PART.IV 모델평가

1. Bias - variance tradeoff

모델의 예측력을 평가하는데 있어서 error 를 이해하는 것은 필수적이다. 모델의 error 를 추정하기 위해 가장 흔히 쓰이는 척도는 MSE(평균제곱오차)인데, MSE 는 세 파트의 error 로 구성되어있다. bias 와 variance, noise 이다.

$$MSE(\theta) = Var(\theta) \ + \ Bias(\theta)^2$$

Noise는 자연적으로 발생하는 error이기 때문에 논외로 하고, MSE를 줄이기 위해서는 bias와 variance를 줄이는 것이 필수적이라 할 수 있다. 그렇다면 bias와 variance는 무엇일까?

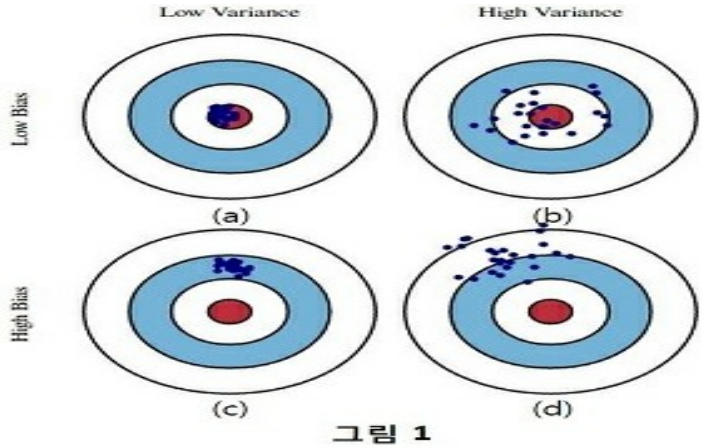
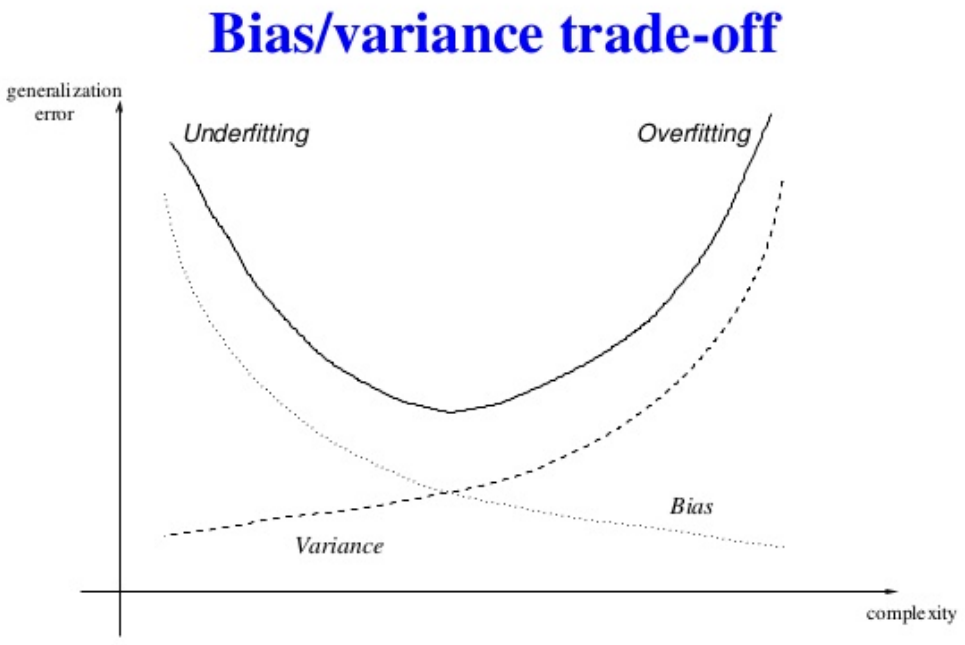


그림 1

붉은 점을 target, 즉 참값이라 하고 푸른 점을 모델로부터 얻은 추정값이라 가정하자.
Bias : 참값과 추정값의 차이를 의미한다. 추정값이 참값과 근접할 수록 좋은 추정값이 된다.(low bias)

Variance : 추정값의 산포도, 즉 퍼진 정도를 의미한다.

위의 그림에서 보여주는 것 처럼, low bias & low variance를 가진 추정값이 좋은 추정값이 된다. 하지만, bias와 variance를 동시에 낮추는 것은 불가능하다. Bias와 variance사이에 tradeoff 관계가 존재하고 있기 때문이다.



