

Deep Into Deep

김성찬

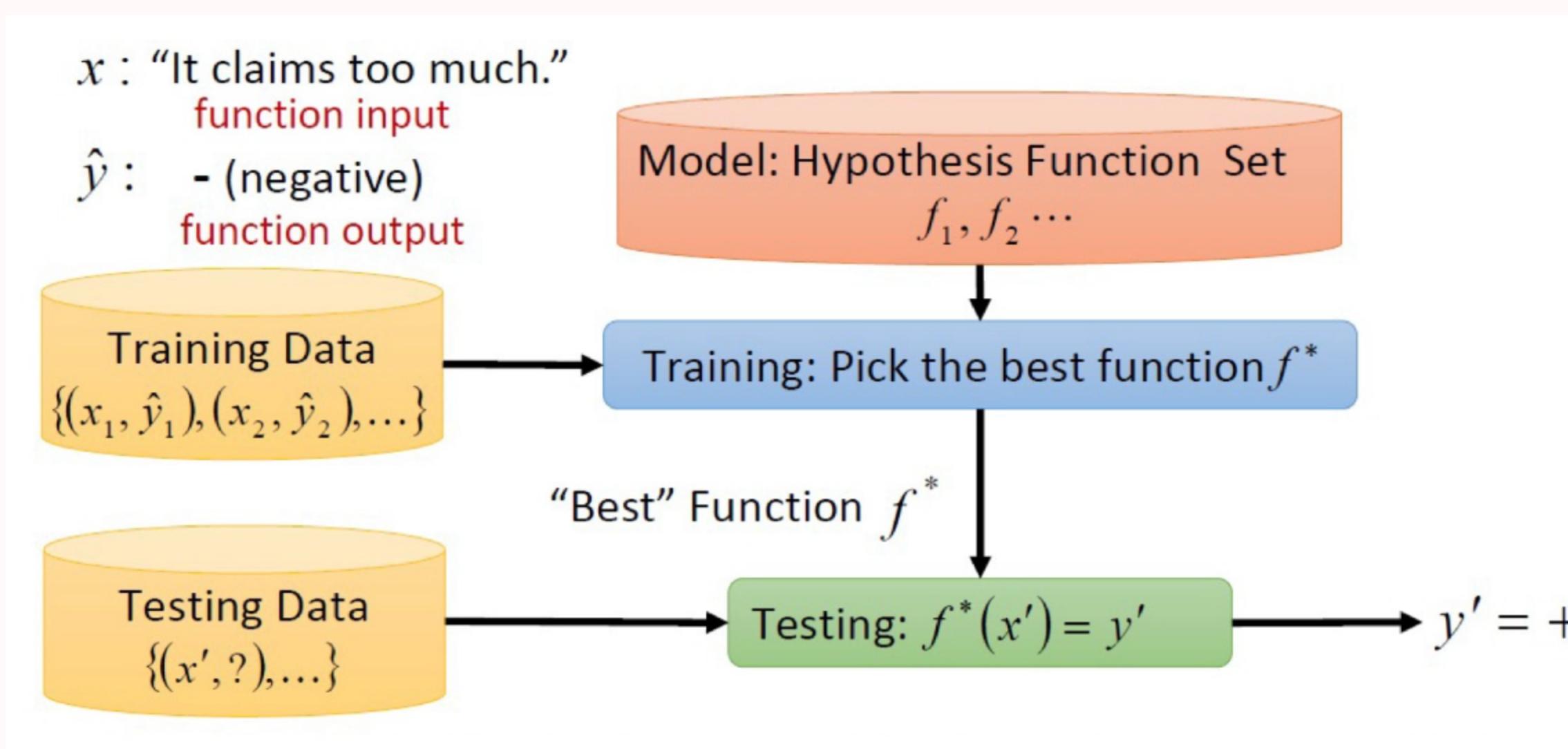
Contents

1. Machine Learning
2. Underfitting and Overfitting
3. Improving Generalization Ability
4. Evaluation and Model Selection
5. Bias-Variance Trade-Off

1. Machine Learning

Training : 주어진 데이터를 가장 잘 근사하는 함수를 찾는 과정

Testing : 학습한 함수를 통해 데이터를 예측하는 과정



2. Underfitting and Overfitting

모델의 Capacity가 너무 제한적일 수도, 너무 풍요로울 수도 있다. too limited or too rich.

너무 제한적인 경우, Underfitting이 발생할 수 있다.

> 데이터를 잘 근사하는 함수를 찾을 수 없다.

너무 풍요로운 경우, Overfitting이 발생할 수 있다.

