기 술 동 향

머신러닝 기술의 핵심 알고리즘과 재료·가공 문제에 적용 I

김영석[1,#]

1. 경북대학교 기계공학부 교수

Review Paper for Key Algorithms of Machine Learning and its Application to Material Processing Problems

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[2-4]. 그 중에서도 AI 기술분야의 하나인 머신러닝(machine learning, 기계학습)은 ANN(artificial neural network, 인공신경망), 딥러닝(deep learning)[1] 등과 함께 복잡한 경계조건 하에서 다양한 변수들이 상호작용하고 있는 재료의 소성·가공 문제들을 해결과 신 제조기술 개발에 효율적으로 적용되고 있는 사례가 많아지고 있다[5-7].

머신러닝은 ANN에서와 같이 복잡한 코딩을 사용하지 않으면서 인간이 학습하는 것과 같이 컴퓨터(machine)가 현상에 대한 데이터를 설정된 알고리즘을 통해 스스로 학습하고 분석 모델을 구축하여 숨겨진 추세를 발견하고 추론과 예측(주로 회귀와 분류)하는 알고리즘을 통칭한다(그림 1). 여기서 회귀(regression)는 학생들의 국어, 영어, 수학 점수를 통해 수능점수를 예측하는 것과 같이 레코드의 연속형 속성의 값을 예측하는 것이고, 분류(classification)은 카드회사에서 회원들의 가입 정보를 통해 신용등급을 알아맞히는 것과 같이 레코드의 범주형 데이터(categorical/discrete class data) 속성의 값을 예측하는 것이다.

AI와 ML의 개념은 수학자 앨런 튜링(Alan Turing)이 1995년에 '경험으로부터 배울 수 있는 기계'를 만들 때 처음 제안하였다. 머신러닝은 1950년대 인공신경망(artificial neural networks, ANN)이 출현하면서 발전을 시작하였으며, 1980년대 후반 이후 상당기간 정체기를 겪었으나 2010년 이후 컴퓨터 성능의 획기적인 개선과 더불어 딥 러닝 (deep learning) 방법의 출현과 함께 다시 한 번 주목 받고 있다. ML은 복잡한 고차원 데이터의 패턴을 발견하고 비선형 관계를 조사하는 데 특히 강점을 갖고 광범위한 과학 연구 및 산업 응용을 위한 강력한 학문으로 부상하였다(그림 1)[8, 9].

이 머신러닝은 그림 1에 나타낸 것과 같이 (1) 문제의 정의·설계기획 (2) 데이터 수집 (3) 데이터 전처리(정제, 가공, 표준화 등) (4) 모델링(학습 모델선

Fig. 1 Workflow of machine learning algorithm

[출처] DOI:10.3389/fphar.2021.720694

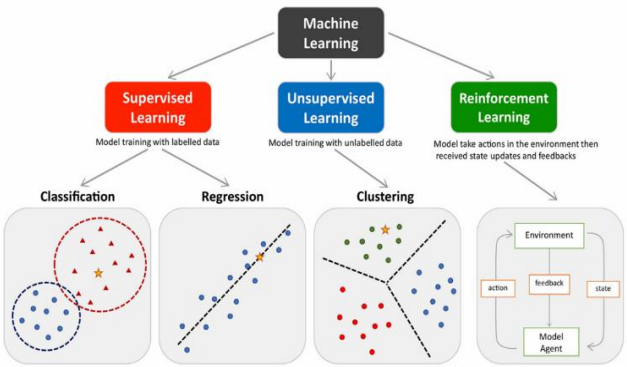


Fig. 2 Classification of machine learning

정, 모델학습, 하이퍼파라미터(초모수) 튜닝) (5) 모델 평가 및 예측(회귀, 분류) (6) 결과의 조치 및 성능개선 작업 등의 일련의 과정을 거쳐 수행된다.

이 머신러닝은 학습 방법에 따라 지도 학습(supervised learning)과 비지도 학습 (unsupervised learning), 강화학습 (reinforcement learning, RL)으로 구분할 수 있다(그림 2). 지도 학습은 입력 값(X data)과 출력 값(y data 또는 label)을 가지고 있는 데이터를 이용한 학습을 통해 경험하지 못한 데이터나 미래의 데이터에 관한 예측을 하는 것을 의미하며, 분류나 회귀 분석에 이용된다.

지도 학습의 대표적 학습 알고리즘은 서포트 벡터 머신(support vector machine, SVM), 의사결정나무(decision tree), k-최근접 이웃(k-nearest neighbors, kNN), 나이브 베이즈(Naive Bayes), 랜덤 포레스트(random forest), 인공신경망(artificial neural networks, ANN), 로지스틱 회귀(logistic regression) 등이 있다.

비지도 학습은 출력 값을 알 수 없는(정답 레이블이 없는) 데이터나 구조를 알 수 없는 데이터를 컴퓨터가 스스로 학습하여 데이터 내부의 패턴과 관계를 찾아내는 학습 알고리즘이다.

비지도학습의 대표적 학습 알고리즘에는 주성분 분석(principal component analysis, PCA), k-평균 군집화 (k-means clustering), GAN(generative adversarial network), DBSCAN (density-based spatial clustering of applications with noise) 등이 있다.

한편 구글의 딥마인드가 개발한 AlphaGo, AlphaS 등에 적용된 것으로 널리 알려진 강화학습은 행동심리학에서 나온 이론으로 분류할 수 있는 데이터가 존재하는 것도 아니고 데이터가 있어도 정답이 따로 정해져 있지 않으며 자신이 한 행동에 대해 보상(reward)를 받으며 학습하는 알고리즘을 말한다.

비지도학습, 지도학습, 강화 학습의 관계에 대해서 2016년 NIPS 학회(neural information processing system, 세계 최대 규모의 신경정보처리시스템학회)에서 뉴욕대학의 얀 르쿤(Yann André LeCun) 교수는 "지능이 케이크라면 케이크의 대부분은 비지도학습이고, 케이크 위의 장식은 지도학습이며, 케이크 위의 체리는 강화 학습이다". 라고 하였다.(LeCun’s cake analogy라고 알려짐)

최근의 다양한 분야에서 디지털 전환(digital transformation, DX)을 넘어 인공지능 전환(AI transformation, AX) 으로의 변화가 가속되고 있다. 이에 제조산업에서도 제조장치에 부착된 각종 센서들로부터 실시간으로 제조 IoT 데이터들을 수집, 분석하여 제조공정을 관리하고 최적화하여 스마트한 공장(smart factory)을 구축하려는 요구가 점증하고 있다 [10].

한편 대규모 빅데이터(1 petabite 정도)를 빠르게 처리할 수 있는 GPU 의 개발이 가속화되고 있다. 이에 따라 자동차 산업 등 연구개발 현장에서 널리 사용되고 있고 유한요소해석과 같은 복잡한 편미분 방정식을 풀어 수치해를 찾는 오프라인 기술과 더불어 소성·가공 산업 현장의 빅데이터들을 실시간으로 분석하여 합리적인 해를 구해가는 통계적 데이터 마이닝 기술에 기초한 AI및 머신러닝의 온라인 기술이 각광받고 있다. 또한 기존의 연속체적인 관점에서 주로 재료의 비선형적인 소성변형 거동을 다루었다면 보다 근본적으로 재료의 미시적인 결정소성학의 관점에서부터 거시적인 소성변형거동을 파악하려는 시도에서 이 머신러닝이 재료과학분야의 새로운 추진력이 되고 있다[11-13].

이 머신러닝 기술은 그 효용성으로부터 프레스 공정, 절삭가공 공정, 용접 공정, 다이캐스팅 등 뿌

리산업 모든 분야에 적용이 확대되어 가고 있다. 국가적으로도 중소벤처기업부가 지원하여 2021년 12월에 구축된 인공지능 중소 벤처 플랫폼(www.kamp-ai.kr)에는 국내 제조현장에서 축적된 제조 데이터를 활용, 중소기업의 설비·공정에서 발생하는 문제를 AI 기술을 접목하여 해결한 사례들과 많은 데이터 셋이 구축되어 있다. 또한 이 플랫폼에는 각 기업들이 해결하려는 제조현장의 다양한 현장이슈(pain point)를 AI 전문가 시스템을 이용하여 코딩 없이 AI 분석을 수행할 수 있는 툴을 제공하고 있다[14].

산업현장에서 주로 접하는 문제, 즉, 수학적인 공식이나 규칙으로 정의되지 않는 다양한 문제들의 해결을 위한 수단으로 머신러닝을 효율적으로 적용하기 위해서는 머신러닝 기술의 핵심 알고리즘[15, 16]에 대한 이해가 필요하다.

본 해설논문에서는 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 2회에 걸쳐서 다음과 같은 핵심 알고리즘에 대해서 알기 쉽게 설명하고 파이썬(python) 프로그램을 통해서 어떻게 알고리즘이 구현되고 있는지를 나타내었다.

(1) Decision Tree (2) Isolation Forest/Random Forest (3) Principal Component Analysis (4) AdaBoost (5) Gradient Boosting Machine (6) Support Vector Machine

이 외에 CatBoost, LightGBM, Logistic Regression, k-Nearest Neighbor, Naive Bayes 등도 자주 사용되는 알고리즘이지만 본 해설논문에서는 생략하기로 한다.

여기서 기술된 알고리즘에 대한 프로그램들을 구글의 Colab(https://colab.research.google.com/) 에서 구동하였다. 본 해설논문에서 다룬 예제들의 상세 프로그램은 깃허브 (https://github.com/ yskim9574/DFclass-2023)에 올려두었다. 참고로 ANN에 대해서는 소성가공학회에 게재된 저자의 해설논문[17, 18]을 참고하기 바란다.

2. 머신러닝 주요 알고리즘

2.1 Decision Tree

결정트리(decision tree)는 분류 작업(DecisionTree Classifier-출력변수가 범주형인 경우)와 회귀 작업

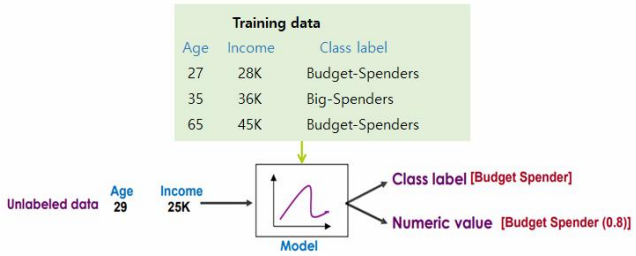


Fig. 3 Example of classification and regression

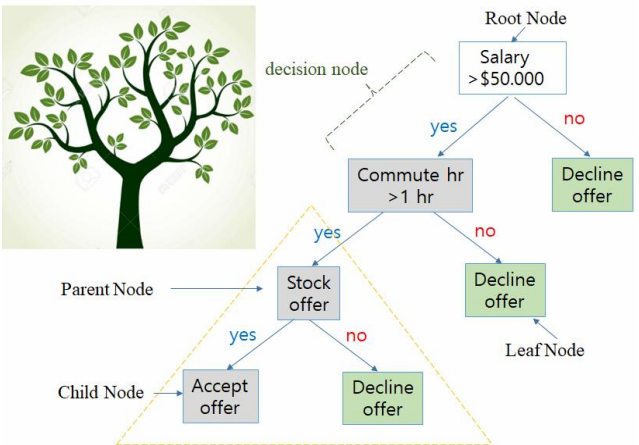


Fig. 4 Example of binary decision tree

(경력직 구직자가 취업할 회사를 선택하는 문제로 연봉이 50,000 불 이상이고 통근 거리가 1 시간 이내이면서 스톡을 주는 회사의 취업 제의를 받아들인다.)

