Multi-Class AdaBoost,” *Statistics and Its Interface*, Vol. 2, pp. 349-360, 2009. https://doi.org/10.4310/sii.2009.v2.n3.a8

[8] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016. https://doi.org/10.1145/2939672.2939785

[9] L. Breiman, “Random Forests,” *Machine Learning*, Vol. 45, pp. 5-32, 2001. https://doi.org/10.1023/A:1010933404324

[10] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *The Annals of Statistics*, pp. 1189-1232, October 2001. https://doi.org/10.1214/aos/1013203451

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, ... and T. Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *31st Conference on Neural Information Processing Systems*, pp. 3149-3157, 2017.

[12] University of California, Irvine. UCI Machine Learning Repository [Internet]. Available: https://archive.ics.uci.edu/ml/index.php.

[13] J. C. Lee, “Comparison of Two Classifiers for Handling Incomplete Data with Decision Trees: C4.5, SVM,” *Journal of Knowledge Information Technology and Systems*, Vol. 18, No. 1, pp. 71-80, 2023. https://doi.org/10.34163/JKITS.2023.18.1.008

[14] J. C. Lee, “Algorithm to Handling Incomplete Data from Decision Tree Using SVM,” *Journal of Knowledge Information Technology and Systems,* Vol. 17, No. 6, pp. 1145-1153, 2022. https://doi.org/10.34163/JKITS.2022.17.6.006

[15] P. Deng, H. Wang, T. Li, S. Horng, and X. Zhu, “Linear Discriminant Analysis Guided by Unsupervised Ensemble Learning,” *Information Sciences*, Vol. 480, pp. 211-221, April 2019. https://doi.org/10.1016/j.ins.2018.12.036

[16] R. Garf, M. Zeldovich, and S. Friedrich, “Comparing Linear Discriminant Analysis and Supervised Learning Algorithms for Binary Classification—A Method Comparison Study,” *Biometrical Journal*, Vol. 66, No. 1, pp. 1-20, January 2024. https://doi.org/10.1002/bimj.202200098

이종찬(Jong Chan Lee)

1988년 : 충남대학교 계산통계학과 (학사)
1990년 : 충남대학교 대학원 전산학과 (석사)
1996년 : 충남대학교 대학원 전산학과 (박사)

1996년~현 재: 청운대학교 컴퓨터공학과 교수
※관심분야 : 딥러닝, 패턴분류, 정보보호, 데이터압축