기 술 동 향

머신러닝 기술의 핵심 알고리즘과 재료·가공 문제에 적용 II

김영석[1,#]

1. 경북대학교 기계공학부 교수

Key Algorithms of Machine Learning and its Application to Material Processing Problems II

Y. S. Kim

1. School of Mechanical Engineering, Kyungpook National University

1. 서 론

최근 사회의 다양한 문제 현상의 파악 뿐 아니라 의료 진단, 자율 주행, 이미지 인식, 신물질 합성, 제조 기술, 금융 통계, 부동산 가격 예측 등 다양한 분야에서 문제 해결을 위한 도구로 AI(artificial intelligence, 인공지능)가 널리 적용되고 있다[1-7]. 그 중에서도 AI 기술분야의 하나인 머신러닝(machine learning, 기계학습)은 ANN(artificial neural network, 인공신경망), 딥러닝(deep learning) 등과 함께 복잡한 경계조건 하에서 다양한 변수들이 상호작용하고 있는 재료의 소성·가공 문제 해결[8-14] 그리고 구동설비의 고장 진단이나 수명 예측 등에 효율적으로 적용되고 있는 사례가 많아지고 있다[15].

이 머신러닝 기술은 그 효용성으로부터 프레스 공정, 절삭가공 공정, 용접 공정, 다이캐스팅 등 뿌리산업 모든 분야에 적용이 확대되어 가고 있다[16]. 국가적으로도 중소벤처기업부가 지원하여 2021년 12월에 구축된 인공지능 중소 벤처 플랫폼(www.kamp-ai.kr)[17a]에는 국내 제조현장에서 축적된 제조 데이터를 활용, 중소기업의 설비·공정에서 발생하는 문제를 AI 기술을 접목하여 해결한 사례들과 많은 데이터셋이 구축되어 있다. 또한 이 플랫폼에는 각 기업들이 해결하려는 제조현장의 다양한 현장 이슈(pain point)를 AI 전문가 시스템을 이용하여 코딩 없이 AI 분석을 수행할 수 있는 툴을 제공하고 있다[17b].

이 머신러닝은 일반적으로 (1)문제의 정의· 설계기획 (2)데이터 수집 (3)데이터 전처리(정제, 가공, 표준화, 학습/평가 데이터 분리 등) (4)모델링(학습 모델선정, 모델학습) (5)모델 평가 및 예측(회귀, 분류) (6) 결과의 조치 및 성능개선 작업 등의 일련의 과정을 거쳐 수행된다[5, 6].

본 해설논문에서는 지난번 해설논문[19]에 이어서 산업현장의 다양한 문제들에 널리 활용되고 있는 머신러닝 기술의 핵심 알고리즘을 통계학적 복잡한 수식을 배제하고 알기 쉽게 설명하여 산업현장 엔지니어들의 동 기술에 대한 이해를 높이고자 한다.

본 해설논문에서는 AdaBoost, Gradient Boosting Machine, Support Vector Machine의 핵심 알고리즘에 대해서 알기 쉽게 설명하였고 저자가 GitHub에 올린 파이썬(python) 프로그램을 통해서 어떻게 알고리

즘이 구현되고 있는지를 나타내었다.

(https://github.com/yskim9574/DFclass-2023/...) 여기서 기술된 알고리즘에 대한 프로그램들은 구글의 Colab(https://colab.research.google.com/)에서 구동하였다.

2. 머신러닝 주요 알고리즘

2.1 AdaBoost

부스팅 알고리즘(Boosting algorithm)은 여러 개의 약한 학습기(weak learner 또는 약한 모형)를 순차적으로 학습-예측하면서 잘못 예측한 데이터에 가중치를 부여해 오류를 개선(오류를 최소화해서)해서 강한 학습기(strong learner)를 만들어가는 학습방식이다. 부스팅 알고리즘은 대표적으로 AdaBoost, GBM (Gradient Boosting Machine), XGBoost 등이 있다.

여기서 AdaBoost 또는 Adaptive Boosting은 Yoav Freund와 Robert Schapire가 제안한 기계학습 메타 알고리즘으로 앙상블 부스팅 분류기 중 하나이다[18a, b]. AdaBoost는 기존의 이진 분류 문제를 더 일반화한 다중 분류에 대한 알고리즘에 기초하고 있고, 분류(classification) 문제와 회귀(regression) 문제에 모두 적용 가능하다.

여기서 앙상블(ensemble)이란 단일 학습 모델을 적용하기 보다는 다양한 학습 모델을 조합 적용하여 더 나은 예측(분류나 회귀) 성능을 얻어내는 방법이다. 이 앙상블은 정확도가 높은 강한 모델을 한 개 사용하는 것보다, 정확도가 낮은 약한 모델을 여러 개 조합하는 방식이 정확도가 높다는 가정 하에 시작된 기법이다. 앙상블은 방식에 따라서 배깅(bagging)과 부스팅(boosting) 방법으로 분류된다.

AdaBoost는 초기 모형을 노드 하나에 두 개의 리프(leaf)를 갖는 그루터기(stump) 형태의 약한 학습기로 설정하고 매 스텝마다 가중값(sample weight)를 이용하여 이전 모형의 약점을 보완하는 새로운 모형을 순차적으로(sequential) 조정한 뒤 최종적으로 이들 약한 학습기들을 선형 결합하여 개별 약한 학습기보다 더 정확하게 분류하는 모형(강한 학습기)을 생성시키는 알고리즘이다(그림 1).

전 호[19]에서 다룬 Random forest에서는 각 트리의 모양은 다르지만 각각 투표할 때의 중요도가 모두 동일하기 때문에 트리를 표현할 때 동일한 크기로 나타내었다. 하지만 AdaBoost에서는 각 트리별

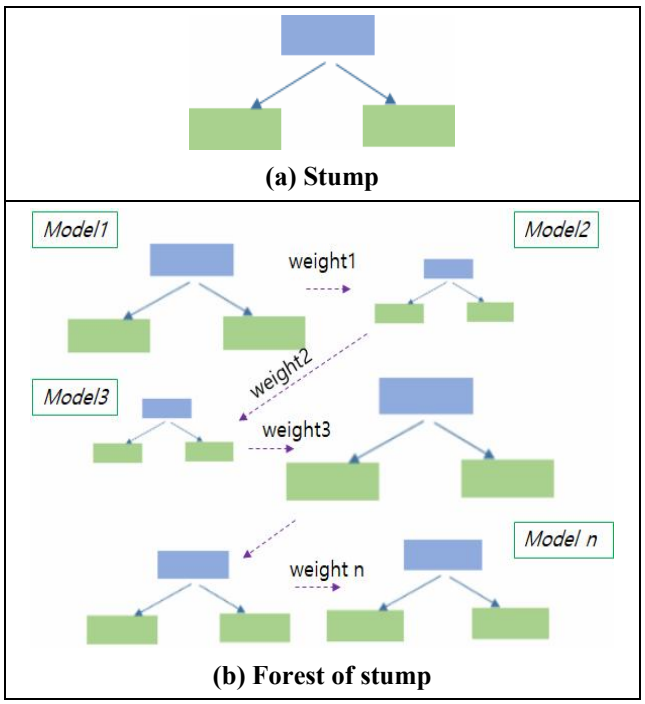


Fig. 1 (a) Stump; (b) Forest of stumps

(주) (a) Stump(그루터기)-노드 하나에 두개의 리프(leaf)를 지닌 트리, (b) Forest of stumps(그루터기 숲) -크기가 큰 것은 가중값이 더 큰 그루터기를 뜻한다. 가중값이 크다 는 것을 Amount of Say가 크다고 표현하며, 결과에 미치는 영향이 크다는 뜻이다.

중요도가 차이가 나기 때문에 그림 1(b)에서와 같이 각 그루터기의 크기가 다르다는 것과 부스팅 특징에 따라 각 그루터기는 생성될 때 이전 그루터기의 정보를 참고하여 종속적이고 연속적으로 모델이 생성되고 있음을 알 수 있다.

AdaBoost는 여러 질문을 통해 데이터를 분류하는 결정트리와 달리, 노드 하나에 두 개의 잎 노드 구조(또는 두 개의 가지 구조 즉, 하나의 속성을 2개의 데이터로 분류하는 구조)를 갖는 그루터기 형태의 결정트리를 가지며 단 하나의 질문(즉, 하나의 변수만 사용)으로 데이터를 분류해야 하기 때문에 약한 모형이다.

위에서 모형의 약점이란 예측을 제대로 하지 못한 데이터가 존재한다는 것을 의미한다.

그림 2은 AdaBoost 알고리즘을 설명한 것이다. 그림에서와 같이 AdaBoost 알고리즘은 다음과 같은 과정을 거쳐 수행된다.

(1) 첫 번째 약한 학습기를 첫 번째 분류기준 D1에서 + 와 - 로 분류하고 오분류된 데이터에 대해

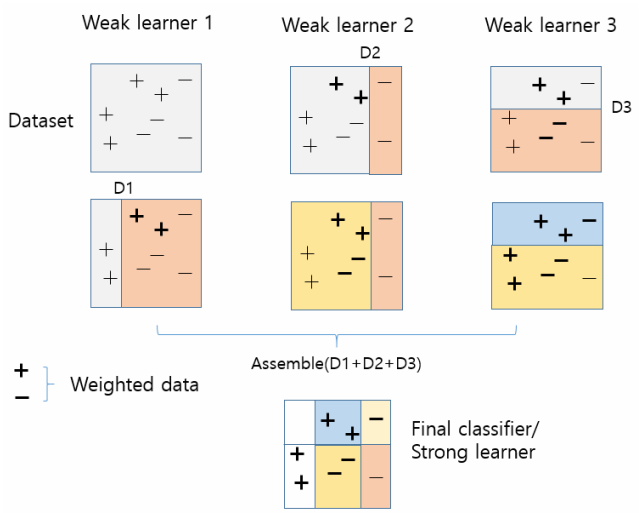


Fig. 2 Mechanism of AdaBoost algorithm

가중치를 부여(두 번째 그림에서 커진 + 표시)하고, (2) 두 번째 약한 학습기를 두 번째 분류기준 D2에서 +와 -로 다시 분류하고 오분류된 데이터에 대해 가중치를 부여(네번째 그림에서 커진 - 표시)하고, (3) 세 번째 약한 학습기를 세번째 분류기준 D3에서 +와 -로 다시 분류해서 오분류 데이터에 가중치를 부여(여섯번째 그림에서 위의 커진 -, 밑의 커진 + 표시)하고, (4) 마지막으로 분리된 이들 약한 학습기들을 결합하여 오분류가 적은 강한 학습기를 만들고 최종 예측 수행(제일 밑 그림)한다.

즉, AdaBoost는 매 스텝마다 이전의 학습 데이터에서 오차가 크거나 오분류된 데이터의 가중값을 크게 하고, 반대로 오차가 작거나 정분류된 데이터의 가중값을 낮게 한다. 이렇게 가중값에 비례하여 새로운 학습 데이터를 복원 추출하여 새로운 학습 데이터를 만들고 모형을 조정해간다.

전보[19]에서 기술한 배깅은 여러 개의 독립적인 약한 학습기에 대해서 각각 독립적으로 값을 예측한 뒤, 그 결과 값을 집계해 최종 결과 값을 예측하는 병렬학습 방식이다. 하지만 부스팅은 처음 약한 학습기에 대해서 예측을 하고 그 예측 결과에 따라 데이터에 가중값이 부여되고, 부여된 가중값이 다음 약한 학습기에 순차적으로 영향을 주는 형태의 팀 워크가 이루어지는 순차학습 방식이다.

[예제 1] 표 1은 [StatQuest: AdaBoost, Clearly Explained]에서 발췌한 Chest Blocked Arteries(막힌 동맥), Patient Weight(환자 체중)에 따른 Heart Disease(심장병) 여부에 대한 데이터이다. AdaBoost 알고리즘을 이용하여 각각의 속성이 결과값 (여기서는 Heart Disease)에 미치는 영향에 대해 검토한다. 또한 가슴 통증과 막힌 동맥이 있고 그리고 체중이 170kg인 환자는 심장병이 있는지 없는지를 판단하는 방법에 대해서 검토해보자. 초기 가중값은 모두 1/8(=0.125)로 동일하게 하여 샘플의 가중값의 합을 1로 한다. 먼저 Chest Pain과 Heart Disease와의 관계, Blocked Arteries와 Heart Disease와의 관계는 의사결정트리에서 설명한 것과 같이 각각 다음과 같이 나타내진다.