위의 그림은 x축을 모델 복잡도, y축을 일반화 에러(새로운 데이터를 넣었을 때 그 참값으로부터의 에러)로 하는 그래프이다.

위의 그림에서 보여주는 것은 모델이 단순할수록 bias는 크지만 작은 variance를 가지고 있고, 모델이 복잡해질 수록 bias는 줄어들지만 variance는 커지는 것을 확인할 수 있다. 즉, bias와 variance를 동시에 줄이는 모델링은 불가능하지만, generalization error, 즉 MSE를 최소화하는 지점을 찾을 수 있다. 즉, 모델은 bias를 충분히 줄일 수 있을 만큼 복잡하여야 하지만, variance가 너무 크지 않도록 단순해야한다.

하지만 어떤 척도로 모델의 일반화 오류에 접근할 수 있을까?

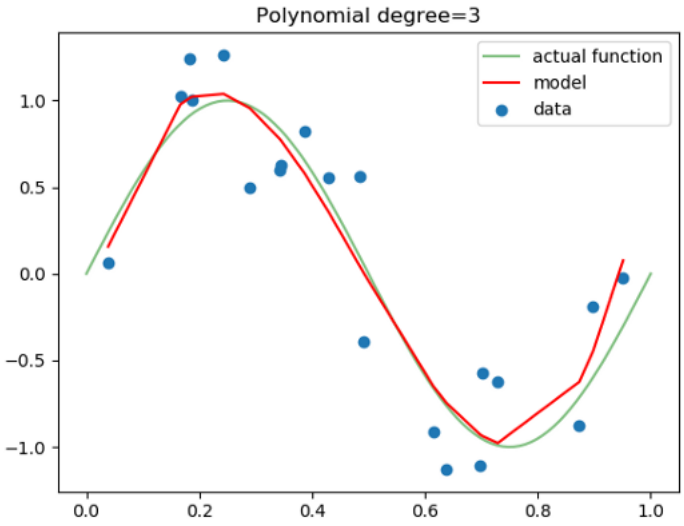
2. Underfitting & Overfitting

Underfitting은 모델이 큰 bias를 가지고 있지만, 작은 variance를 가지고 있는 상태를 의미한다. 즉 데이터를 잘 설명하기 위해 좀 더 복잡한 모델이 필요한 상태이다.

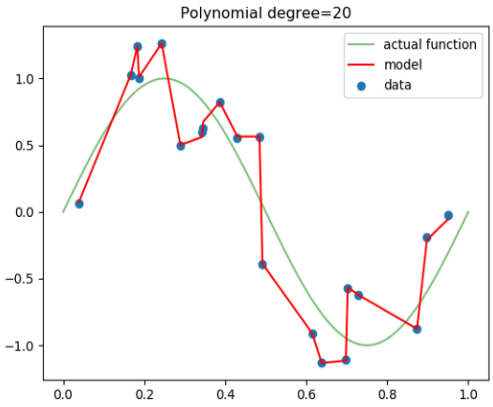
Overfitting은 모델 자체가 가진 bias는 작지만 큰 variance를 가지고 있는 상태이다. 즉 데이터를 너무 잘 설명하기 위해 일반화가능성을 희생한 모델을 의미한다.

구체적인 예시로 overfitting 과 underfitting 의 사례를 살펴보도록 하자.

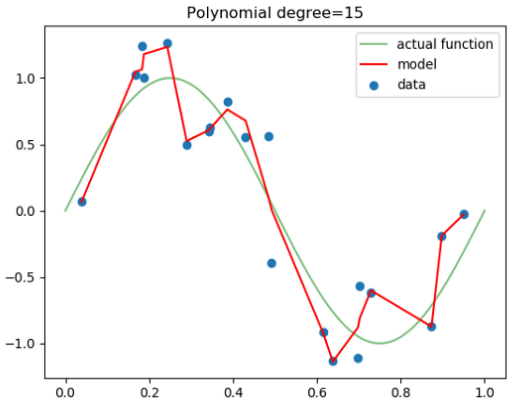
< Desirable Fit >



현재 보여지는 그래프는 3차 함수에 정규난수를 더해준 값을 데이터로 하여 모델을 적합한 결과이다. 정규난수를 더하지 않은 실제 그래프는 녹색선으로 표시되어있고, 데이터는 푸른 색 점으로 표시되어있다. 또한 빨간 색 선은 Cubic regression 모델을 통해 예측한 값을 나타낸다. Cubic regression을 통해 예측한 값은 실제 함수의 그래프와 유사하고, 점들과의 수직거리 또한 큰 차이를 보이지 않고 가장 괜찮은 fit을 보여주고 있다.

