Elements of Statistical Learning

Week 1: Loss Function and Bias-Variance Trade-Off

Basic Setup

통계적 학습에서 관심이 있는 것은 지도 학습으로 입력값 X가 주어졌을 때, 출력값 Y를 잘 맞추기, 또는 비지도 학습으로 차원 축소, 클러스터링 등이 있다. 여기서는 지도 학습, 비지도 학습 등에 대한 내용은 간단하므로 생략하고 지도 학습의 경우에 정의되는 Loss Function과 Bias-Varance Trade-Off를 자세하게 살펴본다. 우선 notation을 정리하자.

- 입력 벡터 $X \in \mathbb{R}^d$는 fixed but unknown 분포인 $P(x)$에서 독립적으로 뽑힌다.
- 각 입력에 대한 출력 값 $Y \in \mathbb{R}$는 조건부 분포인 $P(y \mid x)$에서 뽑힌다.
- 입력 벡터와 출력 값의 training set인 $(x_1, y_1), \cdots, (x_N, y_N)$는 X, Y의 결합 분포인 $P(x, y) = P(y \mid x)P(x)$에서 독립적ㅇ로 뽑힌다.
- Statistical Learning은 훈련 데이터 세트인 $(x_1, y_1), \cdots, (x_N, y_N)$가 주어졌을 때, Y를 가장 잘 예측하는 best function을 $\{f(x; \alpha), \alpha \in \Lambda\}$로부터 찾는 것이다. 여기서 Λ는 임의의 모수 세트이다.

이 setup은 Y가 존재하는 지도 학습에 준한다.

Loss Function and Risk

$(x_1, y_1), \cdots, (x_N, y_N)$가 주어졌을 때, Y를 가장 잘 예측하는 $f(X)$를 찾는 과정에서, '잘 예측하는 것'에 대한 기준이 필요하다. 이를 실제 Y와 함수 f가 X를 받아서 예측하는 $f(X)$와의 '차이의 정도'를 보는 Loss function으로 정의한다. Loss function은 이름에서 유추할 수 있듯이, 손실을 의미하며 손실이 작을 수록 f가 best에 가까움을 의미한다. regression 문제에서 Loss Function은 아래와 같다.

$$L(Y, f(X)) = \begin{cases} (Y - f(X))^2 & \textit{squared error loss} \\ |Y - f(X)| & \textit{absolute error loss} \\ |Y - f(X)|^p & L^p \textit{ error loss} \end{cases} \tag{1}$$

분류 문제에서는 아래의 zero-one loss을 많이 사용한다.

$$L(Y, f(X)) = \begin{cases} 0 & Y = f(X) \\ 1 & Y \neq f(X) \end{cases} \tag{2}$$

앞서 Loss function을 최소로 하는 f가 'best'임을 말했었다. 그런데, $L(Y, f(X))$는 데이터 (X, Y)에서의 손실이다. 즉, 하나의 손실과 유사한 개념이라고 볼 수 있으므로 (정확하게는 Y, X가 확률변수이기 때문에 '손실'이라는 실현된 값은 아니다), 모든 점을 고려하는 손실을 사용해야 하는데, 통계학에서 이러한 개념을 가지고 있는 것이 기댓값이다. $L(Y, f(X))$는 확률 변수로 Y, X를 가지므로 기댓값을 구할 때, 결합 분포를 사용해야 한다. 기댓값을 이용한 Loss를 average loss, 또는 risk functional 이라고 부른다.

$$R(f) = E_{X,Y}\left[L(Y, f(X)\right] = \int L(y, f(x))dP(x, y) \tag{3}$$

지금 하고 있는 논의는 모두 모집단에서 전개되고 있음을 짚고 넘어갈 필요가 있다. 즉, Y, X는 아직 실현된 값이 아닌 확률 변수인 것이다.

Loss Function in Regression Problems

보통 regression 문제에서는 (1)에서 squared error loss을 많이 사용하므로 이를 이용하여 논의를 전개한다. squared error loss을 사용하여 $R(f)$를 다시 적으면 아래와 같다.

$$\begin{aligned} R(f) &= E_{X,Y}\left[(Y - f(X))^2\right] \\ &= E_X E_{Y|X}\left[(Y - f(X))^2 \mid X\right] \end{aligned} \tag{4}$$

'best' f를 찾기 위해, $R(f)$를 최소로 하는 f를 찾으면 된다: $best\ f = \underset{f}{argmin} R(f)$. 그런데 각 $X = x$에서 최소값을 찾으면 자연스럽게 $R(f)$의 최소값도 찾게되므로 $bets\ f = \underset{f}{argmin} E_{Y|X}\left[(Y - f(X))^2 \mid X\right]$. 즉,

$$f(x) = \underset{c}{argmin} E_{Y|X}\left[(Y - c)^2 \mid X = x\right] \tag{5}$$

(5)를 만족하는 $f(x)$는 $f(x) = E(Y \mid X = x)$로 유도된다. 모집단에서 $R(f)$를 최소로하는 가장 좋은 f는 conditional expectation인 것이다.