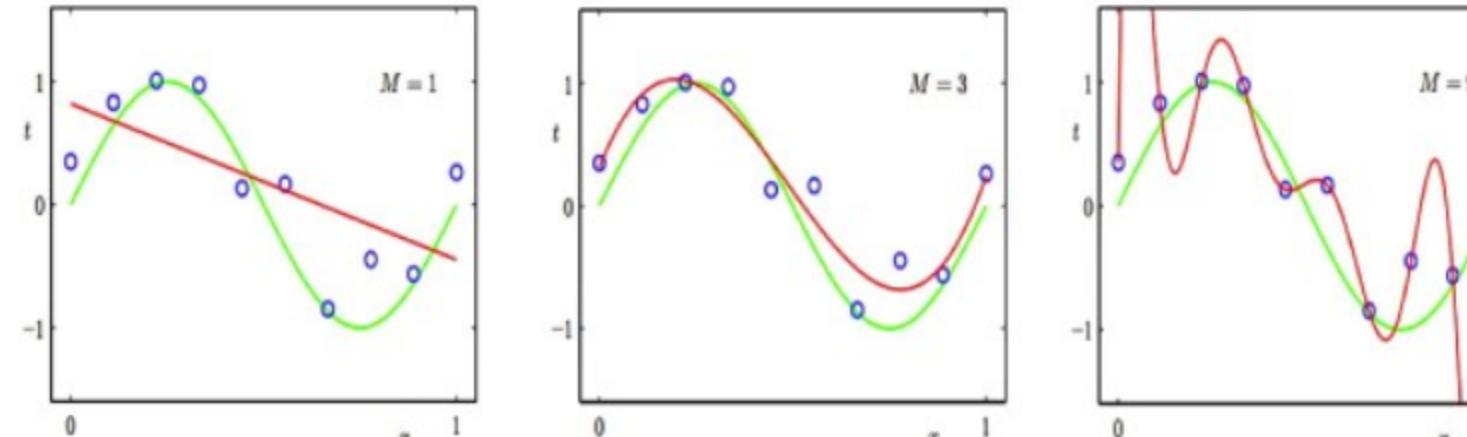
> 우리의 본질적인 목적은 데이터를 통해 기저 함수를 찾는 것이다. 이때 데이터를 통해 기저 함수를 찾을 수 있다는 가정이 있다. 하지만 데이터 속에는 노이즈, 즉 기저 함수를 찾는 데에 방해가 되는 요소 또한 존재한다. 따라서 이러한 노이즈까지 학습하는 경우, 좋은 함수를 얻기 어렵다.

위 두 현상 모두 바람직한 현상이 아니다.

2. Underfitting and Overfitting

Under- and Over-fitting examples

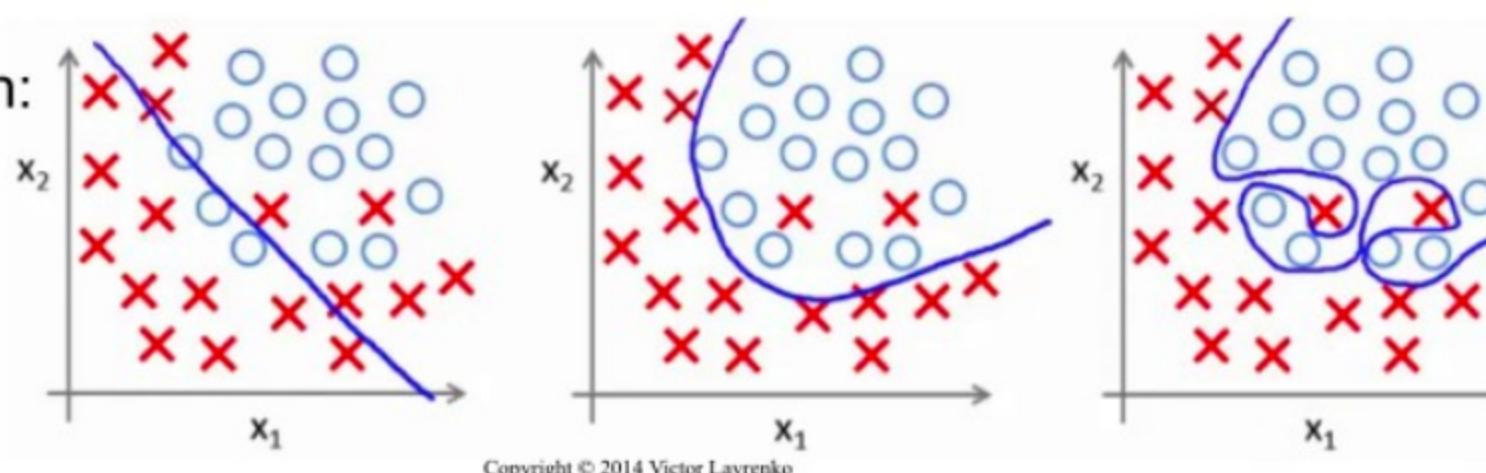
Regression:



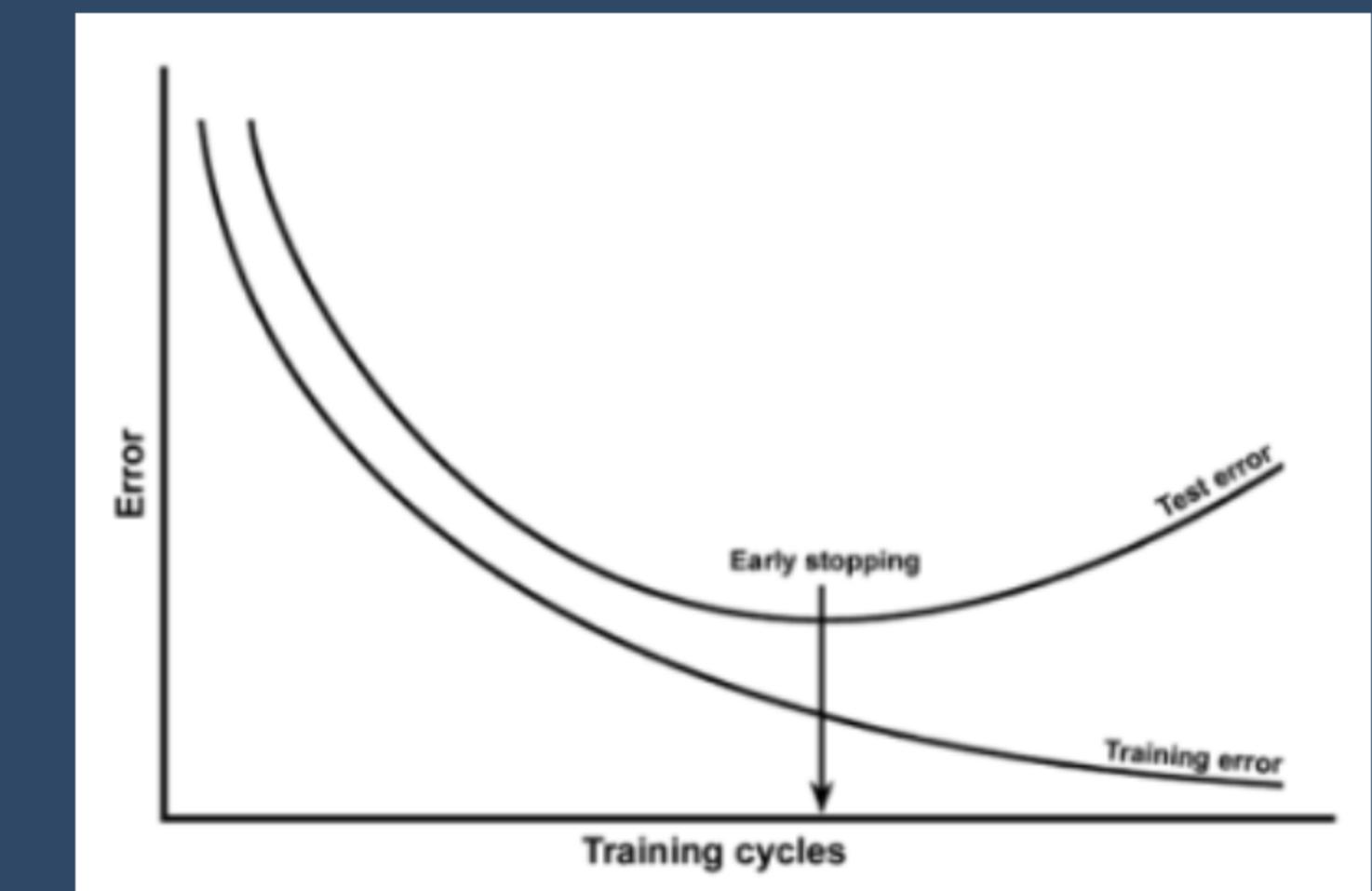
predictor too inflexible:
cannot capture pattern

predictor too flexible:
fits noise in the data

Classification:



Copyright © 2014 Victor Lavrenko



학습 에러와 테스트 에러

2. Underfitting and Overfitting

Underfitting 해결하기

1. 더 오래 학습시키기

Overfitting 해결하기

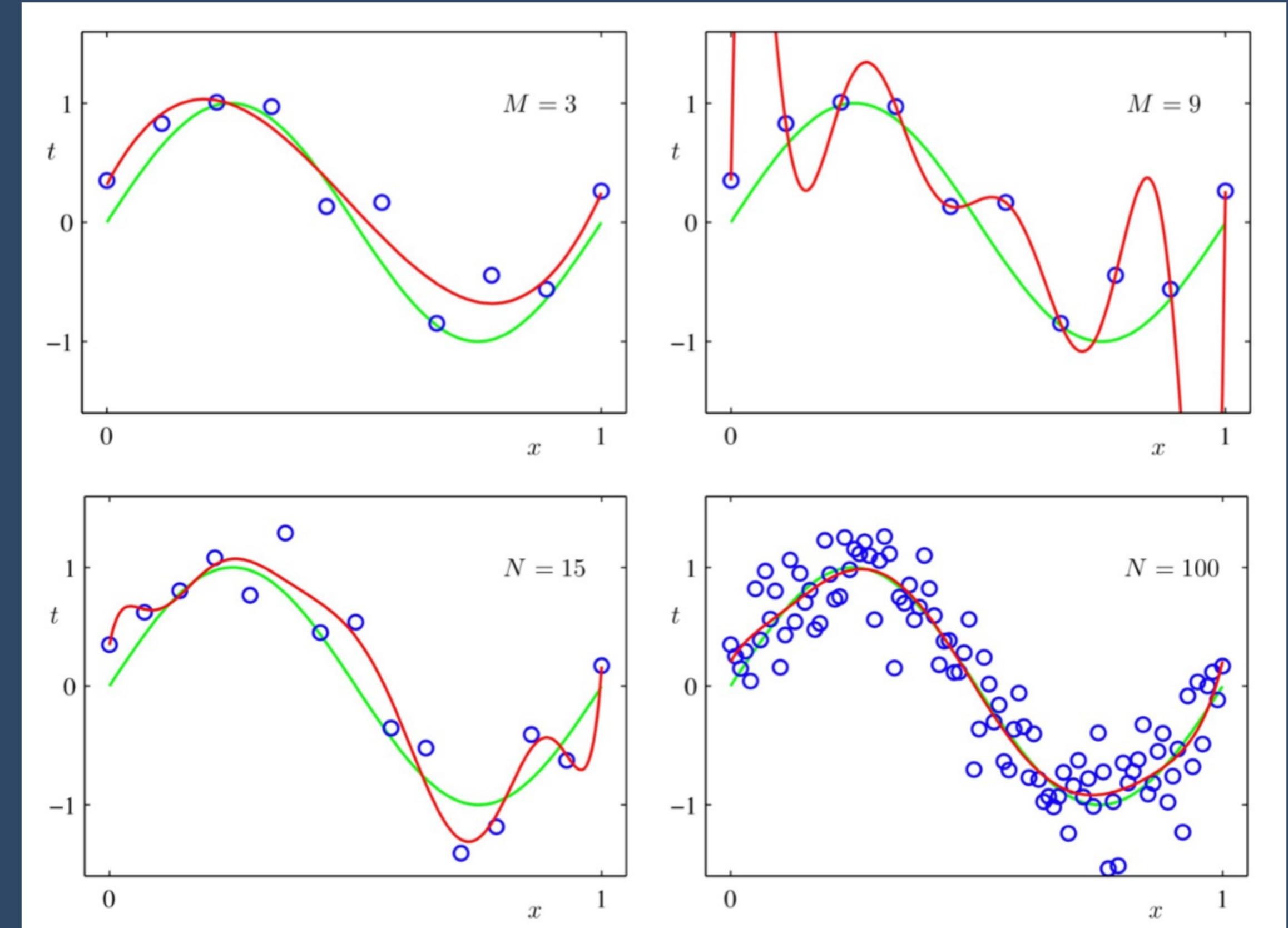
1. 더 많은 학습 데이터로 학습하기
2. 더 적은 차원의 데이터 사용하기
 - feature를 선택한다.
 - 모델 선택 알고리즘
3. Regularization
 - 파라미터 수를 줄인다.
 - 파라미터의 수치적인 크기를 줄인다.