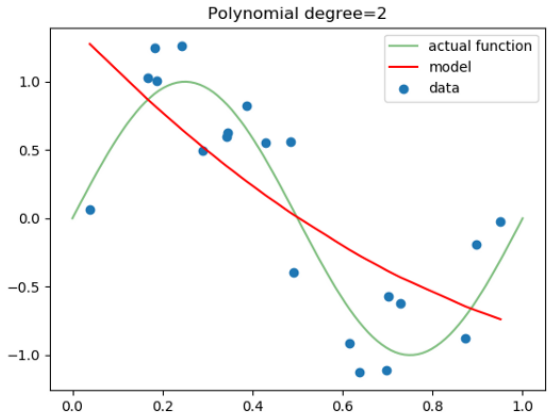


< Overfitting >



3차 함수로 설명할 때 가장 적합한 데이터를 15차 함수를 사용해서 산출한 예측값이 붉은 선으로 그려져있다. 붉은 선과 푸른 점의 수직거리가 작지만(Low bias), 실제 삼차함수보다 복잡한 형태를 가진다.(High variance) 위와 같은 상황에서 생기는 문제점은 새로운 데이터에 대한 일반화가 어렵다는 점이다. 좀 더 극단적인 예를 통해 살펴보자.

붉은 선은 똑같은 데이터를 20차 함수를 사용하여 산출한 예측값이다. 주어진 데이터는 완벽하게 설명하고 있지만, 새로운 데이터에 대해 일반화가 어려운 모델이다. 모델 자체가 주어진 데이터를 설명하기 위해 너무 복잡하게 설정되어 있으므로, 새로운 데이터가 들어갔을때 3차 함수로 적합한 모델(단순한 모델)보다 Overestimate나 Underestimate를 하는 경우가 훨씬 더 많이 생긴다는 점을 짐작할 수 있다.



< Underfitting >

앞서 설명한 똑같은 조건을 quardratic regression을 통해 적합한 예측값이 붉은 색 선으로 나타나있다. 데이터를 설명하기 위해 단순한 모델을 사용하였기 때문에 데이터들과의 수직 거리가 많이 벌어진 상황(High bias)을 보여주고 있다. Data를 설명하기 위해선 좀 더 복잡한 모델이 필요하기 때문에 이러한 예시를 일컬어 "Underfitting"이라고 부른다.

위의 예를 통해 알아본 것처럼 Overfittng이나 Underfitting 된 모델에 미지의 데이터를 넣었을 때 나오는 예측값은 믿을만한 값이 아님을 알 수 있다. 머신러닝의 목적이 미지의 데이터를 일반화하는 것임을 주목할 때, Overfitting이나 Underfitting된 모델은 좋은 모델이라고 평가할 수 없다.

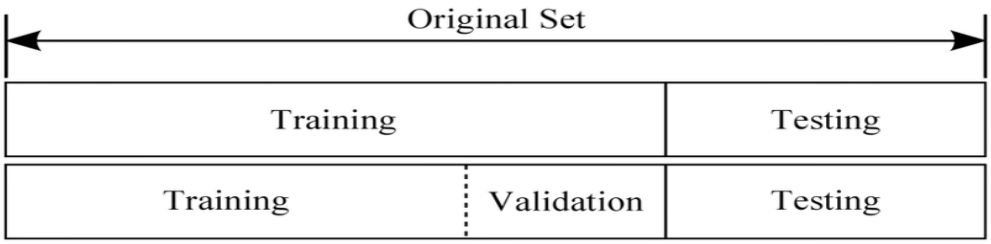
3. Cross - validation

Overfitting을 감지하고 적절한 모델을 찾아감에 있어서 가장 단순한 형태는 cross - validation이다. Cross - validation의 아이디어는 단순하다. 데이터셋을 train set과 test set(hold out sample)으로 나누어서 모델을 학습할 때는 train set만을 사용하고, 학습된 모델을 통해 test set에서 모델의 예측값을 살펴보는 것이다.

여기서 중요한 점은 test set과 train set의 완전한 분리이다. 즉, 모델이 학습할 때는 train set만을 활용하여야 하고, test set에 대한 정보는 완전히 모르고 있어야 한다는 것이다.

그렇다면 어떤 방식으로 train set과 test set을 나눌까?

