MLE & MAP

(Maximum Liklihood Estimation & Maximum A Posterior)

Deep-Byun

발표자 : 김수환

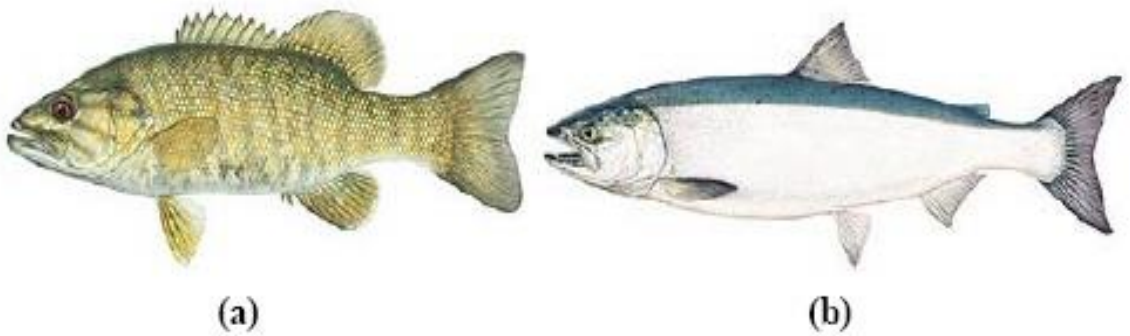
2020.01.17 (FRI)

Bayes Rule

Bayes Rule은 Bayesian Deep Learning에서 가장 기본이 되는 개념이다.
MLE와 MAP를 정리하기 전에 먼저 Bayes Rule에 대해 간단히 살펴보고 넘어가자 !

Bayes Rule

- **농어 vs 연어**

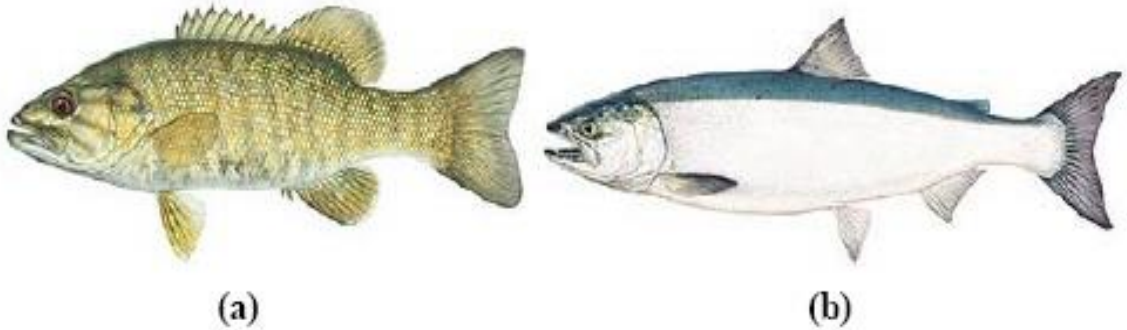


[그림1] (a) 농어 (b) 연어

Bayes Rule을 쉽게 이해하기 위해 낚시를 통해 건져 올린 물고기를 보고
농어인지 연어인지 맞추는 문제를 예로 들어보자 !!
(물고기를 분류하는 기준은 **피부의 밝기**이다)

Bayes Rule

- **수식적 표현**



[그림1] (a) 농어 (b) 연어

이 문제를 수학적으로 정의해보자.

물고기의 피부색의 밝기를 x, 물고기의 종류를 w라고 하자.

물고기가 농어일 사건을 $w = w_1$, 연어일 사건을 $w = w_2$라고 하자.

Ex) 물고기의 피부 밝기가 0.5일 때 그 물고기가 농어일 확률

$$P(w = w_1|x = 0.5) = P(w_1|x = 0.5)$$

Bayes Rule

- **조건부 확률 Review**

$$P(A|B)$$

B일때 A가 발생할 확률

Bayes Rule

- **Posterior (사후확률)**

[그림1] (a) 농어 (b) 연어

- $P(w_1|x) > P(w_2|x)$라면 농어로 분류한다.
- $P(w_2|x) > P(w_1|x)$라면 연어로 분류한다.

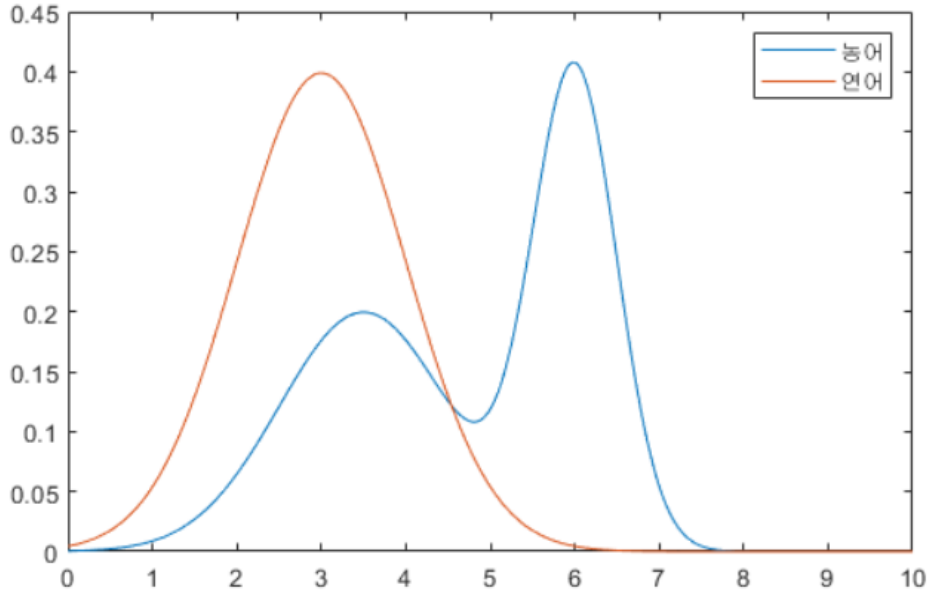
이 문제는 위와 같이 간단하게 정리된다.