Table 1 AdaBoost example for heart disease

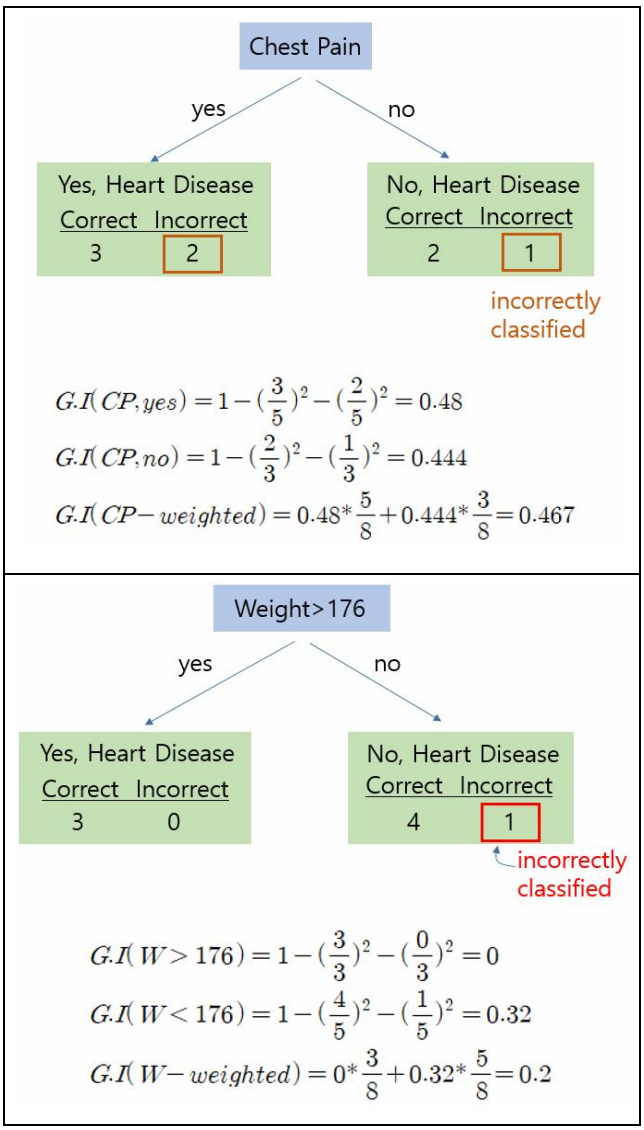
Row No.	Chest Pain	Blocked Arteries	Patient Weight (kg)	Heart Disease	Sample Weight
1	Yes	Yes	205	Yes	1/8=0.125
2	No	Yes	180	Yes	0.125
3	Yes	No	210	Yes	0.125
4	Yes	Yes	167	Yes	0.125
5	No	Yes	156	No	0.125
6	No	Yes	125	No	0.125
7	Yes	No	168	No	0.125
8	Yes	Yes	172	No	0.125

표 1로부터 Chest Pain을 루트노드로 한 경우에 Chest Pain이 있고(yes) Heart Disease가 있는 경우는 (correct) 환자 1,3,4이고 Chest Pain이 있지만(yes) Heart Disease가 없는 경우는 (incorrect) 환자 7,8이다. 또한 Chest Pain이 없고 (yes) Heart Disease가 없는 경우는(correct) 환자 5,6이고 Chest Pain이 없지만(yes) Heart Disease가 있는 경우는 (incorrect) 환자 2이다. 따라서 이 경우에 대한 지니계수는 의사결정트리에서 설명한 방법대로 계산하면 0.467이다. 마찬가지로 Blocked Arteries를 루트로드로 한 경우에 대한 지니계수는 0.5이다.

한편 Patient Weight와 Heart Disease의 관계에 대해서는 Patient Weight를 (125, 156, 167, 168, 172, 180, 205, 210) 순서대로 나열하고 중간값들 (140.5, 161.5, 167.5, 170, 176, 192.5, 207.5)을 기준으로 분리한 경우에 대한 의사결정트리의 지니계수값을 먼저 측정한다. 이 중에서 가장 작은 지니계수값 0.2를 갖는 중간 체중 176(=체중 172와 체중 180의 중간값)를 기준으로 환자를 분리하였다.

체중이 176 보다 큰 경우(Weight>176)로 분리한

Table 2 AdaBoost with decision stumps



경우가 지니계수가 0.2로 가장 작기 때문에 체중 176의 그루터기를 그루터기 숲(forest of stumps)의 제일 첫 그루터기(first stump)로 지정하여 가장 먼저 분리한다(표 2).

이렇게 완성된 그루터기의 분류에 따라 가중값인 Amount of Say α_t는 훈련 오차(training error)를 이용하여 다음 식에 따라 계산한다.

$$Amount\ of\ Say = \alpha_t = \frac{1}{2}\log_e\left(\frac{1 - Total\ error}{Total\ error}\right) \quad (1)$$

여기서 Total error는 오분류된 샘플의 전체 에러이다. Amount of Say는 최종 분류에 있어서 해당 그루터기가 얼마만큼의 영향을 주는가를 뜻하는 의미로 그루터기 성능(Performance of the stump) 또는 영향(Influence) 이라고도 한다.

위 표 2로부터 그루터기들에서 전체 에러(Total Error)는 Weight 속성에서 틀린 정보는 1개만 존재하므로 1/8이 된다.

Blocked Arteries 속성의 경우는 4/8, Chest Pain 속성의 경우는 3/8이다. 따라서 각각의 경우에 Amount of Say는 $0.97(=\frac{1}{2}\log_e\left(\frac{1-1/8}{1/8}\right)\doteq 0.97)$, 0, 0.25이다.

체중이 176보다 큰 경우(Weight > 176)로 분류된 표 2의 아래 그루터기에서 오류 데이터(incorrect))는 오직 하나로 체중이 167인 샘플의 경우이다. 따라서 이 오류 샘플의 가중값을 크게 하고, 반대로 정분류된 나머지 샘플들의 가중값을 낮게 하여 새로운 가중값을 정의하고 다음 그루터기(second stump) 모형을 조정해간다.

가중값은 에러로부터 결정되는 Amount of Say를 이용하여 다음 식으로 나타내진다.

$$\begin{aligned} w_{i,t+1} &= w_t \times e^{\alpha_t} \quad (incorrectly\ classified,\ increase) \\ w_{i,t+1} &= w_t \times e^{-\alpha_t} \quad (correctly\ classified,\ decrease) \end{aligned} \quad (2)$$

따라서 새 가중값은 다음 표 3과 같이 된다.

이 표 3으로부터 오분류된 네 번째 샘플은 가중값이 높아진 반면(0.125에서0.33), 나머지 샘플들은 가중값이 낮아진 것(0.125에서0.05)을 확인할 수 있다. 그런데 모든 샘플에 대해서 새 가중값의 합은 0.68로 1이 되지 않기 때문에 합이 1이 되도록 정규화한 가중값(normalized weight)을 계산한다(표 3).

위의 결과를 바탕으로 새로운 테이블을 만든다. 만드는 방법은 norm_weight의 누적값을 이용해 랜덤하게 샘플링한다. 즉, 위의 표 내부값이 비워진 동일한 크기의 표를 만들어 두고, 0~1 사이에서 랜덤하게 숫자를 택하여 0~0.07 사이의 값이 택해지면 첫 번째 데이터를, 0.07~0.14 (=0.07+0.07) 사이의 값이 택해지면 두 번째 데이터를, 0.14~0.21(=0.14+0.07) 사이의 값이 택해지면 세 번째 데이터를, 0.21~0.7 (=0.21+0.49) 사이의 값이 택해지면 네 번째 데이터를 빈 표에 순서대로 채워가는 방식을 택한다. 이 과정을 복원 추출(boostrap) 과정이라고 한다.

Table 3 Development of an AdaBoost classifier in 1st step

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight	Normalized Weight
Yes	Yes	205	Yes	0.125	0.05	0.07
No	Yes	180	Yes	0.125	0.05	0.07
Yes	No	210	Yes	0.125	0.05	0.07
Yes	Yes	167	Yes	0.125	0.33	0.49
No	Yes	156	No	0.125	0.05	0.07
No	Yes	125	No	0.125	0.05	0.07
Yes	No	168	No	0.125	0.05	0.07
Yes	Yes	172	No	0.125	0.05	0.07

$$w_{i,t+1}(167) = 0.125 \times e^{0.97} \approx 0.33$$

$$w_{i,t+1}(other) = w_t \times e^{-0.97} \approx 0.05$$

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Normalized Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Table 4 Develop an AdaBoost classifier in 2nd step

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	No	210	Yes	0.125
Yes	Yes	167	Yes	0.125
Yes	Yes	205	Yes	0.125
Yes	Yes	172	No	0.125
Yes	Yes	167	Yes	0.125
Yes	Yes	167	Yes	0.125
No	Yes	125	No	0.125
Yes	Yes	167	Yes	0.125

여기서는 0~1 사이에서 0.18, 0.56, 0.04, 0.94, 0.68, 0.32, 0.80, 0.47 순으로 선택되었다고 가정하면 대응하는 데이터를 순서대로 표에 채워는 방식으로 다음 표 4가 얻어진다.

새로운 표 5의 데이터를 살펴보면 중복되는 것도 있는데, 원래 테이블에서 잘못 분류되어 가중값이 높은 샘플이 4번이나 뽑힌 것을 알 수 있다. 여기에 모든 샘플의 가중값을 다시 1/8로 동일하게 한다.

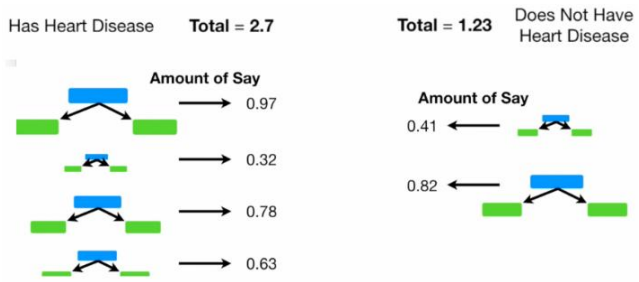


Fig. 3 How to predict in AdaBoost model

이 새로운 학습 데이터 표에 대해서 앞에서 수행하였던 과정을 낮은 훈련 오차가 얻어질 때까지 또는 정해진 횟수만큼 그루터기가 생성될 때까지 순차적으로 반복하여 의사결정트리를 구축한다. 이 후에 테스트 데이터를 보내면 모든 의사결정트리를 통과하고 마지막으로 어떤 클래스가 다수를 차지하는지 확인하고 이를 기반으로 결과값을 예측을 한다.

본 예제의 경우는 6개의 그루터기가 생성되었다고 가정하고 주어진 테스트 데이터(가슴 통증과 막힌 동맥이 있고 체중이 182kg인 환자가 심장병이 있는지 없는지)에 대해서 이렇게 의사결정트리를 통과하면 아래와 같이 각 그루터기 마다의 Amount of Say가 얻어진다.

그림 3에서 Heart Disease가 있다고 판단한 4개 그루터기에 대한 Total Amount of Say는 0.97+0.32+0.78+0.63=2.7이고, Heart Disease가 없다고 판단한 그루터기 다른 2개 그루터기에 대한 Total Amount of Say는 0.41+0.82=1.23이다.

따라서 Heart Disease가 있다고 판단한 환자의 Total Amount of Say가 더 크기 때문에 이 환자는 심장질환이 있는 환자로 분류된다.

