

1.0 scikit-learn

- 파이썬 기반 쉽고 효율적인 머신러닝 라이브러리 제공
- 머신러닝을 위한 다양한 알고리즘 제공
- 텐서플로, 케라스, 파이토치, Mxnet등 딥러닝 전문 라이브러리에 밀려 대중적인 관심은 멀어지고 있지만, 여전히 많은 지지와 호응을 받고 있음

1.0 scikit-learn

scikit-learn algorithm cheat-sheet

1.0 scikit-learn

- 예제 데이터 : sklearn.datasets
- 변수 처리 : sklearn.preprocessing
 - sklearn.feature_selection
 - sklearn.feature_extraction
- 차원 축소 : sklearn.decomposition
- 데이터 분리/검증 : sklearn.model_selection
- 평가 : sklearn.metrics
- 머신러닝 알고리즘 : sklearn.ensemble
 - sklearn.linear_model
 - sklearn.naive_bayes
 - sklearn.svm / sklearn.tree
 - sklearn.cluster

1.0 scikit-learn 데이터 세트

- datasets.load_boston
- datasets.load_breast_cancer
- datasets.load_diabetes
- datasets.load_digits
- datasets.load_iris
- datasets.make_classifications
- datasets.make_blob

1.1 데이터에 대한 이해

■ 과학 기술의 발전 과정

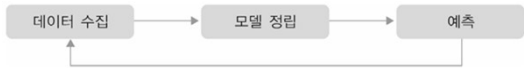


그림 1-8 과학기술의 발전 과정

- 예) 튀코 브라헤는 천동설이라는 틀린 모델을 선택함으로써 자신이 수집한 데이터를 설명하지 못함. 케플러는 지동설 모델을 도입하여 제1, 제2, 제 3법칙을 완성함

■ 기계 학습

- 기계 학습이 푸는 문제는 훨씬 복잡함
 - 예) [그림 1-2]의 '8' 숫자 패턴과 '단추' 패턴의 다양한 변화 양상
- 단순한 수학 공식으로 표현 불가능함
- 자동으로 모델을 찾아내는 과정이 필수

1.2 데이터 생성 과정

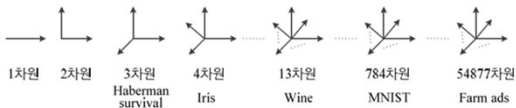
■ 데이터 생성 과정을 완전히 아는 인위적 상황의 예제

- 예) 두 개 주사위를 던져 나온 눈의 합을 x 라 할 때, $y=(x-7)^2+1$ 점을 받는 게임
- 이런 상황을 '데이터 생성 과정을 완전히 알고 있다'고 말함
 - x 를 알면 정확히 y 를 예측할 수 있음
 - 실제 주사위를 던져 $x = \{3,10,8,5\}$ 를 얻었다면, $y = \{17,10,2,5\}$
 - x 의 발생 확률 $P(x)$ 를 정확히 알 수 있음
 - $P(x)$ 를 알고 있으므로, 새로운 데이터 생성 가능

1.2 데이터 생성 과정

■ 실제 기계 학습 문제

- 데이터 생성 과정을 알 수 없음
- 단지 주어진 훈련집합 X, Y 로 예측 모델 또는 생성 모델을 근사 추정할 수 있을 뿐



Haberman survival: $x = (\text{나이}, \text{수술년도}, \text{양성 림프샘 개수})^T$
Iris: $x = (\text{꽃받침 길이}, \text{꽃받침 너비}, \text{꽃잎 길이}, \text{꽃잎 너비})^T$
Wine: $x = (\text{Alcohol}, \text{Malic acid}, \text{Ash}, \text{Alcalinity of ash}, \text{Magnesium}, \text{Total phenols}, \text{Flavanoids}, \text{Nonflavanoid phenols}, \text{Proanthocyanins}, \text{Color intensity}, \text{Hue}, \text{OD280 / OD315 of diluted wines}, \text{Proline})^T$
MNIST: $x = (\text{화소1}, \text{화소2}, \dots, \text{화소784})^T$
Farm ads: $x = (\text{단어1}, \text{단어2}, \dots, \text{단어54877})^T$
그림 1-6 다차원 특징 공간

1.3 데이터베이스의 중요성

■ 데이터베이스의 품질

- 주어진 응용에 맞는 충분히 다양한 데이터를 충분한 양만큼 수집 → 추정 정확도 높아짐
- 예) 정면 얼굴만 가진 데이터베이스로 학습하고 나면, 기운 얼굴은 매우 낮은 성능
→ 주어진 응용 환경을 자세히 살핀 다음 그에 맞는 데이터베이스 확보는 아주 중요함