(DecisionTreeRegressor-출력변수가 연속형인 경우)에 모두 적용 가능한 지도학습 머신러닝 알고리즘으로, 결정 트리는 질문을 던져서 대상을 좁혀나가는 '스무고개' 놀이와 비슷한 개념이다[19]. 이렇게 특정 기준(질문)에 따라 데이터를 구분하거나 분류하는 모델을 결정트리 모델이라고 한다. 결정트리는 자주 의사결정트리라고 불린다.

다음 그림 3은 나이(age), 소득(income)에 따라 소비자 성향을 지도학습에 따라 트리모델로 분류하기 위한 예를 나타낸 것이다. 나이가 29세이고 소득이 25,000불이면 약 80% 정도 확률로 절약 소비자(budget spender)로 분류할 수 있다는 것을 나타낸다.

의사결정트리의 모형은 크게 성장(growing)과 가지치기(pruning) 단계를 통해 만들어진다. 이하에서는 결정트리에서 마디를 2진 분류시키는 CART (Classification And Regression Tree) 알고리즘 의한 성장에 대해서 설명한다(그림 4).

의사결정트리를 포함하여 일반적으로 트리란 노

드로 이루어진 자료구조를 말한다. 이 트리는 하나의 루트 노드를 가지며, 루트 노드는 0개 이상의 자식 노드(child node)를 갖는다. 그 자식 노드 또한 0개 이상의 자식 노드를 가질 수 있다.

이진트리(binary tree) 형태의 결정트리에서는 마디에서 한 번의 분리 때마다 변수 영역을 두 개로 구분한다. 결정트리에서 질문이나 정답을 담은 상자를 노드(node)라고 부르며 맨 처음 분류 기준 (즉, 첫 질문)을 루트 노드(root node)라고 한다. 루트 노드 분리로 인해 발생하는 중간에 있는 노드를 결정 노드(decision node)라고 부르며 이 결정노드는 트리 내의 중간 결정이나 조건을 나타낸다. 주어진 모든 데이터가 분류될 때까지 이 과정을 반복한다.

데이터가 모두 같은 분류 항목에 속하면 제대로 분류가 된 것이므로 더 이상의 트리 성장(생성)을 종료한다. 추가 분리가 불가능한 맨 마지막 노드를 잎 노드(leaf node) 또는 말단 노드(terminal node)라고 부르며, 종종 최종 분류 또는 결과를 나타낸다.

결정트리 모델에서 불순도(impurity)란 해당 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지를 뜻한다. 한 범주에 하나의 데이터만 있다면 불순도가 최소 또는 순도(homogeneity)가 최대이고, 한 범주 안에 서로 다른 두 데이터가 정확히 반반씩 있다면 불순도가 최대 또는 순도가 최소이다.

결정 트리는 불순도를 최소화 또는 순도를 최대화하는 방향으로 분리하며 학습을 진행한다. 불순도의 측도로는 엔트로피, 지니계수, 정보획득, 오차제곱, 카이제곱 통계량, 분산감소량 등이 사용된다.

엔트로피(Entropy)는 시스템의 무질서 또는 무작위성의 척도이며, 열역학의 기본 개념이다. AI에서 이 엔트로피는 해당 범주 안에서의 불순도를 의미하며 다음 식으로 나타내진다.

$$Entropy = E = -\sum_{i=1}^{k} p_i \log_2(p_i) \qquad (1)$$

여기서 p_i 는 한 영역(레코드) 안에 존재하는 데이터 가운데 범주 i에 속하는 데이터의 비율을 나타낸다.

엔트로피가 높다는 것은 불순도가 높은 것이며, 엔트로피가 낮다는 것은 불순도가 낮다는 것이다. 한 범주 안에 서로 다른 데이터가 정확히 반반씩 있다면 엔트로피가 1이 되며 불순도가 최대가 된다. 한편 한 범주 안에 하나의 데이터만 있다면 엔트로피가 0이 되며 불순도는 최소가 된다.

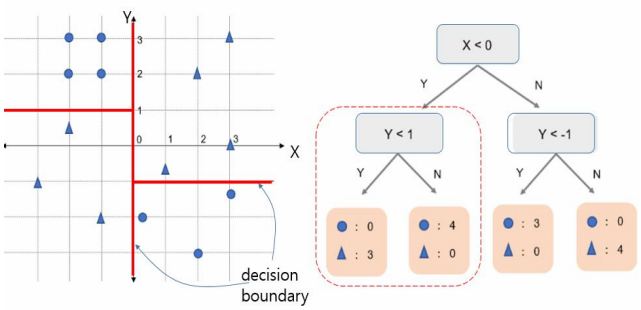


Fig. 5 Geometrical interpretation of decision tree

한편 트리로 분리(분기) 이전의 엔트로피에서 분리 이후의 각 부분 집합의 정보량의 가중 평균 엔트로피(weighted entropy)를 뺀 수치를 정보 이득(information gain, I.G)이라고 한다.

이 정보 이득은 데이터 분류하는 데 해당 속성의 효율성을 계량화하기 위한 척도이며 정보 이득이 클수록 확실성이 증가한다. 결정트리는 정보 이득이 가장 큰 방향으로 학습이 진행되도록 설계한다. 정보 이득이 가장 크다는 것은 해당 범주에 대한 엔트로피가 가장 작다는 의미이며 불순도가 가장 작다는 의미이다. 즉 결정트리를 불순도가 작아지는 방향으로 성장시켜간다는 것이다.

그림 5에는 x,y 평면 상에 7개의 원과 7개의 삼각형 데이터가 혼재하고 있을 때 데이터를 불순도가 0으로 분류해가는 결정트리의 개념을 나타낸 것이다.

결정트리 모델에는 ID3(Iterative Dichotomiser 3), C4.5, C5.0, CART 등 다양한 알고리즘이 존재한다. 이들 알고리즘은 선택의 순간마다 당장 눈앞에 보이는 최적의 상황만을 쫓아 근사적인 최종 해답에 도달하는 탐욕 알고리즘(greedy algorithm)이다.

ID3 알고리즘은 Quinlan[20]에 의해 개발된 결정트리의 가장 초창기 모델로 불순도 지표로 엔트로피를 사용하며 독립변수가 모두 범주형 변수일 경우에 사용 가능한 알고리즘이다. 데이터 마이닝에서 가장 많이 사용되는 알고리즘은 C4.5 또는 C5.0이다.

[예제 1] 다음 예는 도로의 경사 상태 속성(slope), 표면 상태 속성(surface) 그리고 속도 제한 속성(speed limit)에 따른 차량의 속도 구분(speed class)을 나타내는 표이다. 이 표로부터 엔트로피 척도를 이용하여 결정트리를 성장시켜 가는 방법에 대해서 설명한다. 이 표로부터 경사 상태 속성에는 steep(급

Table 1 Information gain for each feature's separation

Predictors			Target
Slope	Surface	Speed limit	Speed
Steep	Bumpy	Yes	Slow
Steep	Smooth	Yes	Slow
Flat	Bumpy	No	Fast
Steep	Smooth	No	Fast

$$E(target) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(0.5\log_2 0.5 + 0.5\log_2 0.5)$$
$$= -(0.5\log_2 (2)^{-1} + 0.5\log_2 (2)^{-1}) = 1$$

경사 상태(slope)를 기준으로 분리하는 경우

		Speed		
		Slow	Fast	
Slope	Steep	2	1	3
	Flat	0	1	1
				4

$$E(slope, steep) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(\frac{2}{3}\log_2 \frac{2}{3} + \frac{1}{3}\log_2 \frac{1}{3}) = 0.9184$$
$$E(slope, flat) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(0\log_2 0 + 1\log_2 1) = 0$$
$$E(slope) = \frac{3}{4} * E(slope, steep) + \frac{1}{4} * E(slope, flat)$$
$$= \frac{3}{4} * 0.9184 + \frac{1}{4} * 0 = 0.6888$$

여기서 $E(slope)$은 경사 상태 속성으로 분리한 후 각 부분 집합의 정보량의 가중 평균 엔트로피이다.
경사 상태 속성을 기준으로 분리를 했을 때는 0.3112만큼의 정보 이득(엔트로피의 감소)이 있다.

$$I.G(target, slope) = E(target) - E(slope) = 0.3112$$

표면 상태(surface)를 기준으로 분리하는 경우

		Speed		
		Slow	Fast	
Surface	Bumpy	1	1	2
	Smooth	1	1	2
				4

$$E(surf, bump) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2}) = 1$$
$$E(surf, smooth) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2}) = 1$$
$$E(surface) = \frac{2}{4} * E(surf.bump) + \frac{2}{4} * E(surf, smooth)$$
$$= \frac{2}{4} * 1 + \frac{2}{4} * 1 = 1$$

표면 상태 속성을 기준으로 분리했을 때는 정보 이득이 0.0으로 정보이득이 전혀 없다.

$$I.G(target, surface) = E(target) - E(surface) = 0$$

속도 제한(speed limit)를 기준으로 분리하는 경우

		Speed		
		Slow	Fast	
Speed limit	Yes	2	0	2
	No	0	2	2
				4

$$E(sl, yes) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(\frac{2}{2}\log_2 \frac{2}{2} + 0\log_2 0) = 0$$
$$E(sl, no) = -(p_{slow}\log_2 p_{slow} + p_{fast}\log_2 p_{fast})$$
$$= -(0\log_2 0 + \frac{2}{2}\log_2 \frac{2}{2}) = 0$$
$$E(sl) = \frac{2}{4} * E(sl, yes) + \frac{2}{4} * E(sl, no) = \frac{2}{4} * 0 + \frac{2}{4} * 0 = 0$$

속도 제한 속성을 기준으로 분리했을 때는 1.0만큼의 정보 이득이 있다.

$$I.G(target, sl) = E(target) - E(sl) = 1$$

경사)와 flat(완만)의 두 가지 클래스가 있으며, 표면 상태 속성에는 bumpy(요철)와 smooth(평탄)의 두 가지 클래스가 있고, 속도 제한 속성에는 yes(속도제한 있음과 no(없음)의 두 가지 클래스가 있다.

경사 상태, 표면 상태, 속도 제한의 각 속성의 경우로 분리하였을 때의 정보이득과 가중평균 엔트로피는 표 1과 같이 계산된다.

따라서 경사 상태 속성, 표면 상태 속성, 속도 제한 속성을 기준으로 한 경우에 각각의 정보이득은 0.3112, 0.0, 1.0이다. 속도제한 속성을 기준으로 분리했을 때 정보 이득이 가장 큰 1.0이었으므로 정보이득이 가장 큰 경우인 속도 제한을 결정노드로 선택하여 첫 분리 점으로 하고 학습이 진행되도록 트리를 성장시켜 간다.

결정트리는 적절한 정지규칙을 만족하거나 자식 노드에 한 가지 속성의 데이터만 존재한다면 (불순도가 0) 더 이상 자식 노드를 나누지 않고 중단한다.

