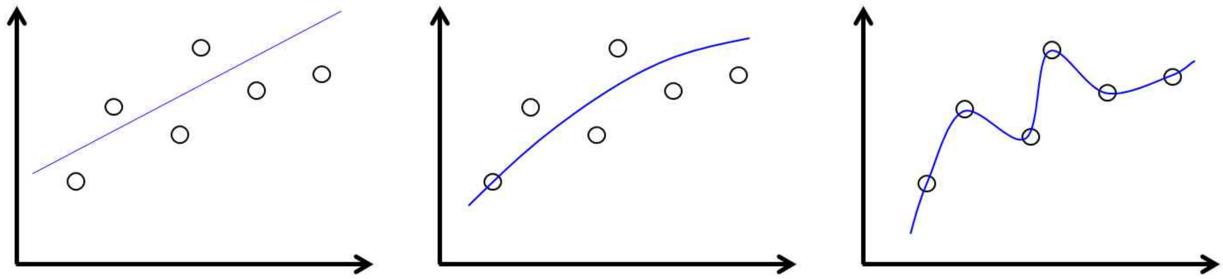
Machine Learning Lecture Note

This lecture note is based on the CS229 lecture materials from Standford University.

Lecture 7 Bias/Variance, Regularization, Train

Bias : expected error created by using a model
Variance : amount that predicted values would change in a different training dataset.

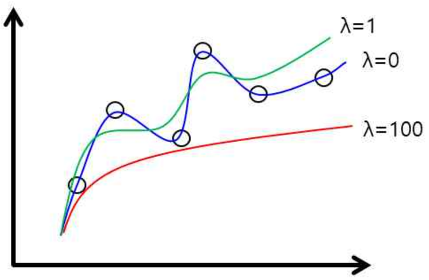


$\theta_0+\theta_1 x$ underfit high bias | $\theta_0+\theta_1 x+\theta_2 x^2$ right | $\theta_0+\theta_1 x+\theta_2 x^2+\cdots+\theta_5 x^5$ overfit high variance

Regularization

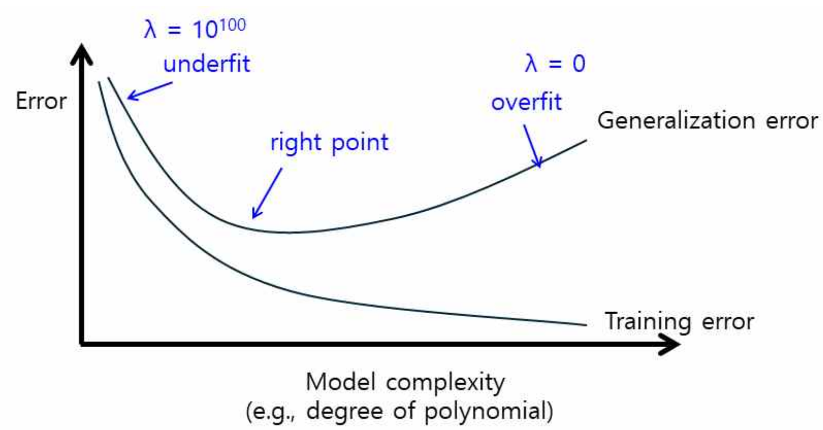
$$\min_\theta \frac{1}{2}\sum_{i=1}^{m} \| y^i-\theta^T x^i \|^2+\frac{1}{2}\lambda \| \theta \|^2$$

$\frac{1}{2}\lambda \| \theta \|^2$: regularization term, 오버피팅되는 모델을 조정한다. λ가 매우 큰 값을 가지면 오버피팅된 모델은 언더피팅의 효과를 가진다.



General form in logistic regression

$$\underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} \log p(y^i|x^i;\theta)-\lambda \| \theta \|^2$$



Train

Data sets: $S=\{S_{train}, S_{dev}, S_{test}\}$

1) Train each model (option for the degree of polynomial) on S_{train}.
   Get some hypothesis h_i
2) Measure the error on S_{dev}.
   Pick a model with the lowest error on S_{dev}.
   ※ S_{train}으로 학습하지 않는 이유: 복잡한 알고리즘은 항상 훈련 데이터 셋에서 더 좋은 성능을 보이기 때문에 overfit 발생 가능성이 높다.
3) Evaluate an algorithm on S_{test}.

Take training set

- historical perspective: 70% train, 30% test; 60% train, 20% dev, 20% test
  대용량 데이터를 가지지 않았을 때 주로 사용하는 방법.
- modern perspective
  대용량 데이터를 학습에 사용하는 경우 알고리즘간 성능을 파악하기 위해서 dev, test 데이터 셋의 비율을 줄인다. (e.g., 90% train, 5% dev, 5% test)

Small data set

m = 100 examples (70 test, 30 dev)

k-fold cross validation:

$S=\{(x^1,y^1),\cdots,(x^{100},y^{100})\}$ k=5 for illustration; k=10 is typical

divide the data set to 5 subset. so there are 20 examples in each subset.

```
For d=1, ..., 5 (degree of polynomial) {   <- k=5 인 예제에서 다음과 같이 수행할 수 있다.
   For i = 1, ..., k,
        Train (fit parameters) on k-1 pieces
        Test on the remaining 1 piece
   Average
}
```

Feature selection

많은 feature가 있는 경우, overfitting을 줄이는 한가지 방법은 task에 가장 유용한 feature의 작은 subset을 찾는 것이다.

Start with $F=\phi$

Repeat:

1) Try adding each feature i to F, and see which single feature addition most improves the dev set performance.

2) Add that feature to F

(example) 5 features: $x_1, x_2, \cdots, x_5$

step 1: $[\phi]$ $\quad F=\{\ \}$

step 2: $\begin{bmatrix}\phi+x_1\\ \phi+x_2\\ \vdots\\ \phi+x_5\end{bmatrix}$ $\quad F=\{x_2\}$

step 3: $\begin{bmatrix}x_2+x_1\\ x_2+x_3\\ \vdots\\ x_2+x_5\end{bmatrix}$ $\quad F=\{x_2, x_4\}$