이 예제에 대한 프로그램을 다음에 올려두었다.

https://github.com/yskim9574/DFclass-2023/blob/main/AdaBoost_heartdisease

2.2 Gradient Boosting Machine

Friedman[21] 등이 제안한 Gradient Boosting Algorithm (GBM)은 Random Forest와 마찬가지로 여러 개의 결정 트리를 결합시킨 앙상블 방법으로, AdaBoost와 유사하지만 경사하강법(Gradient Descent)를 이용하여 가중치 업데이트를 수행하여 최적화된 결과를 얻는 알고리즘이다[3, 20, 21].

이 GBM은 회귀분석 또는 분류분석에 모두 적용

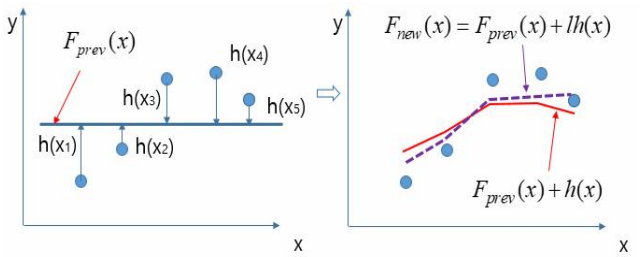


Fig. 4 Gradient boosting machine model

할 수 있다. GBM은 머신러닝 알고리즘 중에서도 표형식 데이터에 대해 가장 예측 성능이 높다고 알려져 있지만 탐욕 알고리즘(Greedy Algorithm)으로 과적합이 빨리 발생할 수 있으며 또한 해석시간이 오래 걸린다는 단점도 있다.

LightGBM, CatBoost, XGBoost 같은 파이썬 패키지는 Gradient Boosting Algorithm을 구현한 패키지들이다.

XGBoost 모델은 그래디언트 부스팅 모델 대비 예측 성능이 우수하며, 계산 효율성이 높고, 병렬 처리가 가능하여 계산 속도가 빠르며, 자체적으로 과적합 규제 기능을 제공하는 장점을 갖고 있다.

CatBoost 모델은 순서형 TBS(target-based statistics)와 순서형 부스팅(ordered boosting)을 통해 범주형 변수를 효과적으로 처리하며, 하이퍼 파라미터 최적화에 민감하지 않고, 시간에 따라 데이터의 분포가 변화하는 경우에도 예측 성능 저하가 낮은 점이 장점인 반면, LightGBM 대비 계산 속도가 느리고, 범주형 특성이 적은 경우 계산 속도나 계산 속도 측면에서 타 그래디언트 부스팅 모델 대비 이점이 없다는 점이 단점이다.

LightGBM 모델은 히스토그램 기반 그래디언트 부스팅 모델로, 장점은 타 그래디언트 부스팅 모델 대비 학습 시간이 가장 짧고, 메모리 사용량이 가장 적은 점이며, 단점은 데이터가 적은 경우 과적합 발생 확률이 상승한다는 점이다.

Gradient boosting은 Gradient descent + boosting 이라고 하기도 한다.

GBM은 Gradient(또는 잔차(residual))을 이용하여 이전 모형(예측함수) $F_{prev}(x_i)$가 실제값 y_i을 정확히 예측하지 못하는 약점을 보완하는 새로운 모형을 순차적으로 적용한 뒤 이들을 선형결합하여 얻어진 모형을 생성하는 지도학습 알고리즘이다. 즉 그림 4에서와 같이

$$\begin{aligned} y_1 &= F_{prev}(x_1) + h(x_1) \\ y_2 &= F_{prev}(x_2) + h(x_2) \\ &\cdots \\ y_n &= F_{prev}(x_n) + h(x_n) \end{aligned} \tag{3}$$

이렇게 h를 찾으면 새로운 예측 함수값은

$$F_{new}(x) = F_{prev}(x) + lh(x), \quad 0 < l < 1 \tag{4a}$$

이다. 즉, $h(x)$를 그대로 더하는 것이 아니라 과적합 방지를 위해 상수 l을 곱하여 더한다. 이 l을 학습률(learning rate, l.r.)이라고 하고, 보통 0.1로 한다.

여기서 순차적으로 적합할 모형 개수를 m이라고 하고, 첫 번째 모형의 예측값을 $F_0(x) = h_0(x)$라고 하면 최종 모형 $F_m(x)$은 다음과 같이 구해진다.

$$F_m(x) = h_0(x) + lh_1(x) + \cdots + lh_m(x) \tag{4b}$$

GBM 알고리즘에서 체중과 같이 목표값(target 또는 label)이 연속형인 회귀 문제인 경우 Gradient Boosting Regressor(GVR)를 사용하고, 목표값이 Yes, No와 같은 분류 문제인 경우 Gradient Boosting Classifier(GBC)를 사용한다.

GBM은 최소화하는 손실 함수를 기반으로 하기 때문에 회귀 문제의 경우 평균 제곱 오차(mean square error, MSE), $L = \frac{1}{2}(y - f(x))^2$, 와 같은 손실 함수를 사용하며, 분류의 경우 로그 우도(log-likehood), $L = -\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$, 와 같은 손실 함수를 사용한다. 여기서 p_i는 $P(y_i = 1|x_i)$ 이다[2-4].

[예제 2] GBM 알고리즘에 대한 설명으로 Height, Gender, Favorite color과 Weight의 관계를 나타낸 표를 다음에 나타내었다. 이 표 5에 대한 분석을 통해서 Height=1.7, Favorite Color=Green, Gender=Female 일 때 Weight가 얼마인지를 예측하는 알고리즘을 이하에 나타내었다.

이 표로부터 처음 예상한 평균 체중은 71.2kg이고 측정한 체중과 예상한 평균 체중과의 차이인 유사 잔차(pseudo residual)을 표 6에 나타내었다.

표 6을 이용하여 Gender=female을 루트노드로 하여 의사결정트리를 전개시켜서 표 7을 얻는다. 이 표 7

Table 5 Gradient boosting machine example for body weight

Height (m)	Favorite Color	Gender	Observed Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Table 6 Process of GBM model

Height (m)	Favorite Color	Gender	Observed Weight	Predicted Average Weight	Pseudo Residual
1.6	Blue	Male	88	71.2	16.8
1.6	Green	Female	76	71.2	4.8
1.5	Blue	Female	56	71.2	-15.2
1.8	Red	Male	73	71.2	1.8
1.5	Green	Male	77	71.2	5.8
1.4	Blue	Female	57	71.2	-14.2

Predicted Average Weight = Sum (Observed Weight)/N
Pseudo Residual(유사 잔차, 擬似残差)
= Observed Weight - Predicted Average Weight

에서 말단 노드의 (-14.2와 15.2), (1.8, 5.8)를 평균값으로 치환한다. 식 (3)과 식 (4)를 이용하여 1st residual를 구하면,

$$1^{st}\ residual = y_i - \left(F_{prev}(x_i) + lh(x_i)\right),\ l = 0.1 \qquad (5)$$

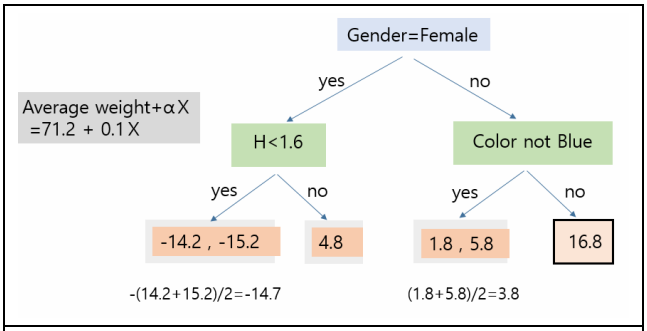
동일한 크기의 트리를 구성하여 이 과정을 반복하여 2nd residual, ··, nth residual을 구해간다. 최대 트리 개수까지 혹은 residual 크기가 더 이상 줄어들지 않을 때까지 이 과정을 반복한다(표 8).

GBM에서는 일반적으로 100개의 트리 (estimator라고 부름)를 사용하며 각 트리의 최대 깊이는 3개 레이어로 한다.

Random Forest와 달리 Gradient Boosting Algorithm는 이전 트리의 오차를 보완하는 방식으로 순차적으로 트리를 만들기 때문에 이전 단계에서 만들어진 트리 모양에 많은 영향을 받는다.

이 예제에서 Height=1.7m, Favorite Color=Green,

Table 7 Process of 1st residual prediction process in GBM model



Height (m)	Favorite Color	Gender	Observed Weight	Predicted Average Weight	Pseudo Residual	1st Residual
1.6	Blue	Male	88	71.2	16.8	15.1
1.6	Green	Female	76	71.2	4,8	4.3
1.5	Blue	Female	56	71.2	-15.2	-13.7
1.8	Red	Male	73	71.2	1.8	1.4
1.5	Green	Male	77	71.2	5.8	5.4
1.4	Blue	Female	57	71.2	-14.2	-12.7

1^{st} residual:	1^{st} predicted:
$88-(71.2+0.1\times16.8)=15.1$	$71.2+0.1\times16.8=72.9$
$76-(71.2+0.1\times4.8)=4.3$	$71.2+0.1\times4.8=71.7$
$56-(71.2+0.1\times(-14.7))=-13.7$	$71.2+0.1\times(-14.7)=69.7$
$73-(71.2+0.1\times3.8)\doteq1.4$	$71.2+0.1\times3.8\doteq71.6$
$77-(71.2+0.1\times3.8)\doteq5.4$	$71.2+0.1\times3.8\doteq71.6$
$57-(71.2+0.1\times(-14.7))\doteq-12.7$	$71.2+0.1\times(-14.7)\doteq69.7$

Gender=Female 인 경우에 GBM회귀분석하면 Weight는 76kg이다. 이 GBM 예제에 대한 프로그램을 다음에 올려두었다.

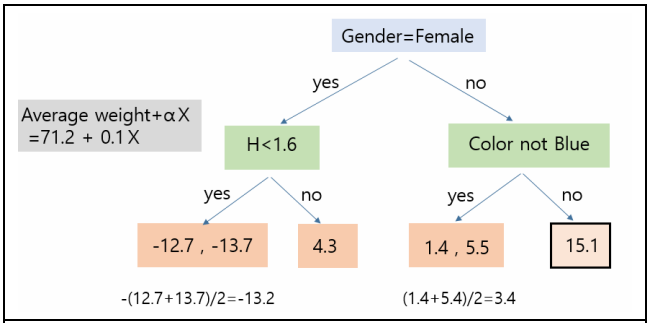
https://github.com/yskim9574/DFclass-2023/blob/main/GBM_weight_1

2.3 Support Vector Machine

Vapnik-Chervonenkis (VC) 이론으로 알려진 서포트 벡터 머신(support vector machine, SVM)은 자료 분류 및 회귀 작업에 사용되는 지도형 기계 학습 알고리즘의 한 유형으로, 패턴 인식, 이미지 분석, 자연어 처리 등 다양한 분야에서 널리 사용되고 있다[4, 22].

SVM은 구체적으로 전통적인 이진 분류(binary classification)를 위한 기법 중 하나로 N차원을 공간에서 데이터들이 주어졌을 때 (N-1)차원으로 데이터들을 두 개의 클래스(class, category 또는 group)로 분류할 수 있는 초평면(hyperplane)을 찾는 분류 기법이다.

Table 8 Process of 2nd residual prediction process in GBM model



Height (m)	Favorite Color	Gender	Observed Weight	Pseudo Residual	1st Residual	2nd Residual	.	Nth Residual
1.6	Blue	Male	88	16.8	15.1	13.6	.	.
1.6	Green	Female	76	4,8	4.3	3.9	.	.
1.5	Blue	Female	56	-15.2	-13.7	-12.4	.	.
1.8	Red	Male	73	1.8	1.4	1.1	.	.
1.5	Green	Male	77	5.8	5.4	5.1	.	.
1.4	Blue	Female	57	-14.2	-12.7	-11.4	.	.