■ 아주 많은 공개 데이터베이스

- 기계 학습의 초파리로 여겨지는 3가지 데이터베이스: Iris, MNIST, ImageNet
- 위키피디아에서 'list of datasets for machine learning research'로 검색
- UCI 리퍼지토리 (2017년11월 기준으로 394개 데이터베이스 제공)

1.3 데이터베이스의 중요성

• Iris 데이터베이스는 통계학자인 피셔 교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃(setosa, versicolor, virginica)을 50송이씩 채취하여 만들었다[Fisher1936]. 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정하여 기록하였다. 따라서 4차원 특징 공간이 형성되며 목표값은 3종류 숫자로 표시함으로써 1, 2, 3 값 중의 하나이다. <http://archive.ics.uci.edu/ml/datasets/Iris>에 접속하여 내려받을 수 있다.

Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
7.0	3.2	4.7	1.4	I. versicolor
6.4	3.2	4.5	1.5	I. versicolor
6.9	3.1	4.9	1.5	I. versicolor
5.5	2.3	4.0	1.3	I. versicolor
6.3	3.3	6.0	2.5	I. virginica
5.8	2.7	5.1	1.9	I. virginica
7.1	3.0	5.9	2.1	I. virginica
6.3	2.9	5.6	1.8	I. virginica



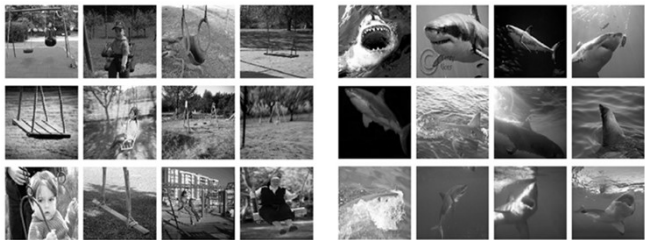
1.3 데이터베이스의 중요성

• MNIST 데이터베이스는 미국표준국(NIST)에서 수집한 필기 숫자 데이터베이스로, 훈련집합 60,000자, 테스트집합 10,000자를 제공한다. <http://yann.lecun.com/exdb/mnist>에 접속하면 무료로 내려받을 수 있으며, 1988년부터 시작한 인식률 경쟁 기록도 볼 수 있다. 2017년 8월 기준으로는 [Oresan2012] 논문이 0.23%의 오류율로 최고 자리를 차지하고 있다. 테스트집합에 있는 10,000개 샘플에서 단지 23개만 틀린 것이다.



1.3 데이터베이스의 중요성

• ImageNet 데이터베이스는 정보검색 분야에서 만든 WordNet의 단어 계층 분류를 그대로 따왔고, 부류마다 수백에서 수천 개의 영상을 수집하였다[Deng2009]. 총 21,841개 부류에 대해 총 14,197,122개의 영상을 보유하고 있다. 그중에서 1,000개 부류를 뽑아 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)라는 영상인식 경진대회를 2010년부터 매년 개최하고 있다. 대회 결과에 대한 자세한 내용은 4.4절을 참조하라. <http://image-net.org>에서 내려받을 수 있다.



(a) 'swing' 부류 (b) 'Great white shark' 부류

그림 4-20 ImageNet의 예제 영상

1.4 데이터베이스 크기와 기계 학습 성능

- 데이터베이스의 왜소한 크기
 - 예) MNIST: 28*28 흑백 비트맵이라면 서로 다른 총 샘플 수는 2^{784} 가지이지만, MNIST는 고작 6만 개 샘플

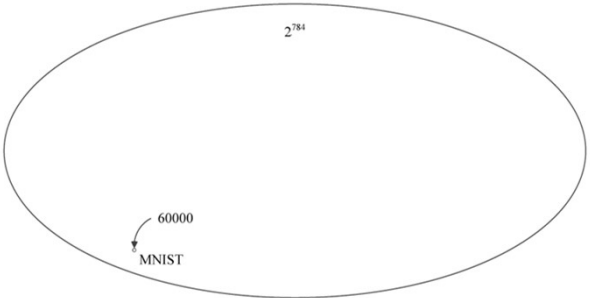




그림 1-9 방대한 특징 공간과 최소한 데이터베이스






1.4 데이터베이스 크기와 기계 학습 성능

■ 왜소한 데이터베이스로 어떻게 높은 성능을 달성하는가?