결정트리의 분리속성 척도로 엔트로피의 정보 이득을 사용할 때에는 속성 값이 많은 것이 결정노드로 선택될 확률이 높다. 이 단점을 보완하기 위해 개선된 척도로 경제학에서 불평등 지수를 나타낼 때 주로 사용하는 다음과 같은 지니계수(Gini Index)가 불순도 측정 지표로 자주 이용된다. 주어진 샘플 셋트 S에 대해서 지니계수는

$$G.I(S) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (2)$$

또한 가중 평균 지니계수는

$$Weighted\ G.I(S) = \frac{|S_1|}{|S|} G.I(S_1) + \frac{|S_2|}{|S|} G.I(S_2) \qquad (3)$$

여기서 k 는 범주의 개수이고 p_i 는 범주 i 에 속해 있는 S에서의 샘플의 비율이다.
예를 들면, 구슬 10개 중에서 파란 구슬이 2개, 빨간 구슬이 8개가 있다면 지니불순도는 0.32이며,

$$1 - \left((\frac{2}{10})^2 + (\frac{8}{10})^2 \right) = 0.32$$

파란 구슬과 빨간 구슬이 각각 5개가 있다면 지니불순도는 0.5이다.

지니 불순도는 집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표이며 CART 알고리즘에서 사용한다. 집합에 있는 항목이 모두 같다면 즉, 불순물이 없이 깨끗하게 분류되어 있다면 지니 불순도는 위의 공식에 따라 최솟값 0을 갖게 되며 이 집합은 완전히 순수하다고 말할 수 있다. 하지만 섞이게 되면 0보다 큰 값을 가지게 되고, 이때 최댓값은 0.5 이다.

Table 2 Dataset example for decision tree

RID	age	student	credit-rating	class buys computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	yes	fair	no
5	middle-aged	no	excellent	yes
6	senior	no	fair	no
7	senior	yes	excellent	yes

의사결정트리는 계산복잡성 대비 높은 예측 성능을 내는 것으로 알려져 있지만 데이터 축에 평행한 결정 경계(decision boundary)만을 갖기 때문에 특정 데이터에만 잘 작동할 가능성이 높다. 이와 같은 문제를 극복하기 위해 같은 데이터에 대해 의사결정나무를 여러 개 만들어 그 결과를 조합하여 예측 성능을 높이는 앙상블 기법이 랜덤포레스트이다. 참고로 kNN의 결정 경계는 직선이 아닌 임의 형태를 가지며 k값이 커질수록 그 경계가 완만한 곡선이 된다.

의사결정트리의 분할이 거대해질수록 트리 구축에 많은 시간이 소요되고 해석에 과적합이 발생할 수 있기 때문에 적당한 수준에서 트리의 성장을 중단하도록 해야 한다. 이를 위해 의사결정트리의 최대 깊이(max_length)를 지정하거나, 최소 마디 수(min_samples_split)또는 최소 샘플 수(min_samples_leaf)를 지정하는 방법 등이 사용된다, 이 방법을 사전 가지치기(pre-pruning)라고 부른다. 또한 과적합을 방지하기 위해 중요하지 않은 노드나 하위 노드를 잘라내는 사후 가지치기(post-pruning) 도 트리의 성능을 향상시키는 데 도움이 된다.

[예제 2] 의사결정트리 알고리즘에 대한 구체적인 예로서 아래 데이터셋에 나타낸 나이(age), 학생 여부(student), 신용도(credit-rating) 에 따른 컴퓨터 구입 여부(buy computer)를 의사결정트리의 CART 알고리즘으로 분류하는 과정을 이하에 설명한다.

이 CART 알고리즘을 이용하여 나이, 학생 여부와 신용도가 정해질 때 컴퓨터를 사는지 안 사는지를 예측할 수도 있다.

이 데이터셋의 경우에 의사결정트리의 처음 분류 기준인 루트노드를 나이로 할지, 학생으로 할지 아니면 신용도로 할지에 따라 결정트리의 전개가 달라진다. 표 3에 루트노드 선택에 따른 트리의 전개를 나타내었고, 가중 평균 지니계수를 이용하여 트리 전개 방법에 대해서 설명한다.

CART 알고리즘은 이진트리 분리를 이용하므로 나이 속성에 따라 분리할 때 youth, middle, senior로 세 가지로 분리하지 않고 youth와 (middle, senior),

Table 3 Decision tree growth based on each feature

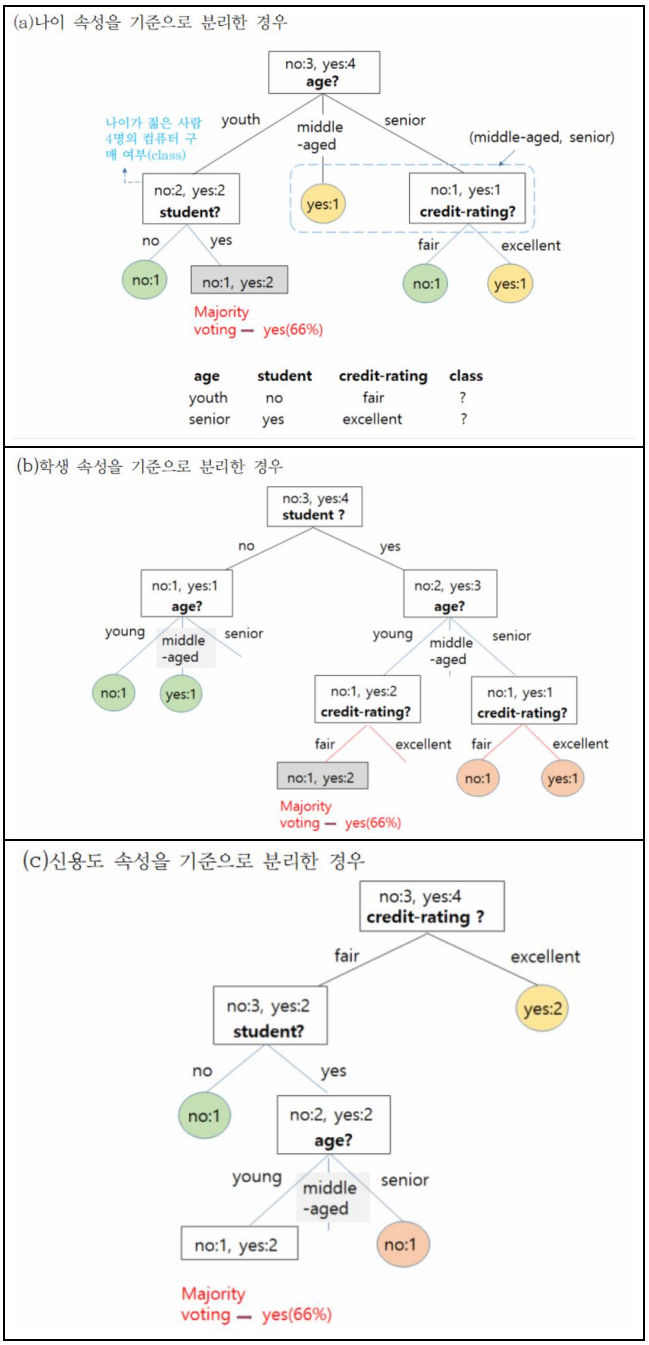
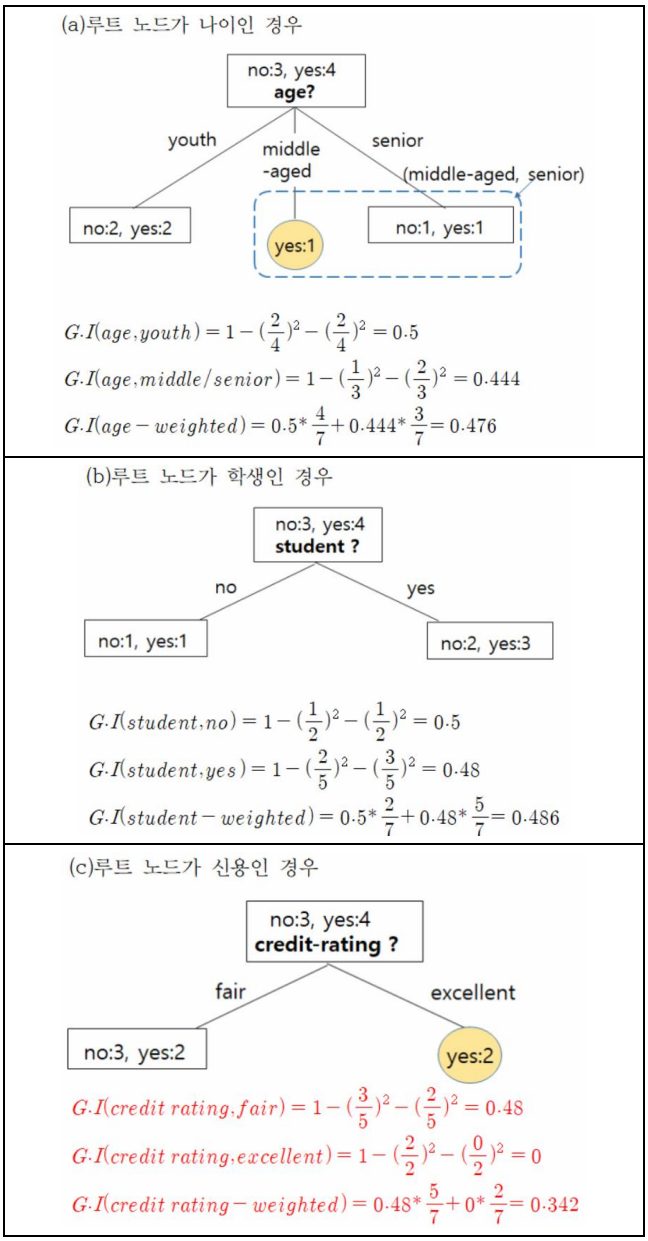


Table 4 Weighted gini index for each root node



middle과 (youth, senior), senior과 (youth, middle)으로 이진트리 분리한다.

아래 예에서는 youth와 (middle, senior)로 분리한 경우만을 나타내었다. 나이 속성의 각 분리에 대해서 지니계수를 계산하고, 또한 학생 속성과 신용도 속성에 대해서도 마찬가지 분리에 대해서 지니계수를 계산한다. 먼저 컴퓨터 구입 여부 class에 대한 14개(yes 9, no 5) 데이터에 대한 지니계수는 $1-(\frac{5}{14})^2-(\frac{9}{14})^2=0.459$ 이다.

Table 5 Dataset example for decision tree

RID	age	student	credit-rating	class buys computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	yes	fair	no
6	senior	no	fair	no

이상의 나이, 학생 여부와 신용도 속성을 기준으로 각각 분리한 경우에 대한 가중 평균 지니계수(weighted Gini index)를 계산한 결과(표 4), 신용도를 fair와 excellent로 나누었을 때 지니계수가 0.342로 가장 작기 때문에 이 분리에서 불순도가 가장 낮아진다는 것을 알 수 있다. 따라서 결정트리에서 신용도를 fair와 excellent로 가장 먼저 분리하며 신용도를 루트노드로 한다. 위의 결정트리 분리과정에서 불순도가 0인 경우, 즉 단일 속성만이 존재하는 경우는 원형으로 표시하고 더 이상 자식 노드를 나누지 않는다.