2. Underfitting and Overfitting

더 많은 데이터와 더 적은 특성
(N 은 샘플의 수, M 은 특성의 수)

특성이 적을수록 모델이 단순해지고,
모델이 적절하게 단순해질수록
더 일반화가 잘되지만,
너무 단순한 모델은
제대로 판단을 내리지 못한다.

데이터가 많을수록
모델에 제약이 많아지고
제약에 의해 모델이 단순해진다.
특성이 많은 모델이라도
데이터가 많으면
모델이 적절하게 단순해질 수 있다.



2. Underfitting and Overfitting

Occam's Razor, "Simple models are better!"

오컴의 면도날 이론의 핵심은 이렇다.

어떤 현상을 설명하기 위한 이론이 복잡한 경우보다 단순한 경우가 해당 현상에 대해 더 잘 설명하는 이론이다.

하지만 단순함의 정도를 파악하고 결정하는 것은 굉장히 어려운 문제이다.

단순함 -> 일반화

복잡함 -> 구체화

명확한 판단을 위해서는 구체적인 근거가 필요하지만, 더 좋은 판단을 위해서는 일반적인 근거가 필요하다.

2. Underfitting and Overfitting

Occam's Razor, "Simple models are better!" -> Regularization

$$\arg \min_W \left\{ \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2 + \lambda \sum_i W_i^2 \right\}$$

w <- w - (Loss + regularization)

파라미터가 작은 값이 되도록 유도하면 더 Simple mapping function이 만들어지고 Overfitting 과적합을 막을 수 있다.

2. Underfitting and Overfitting

Regularization

$$\arg \min_W (L(W) + \Omega(W))$$

첫 번째 항은 손실 함수이고, 두 번째 항은 규제화하는 함수이다.

규제화의 주된 영향은 큰 모델 파라미터를 penalize하여 작게 만드는 것이다.

2. Underfitting and Overfitting

L-2 Regularization

$$\Omega(W) = \lambda \|W\|_2^2 = \sum_i W_i^2$$

lambda는 hyper parameter로 여러 값을 실험해보면서 최적의 값을 찾아야 한다.