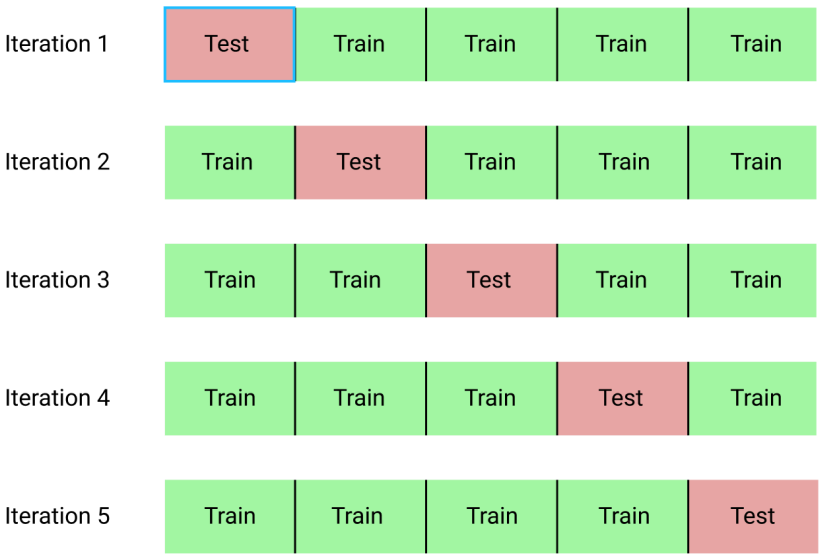
(a) 단순데이터분리



위의 첫 번째처럼 가지고 있는 데이터셋 중 랜덤하게 70-80% 정도를 train set에 할당하고 나머지 20-30% 의 데이터를 test set에 할당하는 방법이다. 만약 가지고 있는 데이터가 충분하다면, 50%의 데이터만 학습에 사용하고(train set) 30% 정도의 데이터를 통해 학습된 모델이 일반화가 가능한지 확인하는 추가적인 데이터셋인 validation set을 활용할 수 있다. 마지막 남은 데이터(test set)로는 일반화가 가능한지 최종적으로 점검해 볼 수 있을 것이다.

(b) k-fold validation

- k-fold validation에서는 단순데이터분리에서 처럼 train set과 test set을 분리한다.



- train set을 k개의 sub-set로 분리한다.
- 1st iteration : 첫 번째 sub-set을 validation에 활용하고 나머지 sub-set을 모델학습에 활용하고 error를 기록한다.
- 2nd iteration : 두 번째 sub-set을 validation에 활용하고 나머지 sub-set을 모델학습에 활용하고 error를 기록한다.
- (반복)
- k th iteration : k 번째 sub-set을 validation에 활용하고 나머지 sub-set을 모델학습에 활용하고 error를 기록한다.
- 총 k개의 error가 기록되어 있는데, error의 mean을 활용하여 모델을 튜닝하며 가장 적은 error를 기록한 모델을 찾는다.
- 해당 모델을 이용하여 hold out sample인 test set으로 최종 모델을 평가하고, evaluation metric을 이용하여 모델 성능을 기록한다. 만약 만족스럽지 않다면 바로 전 단계로 돌아가 모델을 튜닝한다.

위의 그림은 5-fold validation의 예시이다.

단순 데이터분리와 다르게 모든 데이터를 validation으로 활용할 수 있다는 점에서 원본 데이터가 작은 경우 자주 사용하게 되는 model validation method 이다. 또한 overfitting의 염려도 단순데이터 분리보다 덜하다. 하지만 단순 데이터분리보다 오랜 시간이 걸리고 k개의 데이터셋을 분리할때 sampling기법을 활용하여야 좀 더 섬세한 validation이 가능하다.

4. Evaluation metric

모델에 따라서 서로 상이한 평가 metric을 이용한다. 모델별로 가장 자주 활용되는 몇 가지 metric을 소개하고자 한다.

(a) Classification

Classification에서는 거의 모든 평가 metric이 Confusion matrix를 활용한다.

Confusion matrix는 위와 같이 참값과 model이 예측한 값으로 구성된 table이다.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

참값이 positive일때 모델이 positive라고 판단한 결과값의 갯수가 a, 참값이 positive지만 모델이 negative라고 오분류한 경우의 갯수가 c, 참값이 negative일때 model이 positive라고 오분류한 경우의 갯수가 b, 참값이 negative일때 모델이 negative라고 판단한 결과값의 갯수가 d로 표시되어 있다.

Confusion matrix를 통해 다양한 metric에 접근할 수 있다.

