Machine Learning, Fall 2022 - Homework 2

■ 이름 (학번): 김우진 (201820772)

■ 학과: 소프트웨어학과

1 Short questions

1) Supervised learning과 Unsupervised learning 방식의 차이에 대해 설명하고 classification은 어느 쪽에 속하는지 쓰시오.

두 방식의 가장 큰 차이점은 training data에 각 data의 정답(개발자가 의도한 값)을 가리키는 label의 존재여부이다.

supervised learning은 label을 가진 data를 이용해 학습해, test data의 정답을 결정지을 수 있다. 하지만 unsupervised learning은 label이 없는 데이터를 학습해 test data의 정답을 구분지을 순 없지만, test data 사이의 structure을 파악할 수 있다는 장점이 있다.

2) 기계학습에서 데이터를 training/validation/test set 세 부분으로 나누어야 하는 이유에 대해 설명하시오 (즉, 각 데이터 부분의 용도)

training set은, 데이터 간의 패턴을 발견하는 등 내가 만들고자 하는 ML model을 학습시키기 위해 필요한 data이다.

validataion set은 training data에서 쪼개져 생성되며, 현재 training한 model을 평가하기 위해 사용되는 data이다. 즉 매 traing 과정에 관여하며, training된 여러가지 모델 중 가장 좋은 하나의 모델을 고르기 위해 필요한 data이다.

test set은 training->validation cycle을 거쳐 최종 생성된 model의 Accuracy 같은 최종 성능을 평가하기 위해 쓰인다.

3) 기계학습에서 overfitting(과적합)이 의미하는 바를 쓰시오.

training data를 너무 과도하게 학습하여, training data를 제외한 unseen 데이터에 대해 오차가 증가하는 현상을 말한다. 즉, training data에 대해선 꾸준하게 accuracy가 높지만, test(unseen) data에 대해 variance가 높은 것을 말한다. 주로 data가 적은 상태에서 model이 복잡하면 overfitting이 일어나기 쉽다.

2 2차원 상의 데이터 $x=(x_1,x_2)^T$ 를 고차원으로 매핑하는 다음의 함수를 가정하자.

1) 두 벡터 $\boldsymbol{u}=(\boldsymbol{u}_1,\boldsymbol{u}_2)^T$와 $\boldsymbol{v}=(\boldsymbol{v}_1,\boldsymbol{v}_2)^T$를 φ로 각각 매핑한 벡터의 내적에 해당하는 kernel function $\boldsymbol{k}(\boldsymbol{u},\boldsymbol{v})$를 구하시오.

$$\varphi(u)=\begin{pmatrix} u_1^2 \\ u_2^2 \\ \sqrt{2}u_1u_2 \end{pmatrix} \quad \varphi(v)=\begin{pmatrix} v_1^2 \\ v_2^2 \\ \sqrt{2}v_1v_2 \end{pmatrix}$$

$$\therefore \quad \mathrm{K}(u,v)=u_1^2v_1^2+u_2^2v_2^2+2u_1u_2v_1v_2$$

2) SVM에서 kernel trick이 무엇인지 그 장점의 측면에서 서술하시오.

SVM은 linear separation을 통해 각 class를 분류한다. 하지만 현실세계에선 각 class를 linear하게 구분 짓지 못하는 경우가 훨씬 많다. 이러한 비선형 문제를 해결하기 위해, 현재 데이터가 위치한 input space의 각 data를 더 높은 차원인 feature space로 옮기는 방법을 많이 쓴다. 왜냐하면 feature space에서의 선형분류는 input space에서 비선형 분류가 되기 때문에 비선형 문제를 해결할 수 있기 때문이다.

하지만 고차원으로 mapping 해주는 함수를 정의하는 것은 매우 어렵고, 함수를 정의한다 해도 문제를 해결하기 위한 연산량이 너무 많은 경우가 대다수다.

이 때 고차원 공간에 mapping 하지 않아도 input space의 값만을 이용해 값을 계산할 수 있는 kernel function을 사용한다. kernel trick은 kernel function을 이용해 값을 구하는 것을 말하며, kernel trick을 이용하면 고차원 변환과, 많은 연산량 문제를 해결하기 위해 mapping 함수를 알지 못해도 mapping 함수의 내적을 계산할 수 있다.

추가적으로, kernel function을 직접 정의해도 되지만 유명한 kernel function 몇가지를 후보로 올리고 그 중에서 성능이 좋은 것을 선택하면 된다.

3 Random Forest

1) Random forest의 주요 하이퍼마라미터 두 가지를 골라 그 의미를 각각 설명하시오.

1.1. forest를 구성하는 decision tree의 개수

-> forest 모델을 구성하는 의사결정나무의 개수를 의미. 많을수록 좋은 성능이 나올 수 있지만 꼭 그런거 아니다.

1.2. 각 tree를 분할할 때 사용할 data의 feature의 수

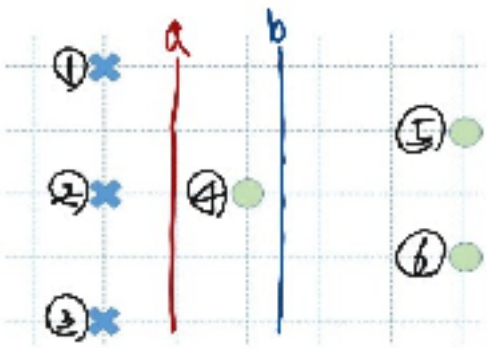
-> node를 split할 때, data의 feature를 전부 사용하는 것이 아닌 이 중 일부분만 랜덤으로 추출해 split하는데, 이 때 추출할 feature의 수.