표 4에서 신용도를 루트노드로 하여 분리 후 단일 속성만으로 구성된 excellent 행(5행과 7행)을 삭제한 새로운 데이터셋에 대한 표 5를 만들고 [no:3, yes:2] 노드를 다시 분리하여야 한다.

이 노드를 student 속성으로 분리할 지 age 속성으로 분리할지를 결정하기 위해 위에서와 같이 각각의 경우에 가중 평균 지니계수를 구해서 가장 작은 지니계수를 갖는 속성으로 트리를 분할한다.

그런데 두 경우에 가중 평균 지니계수값이 같기 때문에 어느 분리도 허용된다. 분리가 완료 될 때까지 이 과정을 반복해간다(표 6).

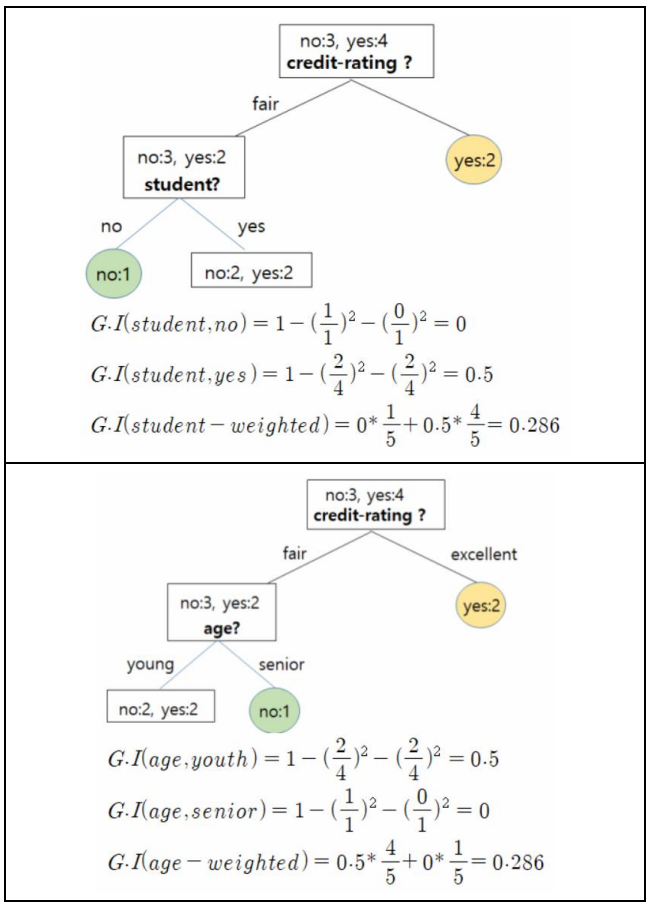
이 문제에 대한 파이썬 코드를 아래 깃허브에 올려 두었다. 파이썬 프로그램에 의한 결과(그림 6)는 앞에서 신용도를 루트 노드로 하여 전개한 트리와 같은 결과를 나타내고 있음을 알 수 있다.

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

2.2 Isolation Forest/Random Forest

비지도학습 머신러닝 알고리즘을 이용한 이상탐지(anomaly detection) 방법의 종류로는 (1) 마할라노비스 거리(Mahalanobis distance)를 이용한 방법, (2) Isolation Forest 방법, (3) z-score 변환 후 이상데이터

Table 6 Weighted gini index for each root node of table 5



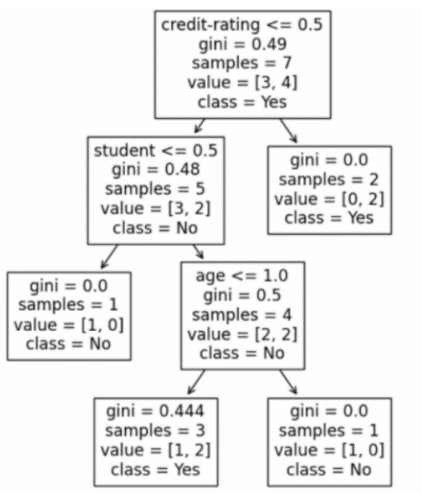


Fig. 6 Decision tree growth for classification study

를 제거하는 방법, (4) IQR(Interquantile Range) 방법, (5) K-Means를 통한 군집화 방법, (6) DBScan를 이용하는 방법 등이 있다.

이 중에서 2008년에 발표된 Isolation Forest는 여러

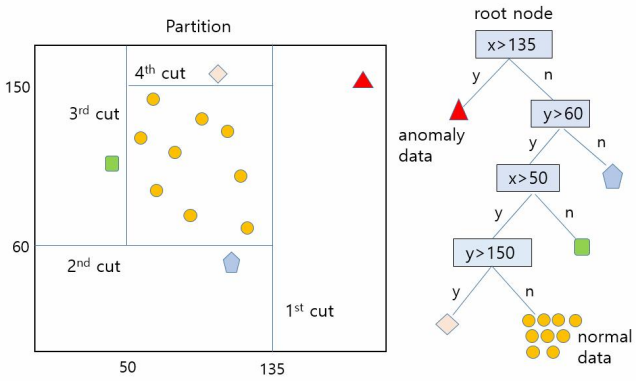


Fig. 7 Isolation Forest algorithm

```
data = np.array([[45,100], [57,120], [63,90], [65,145],
[80,110], [85,75], [90,130], [100,120], [110,95],
[115,70], [110,52], [95,160], [180,152]])
```

개의 이진 의사결정트리(binary decision tree)를 종합한 앙상블 기반의 비지도학습 이상탐지 기법이다[20]. 이 방법은 속성을 랜덤하게 선택하고 해당 속성의 데이터의 최대값과 최소값 사이에서 랜덤하게 데이터를 분리하는 과정을 반복하여 모든 데이터 관측값을 고립(=분리)시키며 고립 정도 여부에 따라 이상값(=이상 데이터)을 판별하는 방법이다(그림 7). 특히, 이 방법은 변수가 많은 데이터에서도 효율적으로 작동하는 장점이 있다.

이 Isolation Forest의 기본 아이디어는 정상 데이터는 고립시키기 어렵기 때문에, 많은 재귀 분할(recursive partitioning)을 통해 말단 노드에 가까운 깊은 깊이(depth, 트리분할 횟수에 해당함) 또는 경로길이(path length)를 가지지만, 이상 데이터는 정상 데이터에 비해 고립시키기 쉽기 때문에, 루트 노드에 가까운 깊이를 가진다는 개념을(즉, 빠르게 분리된다는 것을 의미) 기반으로 한다.

특정 한 개체가 고립되는 말단 노드까지의 깊이를 이상값 점수(outlier score 또는 anomaly score)로 정의하며, 루트 노드로 부터의 평균 경로길이가 짧을수록 이상값 점수는 높아진다.

이상값 여부를 판단하기 위한 점수(score) S 는 다음과 같은 결정함수값(decision function)으로 계산한다.

$$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (4)$$

n 은 랜덤하게 트리에 사용되는 최대 샘플 수

(max_samples), $h(x)$ 는 트리가 고립되는데 사용된 트리분할 횟수인 경로길이, $E(h(x))$ 는 각 트리 별 고립에 사용된 경로길이의 평균에 해당 하는 평균 경로길이, $c(n)$ 는 max_samples에 대한 평균 경로길이를 정규화하기 위한 값으로 각 트리 별로 사용된 데이터 개수 n 에 대해

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \quad (5)$$

$H(n)$ 은 조화 급수의 n항까지의 부분합인 조화수(harmonic number)이며

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.577215 \quad (6)$$

으로 정의된다. 여기서 $E(h) \to c(n)$ 이면 $S \to 0.5$, $E(h) \to 0$ 이면 $S \to 1$, $E(h) \to n-1$ 이면 $S \to 0$ 이다. Score는 $0 \sim 1$ 사이에 분포되며, 1에 가까울수록 이상값일 가능성이 크고 0.5 이하이면 정상 데이터로 판단할 수 있다.

따라서 위 식으로부터 $E(h)$ 가 높으면 점수는 낮고, 반대이면 점수가 높다는 것을 알 수 있다. $E(h)$ 가 높다는 것은 데이터가 고립되기까지 깊이가 깊다는 것이므로 정상데이터일 확률이 높고, $E(h)$ 가 낮으면 데이터가 빨리 고립되기 때문에 이상값일 확률이 높다. 따라서 점수가 높으면 이상 데이터, 낮으면 정상 데이터라고 판단한다. 일반적으로 이상 여부를 판단하기 위한 이상값 점수는 (0.5-결정함수값)를 사용한다.

따라서 Isolation Forest의 알고리즘에서는 (1) 학습 과정에서 입력 데이터 x를 이용해 t개의 하위 샘플(sub-sample)과 트리를 만들어서 앙상블한 후 (2) 평가과정에선 학습과정에서 생성한 t개의 트리에 대해 모든 데이터 포인트 x의 경로길이를 계산하고 계산된 경로길이를 기반으로 각 데이터 포인트의 이상값 점수를 평가한다.

Isolation Forest와 유사한 것으로 Breiman[21]에 의해 제안된 앙상블 학습 모형인 Random Forest는 여러 개의 의사결정트리가 모인 것으로, 하나의 의사결정트리가 오버 피팅되는 경향을 해결하기 위해 배깅기법을 적용하여 훈련 데이터의 약간의 중복을 허용하면서 여러 개의 트리를 만들어 학습을 하고 각 트리의 결과를 결합하여 최종 결과값을 구한다(그림 8).

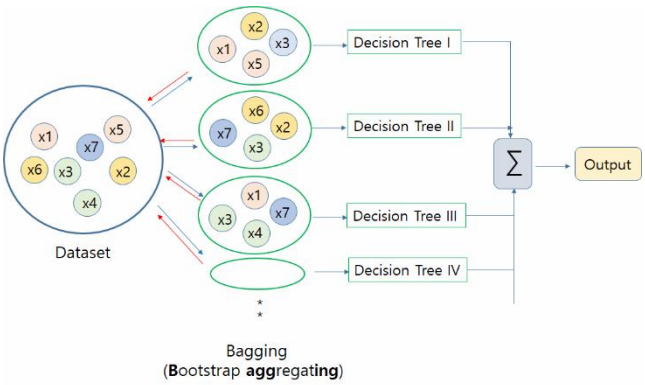


Fig. 8 Example of random forest algorithm

그림에서 배깅(Bagging)은 Bootstrap Aggregating의 약자로, 이 샘플을 여러 번 뽑아(Bootstrap, 브트스트랩) 각 모델을 독립적으로 학습시켜 결과물을 집계(aggregating)하는 방법을 말한다. 통계학에서 모집단의 성질에 대해서 표본(샘플)을 통해 추정할 수 있는 것처럼, 브트스트랩은 표본의 성질에 대해서도 재표집(resampling)을 통해 추정할 수 있다는 것이다. 즉 주어진 표본에 대해서, 그 표본에서 또 다시 재표본을 여러 번 (1,000~10,000번, 혹은 그 이상) 추출하여 표본의 평균이나 분산 등이 어떤 분포를 가지는가를 알아낼 수 있다.

예를 들면. 7개의 데이터셋에서 4개의 서브 데이터셋 샘플을 뽑아서 의사결정트리를 작성하여 통계량을 계산하고 이 샘플을 다시 데이터셋에 복원시킨 후 다시 4개의 샘플을 추출하여 통계량을 계산하는 과정을 n번 반복하여 재표본추출한 값을 집계하여 결과값을 구한다.