x가 주어졌을 때 그 물고기가 class w_i에 속할 확률만 구하면 된다 !!

여기서 우리가 구해야 하는 확률 $P(w_i|x)$를 **Posterior(사후확률)**이라고 부른다.

Bayes Rule

- **Likelihood**



[그림2] 농어와 연어의 피부 밝기 분포

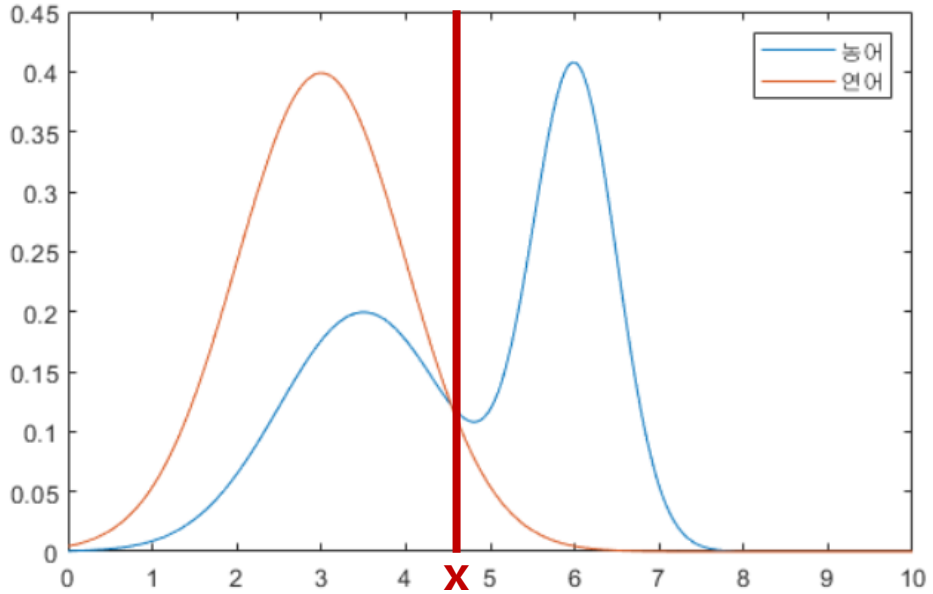
그렇다면 **Posterior**는 어떻게 구하지??

방법이 없으면 농어, 연어를 잡아서 **관찰**해보면 되지 !!

이렇게 **관찰**을 통해 얻은 확률 분포 $P(x|w_i)$를 **Likelihood(가능도)**라고 부른다.

Bayes Rule

- **Likelihood**



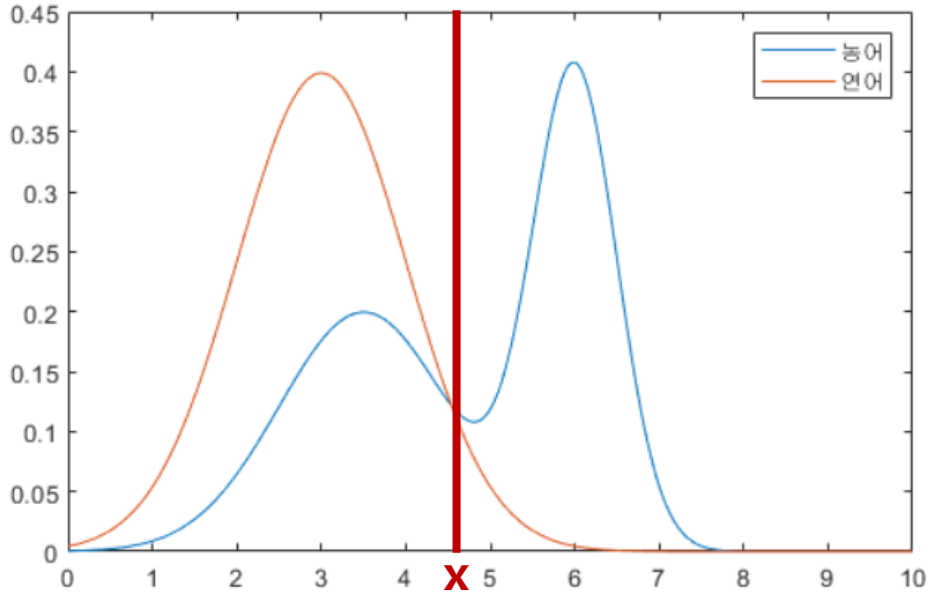
[그림2] 농어와 연어의 피부 밝기 분포

이렇게 그림2와 같은 **Likelihood**를 얻었다고 가정해보자.