▪ 방대한 공간에서 실제 데이터가 발생하는 곳은 매우 작은 부분 공간임

•  ,  와 같은 샘플의 발생 확률은 거의 0

▪ 매니폴드 가정

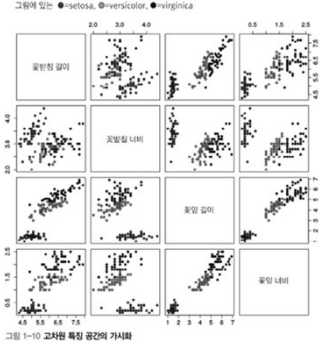
•      와 같이 일정한 규칙에 따라 매끄럽게 변화

1.5 데이터 가시화

■ 4차원 이상의 초공간은 한꺼번에 가시화 불가능

■ 여러 가지 가시화 기법

▪ 2개씩 조합하여 여러 개의 그래프 그림



▪ 고차원 공간을 저차원으로 변환하는 기법들

2.1 과소적합과 과잉적합

■ [그림 1.13]의 1차 모델은 과소적합

▪ 모델의 ‘용량이 작아’ 오차가 클 수밖에 없는 현상

■ 비선형 모델을 사용하는 대안

▪ [그림 1-13]의 2차, 3차, 4차, 12차는 다항식 곡선을 선택한 예

▪ 1차(선형)에 비해 오차가 크게 감소함

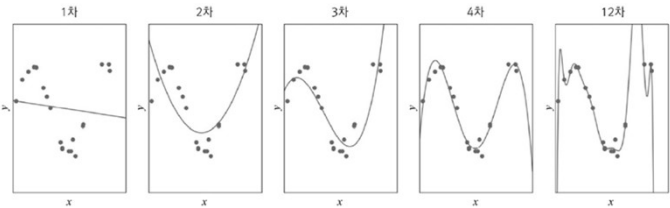


그림 1-13 과소적합과 과잉적합 현상

2.1 과소적합과 과잉적합

■ 과잉적합

▪ 12차 다항식 곡선을 채택한다면 훈련집합에 대해 거의 완벽하게 근사화함

▪ 하지만 ‘새로운’ 데이터를 예측한다면 큰 문제 발생

• x_0 에서 빨간 막대 근방을 예측해야 하지만 빨간 점을 예측

▪ 이유는 ‘용량이 크기’ 때문. 학습 과정에서 잡음까지 수용 → 과잉적합 현상

▪ 적절한 용량의 모델을 선택하는 모델 선택 작업이 필요함

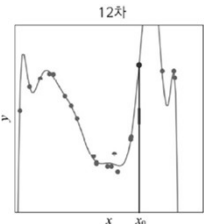


그림 1-14 과잉적합되었을 때 부정확한 예측 현상

4

2.2 편향과 분산

- 1차~12차 다항식 모델의 비교 관찰
 - 1~2차는 훈련집합과 테스트집합 모두 낮은 성능
 - 12차는 훈련집합에 높은 성능을 보이나 테스트집합에서는 낮은 성능 → 낮은 일반화 능력
 - 3~4차는 훈련집합에 대해 12차보다 낮겠지만 테스트집합에는 높은 성능 → 높은 일반화 능력

2.2 편향과 분산

- 훈련집합을 여러 번 수집하여 1차~12차에 적용하는 실험
 - 2차는 매번 큰 오차 → 편향이 큼. 하지만 비슷한 모델을 얻음 → 낮은 분산
 - 12차는 매번 작은 오차 → 편향이 작음. 하지만 크게 다른 모델을 얻음 → 높은 분산
 - 일반적으로 용량이 작은 모델은 편향은 크고 분산은 작음. 복잡한 모델은 편향은 작고 분산은 큼
 - 편향과 분산은 트레이드오프 관계