범주형 데이터(categorical data)는 투표 방식(voting)으로 결과를 집계하며, 연속형 데이터(continuous data)는 평균으로 집계한다.

비지도학습을 기반으로 하는 Isolation Forest는 주로 이상 탐지를 위해 사용되지만 지도학습을 기반으로 하는 Random Forest는 주로 분류 및 회귀 작업에 사용된다.

[예제 3] 그림 8에 나타낸 데이터셋을 이용하여 Isolation Forest를 구성하는 방법과 이상값 점수를 계산하는 방법을 이하에 설명한다. 데이터를 고립시키는 방법을 기하학적으로 설명하기 위해 각 데이터가 x_1, x_2 좌표를 갖는 것으로 하고 표 7과 같은 데이터셋을 이용하였다.

7개의 데이터셋에서 4개씩 서브 데이터셋을 랜덤하게 추출하고 각 서브 데이터셋에서 랜덤하게 선택한 속성의 최대값과 최소값 사이에서 속성에 대한 데이터를 분리하는 과정을 반복한다(표 8-1).

Table 7 Dataset for isolation forest algorithm

No	x_1	x_2
1	1	1
2	2	2.5
3	3	2
4	4	4
5	5	3.5
6	6	3
7	7	3.5

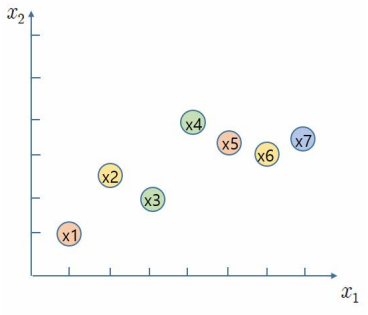
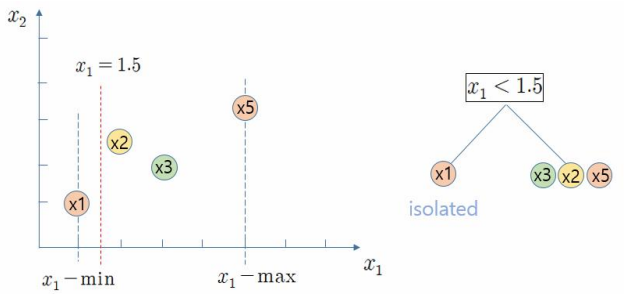
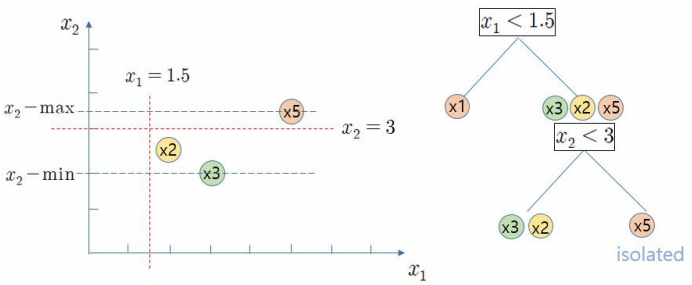


Table 8-1 Isolation forest algorithm for classification of sub-sampling data 1

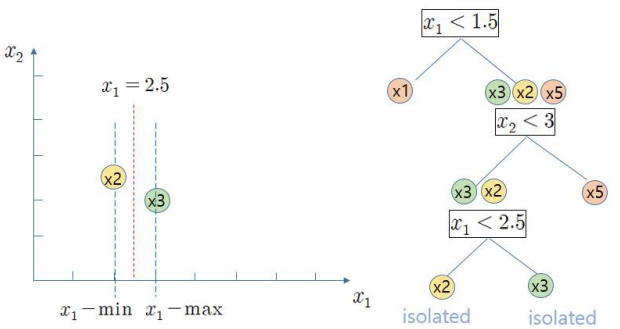
No	x_1	x_2
1	1	1
2	2	2.5
3	3	2
5	5	3.5



No	x_1	x_2
2	2	2.5
3	3	2
5	5	3.5

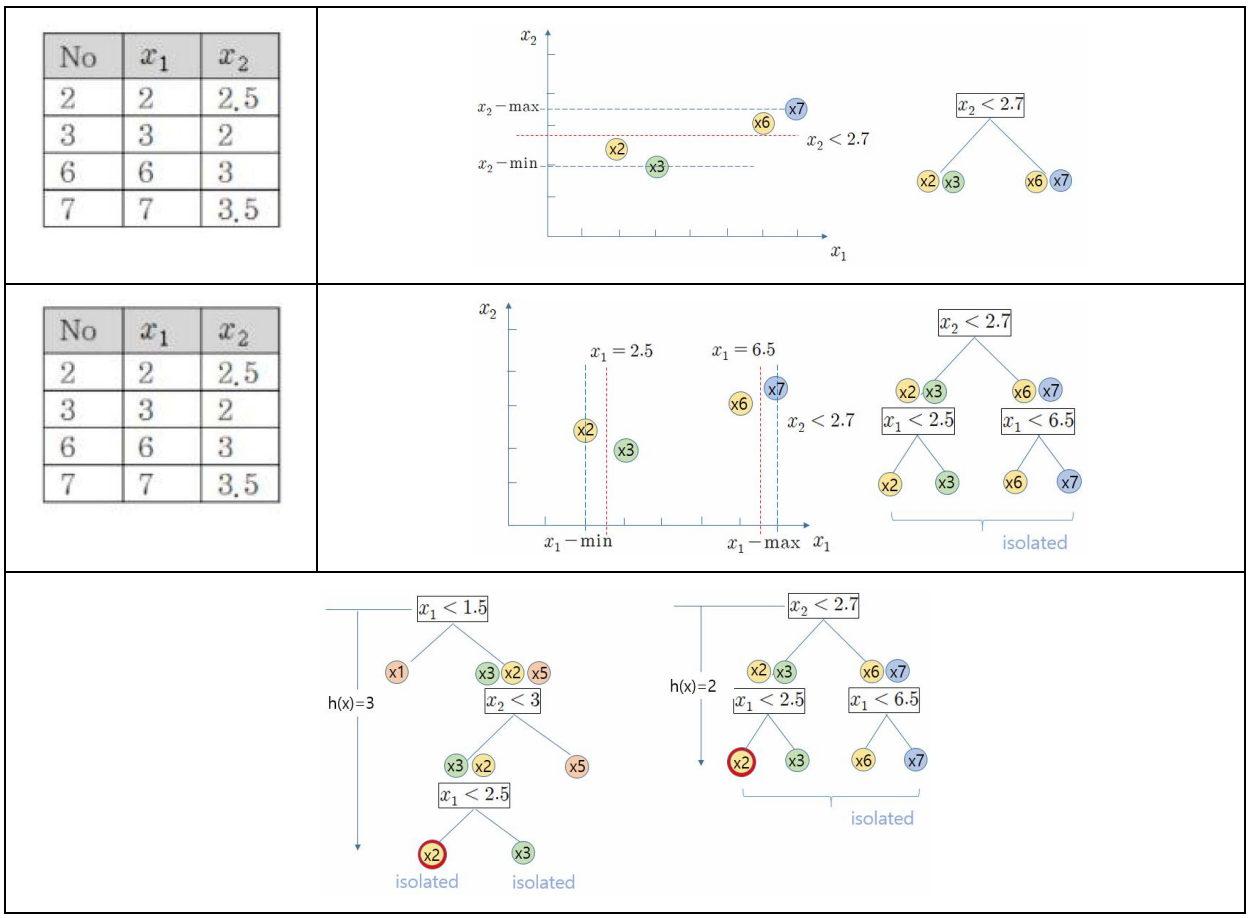


No	x_1	x_2
2	2	2.5
3	3	2



여기서는 그림 7에서 편의 상 7개의 데이터셋에서 위의 2개의 서브 데이터셋 (1,2,3,5)와 (2,3,6,7) 만을 추출하여 2개의 트리를 구성하는 경우에 대해서 설명한다. 첫 번째 서브 데이터셋 (1,2,3,5)을 대상으로 x_1과 x_2 속성 중에서 랜덤하게 속성 x_1를 선택하였다면, $x_1 - \min = 1$, $x_1 - \max = 5$ 사이에 임의의 값 $x_1 = 1.5$를 선택하는 경우에는 첫 번째 트리에서 x1 데이터가 고립된다.

Table 8-2 Isolation forest algorithm for classification of sub-sampling data 2



물론 $x_1 = 2.5$로 택한 경우는 첫 번째 트리에서 x1 데이터가 고립되지 않는다. x1데이터를 제외하고 나머지 데이터를 대상으로 속성 x_2를 선택하였다면, $x_2 - \min = 2$, $x_2 - \max = 3.5$ 사이에 임의의 값 $x_2 = 3$을 선택하는 경우에는 두 번째 트리에서 x5 데이터가 고립된다. 이 과정을 반복하면 3번째 트리에서 x2와 x3까지 모든 데이터가 분리되는 것을 알 수 있다.

다음에 두 번째 서브 데이터셋 (2,3,6,7)을 대상으로 x_1과 x_2속성 중에서 랜덤하게 속성 x_2를 선택하였고 상기 과정을 동일하게 수행한 결과는 다음과 같다(표 8-2).

이 문제에서 이상값이 x2=(2,2.5)이라고 가정한다면 첫 번째 트리에서 x2 말단노드에 해당하는 경로길이가 3이고, 두 번째 트리에서 경로길이는 2이므로 위 식에서 x2 이상값 점수는 다음과 같이 구해진다.

$$c(7) = 2H_5 - \frac{2(7-1)}{7} = 2(2.54) - \frac{2(6)}{7} = 3.18$$

$$H_6 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} = 2.45$$

$$E(h(x)) = \frac{1}{2}(3+2) = 2.5$$

$$S(x,7) = 2^{-\frac{E(h(x))}{c(7)}} = 2^{-\frac{2.5}{3.18}} = 0.58$$

이 문제에서는 데이터 x1이 이상값일 확률이 높다. 그런데 여기서는 데이터 x2가 이상값이라고 가정하고 2개의 트리만 구성하여 이상값 점수를 구하였다. 서브 데이터셋에 대해서 더 많은 트리를 작성하고 위의 과정을 반복하면 x1 데이터가 이상값으로 선택될 확률이 높아진다.

통상 Isolation Forest에서는 서브 데이터셋 크기가 $n = 256$ 정도면 모든 데이터를 다 커버할 수 있다고 하고 트리가 $t = 100$를 넘지 않아도 평균 경로길이

가 수렴이 잘된다고 알려져 있다.

[예제 4] 그림 7에 array(정렬) 형식으로 나타낸 데이터셋에 대해 비지도학습을 이용하여 이상값을 찾기 위한 Isolation Forest 알고리즘을 깃허브의

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

프로그램 수행결과, 그림 9에서 anomaly score가 0.5보다 높은 점수는 이상값을(별 표시), 0.5보다 낮은 점수는 정상값을(원 표시) 의미하며, 점수가 높을수록 이상값의 정도가 더 강하다는 것을 나타낸다. 따라서 이 데이터셋에서는 데이터 [180,152]의 이상값 점수가 0.6451로 가장 높기 때문에 데이터 [180,152]가 이상값일 확률이 가장 높다는 것을 알 수 있다.