그렇다면 그냥 지금 위의 분포에서 x보다 작으면 연어, 크면 농어로 분류하면 되지 않을까?

Bayes Rule

- **Likelihood의 한계**



[그림2] 농어와 연어의 피부 밝기 분포

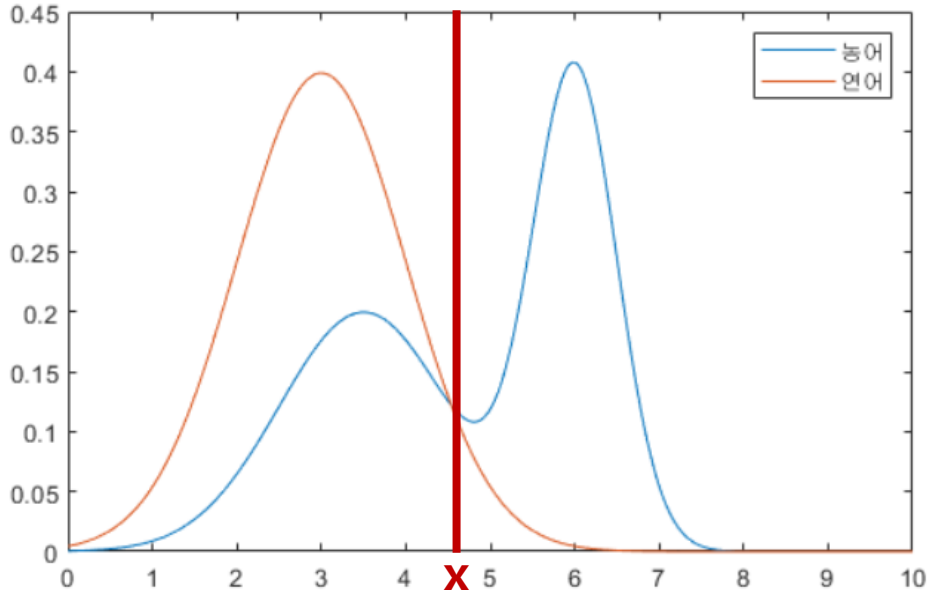
이렇게 그림2와 같은 **Likelihood**를 얻었다고 가정해보자.

그렇다면 그냥 지금 위의 분포에서 x보다 작으면 연어, 크면 농어로 분류하면 되지 않을까?

No

Bayes Rule

- **Likelihood의 한계**

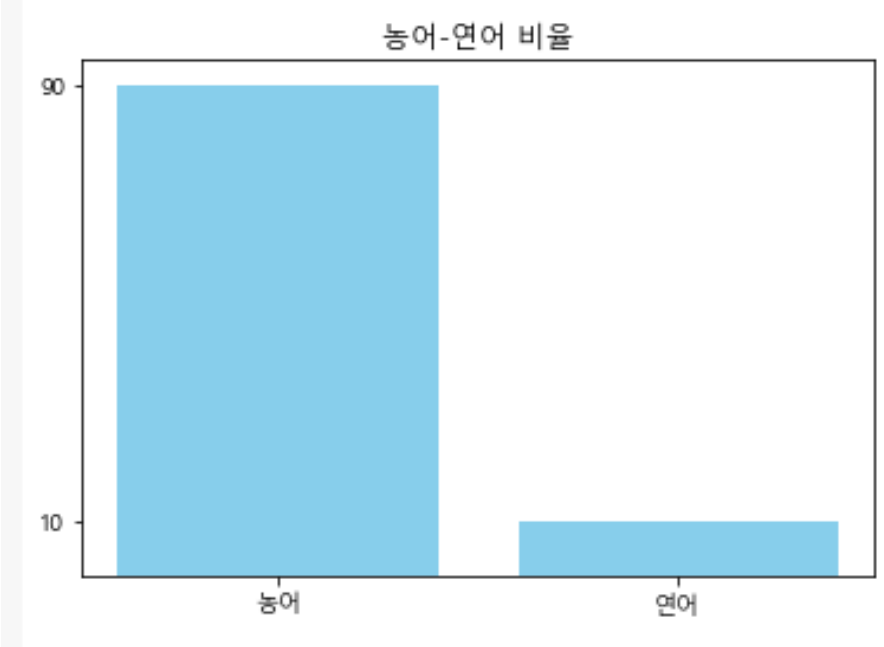


[그림2] 농어와 연어의 피부 밝기 분포

Likelihood에는 애초에 **연어와 농어가 잡힐 확률**이 반영되어 있지 않다 !!
이전과 같은 방법으로 분류를 하기 위해서는 농어와 연어가 똑같은 비율로
바다에 살고 있다는 가정이 있어야한다.