2nd residual:

$88-(71.2+0.1\times16.8+0.1\times15.1)\doteq13.6$

$76-(71.2+0.1\times4.8+0.1\times4.3)\doteq3.9$

$56-(71.2+0.1\times(-14.7)+0.1\times(-13.2))\doteq-12.4$

$73-(71.2+0.1\times3.8+0.1\times3.4)\doteq1.1$

$77-(71.2+0.1\times3.8+0.1\times3.4)\doteq5.1$

$57-(71.2+0.1\times(-14.7)+0.1\times(-13.2))\doteq-11.4$

이렇게 초평면이 찾아지면 새로운 데이터데 주어졌을 때 새로운 데이터가 어느 클래스에 속하는지를 구분한다(그림 5(a)).

이진 분류란 예를 들어, 소비자들의 다양한 정보를 입력 받아 이 소비자가 특정 상품을 구매 할지 안 할지 예측하는 문제나, 신호에 대한 정보를 입력받아 이 정보가 진짜 신호인지 가짜 신호인지 분별하는 문제 등이 이진분류 문제이다. 이진분류 문제에서 출력변수는 0 또는 1로 표기하거나 -1 또는 +1로 표기한다. SVM을 설명할 때엔 이진분류 문제의 출력변수를 주로 -1과 +1로 표현한다.

데이터들이 그림 1과 같이 선형적으로 분리될 필요는 없고 비선형적으로 분리해야 하는 경우도 존재한다(그림 5(b)).

이 SVM에서 사용되는 주요 용어를 정의해둔다.

(1) 초평면(hyperplane)

초평면은 고차원 공간에서 데이터들을 서로 다른 클래스로 완전히 분리하는 최적의 결정 경계(optimal decision boundary)이다. 그림 1에서와 같이 2차원 공간에서 초평면은 데이터들을 두 클래스로 구분하는 선이며, 3차원 공간에서 초평면은 데이터들를 두 클래스로 분리하는 평면이다.

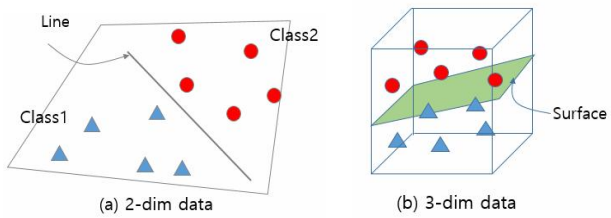


Fig. 5(a) Linearly separable problem in SVM model

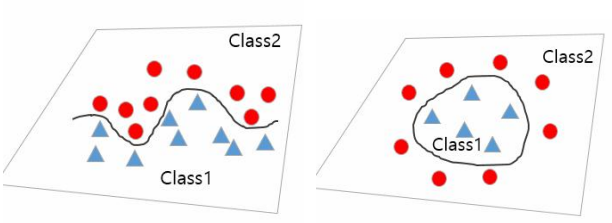


Fig. 5(b) Nonlinearly separable problem in SVM model

이 초평면은 새로운 데이터가 초평면의 어느 쪽에 속하는지를 평가(분류)하고 예측하는데 사용한다.

(2) 마진(margin)

마진(또는 여백)은 경계(초평면)에서부터 각 클래스에서 가장 가까운 데이터 포인트 면까지의 수직거리(폭)이다. SVM의 목표는 분류 오류를 최소화하면서 이 마진을 최대화하는 것이다. 마진이 클수록 분류에 대한 신뢰도가 더 높다는 것을 의미하며, 마진은 클래스가 기능 공간에서 얼마나 잘 분리되어 있는지를 측정한 것이다. SVM은 가장 큰 마진을 가진 초평면을 찾는 알고리즘으로 때때로 최대 마진 분류기(maximum-margin classifier)라고도 한다.

SVM에서 모든 샘플이 마진의 바깥 쪽에 위치하도록 엄격하게 좁은 마진을 잡는 것을 하드 마진(hard margin) 이라고 하고 마진 내에 일부 다른 클래스의 샘플들이 혼재되어 있는 것(overlapping classes)을 허용하도록 넓게 마진을 잡는 것을 소프트 마진(soft margin)이라고 한다.

소프트 마진은 각 훈련 데이터 샘플 (x_i, y_i) 마다 잉여 변수(slack variable) ξ_i를 대응시켜서 샘플이 마진의 폭 안으로 ξ_i 만큼 파고드는 상황을 용인하도

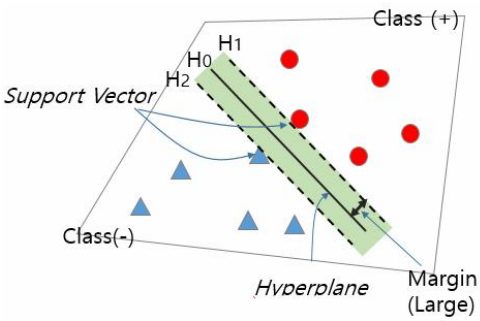


Fig. 6 Definition of hyperplane, margin and support vector in SVM model

록 한 것이다.

즉 소프트 마진은 초평면 근방에서 노이즈 (noise)나 이상치(outlier) 데이터의 발생을 조금 허용하는 여유를 두면서 훨씬 더 많은 정상치 데이터를 최대한 잘 분류할 수 있도록 넓은 마진을 가지는 선형 경계를 만드는 것을 말한다.

이하에서는 하드 마진만을 고려하는 것을 한다.

(3) 서포트 벡터(support vector)

서포트 벡터는 초평면을 경계로 구분된 클래스의 데이터들 중에서 초평면에 가장 가까운 각 클래스의 최외각에 있는 데이터를 말한다.

이 서포트 벡터는 초평면의 위치와 방향을 결정하고 SVM의 분류 정확도에 큰 영향을 미치기 때문에 중요하다. 실제 SVM이란 이름은 초평면을 지원하거나 정의하기 때문에 서포트 벡터의 이름을 따서 명명되었다.

따라서 초평면은 두 클래스의 각각 최외각에 있는 서포트 벡터를 관통하는 두 개의 평행한 경계면(그림 6에서 H_1와 H_2)의 중앙에 위치한 면(H_0)이다. 2차원 공간에서는 선으로 나타난다.

이 서포트 벡터는 초평면과 각 클래스에서 가장 가까운 데이터들 사이의 거리인 마진을 계산하는데 사용되기 때문에 중요하다. 그림 6에 초평면, 마진, 서포트 벡터의 정의를 나타내었다.

그림 6과 같은 데이터를 분류하는데 초평면이 유일한 것은 아니고 그림 7에서와 같이 있어서 초평면의 방향과 마진의 크기가 다른 다양한 초평면이 존재할 수 있다. 그렇지만 그림 6에서 마진이 그림 7의 경우보다 그림 6의 경우가 두 클래스를 더 잘 분류한다는 것을 알 수 있다. 즉 SVM알고리즘은 가장 큰 마진을 가진 초평면을 찾는 것이다.

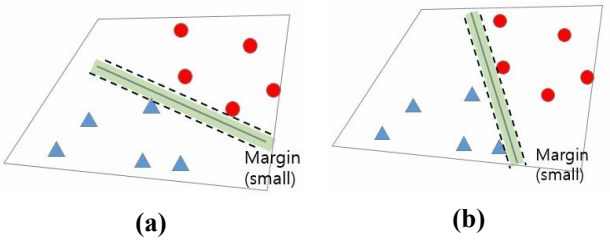


Fig. 7 Small margin example in SVM model

Table 9 Support vector machine example of hard margin problem

No. Data	x(x_1)	y(x_2)	Class(Y_i)
1	1.0	1.0	-1
2	1.7	2.3	-1
3	3.7	3.8	-1
4	4.0	1.7	-1
5	5.3	2.0	-1
6	3.5	6.3	+1
7	4.3	4.9	+1
8	6.1	6.9	+1
9	6.2	4.9	+1
10	7.4	6.8	+1

그림 6의 두 클래스의 구체적인 데이터값, 표 9를 이용하여 초평면과 마진을 구해보자. 그림 6의 두 클래스의 데이터(위쪽 클래스의 원형 데이터와 아래쪽 클래스의 삼각형 데이터)를 분리하는 초평면의 직선 방정식은 $0.49x + y - 6.3 = 0$ 또는 $0.49x_1 + x_2 - 6.3 = 0$ 로 나타내지고 일반식으로는 다음 식으로 표시된다.

$$Ax_1 + Bx_2 + C = 0 \quad (A = 0.49, B = 1, C = -6.3) \qquad (6)$$

이 초평면과 위쪽 클래스의 하나의 원형데이터 (x_1^0, x_2^0) 와의 수직 거리 d는

$$d = \frac{\left| Ax_1^0 + Bx_2^0 + C \right|}{\sqrt{A^2 + B^2}} \qquad (7)$$

또한 식 (6)의 좌변에 $1/\sqrt{A^2 + B^2}$ 을 곱하여 행렬형식으로 나타내면 초평면에 대한 식은 다음과 같이 표현할 수 있다.

$$\left(\frac{A}{\sqrt{A^2+B^2}} \quad \frac{B}{\sqrt{A^2+B^2}}\right)\begin{Bmatrix}x_1\\x_2\end{Bmatrix}+b=0$$
$$\rightarrow (w_1 \; w_2)\begin{Bmatrix}x_1\\x_2\end{Bmatrix}+b=0 \qquad (8)$$
$$\rightarrow w^T x+b=0, \quad \left(b=\frac{C}{\sqrt{A^2+B^2}}\right)$$

여기서 w를 초평면에 수직한 가중값 벡터, b를 편향 상수라고 부른다. 가중값 벡터 w는 서포트 벡터들 사이의 마진을 가장 크게 하는 초평면의 방향을 나타내고, b는 서포트 벡터들 사이의 마진을 가장 크게 하는 결정 초평면의 위치를 나타낸다.

한편 다음 그림 8(a)와 같이 식 (8)에서 우변을 0이 아니고 +1로 하면($w^T x+b=+1$, $Ax_1+Bx_2+C=+1$) 초평면을 기준으로 위쪽 클래스 쪽으로 초평면에 평행한 점선의 경계 H_1으로 나타낸 양의 초평면(positive hyperplane)이, 또한 -1로 하면($w^T x+b=-1$, $Ax_1+Bx_2+C=-1$) 아래 클래스 쪽으로 초평면에 평행한 점선의 경계 H_2으로 나타낸 음의 초평면(negative hyperplane)이 각각 나타나는 것을 알 수 있다.

이 두 점선 간의 수직거리 d는 다음과 같다.

$$d=\frac{2}{\sqrt{A^2+B^2}}=\frac{2}{\|w\|^2} \qquad (9)$$

SVM은 가장 큰 마진 $M\ (=d/2=1/\|w\|^2)$을 가진 초평면을 찾는 것이므로 식 (9)에서 $d(=2/\|w\|^2)$를 최대로 하는 것은 $\|w\|^2/2$를 최소로 하는 것과 같다. 이 경우에 식 (8)은

$$0.49x_1+x_2-6.3=0 \quad (w^T x+b=0)$$

로 표현되므로 이 식에 위쪽 클래스의 원형 데이터 중의 하나인 (6.2, 4.9)를 대입해 보면

$$0.49(6.2)+4.9-6.3=1.638>1 \quad (w^T x+b>1)$$

또한 아래쪽 클래스의 삼각형 데이터 중의 하나인 (4.0,1.7)를 대입해 보면

$$0.49(4)+1.7-6.3=-2.64<-1 \quad (w^T x+b<-1)$$

이다. 만일 아래쪽 클래스에서 H_0와 H_2 사이에 (3.7, 4.3) 인 삼각형 데이터를 생각해보면

$$0.49(3.7)+4.3-6.3=-0.187 \quad (-1<w^T x+b<0)$$

인 것을 알 수 있다.