2) Random forest가 기본 decision tree ensemble(bagged tree) 모델과 구별되는 가장 큰 특징을 한 가지 서술하고 그 특징으로 인한 장점을 설명하시오.

단순히 여러 개의 tree를 만들어 그 결과를 종합해 예측 성능을 높이는 bagging과 달리, random forest는 각 나무를 만들 때 분할에 사용되는 feature을 무작위로 선정해 학습시킨다.

왜냐하면 dataset에 다른 predictor들 보다 영향력이 훨씬 큰 predictor가 있을 시, tree를 여러개 만들어도 feature을 전부 사용하면 항상 root node로 이 predictor가 뽑혀 결과가 다 비슷비슷해지기 때문이다. feature을 random으로 추출하는 random forest의 경우 위와 같은 현상을 방지할 수 있다.

4 다음 2차원 평면 상의 6개의 데이터 샘플에 대해 분류 알고리즘을 적용하기로 하자. 각 샘플의 모양은 두 클래스(o, x) 중 어디에 속하는지를 나타낸다. KNN 알고리즘에서 흔히 사용되는 유클리드 거리를 사용한다고 가정하자.



1) 1-NN을 사용하는 경우 training error를 쓰고 이유를 간략히 설명하시오.

0이다. 자신을 포함한 dataset에서 이웃을 1명만 살펴보므로 항상 자신이 선택되기 때문이다.

2) 3-NN을 사용하는 경우 training error를 쓰고 이유를 간략히 설명하시오.

$\frac{1}{6}$ 이다. 6개의 데이터 중 ④을 제외한 나머지 데이터들은 모두 3-NN 적용 시 정답이기 때문이다.

3) 5-NN을 사용하는 경우 LOOCV error를 쓰고 이유를 간략히 설명하시오.

①~⑥ 데이터 각 1개씩을 validation date로 사용했을 시, ④를 썼을 때를 제외하면 모두 정답이다. 그러므로 총 Accuracy = $\frac{5}{6}$이므로, error = $\frac{1}{6}$

4) Linear SVM을 사용하는 경우 LOOCV error를 쓰고 그 이유를 생성되는 Decision boundary들을 그림에 표시하여 설명하시오.

①, ②, ③, ⑤, ⑥을 Validation data로 사용 시 Ⓐ가 형성되고

④를 validation data로 사용 시 Ⓑ가 형성된다.

Ⓐ -> ①, ②, ③, ⑤, ⑥ 5개의 데이터를 모두 올바르게 분류한다.

Ⓑ -> ④를 올바르게 분류하지 못한다.

∴ Accuracy = (5 + 0) % 6 = $\frac{5}{6}$

error = 1 − $\frac{5}{6}$ = $\frac{1}{6}$

5 0과 1의 두 가지 클래스로 이루어진 이진 분류(binary classification) 문제를 풀기 위한 모델을 학습하여 다섯 개의 테스트 샘플에 적용한 결과, 각 샘플별로 클래스 1에 속할 확률이 아래와 같이 구해졌다고 가정하자.

ID	P(y=1|x)	True Label
1	0.95	1
2	0.85	0
3	0.75	1
4	0.65	1
5	0.45	0
6	0.35	1
7	0.15	0

1) P(y=1|x)>0.7인 경우에 클래스 1로, 아닌 경우 0으로 예측하기로 할 경우의 Sensitivity, Specificity를 구하시오(단, 정답 클래스는 위 테이블의 'True Label' 열에 주어져 있다).

①, ②, ③번 데이터는 positive로, 나머지 데이터는 negative로 예측한다.

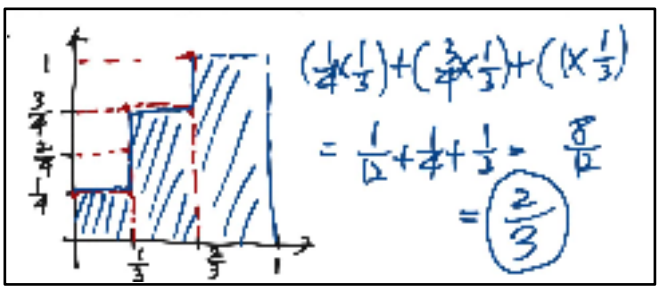
그러면 TP = 2, FN = 2, FP = 1, TN = 2이다.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} = \frac{2}{2+2} = \frac{1}{2}$$

$$\text{Specificty} = \frac{\text{FP}}{\text{FP+TN}} = \frac{2}{1+2} = \frac{2}{3}$$

2) 위 표를 기반으로, 클래스 1로 예측하는 기준이 되는 threshold (예를 들면 1)번에서는 0.7)를 바꾸어 가면서 반복적으로 Sensitivity, Specificity를 구할 경우 ROC curve를 그릴 수 있다. 주어진 결과에 해당하는 ROC curve 상의 (0, 0)과 (1, 1)을 제외한 주요 좌표 6개를 적고 AUC(Area under the ROC curve) 값을 구하시오.

AUC 그래프 및 값:



주요 좌표 6개: $(0, \frac{1}{4})$, $(\frac{1}{3}, \frac{1}{4})$, $(\frac{1}{3}, \frac{1}{2})$, $(\frac{1}{3}, \frac{3}{4})$, $(\frac{2}{3}, \frac{3}{4})$, $(\frac{2}{3}, 1)$

(ID 1에서부터 시작되는, Threshold: 1, 2, 3, 4, 5, 6 순서대로 기술한 것임)

3) (True or False) ROC curve가 (0,1)점을 지나는 경우 그 분류기는 ROC curve를 만드는 데 사용된 테스트 데이터들을 항상 완벽하게 분류한다 (즉, error = 0).

AUC = 1이 되므로 True이다.