그 다음으로는 데이터 [45,100]과 [110,52]도 이상값일 확률이 높다. 그림 9에서 이상값 점수가 높은 4개 데이터가 별 표시로 나타나고 있음을 알 수 있다.

한편 지도학습의 대표적인 알고리즘인 RandomForest를 이용하여 새로운 데이터들이 주어졌을 때 이 데이터가 정상 데이터인지 이상 데이터인지를 분류하기 위한 프로그램을 다음 깃허브 주소에 올려두었고

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

프로그램 수행 결과를 그림 10에 나타내었다.

그림 10에서 작은 원(정상 데이터)과 작은 별(이상 데이터)은 그림 9의 결과로부터 얻은 지도학습에 사용한 데이터이며 큰 원과 큰 별은 이상 유무를 평가하기 위해 사용한 네 개의 새 데이터이다. 새 데이터 중에서 큰 별로 표시된 [142,155] 데이터만이 이상 데이터이고 나머지 세 개 데이터는 정상 데이터인 것으로 분류되었다.

2.3 Principal Component Analysis

머신러닝에서 차원 축소(dimension reduction)란 많은 변수(또는 속성, feature)로 구성된 다차원 데이터셋의 차원을 축소해 새로운 차원의 데이터셋을 생성하는 것을 말한다. 일반적으로 차원이 증가할수록, 즉 속성이 많아질수록 예측 신뢰도가 떨어지고, 과적합이 발생하고, 개별 속성간의 상관관계가 높아질 가능성이 있다. 여기서 차원이란 속성의 개수를 의미한다.

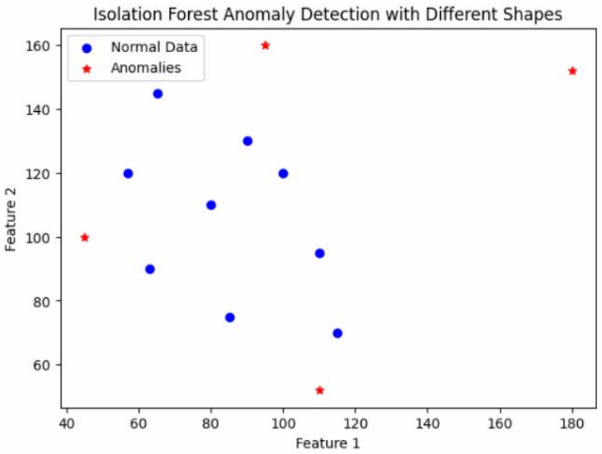


Fig. 9 Isolation forest algorithm result for anomaly detection

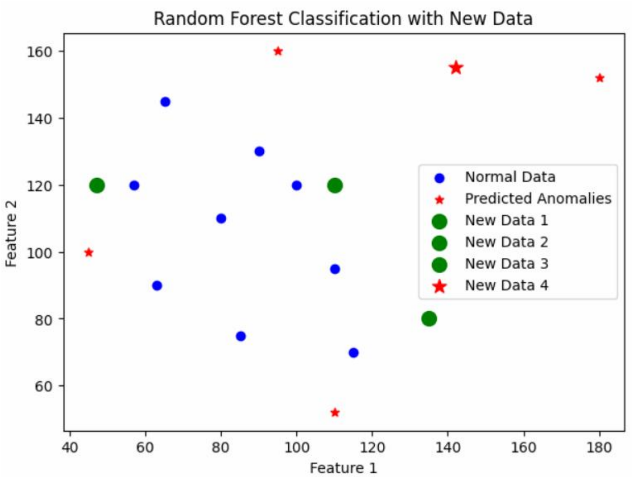


Fig. 10 Random forest algorithm result for newdata classification

new_data=np.array([[47,120],[110,120],[135,80],[142,155]])

물론 높은 차원의 데이터셋을 낮은 차원의 데이터셋으로 축소하여 시각화하면 일부 데이터 정보의 손실을 피할 수 없지만 데이터의 노이즈를 줄이면서 데이터의 주요 특징을 쉽게 파악할 수 있고 또한 메모리 절약으로 계산 효율성을 향상시킬 수 있는 장점이 있다. 이렇게 고차원에서 더 낮은 차원으로 데이터 차원을 축소하는 대표적인 방법으로는 주성분 분석(principal component analysis, PCA)과 선형판별 분석(Linear Discriminant Analysis, LDA)이 있다[9, 21-24].

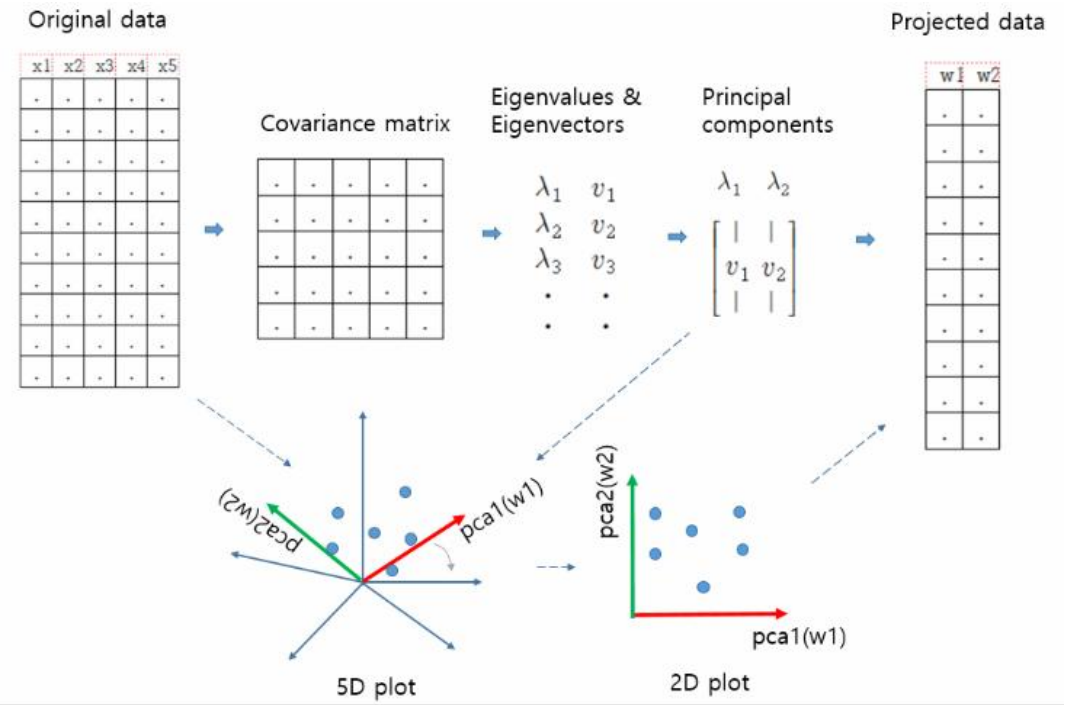


Fig. 11(a) Process of principal component analysis [22]

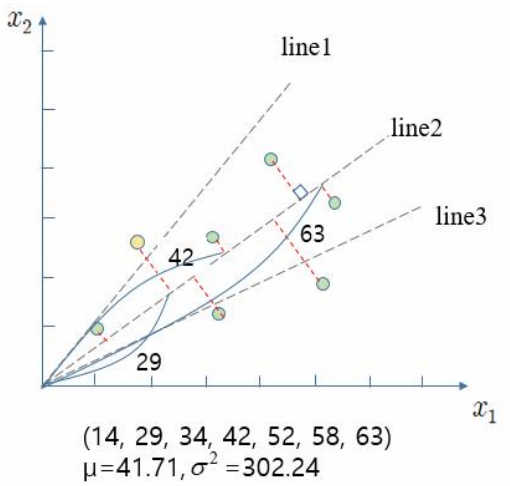


Fig. 11 (b) Select the axis with maximum variance

(예를 들면, 원점을 중심으로 line1, line2, line3 의 새로운 축을 잡고 데이터들을 이들 각 축에 정사영하였을 때 분산값이 최대인 축을 선택한다.

그림 11(b)에서 line2 축에 대한 투영 데이터는 (14,29,34,42,52,58,63)이고 이 경우에 분산값은 302.24 이다. Line3 축에 대한 투영 데이터는 (13, 21,34,40.53,54,62) 이고 분산값은 294.47 이다. line1 축에 대한 투영 데이터는 (15,31,31,41,47,58, 60) 이고 분산값은 259.96 이다.)

PCA는 고차원의 데이터를 저 차원의 데이터로 축소시키는 차원 축소 방법 중 하나이다. 머신러닝을 할 때 훈련 데이터의 속성이 많은 경우가 있는데 이 경우에 모든 속성이 결과에 중요한 영향을 끼치는 것은 아니다. 결과에 영향을 크게 미치는 속성도 있고 그렇지 않은 속성들도 존재한다.

여러 개의 속성 중에서 결과에 크게 영향을 미치는 주요한 속성 몇 개만을 선택하여 머신러닝 학습에 사용하려는 개념이 PCA 분석이다.

따라서 이 PCA 방법은 복잡한 데이터 세트의 패턴을 포착하고 모델링에서 발생할 수 있는 어려움을 줄이기 위해 예측 모델을 생성할 때 첫 번째 단계로 일반적으로 적용된다[23].

예를 들면 판재의 프레스 가공에서 금형 형상, 판재와 금형의 마찰조건, 프레스 속도, 금형 온도, 판재의 소성특성 등이 가공 성패에 미치는 영향을 검토하는 문제에서와 같이 5개의 속성(feature)이 있고 하나의 레이블(label)이 존재하는 문제를 생각해보자 (그림 11).

이 경우에 5개의 속성과 1개의 레이블을 매핑시키는 그래프를 그리려면 5차원의 그래프가 있어야 하지만 기하학적으로 3차원이 넘어가면 데이터의

시각화가 불가능하다. 따라서 이 경우에 중요한 2개의 속성만을 선택한다면 2차원의 그래프 또는 도형을 작성할 수 있다.

구체적으로 PCA는 여러 변수 간에 존재하는 상관관계를 이용해 이를 대표하는 주성분을 추출해 차원을 축소하는 기법으로, 데이터를 축에 정사영(orthogonal projection) 했을 때 가장 큰 분산을 갖는 데이터의 축(high variance axis)을 찾아 그 축으로 차원을 축소하는 것이다.

그림 11(b)에 표시한 데이터를 예를 들면, 원점을 중심으로 line1, line2, line3의 새로운 축을 잡고 데이터들을 이들 각 축에 정사영하였을 때 분산값이 최대인 축을 선택한다. 이 세 선 중에서는 line2가 가장 큰 분산을 갖지만 엄밀하게 최대 분산을 갖는 축은 아래에서 설명하는 것과 같이 공분산 행렬의 고유벡터 방향이 된다(그림 12(a)).

주성분분석에서는 새로운 축에서 분산이 최대가 될수록 원래 정보를 잘 반영한다.