각 데이터들의 클래스는 다음과 같이 부호함수

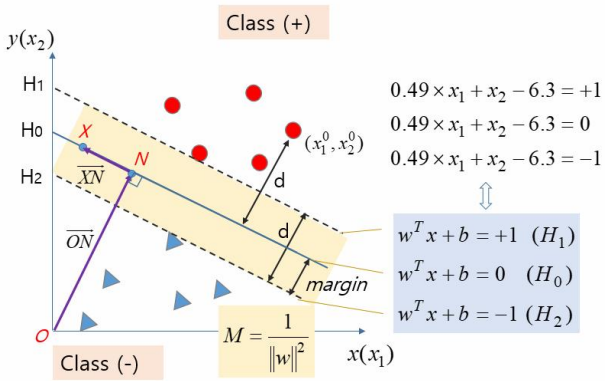


Fig. 8 (a) Definition of classification of data by linear support vector boundary

(sign function)로 구별할 수 있다.

$$Y_i=\begin{cases}+1 \text{ if } w^T x+b\geq 0 \ (Class\ +1)\\ -1 \text{ if } w^T x+b<0 \ (Class\ -1)\end{cases} \qquad (10)$$

따라서 표 3에서 데이터 (4.0, 1.7)를 포함한 위쪽 클래스는 $Y_i=-1$의 레이블을 갖고(초평면을 기준으로 아래 쪽에 존재, 음의 방향), 데이터 (6.2, 4.9)를 포함한 아래쪽 클래스는 $Y_i=+1$의 레이블을 갖는다(초평면을 기준으로 위쪽에 존재, 양의 방향).

이 초평면의 직선방정식 (8)은 다음과 같이 벡터의 스칼러 곱으로 유도할 수 있다. 즉, 초평면의 직선방정식, 식 (6)에 대한 단위 법선벡터가

$$n=\frac{A}{\sqrt{A^2+B^2}}i+\frac{B}{\sqrt{A^2+B^2}}j \qquad (10)$$

이고 이 단위 법선벡터와 초평면이 만나는 점N의 위치를 (x_1^n, x_2^n)라고 하고 초평면 상의 임의 점X의 위치를 (x_1, x_2)라고 하자. 벡터 ON과 벡터 XN은 직교하므로 이 두 벡터의 스칼러 곱은 영이어야 하는 조건으로부터 식 (6)이 얻어진다.

$$0=ON\cdot XN$$
$$=\left(\frac{A}{\sqrt{A^2+B^2}}i+\frac{B}{\sqrt{A^2+B^2}}j\right)\bullet\left((x_1-x_1^n)i+(x_2-x_2^n)j\right)$$
$$=\frac{A}{\sqrt{A^2+B^2}}x_1+\frac{B}{\sqrt{A^2+B^2}}x_2-\left(\frac{A}{\sqrt{A^2+B^2}}x_1^n+\frac{B}{\sqrt{A^2+B^2}}x_2^n\right)$$
$$=\frac{A}{\sqrt{A^2+B^2}}x_1+\frac{B}{\sqrt{A^2+B^2}}x_2+b \quad (\therefore w^T x+b=0)$$
(11)

이 초평면을 정의하기 위해서는 초평면에 수직한 가중값 벡터 w 와 편향 상수 b 의 두 개의 매개변수가 필요하다.

SVM에서 클래스 +1과 클래스 -1의 데이터들 사이에서 식 (9)의 마진 M 을 최대로 하는 초평면을 찾는 것은 $\|w\|^2/2$ 를 최소로 하는 것과 같다. 또한 클래스 $Y_i=+1$ 인 데이터는 $w^Tx+b\geq 1$ 이며, 클래스 $Y_i=-1$ 인 데이터는 $w^Tx+b<-1$ 이므로 두 식을 하나로 다음과 같이 나타낼 수 있다.

$$\begin{aligned} &Y_i(w^Tx+b)\geq 1 \\ &Y_i(w^Tx+b)-1\geq 0 \end{aligned} \tag{12a}$$

이 식을 결정조건(decision rule)이라고 부른다. 만약 경계선에 데이터가 걸쳐있다면

$$Y_i(w^Tx+b)-1=0 \tag{12b}$$

이 식을 제약조건(constraints)이라고 한다.
따라서 하드마진의 SVM 문제는 제약조건 (12)를 만족하며 다음과 같이 손실함수(loss function)를 최소화하는 문제로 정의 할 수 있다.

$$\begin{aligned} &\text{Minimize loss function}: \ L=\frac{1}{2}\|w\|^2 \\ &\text{Subject to}: \ Y_i(w^Tx+b)-1=0 \ \ , i=1,2,\cdots,N \end{aligned} \tag{13}$$

따라서 SVM 문제는 식 (12)의 제약조건에 라그랑지 승수법(Lagrange multiplier method)을 적용하여 다음의 볼록손실함수(convex loss function) L_p 를 최소화하는 문제로 간주할 수 있다.

$$\begin{aligned} &\min \ L_p(w,b,\alpha_i)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i[Y_i(w^Tx_i+b)-1] \\ &\alpha_i\geq 0, \quad i=1,2,\cdots,N \end{aligned} \tag{14}$$

여기서 α_i 는 라그랑지 승수(Lagrange multiplier)이다. L_p 는 라그랑지 프라이멀 함수(Lagrange primal function)라고도 불린다.

위 식에서 x_i, Y_i 를 대입하여 L_p 값을 최소화하기 위해 Karush-Kuhn-Tucker 조건(KKT condition) 을 적용하고

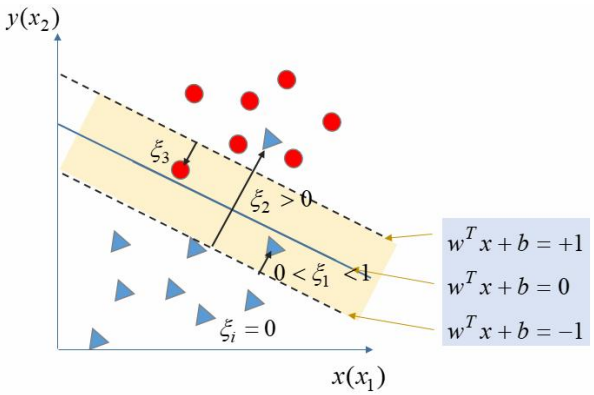


Fig. 8 (b) Definition of classification of data by soft margin SVM problem

$$\begin{cases} \dfrac{\partial L_p}{\partial b}=0 \\ \dfrac{\partial L_p}{\partial w}=0 \end{cases} \Rightarrow \begin{cases} \displaystyle\sum_{i=1}^{N}\alpha_i Y_i=0 \\ \displaystyle w=\sum_{i=1}^{N}\alpha_i Y_i x_i \end{cases} \tag{15}$$

식 (15)에서 가중값 벡터 $w(=\sum_{i=1}^{N}\alpha_i Y_i x_i)$ 를 식 (14)에 적용하면 식 (14)의 L_p 를 최소화하는 문제는 다음과 같은 라그랑지 듀얼 함수(Lagrange dual function) L_D 를 최대화하는 문제가 된다.

$$\begin{aligned} &\max \ L_D(\alpha_i)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j Y_i Y_j x_i^T x_j \\ &\text{subject to } \sum_{i=1}^{N}\alpha_i Y_i=0 \ \text{ and } \ \alpha_i\geq 0 \end{aligned} \tag{16}$$

가중값 벡터 $w(=\sum_{i=1}^{N}\alpha_i Y_i x_i)$ 를 구하고 $w^Tx+b=\pm 1$ 를 이용하여 편향 상수 b 값을 구한다.
한편 그림 8(b)에 나타낸 소프트 마진의 SVM 문제는 다음과 같이 정의 할 수 있고[5, 7]

$$\begin{aligned} &\text{Minimize loss function}: \ L=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i \\ &\text{Subject to}: \ \xi_i\geq 0, \ \ Y_i(w^Tx+b)-(1-\xi_i)=0 \ , \ \forall i \end{aligned} \tag{17}$$

라그랑지 함수는

$$\begin{aligned} L_p=&\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i[Y_i(w^Tx_i+b)-(1-\xi_i)] \\ &-\sum_{i=1}^{N}\mu_i\xi_i \end{aligned} \tag{18}$$

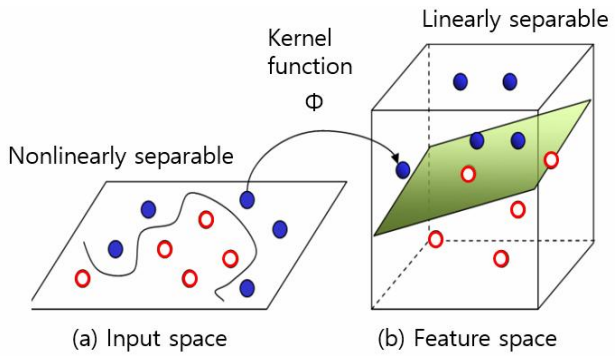


Fig. 9 Mapping of the data from the input space to a high-dimensional feature space

이 식에 Karush-Kuhn-Tucker 조건(KKT condition)을 적용하면 다음과 같이 정리된다.

$$\frac{\partial L_p}{\partial b}=0 \Rightarrow \sum_{i=1}^{N}\alpha_i Y_i=0$$
$$\frac{\partial L_p}{\partial w}=0 \Rightarrow w=\sum_{i=1}^{N}\alpha_i Y_i x_i \quad (19)$$
$$\frac{\partial L_p}{\partial \xi_i}=0 \Rightarrow \alpha_i=C-\mu_i$$

여기서 정칙화 비용(regularization cost) $C\geq 0$ 는 얼마나 강한 페널티를 줄지를 결정하는 하이퍼 파라미터이다.

식 (12)을 도출하는 과정에서 우변에 1.0을 사용하였는데 이것은 특별한 의미는 없고 수학적 편의를 위해 사용한 것이다. 식 (13)의 손실함수에서 이 값 1.0은 상수이기 때문에 편미분 과정에서 영이 된다.

위에서는 선형으로 분리 가능한 문제에 대한 SVM의 핵심알고리즘에 대해서 논하였다. 이 선형 SVM은 데이터 분할 경계가 복잡한 경우에는 분류 성능이 떨어지는 문제점이 있다. 이 때는 커널 매핑 함수(Kernel mapping function)를 통해 비선형 비모수 분할선을 정의하여 문제를 해결한다[22, 23].

여기서 커널은 저차원 입력 공간(input space)을 가져와 이를 고차원 공간(feature space)으로 변환한다. 즉, 분리 불가능한 문제에 차원을 더 추가하여 분리 가능한 문제로 변환하는 역할을 한다.

많이 사용하는 커널 함수로는 Linear Kernel, Polynomial Kernel, Gaussian radial basis function(RBF) Kernel, Bessel function Kernel, Sigmoid Kernel 등이 있다.

SVM을 이용한 머신러닝에서 이산 범주용 데이터

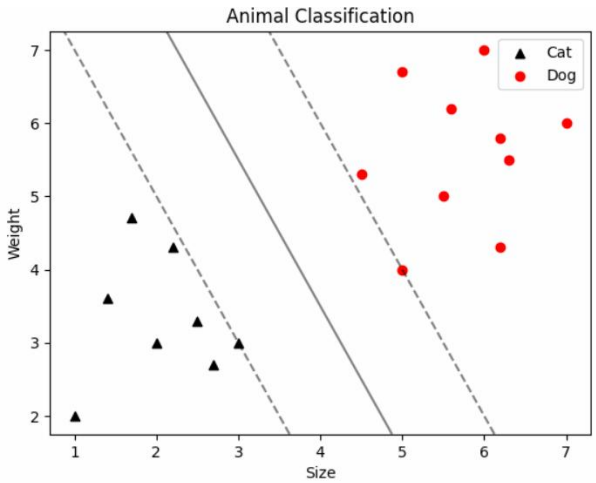


Fig. 10 Support vector machine to classify the dog and cat

의 분류에서는 SVC(support vector classifier)를, 연속한 데이터의 회귀에서는 SVR (support vector regressor)를 사용한다. SVR은 SVM의 많은 장점을 공유하는 방식으로 데이터에 연속값 함수를 적용한다. sklearn.svm의 SVR알고리즘에서는 RBF kernel을 기본으로 사용한다.

위에서 설명한 SVM은 전통적으로 이진 분류에 널리 사용되지만 다중 클래스 분류에도 활용된다. 이진 분류에 사용되는 SVM을 이진SCM, 다중 분류에 사용되는 SVM을 다중클래스 SVM(multiclass support vector machine, mSVM)이라고 부른다[25].