Bayes Rule

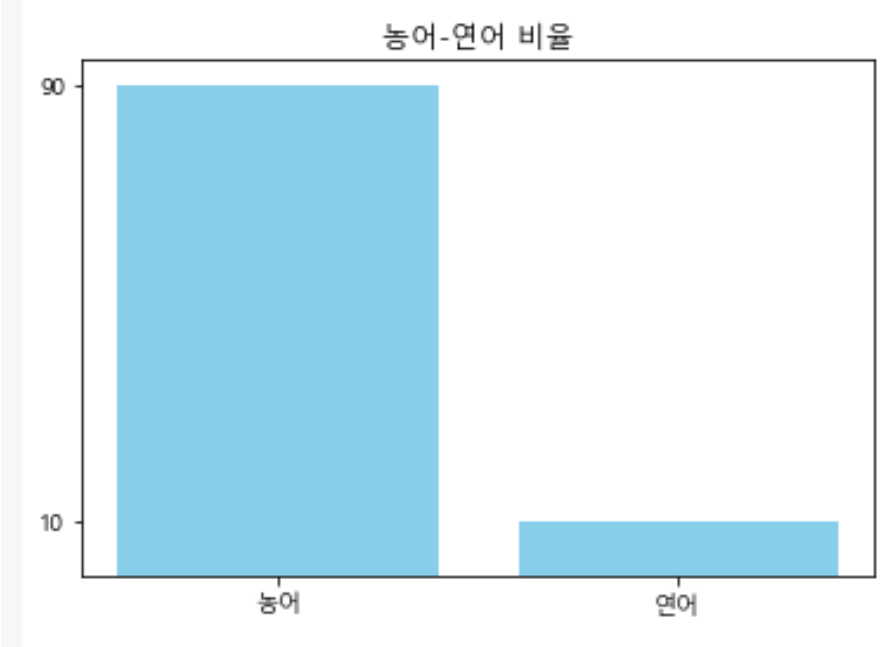
- **Likelihood의 한계**



특정 x에 대해 연어의 Likelihood가 농어의 Likelihood보다 크다 하더라도,
애초에 연아가 매우 희귀하다면 우리는 이 물고기가 농어일 가능성도 고려해야 한다.

Bayes Rule

- **Prior (사전확률)**



결론적으로, x와 관계 없이 **농어가 잡힐 확률 ($P(w_1)$)과 연어가 잡힐 확률 ($P(w_2)$)을 알아야 한다.**

이 값을 우리는 Prior(사전확률)이라고 하며, 우리가 이미 갖고 있는 사전 지식에 해당한다.

Prior는 보통 우리가 사전 지식을 이용해 정해줘야 하는 경우가 많다.

Bayes Rule

- **정리**

- **Posterior($P(w_i|x)$)**: 피부 밝기(x)가 주어졌을 때 그 물고기가 농어일 확률 또는 연어일 확률. 즉 단서가 주어졌을 때, 대상이 특정 클래스에 속할 확률. 우리가 최종적으로 구해야 하는 값이다.
- **Likelihood($P(x|w_i)$)**: 농어 또는 연어의 피부 밝기(x)가 어느 정도로 분포되어 있는지의 정보. 즉 각 클래스에서 우리가 활용할 단서가 어떤 형태로 분포 돼 있는지를 알려준다. Posterior를 구하는 데 있어서 매우 중요한 단서가 된다.
- **Prior($P(w_i)$)**: 피부 밝기(x)에 관계 없이 농어와 연어의 비율이 얼마나 되는지의 값. 보통 사전 정보로 주어지거나, 주어지지 않는다면 연구자의 사전 지식을 통해 정해줘야 하는 값이다.

지금까지 우리는 세 가지 종류의 확률을 알아봤다.
정리해보자.

Bayes Rule

- **Bayes Rule**

우리의 목적은 P**osterior**를 구하는 것이며, 이 값은 **Likelihood**와 **Prior**를 이용하면 구할 수 있다.
고등학교 때 배운 조건부 확률의 정의를 떠올려보자.

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

이를 살짝 변형하면 다음과 같은 식을 얻을 수 있다.

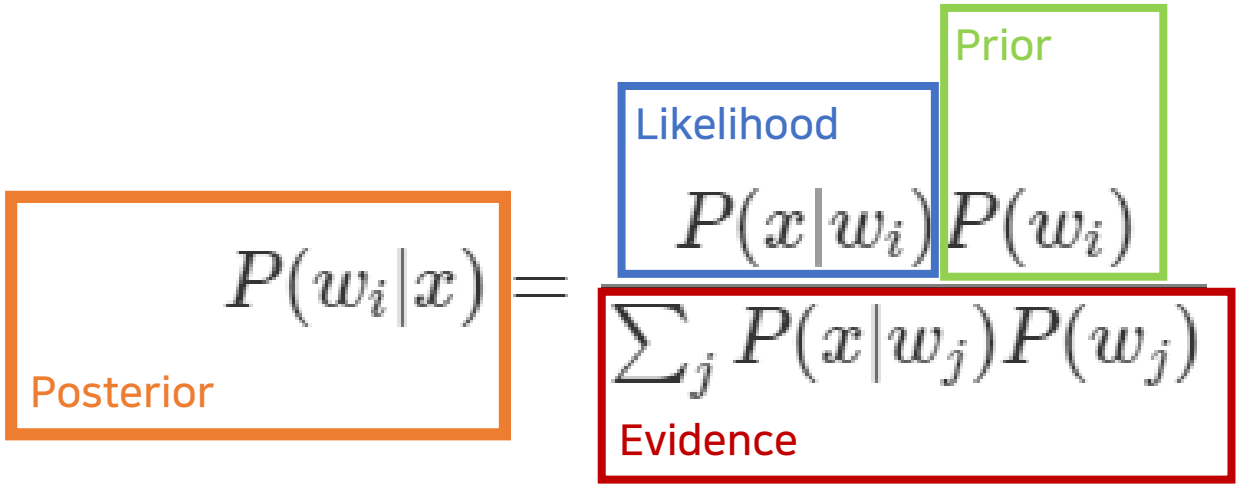
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)}$$