lambda의 값이 큰 경우 vs. lambda의 값이 작은 경우

2. Underfitting and Overfitting

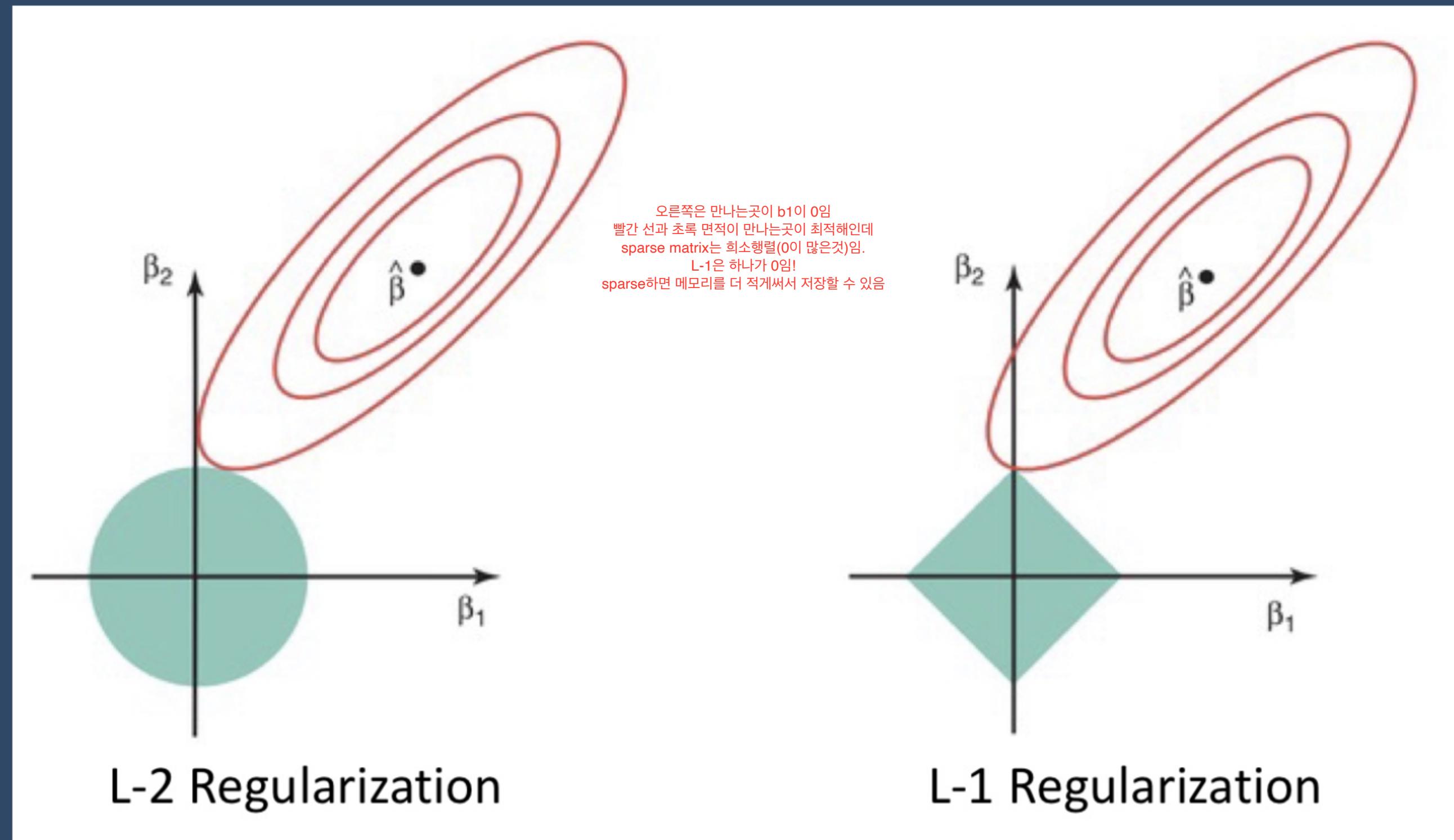
L-1 Regularization

$$\Omega(W) = \lambda \|W\|_1 = \sum_i |W_i|$$

L-1 Regularization의 최적해는 L-2 Regularization과 비교할 때 sparse한 경우가 많다.

2. Underfitting and Overfitting

L-2 vs. L-1



3. Improving Generalization Ability

모델의 성능이 좋지 않다면

1. 더 많은 학습 데이터를 구한다.

2. 특성의 수를 조절한다.

- 불필요한 특성이 있는 경우, 학습에 좋지 않으므로 제거한다.
- 유의미한 특성을 추가한다.
- polynomial features와 같은 고차원의 특성을 추가한다.

3. 하이퍼파라미터를 조정한다.

- lambda가 너무 큰 경우, 제대로 학습이 이루어지지 않으므로 lambda의 값을 줄인다.
- lambda가 너무 작은 경우, 과대적합이 발생할 수 있으므로 lambda의 값을 키운다.

4. Evaluation and Model Selection

Evaluating a Mapping Function

모델의 학습과 평가에는 모두 데이터가 필요하다.

이때 학습과 평가에 같은 데이터를 사용하는 것은 교과서에 있는 문제를 시험에 출제하는 것과 같다.
교과서를 제대로 봤다면 다 맞을 수 있는 것이다.

따라서 학습을 위한 데이터와 평가를 위한 데이터를 각각 만드는 것이 좋다.

$$D_{train} = \{(x_1, y_1), (x_2, y_2), \dots, (x_{N_{train}}, y_{N_{train}})\}$$
$$D_{test} = \{(x_1^{test}, y_1^{test}), \dots, (x_{N_{test}}^{test}, y_{N_{test}}^{test})\}$$

4. Evaluation and Model Selection

$$L_{test}(W) = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} (f_W(x_n^{test}) - y_n^{test})^2$$

$$L_{test}(W) = -\frac{1}{N_{test}} \sum_{n=1}^{N_{test}} [y_n^{test} \log(f_W(x_n^{test})) + (1 - y_n^{test}) \log(1 - f_W(x_n^{test}))]$$

4. Evaluation and Model Selection

Model Selection with hyperparameters

1. 특성의 수를 변화시키면서 모델을 학습과 검증해본다.
 2. lambda와 같은 값을 변화시키면서 모델을 학습, 검증해본다.
- + 딥러닝에서는 선형 회귀보다 모델이 더 복잡하고 학습시키는 알고리즘도 다양하기 때문에 여러가지 조합이 나올 수 있고 때문에 모델을 분석하는 것이 더 중요해진다. (모든 경우의 수를 모두 탐색할 수는 없기 때문에)