SVM은 최근에 다양한 재료가공 문제 해결에 적용되고 있다[25-30].

SVM 모델을 이용하여 동물의 크기와 무게로 개와 고양이를 분류한 예제에 대한 프로그램을 https://github.com/yskim9574/DFclass-2023/blob/main/SVM_catdog_classifier 에 올려두었다.

이렇게 support vector과 margin을 도출하여 두 동물을 분류하고 동물의 크기와 무게가 주어졌을 때 어떤 클래스에 속하는 지를 판단할 수 있다. 예를 들면, 크기가 4, 무게가 5인 동물은 개로 판단된다. (그림 10)

이 SVM은 다른 차량, 보행자, 교통 표지판 등과 같은 물체(객체) 감지 및 분류와 같은 작업을 위한 이미지 처리 및 컴퓨터 비전에 자주 사용되어 왔다. 이 알고리즘은 자율주행기술(autonomous driving technology)에서 전방에 사람이 있는지 없는지를 판별하는데도 사용될 수 있다.

[**예제** 3] 점진판재성형(incremental sheet forming, ISF)은 금형이 없이 판재를 복잡한 형상으로 성형할 수 있는 방법으로 최근 활발히 연구되고 있는 기술분야이다. ISF는 기존의 프레스 성형과 비교하여 프로토타입이나 소규모 생산에 적합한 기술로 알려져 있다.

그러나 프레스 성형된 제품과 비교하면 단점점진성형(Single Point Incremental Forming, SPIF)으로 성형된 부품은 베개 효과(pillow effect)와 벽 변위라고 부르는 형상 불량이 자주 발생하고 있다.

따라서 충분한 품질과 결함이 없는 SPIF 부품을 생산하려면 최적의 공정 매개변수를 선택해야 한다.

두께 0.22mm의 AlMn1Mg1 판재에 대한 점진성형공정(incremental sheet forming, ISF)을 수행하여 얻은 성형된 부품의 표면 거칠기 Ra에 대한 데이터셋에 ANN과 다양한 커널을 적용한 SVR을 적용하여 ISF 공정 매개변수의 영향을 평가하고 표면 거칠기를 예측한 결과를 그림 11에 나타내었다[27a].

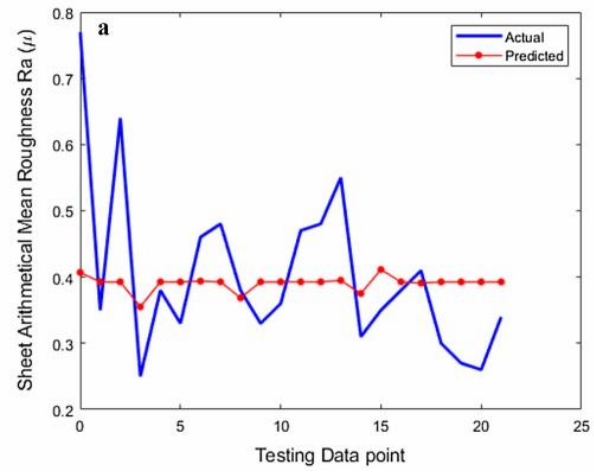
이 결과 ANN의 경우는 성형제품에 대한 표면 거칠기 예측이 우수하나 SVR의 경우는 거칠기에 대한 예측이 좋지 않다는 것을 알 수 있다.

또한 다양한 머신러닝 알고리즘을 적용하여 공구 형상, 공구 재료, 공구의 반경, 공구의 거칠기 등이 베개 효과와 벽 변위에 미치는 영향을 검토한 결과[27b] Levenberg-Marquardt (LM) optimizer를 적용한 MLP(multilayer perceptron) 가 두 경우에 모두 가장 우수한 성능을 보였고, 다음에 CatBoost와 GBR가 우수한 성능을 보였다(그림 11(c)). MLP는 ANN모델에서 다수의 은닉층을 갖고 있는 모델을 말한다.

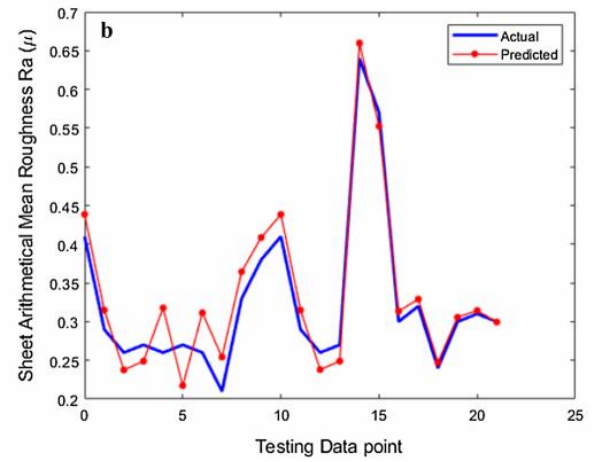
또한 공구 재료와 공구 형상이 베개 효과에 가장 큰 영향 미치는 상대적 중요 요소(relative importance, RI)이며, 공구의 거칠기와 공구 형상이 벽 직경에 가장 영향을 주는 것을 나타났다.

이 예제에서 시사하는 것과 같이 머신러닝의 다양한 알고리즘(모델)은 소성가공의 결함이나 특성 평가에서 많은 차이를 보이는 것을 알 수 있다.

[예제 4] 문헌 [28]에서는 소성가공 문제에 적용할 최적의 머신러닝 모델을 선택하기 위한 실증적 연구를 수행하였다. Mild steel, DP600, HSLA340 3가지 강종을 대상으로 다양한 조건 하에서 U-채널 성형공정에 대한 실험과 유한요소해석을 수행하여 MLP,



(a) SVR



(b) ANN

Fig. 11 Actual and predicted testing data for surface roughness in ISF experiments: (a) SVR; (b) ANN [27a]

RF, CART, NB, SVM 등 다양한 비모수적 모델에서 스프링백과 최대 두께감소율 발생에 대한 평가 성능을 측정하여 비교한 결과를 표 11에 나타내었다.

각각의 머신러닝 모델에서 분류를 목적으로 하는 경우에는 정확도(Accuracy), 정밀도(Precision), 재현률(Recall), 정밀도와 재현율의 조화 평균값(F1-score), ROC곡선의 밑면적(AUC) 등의 측정항목으로 주로 사용된다.

제한된 실험결과 이지만 DP600 강의 스프링백 평가에는 SVM모델이 가장 우수하지만 최대 두께 감소율 평가에는 RF 모델이 더 적합하다는 것을 알 수 있다.

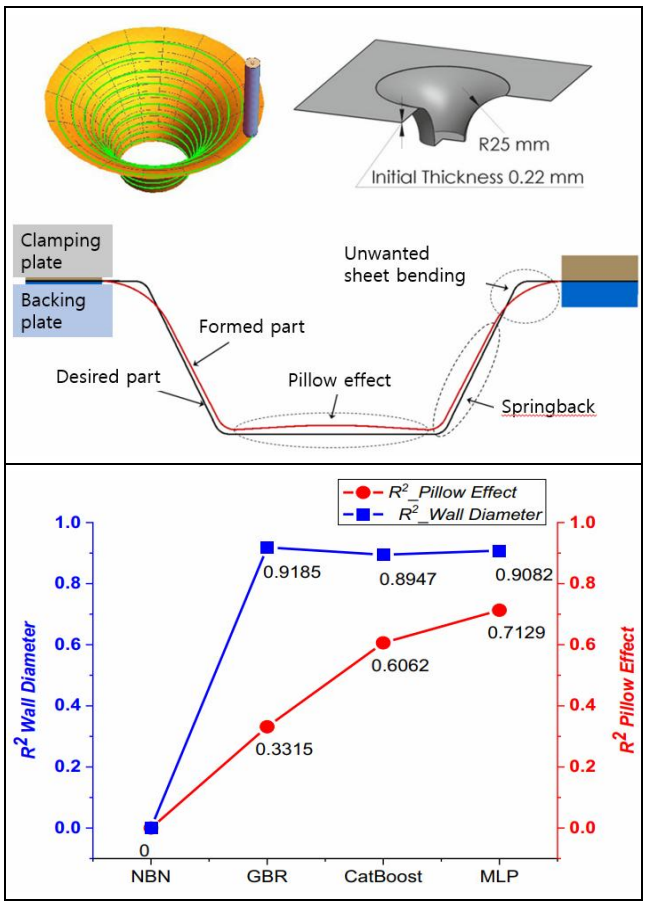


Fig. 11 (c) R^2 values for predicted results (pillow effect and wall diameter)

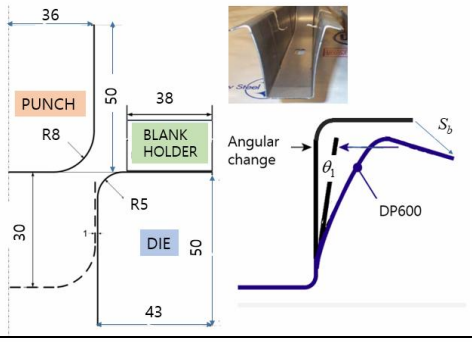
$$R^2 = \frac{SS_{tot} - SS_{res}}{SS_{tot}}$$

(주)

$$SS_{tot} = \sum_{i=1}^{N} (y_i^{\text{target}} - \bar{y})^2, \quad SS_{res} = \sum_{i=1}^{N} (y_i^{\text{target}} - y_i^{\text{predit}})^2$$

[예제 5] 판재 블랭킹 가공공정에서 공구의 마모 상태를 분류하기 위해 다중클래스 SVM을 적용한 결과를 그림 12에 나타내었다. 공구 마모상태가 잘 분류되고 있음을 알 수 있고, 공구 마모 분류의 정확도는 센서에 의한 하중의 시계열 데이터 수집 방법(data acquisition), PCA 등의 데이터 전처리(preprocessing) 및 특징 추출(feature extraction)의 정확성에 민감하게 영향을 받는 것으로 평가되었다[25]. 또한 다양한 머신러닝 알고리즘 - LDA, RF, mSVM, NB, kNN -들의 정확성을 평가한 결과, 선형 커널 함수를 이용한 mSVM이 가장 정확성이 높다는 것을 보였다.

Table 11 Comparisons of machine learning model for U-channel forming problem [28]



Material	Springback *sb* [mm]		Maximum thinning *th* [%]	
	BHF=4.9kN	BHF=19.6kN	BHF=4.9kN	BHF=19.6kN
Mild Steel	6.165	2.601	2.82	9.62
DP600	11.151	8.522	2.07	5.83
HSLA340	8.747	2.70	5.115	7.66

Algorithms	Accuracy		Precision		Recall		F1-Score		AUC	
	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std
DP600 - Springback Results										
MLP	91%	0.010	92%	0.010	91%	0.010	91%	0.010	91.90%	0.010
CART	87%	0.017	87%	0.018	87%	0.018	87%	0.018	86.54%	0.015
NB	84%	-	85%	-	84%	-	84%	-	85,11%	-
RF	86%	0.024	86%	0.028	85%	0.024	85%	0.024	84.76%	0.022
SVM	92%	-	92%	-	92%	-	92%	-	92.01%	-
DP600 - Maximum Thinning Results										
MLP	94%	0.010	95%	0.006	93%	0.010	93%	0.010	95.20%	0.008
CART	91%	0.009	92%	0.010	91%	0.010	91%	0.010	92.11%	0.010
NB	94%	-	94%	-	94%	-	94%	-	94.12%	-
RF	97%	0.015	97%	0.015	97%	0.015	97%	0.015	96.42%	0.015
SVM	92%	-	92%	-	92%	-	92%	-	90%	-

㈜ 실제 클래스와 예측한 클래스의 매칭을 이용하여 분류 모델의 성능을 평가하는 도구인 혼동행렬(confusion matrix) 을 아래에 나타내었다.