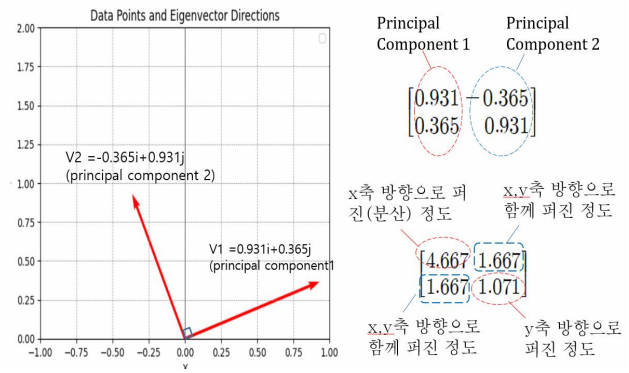
이 축(첫 번째 축이라고 함)을 주성분이라고 하고, 두 번째 축은 이 축에 직각이 되는 축으로 한다. 이 두 가지 주성분을 x, y 평면에 플롯하고 클래스 레이블별로 각 데이터를 구별하여 나타낸다. 그러면 유사한 항목이 함께 클러스터링된(clustering, 군집) 데이터 분포를 쉽게 파악할 수 있다(그림 12(b)).

한편 LDA는 클래스 간의 분산을 최대화하고 클래스 내의 분산을 최소화하면서 정보의 손실을 최소화하는 방향으로 차원을 축소한다.

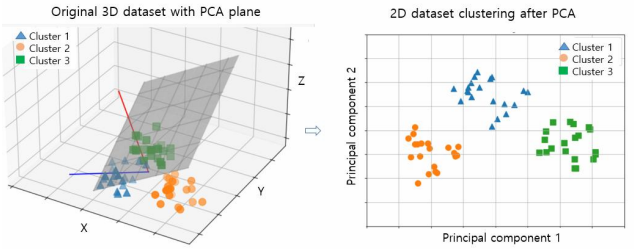
일반적으로 속성의 개수인 차원이 증가할수록 데이터 간의 거리가 기하급수적으로 증가하기 때문에 희소한 구조를 가지게 되고 모델의 예측 신뢰도가 떨어지게 된다. 따라서 차원 축소를 할 경우 학습 데이터의 크기가 줄어들어서 학습 속도가 빨라진다.

이 P/CA 알고리즘은 다음과 같은 과정을 거친다.

(1) 데이터 입력, (2) 평균 μ와 표준 편차 σ 계산, (3) 데이터를 $z=\frac{x-\mu}{\sigma}$로 표준화(z-score normalization), (4) 공분산 행렬(Covariance matrix)을 계산, (5) 공분산 행렬의 고유 벡터와 고유 값을 계산, (6) 고유 값을 내림차순으로 정리하고 고유 벡터를 추출하여 주성분을 선택, (7) 주성분을 이용해서 표준화된 데이터를 선형 변환하여 축소된 차원에서 새로운 데이터 셋 도출.



(a)



(b)

Fig. 12 (a) Mathematical meaning of eigenvector and eigenvalue obtained from eigenvalue decomposition of covariance matrix; (b) Application of PCA to the 3D dataset showing 2D dataset clustering

[https://github.com/yskim9574/DFclass-2023/blob/main /PCA_3Dto2D]

(주) 공분산과 피어슨 상관계수

정규분포하고 있는 두 확률변수 X 와 Y 의 분포로부터 크기 n 인 표본을 추출하여 얻은 관측치를 $(x_1,y_1),,,(x_n,y_n)$ 이라 하고, 평균값을 $(\bar{x},\bar{y})$라고 할 때 모집단에 대한 모상관계수 ρ 의 점추정치는 다음과 같이 나타내진다.

$$\rho=\frac{\sum_1^n(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_1^n(x_i-\bar{x})^2}\sqrt{\sum_1^n(y_i-\bar{y})^2}}=\frac{Cov(x,y)}{s_x s_y} \quad (6)$$

이 표본상관계수는 피어슨의 상관계수(Pearson's correlation coefficient)라고 불리기도 한다. 여기서

$$Cov(X,Y)=\frac{1}{n-1}\sum_1^n(x_i-\bar{x})(y_i-\bar{y})$$
$$s_x^2=\frac{1}{n-1}\sum_1^n(x_i-\bar{x})^2, s_y^2=\frac{1}{n-1}\sum_1^n(y_i-\bar{y})^2, \quad (7)$$

여기서 $Cov(X,Y)$ 는 두 확률변수 X 와 Y 의 분포가 결합확률분포를 이룰 때 해당 분포의 공분산(covariance)이다. 이 공분산은 두 변량 간에 연관성을 나타내는 척도이다. 또한 s_x 와 s_y 는 두 변수 X,Y 에 대한 표본 표준편차이다.

한편 공분산 행렬(covariance matrix)는 변수들 사이의 공분산을 행렬 형태로 나타낸 것이다. 공분산 행렬은 정방행렬(square matrix)이자 전치(transpose)를 시켰을 때 동일한 행렬이 나타나는 대칭행렬(symmetric matrix)인 특징이 있다. 또, 공분산 행렬은 두 변수가 각각의 평균으로부터 변화하는 방향 및 크기(분포)를 나타낸 분산이며, 행렬의 대각항들은 단일 변수의 퍼짐정도를 나타내는 분산을 의미한다.

$$Cov = \frac{1}{n-1} XX^T \quad (= A) \tag{8}$$

이 정방행렬인 공분산 행렬을 A 라고 하고 이를 고유값 분해(eigenvalue decomposition)하면 다음과 같이 표현된다.

$$A = Q\Lambda Q^T$$
$$Q = [V_1\ V_2\ V_3],\ \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{9}$$

여기서 $V_1, V_2, ..$ 은 정방행렬 A 의 고유값 $\lambda_1, \lambda_2, ...$ 에 대응하는 고유벡터로 열벡터이다. 각 열벡터는 표준정규분포 데이터를 얼마만큼 회전시켰는지에 대한 정보를 나타내며, 구체적으로는 주성분방향(principal component axis (PC), 간단히 주축)을 나타낸다.

통계 및 데이터 분석, 특히 주성분 분석에서 데이터 포인트 세트의 주축은 데이터가 가장 많이 변하는 방향을 의미한다. 또한 데이터들을 이 주축으로 정사영시키고 그 결과를 x축에 수평되게 회전시키면 차원이 축소된 새로운 데이터(projected data, 이 경우는 1차원 데이터)가 얻어지기 때문에 데이터의 특징을 파악하기 쉬워진다(n차원에서 2차원 또는 2차원에서 1차원으로 차원 축소).

한편 대각행렬 Λ 의 대각 성분들은 각 주성분 방향으로 얼마만큼 데이터가 분산되어 있는지를 나타낸다. 즉, 공분산 행렬의 고유값 분해는 표준정규분포의 데이터가 얼마나 분산되고 회전했는지를 벡터로 표현해준다.

공분산 행렬을 식 (8)에 적용하여 구해지는 행렬을 피어슨의 상관행렬(Pearson's correlation matrix) 라고 한다.

[예제 5] 예제 3에 나타낸 7개의 2차원 데이터를 1차원 데이터로 차원 축소하는 상세한 과정을 아래에 나타내었고 PCA알고리즘에 대한 파이썬 프로그램을 https://github.com/yskim9574/DFclass-2023/blob/main/Covariance _matrix1에 올려두었다.

$$X = \begin{bmatrix} 1 & 1 \\ 2 & 2.5 \\ 3 & 2 \\ 4 & 4 \\ 5 & 3.5 \\ 6 & 3 \\ 7 & 3.5 \end{bmatrix} (2D-data), \quad X = X - mean(X) = \begin{bmatrix} -3 & -1.78 \\ -2 & -0.28 \\ -1 & -0.78 \\ 0 & 1.22 \\ 1 & 0.72 \\ 2 & 0.22 \\ 3 & 0.72 \end{bmatrix}$$

$$Cov = \frac{1}{n-1} XX^T$$

$$= \frac{1}{7-1} \begin{bmatrix} -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -1.78 & -1.28 & -0.78 & 1.22 & 072 & 0.22 & 0.72 \end{bmatrix} \begin{bmatrix} -3 & -1.78 \\ -2 & -0.28 \\ -1 & -0.78 \\ 0 & 1.22 \\ 1 & 0.72 \\ 2 & 0.22 \\ 3 & 0.72 \end{bmatrix}$$

$$= \begin{bmatrix} 4.667 & 1.667 \\ 1.667 & 1.071 \end{bmatrix}$$

eigen value decomposition of covariance matirx :

$$\begin{bmatrix} 4.667 & 1.667 \\ 1.667 & 1.071 \end{bmatrix} = \begin{bmatrix} 0.931 & -0.365 \\ 0.365 & 0.931 \end{bmatrix} \begin{bmatrix} 5.32 & 0 \\ 0 & 0.4181 \end{bmatrix} \begin{bmatrix} 0.931 & 0.365 \\ -0.365 & 0.931 \end{bmatrix}$$

$$\rho = \frac{Cov}{s_x s_y} = \frac{\begin{bmatrix} 4.667 & 1.667 \\ 1.667 & 1.071 \end{bmatrix}}{2.16 \times 1.035} = \begin{bmatrix} 2.087 & 0.745 \\ 0.745 & 0.479 \end{bmatrix} \Rightarrow \begin{bmatrix} 1.0 & 0.745 \\ 0.745 & 1.0 \end{bmatrix}$$

$$XB = \begin{bmatrix} 1 & 1 \\ 2 & 2.5 \\ 3 & 2 \\ 4 & 4 \\ 5 & 3.5 \\ 6 & 3 \\ 7 & 3.5 \end{bmatrix} \begin{bmatrix} 0.931 \\ 0.365 \end{bmatrix} = \begin{bmatrix} 1.296 \\ 2.775 \\ 3.523 \\ 5.184 \\ 5.933 \\ 6.681 \\ 7.795 \end{bmatrix} \quad (1D-projected\ data)$$

프로그램 수행결과, 공분산 행렬의 고유값 분해로부터 그림 13에서와 같이 주성분(방향)을 알 수 있고 이 축에 2차원 데이터를 정사영한 데이터를 x축으로 회전시키면 그림 14와 같은 1차원 데이터가 얻어진다.

통상 머신러닝에서는 입력 데이터들을 z-score를 이용하여 표준화(평균이 영이고 편차가 1의 값을 갖도록)하는 것이 바람직하다. 이 경우에 대한 파이썬 프로그램은 깃허브에 올려두었다.

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

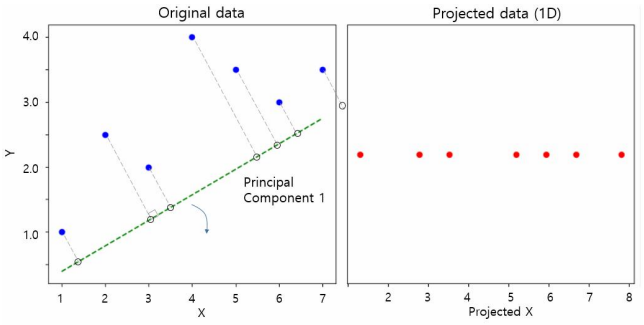


Fig. 13 Example of dimension reduction from PCA

data = np.array([(1,1), (2,2.5), (3,2), (4,4), (5,3.5), (6,3), (7,3.5)])

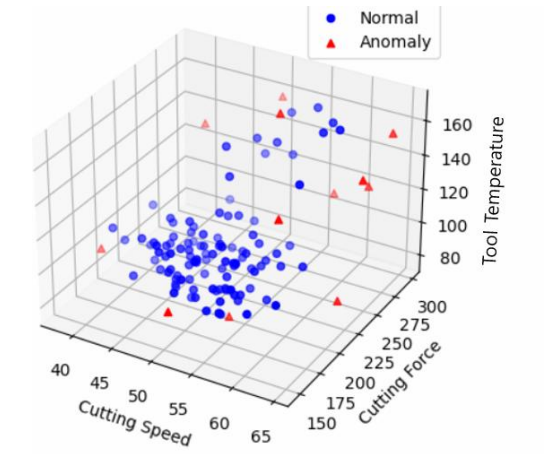
데이터를 표준화하는 경우에 제1 주성분만을 이용하고 1차원으로 축소된 데이터를 역 변환하여 원래의 데이터를 예측한 결과는 원래 데이터와 정확히 일치하지는 않지만 데이터의 특성을 잘 반영한다.