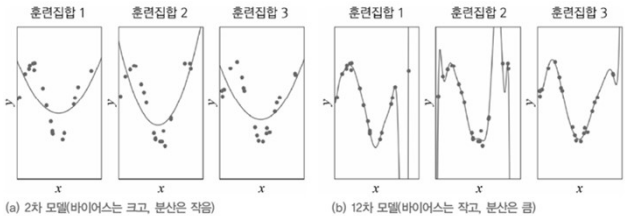


그림 1-15 모델의 바이어스와 분산 특성

2.2 편향과 분산

- 기계 학습의 목표
 - 낮은 편향과 낮은 분산을 가진 예측기 제작이 목표. 즉 왼쪽 아래 상황

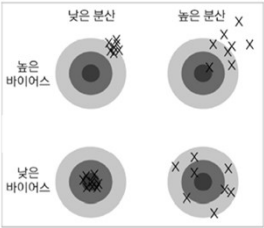
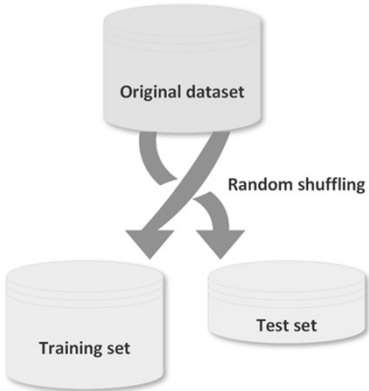


그림 1-16 바이어스와 분산

- 하지만 편향과 분산은 트레이드오프 관계
- 따라서 편향을 최소로 유지하며 분산을 최대한 낮추는 전략 필요

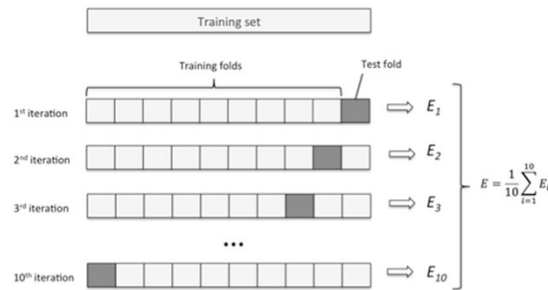
2.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

- 검증집합을 이용한 모델 선택
 - 훈련집합과 테스트집합과 다른 별도의 검증집합을 가진 상황



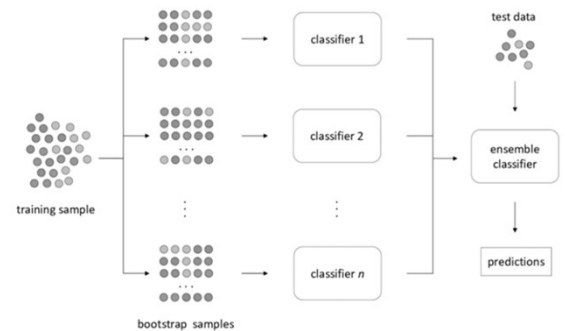
2.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

- 교차검증cross validation
 - 비용 문제로 별도의 검증집합이 없는 상황에 유용한 모델 선택 기법
 - 훈련집합을 등분하여, 학습과 평가 과정을 여러 번 반복한 후 평균 사용



2.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

- 부트스트랩boot strap
 - 난수를 이용한 샘플링 반복



2.4 모델 선택의 한계와 현실적인 해결책

- 현실에서는 아주 다양
 - 신경망, 강화 학습, 확률 그래픽 모델, SVM, 트리 분류기 등이 선택 대상
 - 신경망 - MLP, 깊은 MLP, CNN 등 아주 많음
- 현실에서는 경험으로 큰 틀 선택한 후
 - 모델 선택 알고리즘으로 세부 모델 선택하는 전략 사용
 - 예) CNN을 사용하기로 정한 후, 은닉층 개수, 활성화함수, 모멘텀 계수 등을 정하는데 모델 선택 알고리즘을 적용함

2.4 모델 선택의 한계와 현실적인 해결책

- 이런 경험적인 접근방법에 대한 『Deep Learning』 책의 비유

“To some extent, we are always trying to fit a square peg(the data generating process) into a round hole(our model family). 어느 정도 우리가 하는 일은 항상 둥근 홈(우리가 선택한 모델)에 네모 막대기(데이터 생성 과정)를 끼워 넣는 것이라고 말할 수 있다[Goodfellow2016(222쪽)].”
- 현대 기계 학습의 전략
 - 용량이 충분히 큰 모델을 선택 한 후, 선택한 모델이 정상을 벗어나지 않도록 여러 가지 규제(regularization) 기법을 적용함
 - 예) [그림 1-13]의 경우 12차 다항식을 선택한 후 적절히 규제를 적용