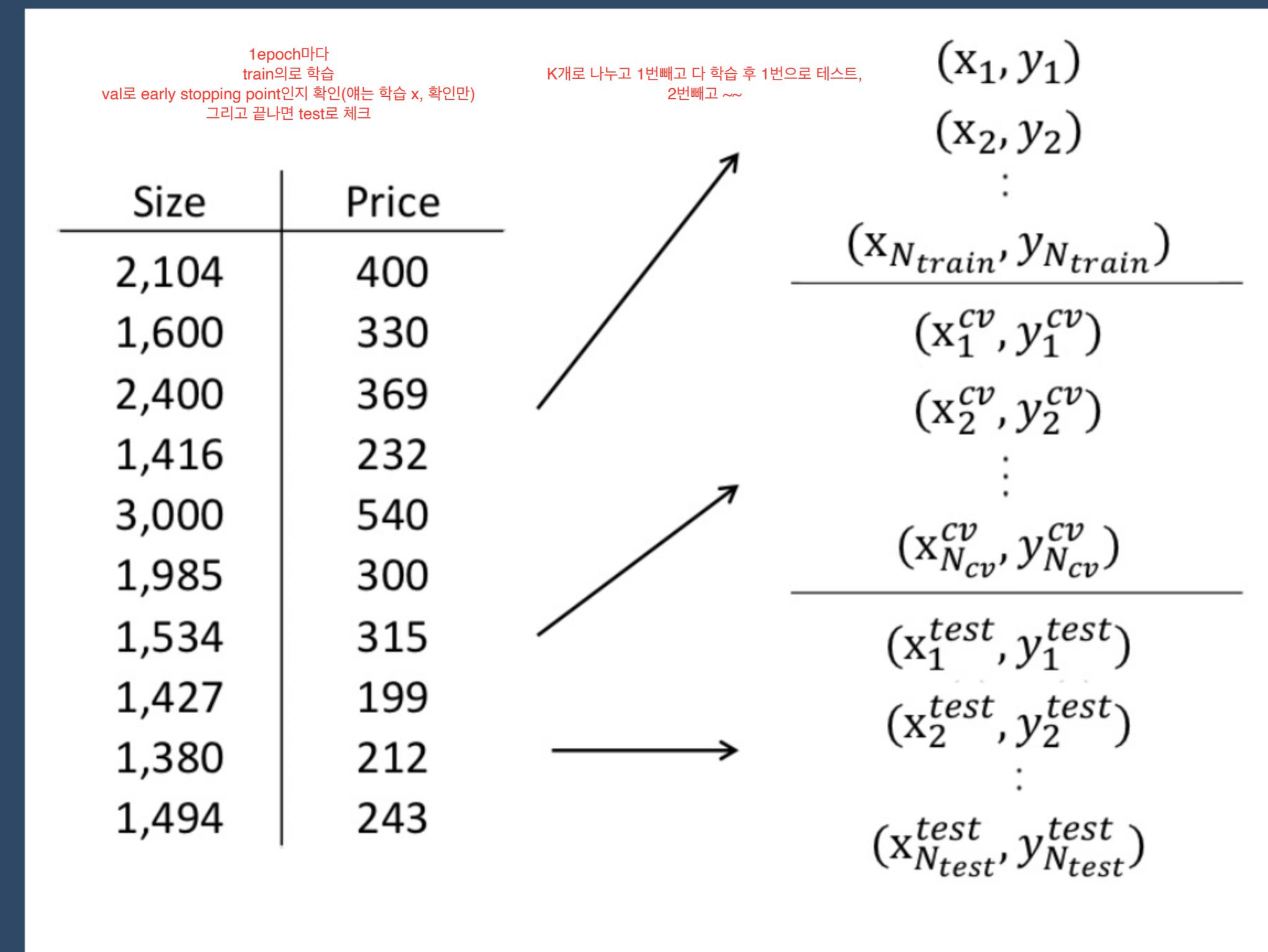
4. Evaluation and Model Selection

(K-Fold) Cross Validation

1. Train Dataset

2. Validation Dataset

3. Test Dataset



5. Bias-Variance Trade-Off

$$y = f(x) + e$$

$$e \sim \mathcal{N}(0, \sigma^2)$$

$$E[y] = f(x)$$

$$y = f(x)$$

$$\hat{y} = \hat{f}(x)$$

$$E[X^2] = Var[X] + \{E[X]\}^2$$

$$\begin{aligned}Var[f] &= E[(f - E[f])^2] = 0 \\E[f] &= f\end{aligned}$$

$$Var[e] = \sigma^2$$

$$\begin{aligned}Var[y] &= E[(y - E[y])^2] \\&= E[(y - f)^2] \\&= E[(f + e - f)^2] \\&= E[e^2] \\&= Var[e] + \{E[e]\}^2 \\&= \sigma^2\end{aligned}$$

bias-variance link: <https://velog.io/@cleansk/y/%EC%9D%B8%EC%82%AC%EC%9D%B4%EB%93%9C-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-Bias-Variance-Trade-Off>

5. Bias-Variance Trade-Off

$$\begin{aligned} E[(y - \hat{f})^2] &= E[y^2 + \hat{f}^2 - 2y\hat{f}] \\ &= E[y^2] + E[\hat{f}^2] - 2E[y\hat{f}] \\ &= Var[y] + \{E[y]\}^2 + Var[\hat{f}] + \{E[\hat{f}]\}^2 - 2E[(f + e)\hat{f}] \\ &= Var[y] + Var[\hat{f}] + f^2 + \{E[\hat{f}]\}^2 - 2E[f\hat{f}] \\ &= \sigma^2 + Var[\hat{f}] + (f - E[\hat{f}])^2 \\ &= \sigma^2 + Var[\hat{f}] + bias[\hat{f}]^2 \end{aligned}$$

5. Bias-Variance Trade-Off

모델의 오류는 2개의 하위 요소로 구분할 수 있다: Bias에 의한 오류와 Variance에 의한 오류

Bias에 의한 오류: 실제 정답 값과 모델이 예측한 값의 차이로부터 계산되는 오류, 이 오류는 말 그대로 모델이 예측한 값이 정답인지 아닌지에 의해 결정된다.