Loss Function in Classification Problems

$Y \in \mathcal{G} = \{1, \cdots, K\}$일 때,

$$\begin{aligned} R(f) &= E_{X,Y}\left[(Y - f(X))^2\right] \\ &= E_X E_{Y|X}\left[(Y - f(X))^2 \mid X\right] \\ &= E_X\left[\sum_{k=1}^{K} L(k, f(X))P(Y = k \mid X)\right] \end{aligned} \tag{6}$$

마찬가지로, $R(f)$를 최소화하는 것은 $\sum_{k=1}^{K} L(k, f(X))P(Y = k \mid X)$을 최소화하는 것과 동일하다.

best f를 찾아보자.

$$
\begin{aligned}
f(x) &= \underset{g \in \mathcal{G}}{argmin} \sum_{k=1}^{K} L(k,g) P(Y=k \mid X=x) \\
&\underset{g \in \mathcal{G}}{argmin} \sum_{k \neq g} P(Y=k \mid X=x) \\
&= \underset{g \in \mathcal{G}}{argmin} \left[1 - P(Y=g \mid X=x)\right] \\
&= \underset{g \in \mathcal{G}}{argmax} P(Y=g \mid X=x) \qquad (7)
\end{aligned}
$$

best f는 (7)과 같이 유도됐는데, (7)은 Bayes classifier로 알려져 있다. 이는 클래스가 g일 확률을 최대로 만드는 classifier f가 Loss 측면에서 가장 좋다는 의미이다.

Empirical Risk Minimization (ERM)

위의 논의들은 모두 모집단에서 진행됐다. 다시 말해, $P(x,y)$가 알려져있지 않기 때문에 $R(f)$를 구할 수 없다. 따라서 훈련 데이터 $(x_1,y_1),\cdots,(x_N,y_N)$을 통해서 $R(f)$을 estimate해야 한다. 보통 통계학에서 기댓값을 추정할 때, 표본평균을 사용한다. 이는, LLN에 의해서 표본 평균이 기댓값으로 수렴할 뿐만 아니라, MVUE 등 좋은 성질은 모두 가지고 있기 때문이다.

$$
R_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) \qquad (8)
$$

Bias-Variance Trade-off

테스트 데이터 x_0의 MSE는 아래와 같이 분해된다.

$$
\begin{aligned}
MSE(x_0) &= E\left[\left(\hat{f}(x_0) - f(x_0)\right)^2\right] \\
&= E\left[\left(\hat{f}(x_0) - E\left[\hat{f}(x_0)\right] + E\left[\hat{f}(x_0)\right] - f(x_0)\right)^2\right] \\
&= E\left[\left(\hat{f}(x_0) - E\left[\hat{f}(x_0)\right]\right)^2\right] + E\left[\left(E\left[\hat{f}(x_0)\right] - f(x_0)\right)^2\right] \\
&= Var(\hat{f}(x_0)) + Bias(\hat{f}(x_0))^2 \qquad (9)
\end{aligned}
$$

(9)는 MSE가 variance와 bias의 제곱으로 분해된다는, 유명한 결과이다. 이 결과를 해석해보면, $\hat{f}(x_0)$의 variance가 증가할수록 bias는 작아진다. MSE는 한정되어 있기 때문이다. 반대로 $\hat{f}(x_0)$의 bias가 증가할수록 variance는 작아진다. 즉, variance와 bias는 서로 상충되는 관계를 가진다. 가장 좋은 상황은 bias도 작고 variance도 작은 상황이다. 하지만 MSE를 분해해보면 이 둘을 동시에 작게 만들수는 없음을 알 수 있다.

Expected Prediction Error (test or generalization error) at x_0

$$
\begin{aligned}
EPE(x_0) &= E\left[\left(Y - \hat{f}(x_0)\right)^2\right] \\
&= E\left[\left(Y - f(x_0) + f(x_0) - \hat{f}(x_0)\right)^2\right] \\
&= E\left[(Y - f(x_0))^2\right] + E\left[\left(f(x_0) - \hat{f}(x_0)\right)^2\right] - 2E\left[(Y - f(x_0))\left(f(x_0) - \hat{f}(x_0)\right)\right] \\
&= \sigma_\epsilon^2 + MSE(x_0) \qquad (10)
\end{aligned}
$$

(10)에서 σ_ϵ^2는 unknown constant이므로, test error, 즉 $EPE(x_0)$는 $MSE(x_0)$와 비슷한 경향을 보임을 알 수 있다.