Accuracy : 정분류 비율로 classifier에서 가장 많이 쓰이는 평가 metric이다. - (a+d)/(a+b+c+d)

Positive predictive value : 예측값이 positive일때 정분류 비율 - a/(a+b)

Negative predictive value : 예측값이 negative일때 정분류 비율 - d/(c+d)

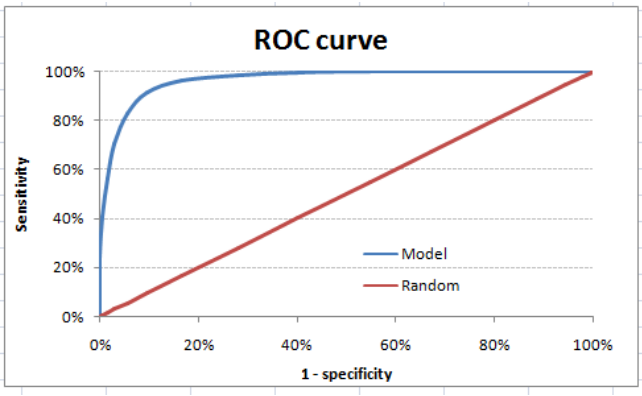
Sensitivity : 참값이 positive일때 정분류 비율 - a/(a+c)

Specificity : 참값이 negative일때 정분류 비율 - d/(b+d)

앞서 소개한 것처럼, Classifier에서는 Accuracy를 가장 흔히 사용한다.

Binary classifier인 경우, 가능한 cutoff에 대한 sensitivity와 specificity를 활용한 metric도 존재한다.

ROC curve



ROC curve는 가능한 cutoff point에서의 y축을 sensitivity, x축을 (1-specificity)로 하는 그래프이다.

위의 그림처럼 파란색 선처럼 y축과 그래프 상단에 근접할 수록 좋은 모델이 좋은 예측력을 가지고 있다고 판단하고, 빨간색 선은 random guessing으로 빨간색선과 비슷하다면 모델의 예측력이 random guessing과 다를 것이 없다고 판단한다.

AUC

AUC는 Area Under Curve의 약자로, ROC curve아래의 영역을 나타낸다. 이 영역은 0-1의 값을 가지며, 모델의 예측력으로 해석이 된다. 완벽한 예측을 할때 ROC Curve는 y축과 그래프 상단에 달라 붙은 형상을 보일 것이며, 가로 세로 길이를 100%, 즉 1로 가지는 정사각형의 넓이가 될 것이다. 즉 완벽한 예측을 할 경우 AUC는 1이 된다.

(b) Regression

SSE

Regression에서의 evaluation metric으로 SSE(error sum of square)를 활용할 수 있다. SSE는 $\sum(y_i - y_{i(pred)})^2$ 으로 얻어진다. 즉 참값과 예측값의 수직거리의 제곱의 합으로 얻어지는 것이다. SSE를 총 관측치의 갯수로 나누어 준 것, 즉 SSE/N은 MSE로 일컬어진다. 즉 작은 MSE 를 산출하는 모델이 좋은 모델이 되는 것이다.

RMSE

$$. MSE = \frac{SSE}{number\ of\ obs}, and\ RMSE\ is\ given\ by\ \sqrt{MSE}$$

마찬가지로, 작은 RMSE값은 좋은 모델을 의미한다. 하지만 Outlier에 민감하므로 유의하여야한다. RMSE는 train set과 test set간의 performance를 비교하기 위해 많이 사용된다. train set과 test set에서의 RMSE가 비슷하다면, train set과 test set이 비슷한 수준의 performance를 보인다고 해석할 수 있고, 이 값들이 충분히 작다면 Optimal model을 찾았다고 판단할 수 있다. 하지만 train set에서 RMSE값이 충분히 작은데 test set에서의 RMSE값이 크다면 Overfitting이나 outlier의 존재를 의심해야 한다.

데이터의 Scale에 dependent한 metric이라는 단점이 있다.

R^2

R^2(1-SSE/SST, where SST is total sum of square, given by $\sum(y_i - y^-)^2$)도 regression model 의 performance 를 평가하는 척도로 사용될 수 있다. 데이터의 Scale 에 관계없이 0~1 의 값을 뱉어내는 장점이 있다. Prediction 이 지나치게 크게 벗어날 경우 R^2가 음수가 나올 수 있다. 변수의 개수를 고려하여 adjusted-R2 score 를 사용하기도 한다.

사진출처 및 참조

https://towardsdatascience.com/regularization-the-path-to-bias-variance-trade-off-b7a7088b4577

https://angel.co/projects/465697-machine-learning-bias-and-variance-tradeoff

https://my.oschina.net/Bettyty/blog/751627

https://discuss.analyticsvidhya.com/t/explain-what-is-confusion-matrix-and-how-it-is-used-to-validate-classification-models/8396

https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/

http://it.plusblog.co.kr/221238399644

https://techdifferences.com/difference-between-classification-and-clustering.html

Introduction to Machine Learning, E.Alpaydin