

[ML-11] K-Nearest Neighbors

Recap: supervised learning

1) Setup 지도학습을 해결할 문제를 정한다.

2) Data Collection

training과 test에 사용할 올바른 장성과 함께 정된 데이터를 준비한다.

3) Representation

준비한 데이터를 학습 모델에 사용할 방식으로 표현 방법을 선택한다.

4) Modeling

학습에 사용할 hypothesis class를 선택한다.

5) Learning/estimation

학습을 통해 최적의 hypothesis class를 찾는다.

6) Model selection.

다른 모델을 시도해보면서 최적의 모델을 찾는다.

Nearest neighbor classifier.

Nearest Neighbor의 핵심 개념은 주변에 있는 가장 가까운 데이터에 의해 값이 정해지라는 것이다. 그리고 가까운 이웃의 수나 이웃까지의 거리를 계산하는 것에는 다양한 방법이 존재한다.

- Distance metrics

$$L_2\text{-norm: } D(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$

$$L_1\text{-norm: } D(x, x') = \sum_i |x_i - x'_i|$$

$$L_\infty\text{-norm: } D(x, x') = \max(|x_i - x'_i|)$$

$$\text{Scaled } L_2\text{-norm: } D(x, x') = \sqrt{\sum_i \sigma_i^2 (x_i - x'_i)^2}$$

$$\text{Mahalanobis distance: } D(x, x') = \sqrt{(x - x')^T A (x - x')}$$

KNN Classification

→ KNN은 K개의 이웃을 통해 답을 찾아내려 하는 것이다.



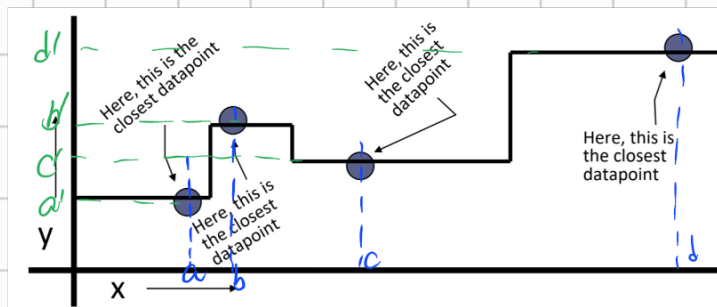
먼저 K값을 살펴보면 test example과 가장 가까운 데이터 3개 중 가장 많은 수를 차지하는 데이터로 답을 유추하는 것이다.

가장 가까운 데이터 3개, 가장 가까운 데이터 2개 가 있으면 K=3일 때는

test = 가장 가까운 데이터이다.

K=5일 때 → 가장 가까운 데이터 3개, 가장 가까운 데이터 2개 → test = 가장 가까운 데이터이다.

KNN Regression



간단하게 1-NN regression을 보면 위와 같다.

$$h(a) \rightarrow \frac{a+b}{2} \rightarrow \text{가장 가까운 이웃이 a이면 } \lfloor \frac{a+b}{2} \rfloor$$

$$\frac{a+b}{2} \leq x < \frac{b+c}{2} \rightarrow \lfloor \frac{a+b}{2} \rfloor$$

$$\frac{b+c}{2} \leq x < \frac{c+d}{2} \rightarrow \lfloor \frac{b+c}{2} \rfloor$$

$$\frac{c+d}{2} \leq x \rightarrow \lfloor \frac{c+d}{2} \rfloor$$

→ 1-NN의 경우 위와 같이 모든 training data에 맞추기 때문에

가장 가까운 이웃의 이웃이 불필요한 noise 있는 경우에도 학습이 이루어져

over fitting에 해당한다.

→ 따라서 K개의 이웃을 통해 값을 추측하는 KNN을

주로 사용한다.

Problems with Instance-Based Learning

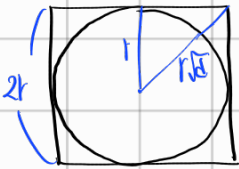
Expensive

- KNN 같은 경우 Learning 과정이 없기 때문에 대부분의 실제 작업은 test 과정에서 이루어짐
- 모든 test sample 마다 모든 dataset을 통해 search를 해야해서 비효율적.

Doesn't work well when large number of irrelevant features.

- noisy feature와 같이 연관이 없는 데이터가 많을수록 결과 좋지 않음.

Curse of Dimensionality



$$\frac{V_{\text{hypersphere}}}{V_{\text{hypercube}}} = \frac{\pi^{d/2}}{2^d \Gamma(d/2 + 1)} \rightarrow \text{d가 커지면 0으로 갈}$$

⇒ hypercube가 커질수록 cube의 corner에 거의 속하지 않음.

→ 이러한 문제들을 해결하기 위해 Dimension을 줄이거나 Random projection을 한다.

Hyper parameter selection

- test dataset은 hyper parameter를 조정하는 것은 의미가 없음.
- 최대한 training set의 일부를 validation set으로 남겨 hyper parameter를 조정한다.