		Predicted Level	
		True	False
Actual Level	True	True Positive (TP)	False Negative (FN) – Type II error
	False	False Positive (FP) – Type I error	True Negative (TN)

모델의 결과를 이용하여 혼동행렬을 작성하고 그 결과값들을 이용하여 모델을 평가하기 위해 정확도, 정밀도, F1-점수 등의 지표를 계산한다. F1-점수(F1-score)는 정밀도 (Precision)와 재현률 (Recall)의 조화평균이며, 다음과 같이 정의된다.

$$F1 = 2 \times \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

$$precision = \frac{TP}{TP+FP}, \quad recall = \frac{TP}{TP+FN}$$

정밀도는 모델이 True 로 예측한 데이터 중 실제로 True 인 데이터의 비율이고, 재현율은 실제로 True 인 데이터 중에서 모델이 True 라고 예측한 데이터의 비율이다.

이 재현률은 통계학에서는 민감도(sensitivity)로, 그리고 다른 분야에서는 hit rate 라는 용어로도 불린다.

일반적으로 $F1$ 값이 높으면 모델 성능이 좋다고 볼 수 있다. 또한 정확도(Accuracy)는 전체 데이터 중 올바르게 예측한 데이터 비율로

$$accuracy = \frac{TP+TN}{TP+FN+TN+FP}$$

또한 효율(Efficiency)는

$$efficiency = \frac{TP}{TN+FP}$$

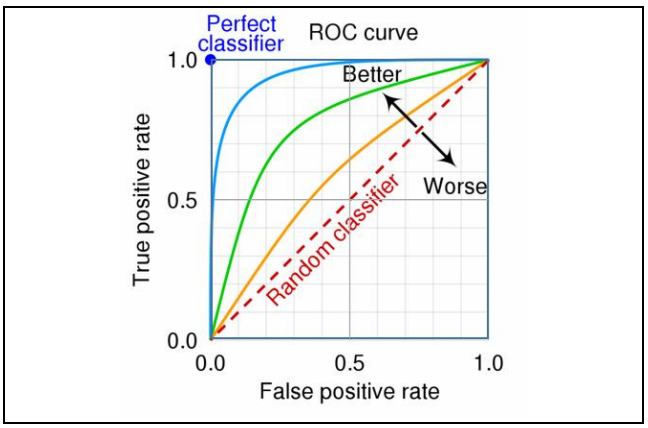
로 정의된다.

여기서 TP 는 실제로 True 이며 모델이 True 로 예측한 값, FP 는 실제로 False 이지만 True 로 예측한 값, FN 은 실제로 True 이지만 False 로 예측한 값, TN 은 실제로 False 이며 false 로 예측한 값 이다.

예를 들면, 실제 개가 12마리, 고양이가 9마리가 있는 디지털 사진을 보고 AI 프로그램이 8마리를 개로 식별(예측)하였고 가정해보자. 그런데 예측한 결과에서 실제는 개가 5마리이고 고양이가 3마리이다. 여기서 개로 식별된 8개 요소 중 실제로는 5개만 개(TP)이고 나머지 3개는 고양이(FP)이다. 또한 실제 7마리의 개는 예측에서 개가 아닌 것으로 누락되었고(FN), 6마리의 고양이는 올바르게 제외되었다(TN). 따라서 이 경우에 AI 예측의 정밀도는 5/8 이고, 재현률은 5/12이다.

한편 AUC 는 종속변수 값을 무엇으로 예측할 지 기준이 되는 cutoff(threshold) 값을 0부터 1까지 변화시키면서 테스트 했을 때, 각각의 경우에 대한 TPR(true positive rate, 양성률)과 FPR(false positive rate, 위양성률)의 비율을 그림으로 나타낸 것이다. FPR 은 실제 False 인 샘플 중 분류모델이 True 로 판정한 비율이고, TPR 은 실제 True 인 샘플 중 분류모델이 True 로 판정한 비율을 말한다. TPR 은 Recall 과 같다.

TPR값이 커질수록 모델의 성능이 좋아지고 반대로FPR 값이 커질수록 모델의 성능이 나빠진다. AUC값은 ROC(receiver operating characteristic) 곡선의 밑면적이며 모델의 성능이 좋을수록 곡선이 전체적으로 좌측 상단으로 넓어지면서, AUC 면적이 커진다. 최대값은 1 이다.



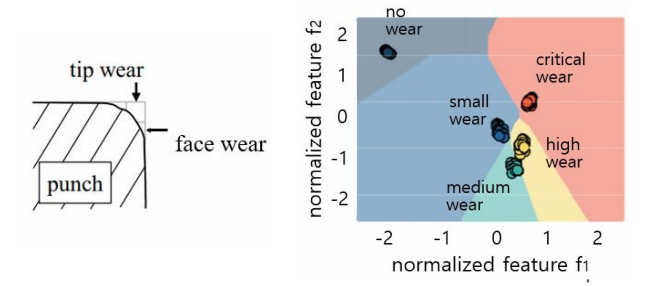


Fig. 12 Tool wear(a) and mSVM classifying five wear states [25]

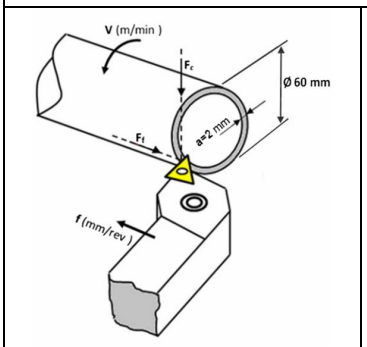
[예제 6] AISI 316L 오스테나이트 스테인레스 봉을 선삭가공할 때 공구동력계(Kistler Piezoelectric dynamometer)로 측정한 가공력에 대한 데이터[30]를 이용하여 절삭속도(cutting speed), 이송속도(feed rate), 절삭깊이(depth of cut) 등의 데이터로부터 가공력을 회귀분석한 연구결과를 아래에 나타내었다.

여기서 절삭깊이는 2mm, 인서트의 경사각(rake angle)은 -6°, 여유각 (clearance angle)은 5°, 가공물의 회전속도는 6500rpm, 노즈 반지름 (nose radius)는 0.8mm 이다.

그림 13(a)-(c)에 ANN, SVR, GBM 세 경우에 대해서 해석 정밀도를 비교하였다. SVR과 GBM 모델은 ANN모델의 프로그램 구성과 거의 동일하지만 model 정의 부분와 학습 부분이 달라진다. 이 선삭가공에 대한 회귀해석에서는 RMSE와 R2의 측정항목으로 평가한 가공력 예측 정밀도는 GBM과 MLP이 SVR 보다 상대적으로 우수한 것을 알 수 있다.

SVM은 의사결정트리 및 신경망과 같은 타 분류 알고리즘에 비해 많은 장점을 가지고 있다. SVM 훈련에는 볼록 손실함수의 최적화가 포함되기 때문에 역전파 신경망의 경우처럼 국소 최소값에 갇힐 위

```
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean squared error
from sklearn.metrics import r2_score
import pandas as pd

df = pd.read_csv('lathecutting.csv')

# Distinguish between input and output columns in a
data frame
X = df.drop('cutting force', axis=1)
y = df['cutting force']

# Split the data into training and testing sets
x train, x test, y train, y test = train test split(X,
y, test_size=0.3, shuffle=True, random_state=None)
x_val, x_test, y_val, y_test = train_test_split(x_test,
y test, random state=None, test size=0.5)

# Define the model
model = Sequential([
    Dense(units=10, input_dim=X.shape[1],
activation='relu'),
    Dense(units=30),
    Dense(units=1, activation='sigmoid')
])

# Compile the model
model.compile(loss='mse', optimizer='adam',
metrics=['mse'])

# Train the model
history = model.fit(x train, y train, epochs=500,
batch_size=1, validation_data=(x_val, y_val),
verbose=1)

# Model evaluation
yhat = model.predict(x_train)
y pred = model.predict(x test)
y_val_hat = model.predict(x_val)

# Calculate RMSE
def RMSE(y_train, yhat):
    return np.sqrt(mean_squared_error(y_train, yhat))

# Calculate R^2
r2_yhat = r2_score(y_train, yhat)

print("RMSE:", RMSE(y_train,
yhat).round(3)),print("R2:", r2_yhat.round(3))
```




```
y_test:
  0.19513
  0.31513
  0.63513
  0.22486
  0.76486
  0.48108
  0.68432
  0.73675
  0.30756
```

```
y_pred:
[[0.16929]
 [0.28979]
 [0.63096]
 [0.20972]
 [0.82288]
 [0.49772]
 [0.67734]
 [0.77515]
 [0.25773]]

RMSE:
0.00916
R2: 0.99818
```

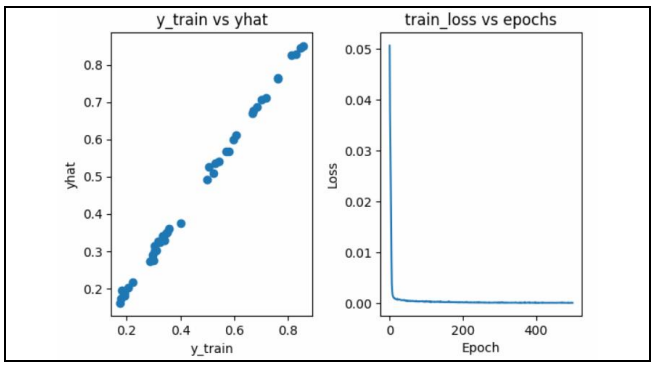


Fig. 13 (a) Lathe cutting force prediction in MLP method

(https://github.com/yskim9574/DFclass-2023/blob/main/ANN_lathecutting)

```
……
from sklearn.ensemble import GradientBoostingRegressor
……
# Define the model
model = GradientBoostingRegressor(n estimators=100,
learning_rate=0.1, max_depth=3, random_state=None)
# Train the model
model.fit(x_train, y_train)
……
```

```
RMSE: 0.00536, R2: 0.99942
```

Fig. 13 (b) Lathe cutting force prediction in GBM method

```
……
from sklearn.svm import SVR
……
# Define the SVR model
model = SVR(kernel='rbf', C=100, gamma=0.1,
epsilon=0.1)
# Train the model
model.fit(x train, y train)
……
```

```
RMSE: 0.05582, R2: 0.92765
```

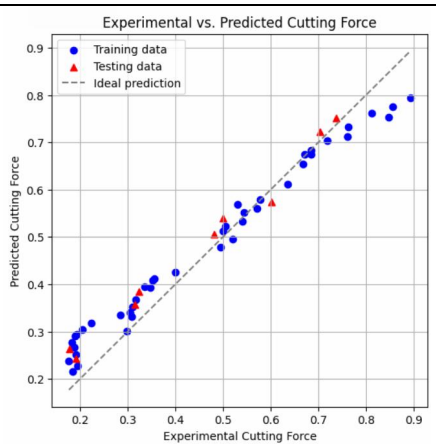


Fig. 13 (c) Lathe cutting force prediction in SVR method

험이 없고, 과적합이 덜 발생한다. 또한 서포트벡터 경계면 주변의 샘플만 학습에 사용하기 때문에 학습에 필요한 데이터 수가 적으며 신경망과 달리 최적화 해야 할 파라미터의 수도 훨씬 적다.

SVM의 또 다른 장점은 특정 커널 선택을 통해 다양한 기계 학습 아키텍처를 생성할 수 있는 통합 프레임워크를 제공하는 것이다.

위에서 논한 것과 같이 SVM은 특정 분류 작업에 유용하지만 실시간 처리가 필요한 경우에 대규모의 고차원 데이터를 처리하는 데에는 한계가 있다. 이런 경우는 방대한 양의 데이터에서 기능 계층을 자동으로 학습할 수 있는 딥 러닝 (deep learning) 방법이 보다 효과적이다. 한편 회귀 목적에 사용되는 SVR은 원래 분류 목적으로 개발된 SVM을 개선한 것이기 때문에 타 모델보다 회귀문제에 대한 해석 정밀도는 떨어지는 것으로 알려져 있다.

이상에서 논한 것 같이 기존에 전문가의 경험에 의존하여 해결하였던 제조현장의 문제들을 데이터 기반 통계기술을 접목한 머신러닝 기법을 적용하여 효율적으로 해결할 수 있다는 것을 확인하였다.