그리고 A 대신 w, B대신 x를 넣으면 다음과 같이 된다 !

$$P(w_i|x) = \frac{P(x|w_i)P(w_i)}{\sum_j P(x|w_j)P(w_j)}$$

Bayes Rule

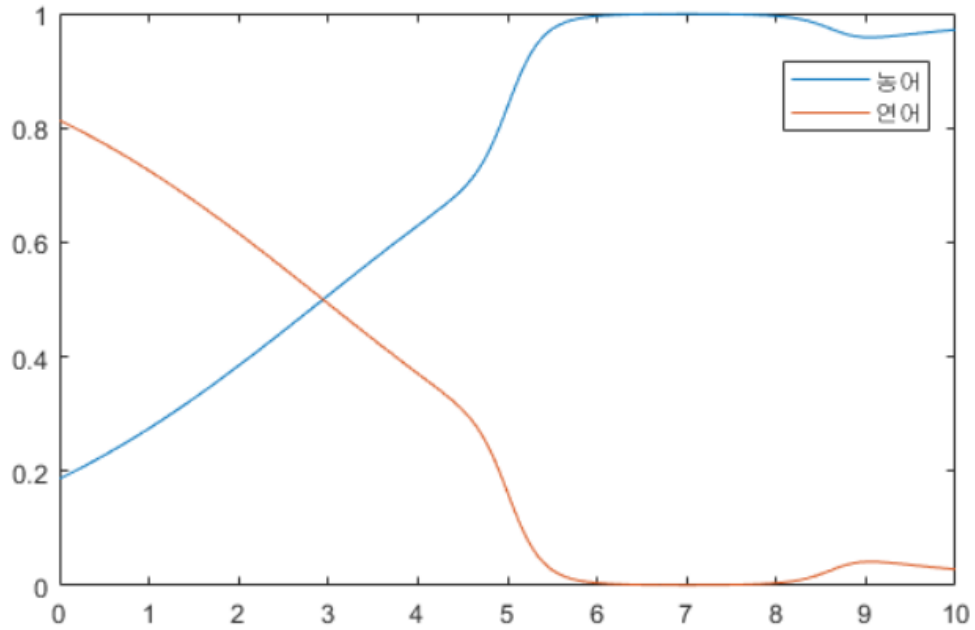
- **Bayes Rule**



좌변은 우리가 구하고자 하는 Posterior이고, 우변의 분자는 Likelihood와 Prior의 곱이며,
우변의 분모는 Evidence라고 보통 부르는데, 이 또한 Likelihood와 Prior들을 통해 구할 수 있다.
이 식을 우리는 **Bayes Rule** 또는 **Bayesian Equation**으로 부른다.

Bayes Rule

- **Bayes Rule**



위 그래프는 Bayes Rule에 따라 농어와 연어의 Posterior를 구한 결과이다.

이제 우리는 Posterior가 큰 쪽을 고르면 된다 !!

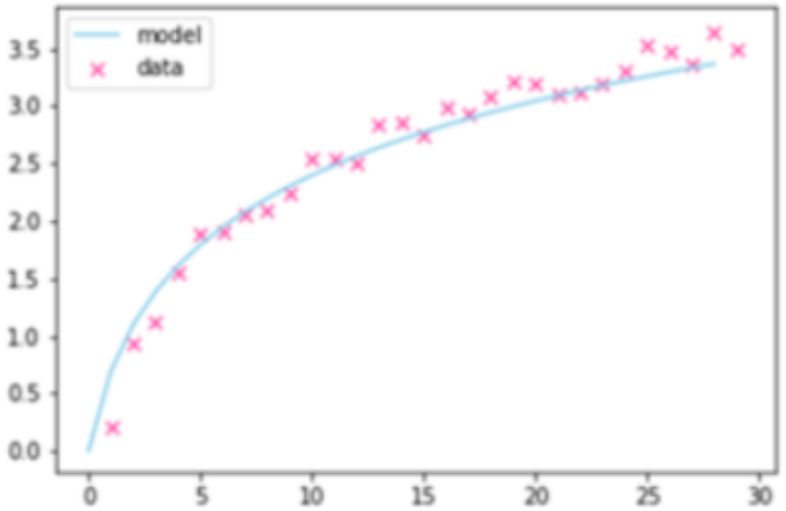
MLE & MAP

Bayes Rule 방식의 가장 큰 단점 중 하나는 Likelihood의 Probability Distribution을 알아야 한다는 점이다.
몇 개의 파라미터로 이루어진 함수로 모델링을 해서 데이터를
가장 잘 설명하도록 파라미터를 구해낼 수 있다면 어떨까???