위에서는 정렬형식의 데이터를 이용하였다. 테이블 형식의 **.csv 파일을 pandas에서 불러들여 동일한 해석을 수행하는 경우에 대해서는 깃허브에 올린 다음 자료를 참고하라.

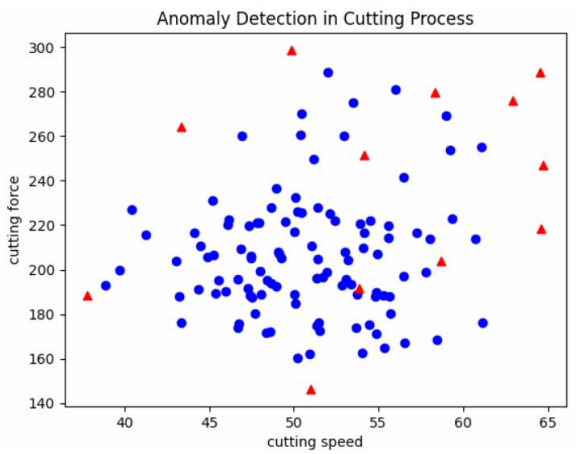
https://github.com/yskim9574/DFclass-2023/blob/main/Covariance_matrix2

동일한 고찰로부터 그림 11과 같이 변수가 $x_1, x_2, , , x_5$ 로 5개이고 각 변수에 대해서 12개의 데이터가 존재하는 크기 행렬의 데이터셋에 대해서도 크기의 공분산 행렬을 구하고, 이 공분산 행렬을 고유값 분해하여 고유값 $\lambda_1, \lambda_2, , , \lambda_5$ 를 구하고 대응하는 고유벡터 $V_1, V_2, , , V_5$ 를 구한다. 그런데 데이터셋의 특성을 가시적으로 파악하기 위해서는 3차원 또는 2차원 공간에 표시하여야 하므로 편의상 2차원 공간에 나타내기 위해서는 크기가 큰 순서대로(sorted) 2개의 고유값과 대응하는 고유벡터만을 이용한다. 5×2 크기 행렬의 이 고유벡터를 12×5 크기 행렬의 데이터셋에 작용시키면(행렬 곱) 변수가 w_1, w_2 인 12×2 크기 행렬이 얻어진다. 따라서 5차원의 데이터가 2차원의 데이터로 차원 축소가 이루어지기 때문에 2차원 평면상에 새로운 데이터셋을 가시화할 수 있고 데이터의 특징 파악이 용이해진다.

이 차원 축소방법을 신경망 방법에 연계 적용하여 비선형 과정인 소성가공 공정의 수치해석에 많은 시간이 소요되는 단점을 해결하기 위한 방법(dimension-reduced neural networks, DR-NNs)으로도 사



(a)



(b)

Fig. 14 (a) Original 3-dimentional dataset; (b) PCA reduced 2-dimentional dataset of lathe cutting problem

[https://github.com/yskim9574/DFclass-2023/blob/main/IsolationForest_cutting_process]

용되고 있다[25].

[예제 6] 다음은 공구속도, 공구력 그리고 공구온도를 변화시켜 절삭가공 실험을 수행하여 DataFrame (데이터프레임, df) 형식의 절삭 데이터를 대상으로 정상 가공이 이루어진 경우에는 1(원형 형상), 이상 가공이 발생한 경우에는 -1(삼각형 형상)로 표시하고 PCA와 Isolation Forest 알고리즘 적용한 예이다. 이 경우에 새로운 가공 데이터들(new_data)에 대해서 Isolation Forest 알고리즘을 적용하여 정상 가공인지 이상 가공인지를 구분하는 프로그램과 그 결과를 아래에 나타내었다(그림 14).

여기서는 미리 PCA 분석을 통해서 cutting_speed와 cutting_force가 순서대로 가장 큰 고유값을 갖는 것이 확인 되었기에 이들만을 주성분값으로 하는 2차원 공간에 나타내었다.

cutting_speed, cutting_force와 tool_ temperature의 3차원 공간에서는 이상 데이터들이 정상 데이터들의 외부에 존재하는 것이 확인 되지만 2차원 공간으로 축소하여 나타낸 경우는 이상 데이터들이 정상 데이터들의 내부에도 존재하는 것으로 나타날 수 있다.

이 문제에서 new_data = pd.DataFrame({
'cut_speed':[51.4,56.7,61.7,45.8,66.2],
'cut_force':[204.4,215.6,206.8,218.4,241.1],
'tool_temp':[98.1,101.3,99.7,112.3,134.3] })
를 적용하여 가공이 정상인지 이상인지를 평가한 결과 가공속도, 가공력, 공구온도가 각각 66.2, 241.1, 134.3인 경우만이 가공불량이 발생하는 것으로 예측되었다.

한편 머신러닝의 알고리즘들은 그 적용 결과에 따라 예측 정확도(accuracy), 정밀도(precision), 재현률(recall), F1-점수(F1-score) 등이 차이가 날 수 있다. 에를 들면 CNC 설비에 기본 내장된 센서들로부터 추출된 데이터에 다양한 머신러닝 알고리즘을 적용하여 실시간 CNC가공 완제품의 불량 여부를 예측한 연구[26]나 CNC 공구 마모도 예측 연구[27]의 예에서와 같이 문제에 따라 가장 효과적인 머신러닝 알고리즘을 선택하는 것이 바람직하다.

3. 결 론

본 해설논문에서는 최근 다양한 분야에서 널리 사용되고 있는 머신러닝 기술의 핵심 알고리즘의 원리를 알기 쉽게 설명하였다. 이들 알고리즘은 데이터 패턴을 수식화할 수 없는 데이터들의 통계적 처리(분류와 회귀)를 위해서 근년에 개발된 것으로 재료의 소성가공 분야 이외에도 다양한 사회현상과 문제해결을 위한 수단으로 적용이 점증하고 있다.

본 해설논문은 통계 이론적으로 수식화된 각 알고리즘에 근거하여 저자가 경북대학교 경영대학원에서 강의한 자료를 바탕으로 관련 연구자 및 산업현장 엔지니어들에게 각 알고리즘의 원리를 알기 쉽게 설명하여 머신러닝 기술을 산업현장의 문제해결에 잘 활용할 수 있도록 하기 위한 것이다.

이번 호에서는 Decision Tree, Isolation Forest/ Random Forest, Principal Component Analysis를 다루었고 다음 호에서는 AdaBoost, Gradient Boosting Machine, Support Vector Machine에 대해서 다룰 예정이다.

본 해설논문에서 참고한 자료[3, 4, 28]과 KOCW(Korea Open Course Ware) 공개강의 자료[29]에 AI와 머신러닝의 주요 알고리즘 그리고 통계에 대한 다양한 기초지식을 알기 쉽게 설명하고 있다. 관심이 있는 사람들에게 추천한다.

REFERENCES

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

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

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

[4] Michele di Nuzzo, 2021, Data Science and Machine Learning: From Data to Knowledge, ISBN-13979-8779849456

[5] H.B. Moon, et al., Development of Artificial Intelligence Constitutive Equation Model Using Deep Learning, Trans. Mater. Process., 30(4) (2021), 186-194, https://doi.org/10.5228/KSTP.2021.30.4.186

[6] T. Lehrer, et al., A simulation-based classification and regression approach for drawability assessment in deep drawing, Int. J. Mater. Form., 16(56) (2023), https://doi.org/10.1007/s12289-023-01770-3

[7] 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

[8] K.J. Lee, 2017, Statistical Machine Learning (Bigdata/R/Python), Kyowoo, ISBN 9791125101871

[9] S. Raschka,V. Mirjalili, 2021, Python Machine Learning 3rd Edition(H.S.Park, Korean Translation, Gilbut), ISBN9791165215187

[10] 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

[11] 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

[12] M. Mińkowski , L. Laurson, Predicting elastic and plastic properties of small iron polycrystals by machine learning, Scientific Reports, 13, (2023), 13977, https://doi.org/10.1038/s41598-023-40974-0

[13] Pham, Q. Tuan; Le, 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. Tecgnol., 120(1-4) (2022), DOI:10.1007/s00170-022-08698-z

[14] Ministry of SMEs and Startups of Korea, Precision processing resource optimization AI dataset, analysis practice guidebook. www.kamp-ai.kr
(a) https://www.kamp-ai.kr/mnt/dataset/374950177.pdf
(b) https://www.kamp-ai.kr/mnt/dataset/458329292.pdf
(c) https://www.kamp-ai.kr/aidataDetail?AI_SEARCH=&page=3&DATASET_SEQ=50&EQUIP_SEL=&GUBUN_SEL=&FILE_TYPE_SEL=&WDATE_SEL=

[15] A. Geitgey, https://www.machinelearningisfun.com/

[16] Andrew Ng https://www.coursera.org/learn/machine-learning

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

[18] 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.

[19] J.R. Quinlan, Simplifying decision trees, Int. J. Man-Machine Studies, 27 (1987), 221-234, https://doi.org/10.1016/S0020-7373(87)80053-6

[20] J.R. Quinlan, Introduction of Decision Trees, Machine Learning, 1 (1986), 81-106, http://dx.doi.org/10.1007/BF00116251

[21] L. Breiman, Random forests, Machine Learning, 45 (2001), 5-32, https://doi.org/10.1023/A:1010933404324

[22] https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d

[23] I.T. Jolliffe, 2002, Principal component analysis, 2nd edn. New York, NY: Springer-Verlag.

[24] I. T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Philil. Trans. R. Soc. A374 (2016), 20150202, https://doi.org/10.1098/rsta.2015.0202

[25] CKJ Hou, K. Behdinan, Neural networks with dimensionality reduction for predicting temperature change due to plastic deformation in a cold rolling simulation, Artificial Intelligence Engng. Design, Analy. Manuf., 37 (2023), 1-13, https://doi.org/10.1017/ S0890060422000233

[26] Y.H. Han, Prediction Model of CNC Processing Defects Using Machine Learnig, J. Korea Converg. Soc., 13(2) (2022), 249-255, DOI : https://doi.org/10.15207/JKCS.2022.13.02.249

[27] K.B. Lee, et al., A Study on the Prediction of CNC Tool Wear Using Machine Learning Technique, J. Korea Convergence Soc., 10(11) (2019) 15-21, https://doi.org/10.15207/JKCS.2019.10.11.015

[28] https://www.youtube.com/@statquest

[29] http://www.kocw.net/home/search/kemView.do?kemId=1481215&ar=relateCourse