그러나 이 머신러닝 또는 AI 기술을 효율적으로 사용하기 위해서는 무엇보다도 품질이 우수한 데이터 확보가 우선되어야 한다. 우수한 품질이란 6가지 품질지수 - 완전성, 유일성, 유효성, 일관성, 정확성, 무결성 -가 향상된 데이터를 말한다. 실제 제조현장에서 받아들이는 공정 데이터와 IoT 센서 데이터의 시계열 데이터(sequential data)들에는 결측치(missing value)와 이상치(outlier) 등 불량 데이터들이 자주 포함되어 있다. 따라서 이들 수집된 데이터를 전처리하고 특징추출 작업을 거쳐 양질의 데이터셋을 확보하는 작업에 많은 노력과 시간이 소요된다[31a, b].

이렇게 정제된 데이터셋을 활용하여 머신러닝 기법을 적용하는데 있어서 다양한 알고리즘이 존재하기 때문에 위에서 열거한 예제에서 기술한 것과 같이 해결하려는 문제에 적합한 최적의 알고리즘을 선택하여야 한다.

국내 제조현장을 디지털 전환(digital transformation, DX)하여 제조업의 경쟁력 강화를 위해서는 머신러닝 기술을 통한 실시간 현장 데이터 모니터링과 시퀀스 데이터(sequential data) 또는 시계열 데이터(time-series data) 분석결과를 가시화하여 인사이트(insight-의사결정을 내릴 수 있는 통찰력)를 제공하는 시스템 구축이 필요할 것으로 판단된다.

3. 결 론

본 해설논문에서는 지난 호에 이어서 최근 다양한 사회과학 분야와 재료·가공기술 분야에 널리 사용되고 있는 AI 기술중의 하나인 머신러닝의 핵심 알고리즘 중 AdaBoost, Gradient Boosting Machine, Support Vector Machine 을 알기 쉽게 설명하였다.

본 해설논문에서는 각각 기술의 통계학적 복잡한 수식을 가능한 배제하고 개념적으로 각 알고리즘을 쉽게 설명하고자 하였다. 머신러닝에 대해 보다 공부하고자 하는 사람은 앤드류 응(Andrew Ng)의 유튜브 강의[20]와 머신러닝의 여러 알고리즘과 통계이론을 알기 쉽게 설명하는 유튜브 Statquest[32], Colah's Blog[33] 그리고 관련 전문서적[2-5, 34] 등을 참고하기 바란다.

머신러닝을 공부하고 여러 알고리즘의 성능을 평가하는데 활용되고 있는 다양한 테이블 형식의 데이터셋들이 세계 최대의 데이터 과학 커뮤니티인 kaggle.com[35a, b, c, d]에, 그리고 진동에 대한 시계열 데이터가 [35e, f, g]에 공개되어 있다. 한편 국내에서도 정부주도로 공공데이터포털이 운용되고 있어 다양한 분야의 빅데이터가 제공되고 있다. (www.data.go.kr) 관심이 있는 사람은 참고하기 바란다.

한편 제조현장의 기계 또는 대형 구조물에 부착된 센서로부터 실시간적으로 얻어지는 IoT 데이터나 언어와 같이 동적 특성을 갖는 시퀀스 데이터의 분석과 처리에 순환 신경망 (recurrent neural network, RNN)과 RNN을 보완한 장단기메모리(long short-term memory networks, LSTM) 방법과 시계열 데이터의 이상 탐지(anomaly detection)에 LSTM Autoencoder 방법이 자주 사용된다.

이들 모델은 이전에 나온 정보를 기억하면서 다음 데이터들을 학습할 수 있다는 특징 때문에 시퀀스 데이터 뿐 아니라 텍스트 데이터를 처리할 때 자주 사용된다[2-5, 33, 34]

다음 호에는 RNN, LSTM, LSTM Autoencoder 그리고 의료 영상과 같은 이미지 인식, 컴퓨터 비전에 의한 객체 검출, 오디오 신호데이터 분리 등에 널리 사용되는 합성곱 신경망(convolution neural network, CNN 또는 ConvNet라고도 부름)에 대해서 다룰 예정이다.

REFERENCES

[1] Y. LeCun, Y. Bengio, Geoffrey Hinton, Deep learning, Review Nature. 28(521) (2015) 436-444, https://doi.org/10.1038/nature14539

[2] D. P. Kroese, et al, 2023, Data Science and Machine Learning; Mathematical and Statistical Methods, ISBN-13978-1138492530

[3] M. di Nuzzo, 2021, Data Science and Machine Learning: From Data to Knowledge, ISBN-13979-8779849456

[4] V.N. Vapnik, 1998, Statistical Learning Theory. New York: Wiley, ISBN-978-0471030034

[5] E. Alpaydin, 2016, Introduction to Machine Learning. The MIT Press, Cambridge, Massachusetts, USA; London, England, 3rd edn.

[6] S.W. Bae, J.S. Yu, Estimation of the apartment housing price using the machine learning methods: The case of Gangnam-gu, Seoul, J. Korea Real Estate Analysts Assoc., 24(1) (2018), 69-85, http://dx.doi.org/10.19172/KREAA.24.1.5

[7] T. Hastie, R. Tibshirani, J. Friedman, 2009, The elements of statistical learning, data mining, inference, and prediction, Springer 2nd Ed., ISBN 978-0-387-84857-0

[8a] Y.S. Kim, J.J. Kim, Basics of Artificial Neural Network and its Applications to Material Forming Process I, Trans. Mater. Process., 30(4) (2021), 201-210.

[8b] Y.S. Kim, J.J. Kim, Basics of Artificial Neural Network and its Applications to Material Forming Process II, Trans. Mater. Process., 30(6) (2021), 311-322.

[9] K. Kim et al., Prediction of hardness for cold forging manufacturing through machine learning, Trans. Mater. Process., 32(6) (2023), 329-333, http://data.doi.or.kr/10.5228/KSTP.2023.32.6.329

[10] K.P. Kim et al., A study on improving formability of stamping processes with segmented blank holders using artificial neural network and genetic algorithm, Trans. Mater. Process., 32(5) (2023), 276-286, https://doi.org/10.5228/KSTP.2023.32.5.276

[11] M. Soori et. al., Machine learning and artificial intelligence in CNC machine tools, A review, https://doi.org/10.1016/j.smse.2023.100009

[12a] V.C. Do, Y.S. Kim et al., Effect of hole lancing on the forming characteristics of single point incremental forming, Procedia Engng., 84(2017) 35-42, https://doi.org/10.1016/j.proeng.2017.04.068

[12b] Pham Q. Tuan, Y.S. Kim et al., A machine learning-based methodology for identification of the plastic flow in aluminum sheets during incremental sheet forming processes, Int. J. Adv. Manuf. Technol., 120 (2022) 3559-3584, https://doi.org/10.1007/s00170-022-08698-z

[13] H. Salmenjoki, M. J. Alava, L. Laurson, Machine learning plastic deformation of crystals, Nature Communications, 9, (2018) 5307, https://doi.org/10.1038/s41467-018-07737-2

[14] T. Tancogne-Dejean et al., Recurrent neural network modeling of the large deformation of lithium-ion battery cells, Int. J. Plast., 146 (2021), 103072, https://doi.org/10.1016/j.ijplas.2021.103072

[15] X. Chen et al., Research on fault early warning and the diagnosis of machine tools based on energy fault tree analysis, Proc. IMech. Engng., Part B: J Eng. Manuf., 233(11) (2019) 2147-2159, https://doi.org/10.1177/0954405418816848

[16] F. Tao et al, Data-driven smart manufacturing, J. Manuf. Systems-C, 48(7) (2018), 157-169, https://doi.org/10.1016/j.jmsy.2018.01.006

[17a] www.kamp-ai.kr

[17b] Ministry of SMEs and Startups, Precision processing resource optimization AI dataset, analysis practice guidebook, 2021.

[18a] Y. Freund, R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comp. Syst. Sci., 55(1), (1997) 119-139, https://doi.org/10.1006/jcss.1997.1504

[18b] Y. Freund, R.E. Schapire., A short introduction to boosting, J. Japan. Soc. Artificial Intell., 14(5) (1999) 771-780 (In Japanese, translation by Naoki Abe.)

[19] Y.S. Kim, Review paper for key algorithms of machine learning and its application to material processing problems I, Trans. Mater. Process., 33(1)

(2024), 50-67.
[20] Andrew Ng, https://www.coursera.org/learn/machine-learning
[21] J.H. Friedman, Greedy function approximation: A gradient boosting machine, Ann. Statist. 29 (5) 1189-1232, https://doi.org/10.1214/aos/1013203451
[22] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning, 20, (1995), 273-297, https://doi.org/10.1007/BF00994018
[23] F. Parrella, On-line support vector regression, Thesis, Department of Information Science Univ. Genoa Italy June 2007.
[24] A.J. Smola, B. Schölkopf, 1998. A Tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, Univ. London, UK.
[25] C. Kubik et al., Smart sheet metal forming: importance of data acquisition, preprocessing and transformation on the performance of a multiclass support vector machine for predicting wear states during blanking , J. Intell. Manuf., 33 (2022) 259-282, https://doi.org/10.1007/s10845-021-01789-w
[26] P.E. Romero et al., Use of the support vector machine (SVM) algorithm to predict geometrical accuracy in the manufacture of molds via single point incremental forming (SPIF) using aluminized steel sheets, J. Mater. Res. Technol., 15 (2021), 1562-1571, https://doi.org/10.1016/j.jmrt.2021.08.155
[27a] S.M. Najm, I. Paniti, Predict the effects of forming tool characteristics on surface roughness of aluminum foil components formed by SPIF using ANN and SVR. Int. J. Prec. Engng. Manuf., 22(1), (2021) 13-26, https:// doi.org/10.1007/s12541-020-00434-5
[27b] S. M. Najm, I. Paniti, Investigation and machine learning-based prediction of parametric effects of single point incremental forming on pillow effect and wall profile of AlMn1Mg1 aluminum alloy sheets, J. Intell. Manuf. 34 (2023) 331-367, https://doi.org/10.1007/s10845-022-02026-8
[28] M. Dib et al., Model prediction of defects in sheet metal forming processes, In: Pimenidis, E., Jayne, C. (eds) Engineering Applications of Neural Networks. EANN 2018. Comm. Comp. Info. Sci., 893, https://doi.org/10.1007/978-3-319-98204-5_14
[29] J. Wang et al., A neural networks approach to investigating the geometrical influence on wrinkling in sheet metal forming. J. Mater. Process. Technol. 105, (2000), 215-220, https://doi.org/10.1016/S0924-0136(00)00534-3
[30] F. Kara et al., ANN and multiple regression method-based modelling of cutting forces in orthogonal machining of AISI 316L stainless steel, Neural Comput. Appl., (2014) https://doi.org/10.1007/s00521-014-1721-y
[31a] J.Y. Lee, Technology for collecting, processing, analyzing, and utilizing data for intelligent die-casting processes, J. Korean. Soc. Manuf. Eng., 29(6) (2020), 441-448, https://doi.org/10.7735/ksmte.2020.29.6.441
[31b] J.Y. Lee, Development of intelligence data analytics system for quality enhancement of die-casting process, J. Korean Soc. Precis. Eng., 37(4) (2020), 247-254, https://doi.org/10.7736/JKSPE.019.136
[32] https://www.youtube.com/@statquest
[33] https://colah.github.io/
[34] S. Raschka, V. Mirjalili, 2021, Python Machine Learning, 3rd Edition (H.S. Park, Korean Translation, Gilbut) ISBN9791165215187
[35a] https://www.kaggle.com/c/titanic/data
[35b] https://www.kaggle.com/datasets/uciml/iris
[35c] https://www.kaggle.com/code/agileteam/t2-2-pima-indians-diabetes
[35d] https://www.kaggle.com/datasets/altavish/boston-housing-dataset
[35e] http://www.timeseriesclassification.com/description.php?Dataset=FordA
[35f] www.kamp-ai.kr「Ford Engine Vibration AI Dataset」Analysis Practice Guidebook
[35g] https://fordatis.fraunhofer.de/handle/fordatis/151.2?mode=full&locale=en