이러한 방식을 이용하는 대표적인 알고리즘이 **Deep Learning**이다.
Deep Learning의 기본적인 Loss Function들은 대부분 MLE와 MAP를 통해 증명된다.
한번 MLE와 MAP에 관해 알아보자 !!

Maximum Likelihood Estimation

- **문제의 정의**



< Regression >

Regression 문제를 생각해보자.

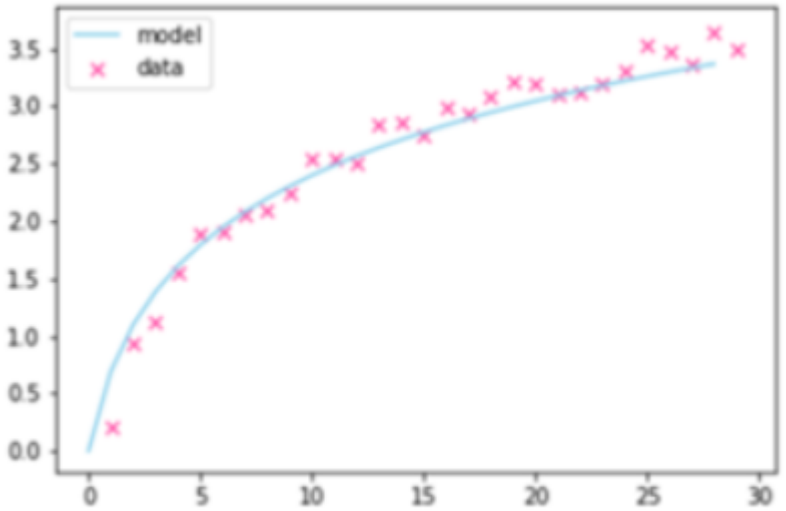
예를 들면 **키를 보고 몸무게를 예측하는 모델**이라고 하자.

Regression으로 모델을 만든다면 다음과 같은 모델이 나오게 될 것이다.

$$t = y(x|w)$$

Maximum Likelihood Estimation

- **문제의 정의**



< Regression >

그렇다면 우리는 항상 $t = y(x|w)$ 라고 말할 수 있을까?

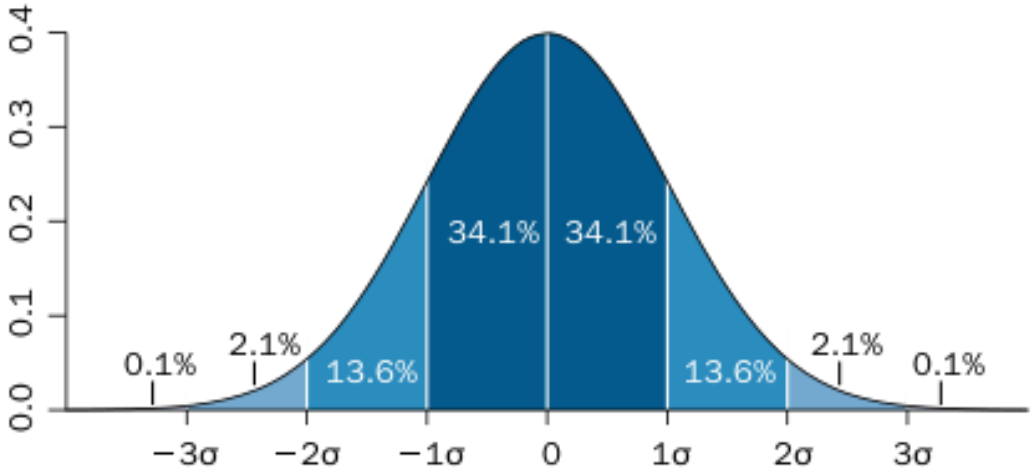
No !!

"실제 몸무게(t)는 내가 예측한 몸무게(y)일 확률이 가장 높지만, 아닐 수도 있어!"

위의 문장이 더 정확한 말일 것이다.

Maximum Likelihood Estimation

- **MLE**



Y를 평균으로 하고 σ를 표준편차로 하는 정규분포

$$t \sim N(y(x|w), \sigma^2)$$

$$p(t|x, w, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-y(x|w))^2}{2\sigma^2}}$$