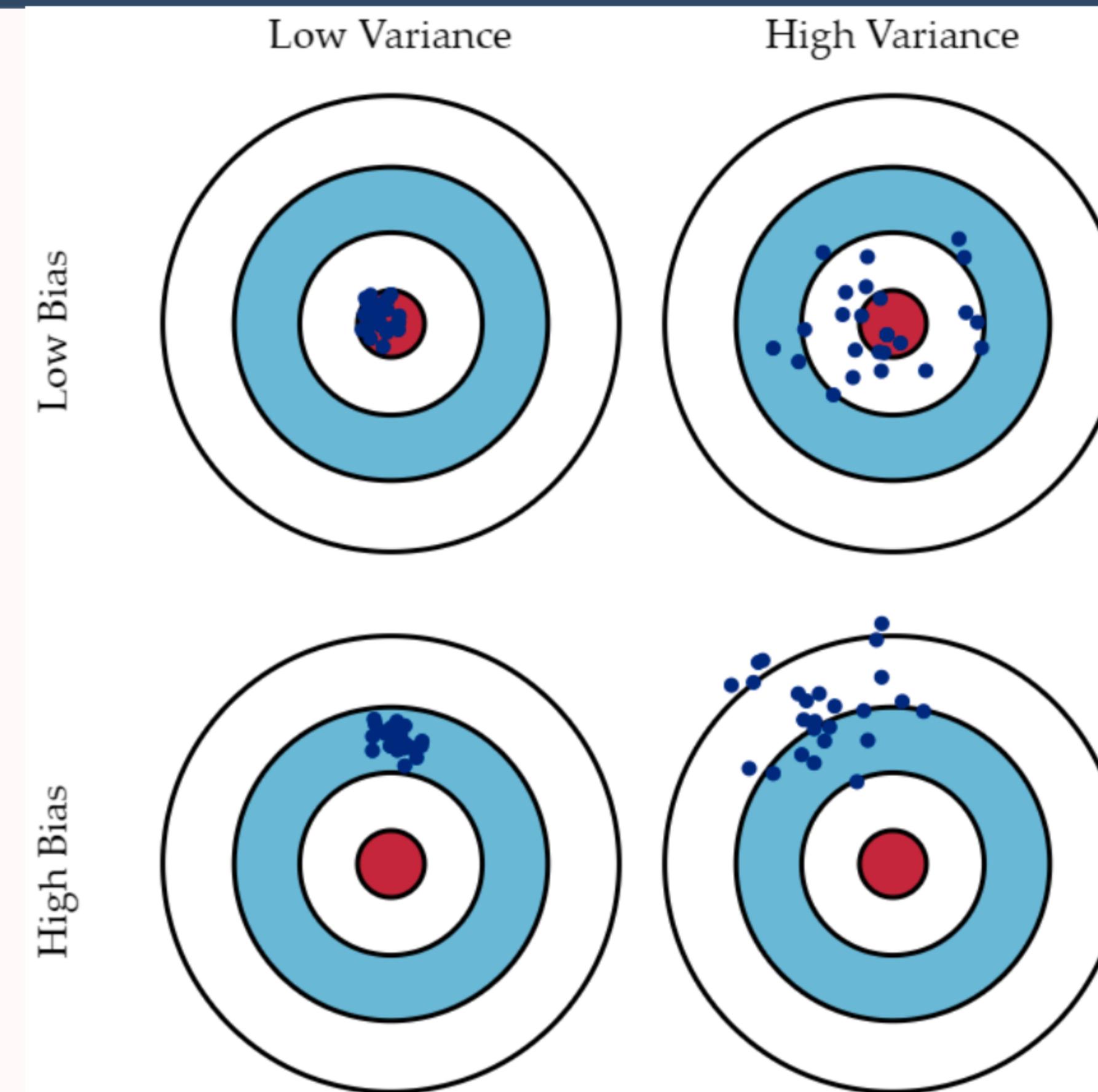
Variance에 의한 오류: 주어진 학습 데이터에 의해 달라지는 모델의 예측값의 차이로부터 계산되는 오류

모델은 어떤 학습 데이터를 통해 학습하느냐에 따라 다른 결과를 가져온다.

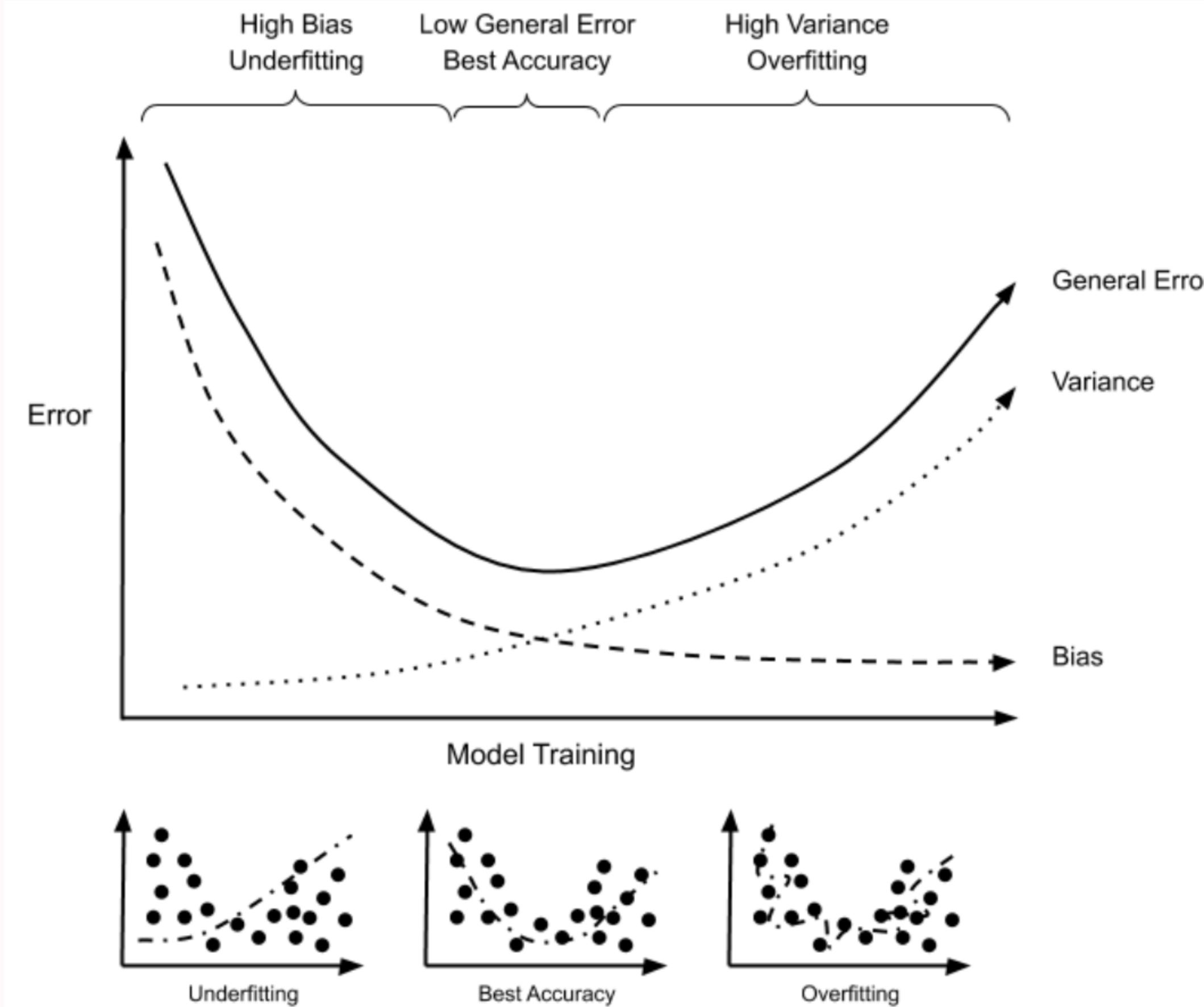
A라는 데이터로부터 학습하여 얻은 모델 a와 B라는 데이터로부터 학습하여 얻은 모델 b는 특정한 입력 x 에 대해 다른 예측을 내놓을 수 있다.