2.6 데이터 확대

■ 데이터를 더 많이 수집하면 일반화 능력이 향상됨

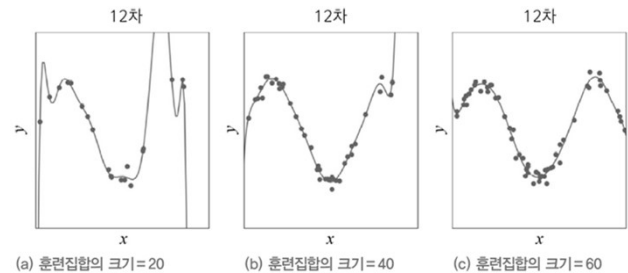


그림 1-17 데이터를 확대하여 일반화 능력을 향상함

2.6 데이터 확대

- 데이터 수집은 많은 비용이 듭
 - 실측 자료ground truth를 사람이 일일이 레이블링해야 함
- 인위적으로 데이터 확대
 - 훈련집합에 있는 샘플을 변형함
 - 약간 회전 또는 와핑 (부류 소속이 변하지 않게 주의)



그림 5-24 필기 숫자 데이터의 다양한 변형

2.7 가중치 감쇠

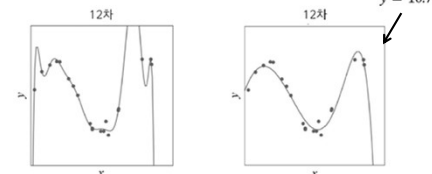
■ 가중치를 작게 조절하는 기법

- [그림 1-18(a)]의 12차 곡선은 가중치가 매우 큼

$$y = 1005.7x^{12} - 27774.4x^{11} + \dots - 22852612.5x^1 - 12.8$$

- 가중치 감쇠는 개선된 목적함수를 이용하여 가중치를 작게 조절하는 규제 기법
 - 식 (1.11)의 두 번째 항은 규제 항으로서 가중치 크기를 작게 유지해줌

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (f_{\theta}(x_i) - y_i)^2 + \lambda \|\theta\|_2^2 \tag{1.11}$$



(a) 가중치 감쇠 적용 안 함(식 (1.8)의 목적함수) (b) 가중치 감쇠 적용함(식 (1.11)의 목적함수)

그림 1-18 가중치 감쇠에 의한 규제 효과

2.8 모델 성능 평가

■ 혼동 행렬confusion matrix

- 모델의 성능을 평가할 때 주로 사용
- 분류문제를 학습한 모델을 평가할 때 사용

Confusion Matrix

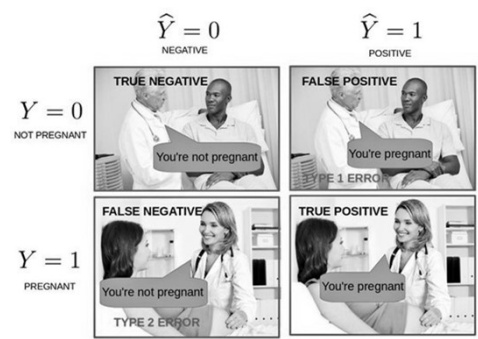
		Predicted	
		Positive	Negative
Actual	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

정밀도

민감도
특이도
정확도

2.8 모델 성능 평가

■ 혼동 행렬confusion matrix



2.8 모델 성능 평가

■ 정확도(Accuracy)

- 맞는 것을 맞다고, 틀린 것을 틀리다고 올바르게 예측한 것
- 혼동행렬의 대각선 부분

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

■ 정밀도(Precision)

- 모델의 예측 값이 얼마나 정확하게 맞는지 예측되었는지 알아봄
- 혼동행렬 1열 부분

$$Precision = \frac{TP}{TP + FP}$$

2.8 모델 성능 평가

■ 민감도(Sensitivity)

- 맞는 것 중 맞다고 예측된 것들의 비율 = 재현율
- 혼동행렬의 1행 부분

$$Recall = \frac{TP}{TP + FN}$$

■ 특이도(Specificity)

- 틀린 것 중 틀리다고 예측된 것들의 비율
- 혼동행렬의 2행 부분

$$Specificity = \frac{FP}{TN + FP}$$

2.8 모델 성능 평가

■ F1 Score

- 정밀도도 중요하고 재현율도 중요하지만 무엇이 중요한지 고민
- 이 두 값을 조화평균으로 계산한 결과값

$$F_1 = 2 \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

2.8 모델 성능 평가

■ ROC 곡선

- 수신자 판단 곡선 Receiver Operation Characteristic Curve
- 세계 2차 대전 통신 장비 성능 평가를 위해 고안된 수치
- 의학분야에 많이 사용되지만, 머신러닝의 이진 분류 모델 예측 성능 평가에도 사용
- x축은 특이도(음성 판단)로, y축은 민감도(양성 판단)로 설정한 관계를 그래프로 표현