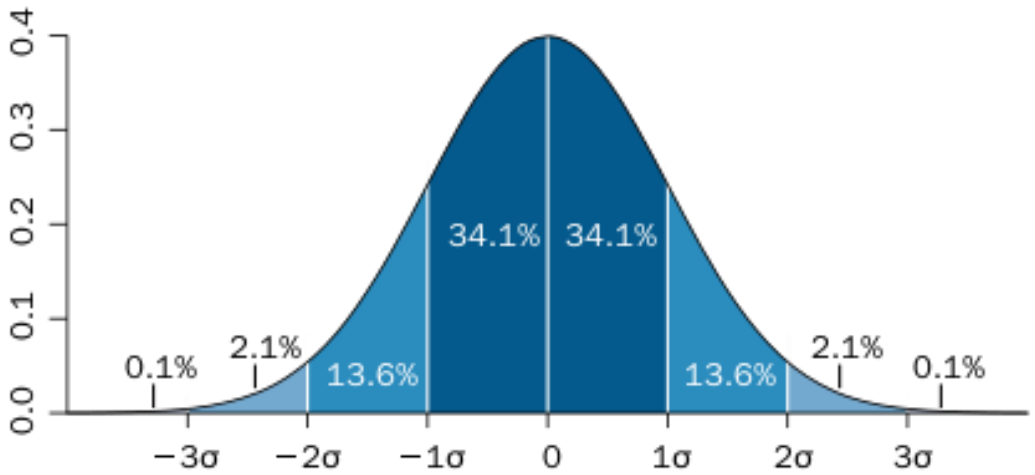
이를 조금 더 수학적인 표현으로 말하면

"실제 몸무게(t)는 내가 예측한 몸무게(y)를 평균으로 하고 특정 값 σ를 표준편차로 하는 정규 분포를 따른다"

고 할 수 있다.

Maximum Likelihood Estimation

- **MLE**



여기서 σ는 무엇을 의미할까?

σ는 **우리가 한 예측이 얼마나 불확실한지의 정도**를 나타낸다.

σ는 풀려는 문제의 특성에 따라 설정되는 상수이다.

Maximum Likelihood Estimation

- **MLE**

Y를 평균으로 하고 σ를 표준편차로 하는 정규분포

$$p(t|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-y(x|w))^2}{2\sigma^2}}$$

앞에서 정규분포 식에 다시 주목해보자 !

P(t|x)의 의미는 **키가 x일 때 실제 몸무게가 t일 확률**이다.

그렇다면 데이터셋이 위와 같이 구성될 확률 p(D)는 어떻게 구할 수 있을까?

Maximum Likelihood Estimation

- **MLE**

데이터셋이 키가 x일 때 몸무게가 t일 확률

$$p(D) = \prod_{i=1}^{N} p(t_i|x_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t_i - y(x_i|w))^2}{2\sigma^2}}$$

p(D) 값은 w에 따라 다르게 구해지기 때문에 p(D|w)라고 하자.

그렇다면 데이터셋이 위와 같이 구성될 확률 p(D)는 어떻게 구할 수 있을까?

⇒ 이를 다시 말하면 "키가 x_1일 때 몸무게가 t_1이고 ... 키가 x_N일 때 실제 몸무게가 t_N일 확률"이다.

이는 데이터가 독립이라고 했을 때 곱의 법칙을 통해 위와 같이 구할 수 있다.

그렇다면 몸무게를 가장 잘 예측하는 모델은 다음과 같을 것이다.

"키가 x_1일 때는 몸무게가 t_1일 확률이 가장 높다고 말하고, ... 키가 x_N일 때는 몸무게가 t_N일 확률이 가장 높다고 말하는 모델"

즉, **p(D|w)가 최대가 되는 모델**, **p(D|w)를 최대로 해주는 w를 찾는 것**이다.

Maximum Likelihood Estimation

- **MLE의 계산**

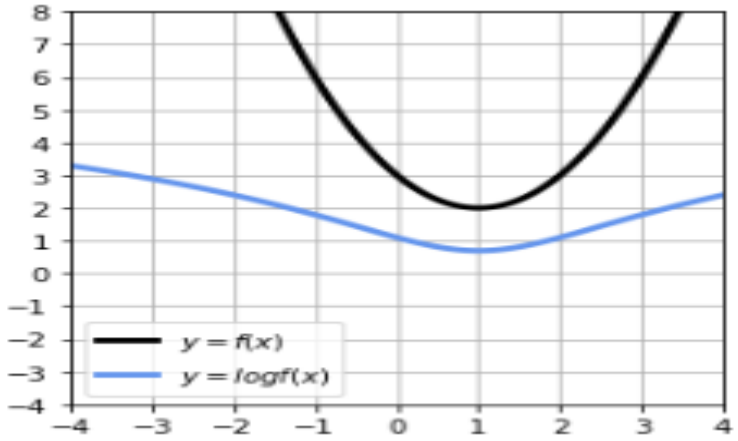
$$\text{likelihood} = p(D|w) = \prod_{i=1}^{N} p(t_i|x_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t_i - y(x_i|w))^2}{2\sigma^2}}$$

그렇다면 다음 식을 최대로 하는 w를 찾아보자 !

이런 문제에서 우리는 주로 log를 취해 준다.
많은 이유가 있지만 대표적으로 2가지 이유 때문이다.

Maximum Likelihood Estimation

- **MLE의 계산**



먼저 likelihoo가 최소 / 최대면, log를 취해주더라도 같은 지점에서 최소 / 최대가 된다.
즉, 어떠한 최소 / 최대가 되는 해를 구할 때 log를 취해줘도 문제가 생기지 않는다.

또한 log의 특성상 복잡한 곱셈 연산을 log간의 간단한 덧셈 연산으로 바꿔준다.