이 오류는 데이터의 변화에 따른 모델 변동성에 대한 오류이다.

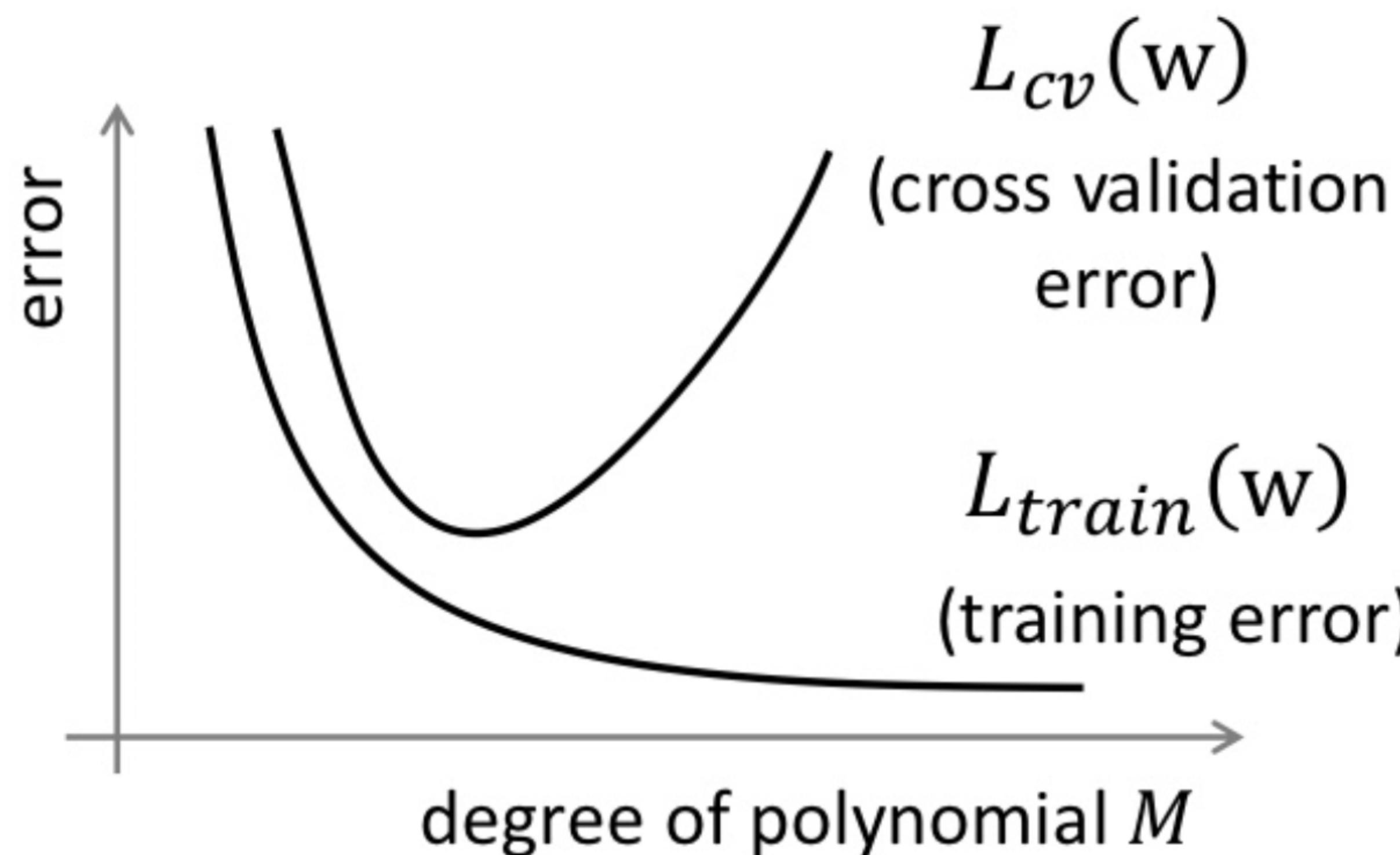
5. Bias-Variance Trade-Off



5. Bias-Variance Trade-Off



5. Bias-Variance Trade-Off



5. Bias-Variance Trade-Off

더 생각해볼 것들

1. bias-variance with linear regression and polynomial features
2. bias-variance with linear regression and regularization
3. bias-variance with linear regression and dataset size

Thank you for listening

Deep Into Deep