Maximum Likelihood Estimation

- **MLE의 계산**

$$\log \text{likelihood} = \log\left(p(D|w)\right) = \sum_{i=1}^{N}\{-\log\left(\sqrt{2\pi}\sigma\right) - \frac{\left(t_i - y(x_i|w)\right)^2}{2\sigma^2}\}$$

다시 식으로 돌아와서 log를 씌워주면 다음과 같은 식이 전개된다.

이때 최대가 되는 w를 찾아주면 되는데, σ와 π는 상수 값이므로 이를 생략하면 다음 식이 나오게 된다.

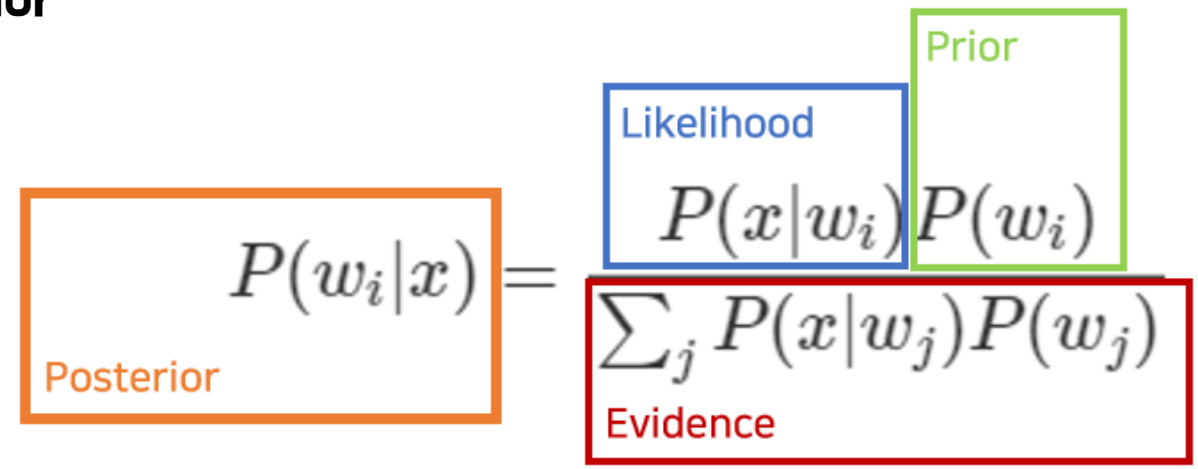
$$\sum_{i=1}^{N}\left(t_i - y(x_i|w)\right)^2$$

예측값과 실제 값의 차이의 제곱인 Loss 함수가 튀어나왔다.

이는 딥러닝에서의 Loss를 최소화 시키는 것은 Likelihood를 최대한 시키는 일이라는 말이 된다 !!

Maximum A Posterior

- **Maximum A Posterior**



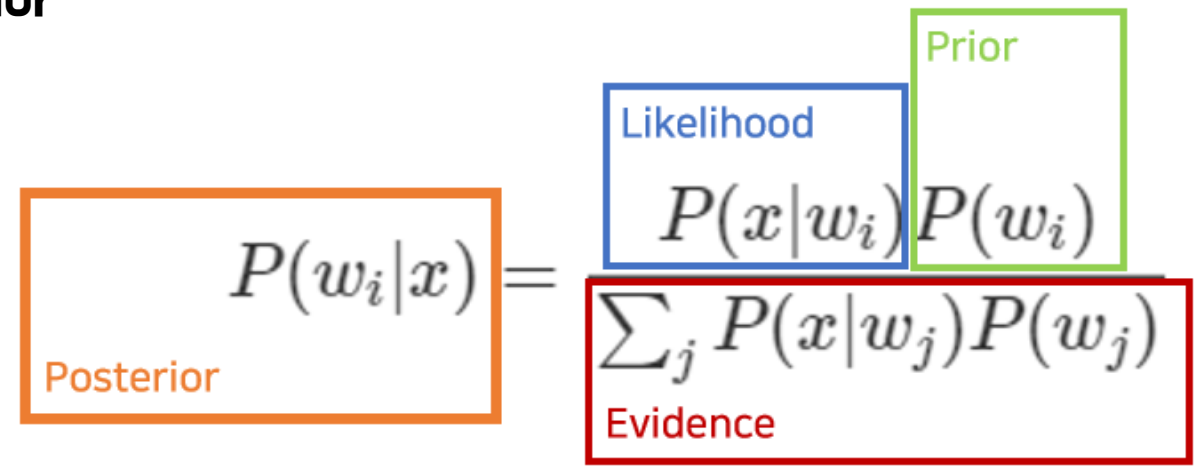
Maximum Likelihood Estimation이 **Likelihood**를 최대화 시키는 작업이었다면,
Maximum A Posterior는 이름 그대로 **Posterior**를 최대화 시키는 작업이다.

Likelihood와 **Posterior**의 차이는 사전 지식인 **Prior**의 유무이다.

즉 구하고자 하는 대상을 철저히 데이터만 이용하고 싶다면 ? **MLE** !!
데이터와 더불어 갖고 있는 사전지식까지 반영하고 싶다면 ? **MAP** !!

Maximum A Posterior

- **Maximum A Posterior**



그렇다면 **Prior**를 적용해서 좋은 점은 무엇일까?

만약 **매우 강력한 사전 지식**을 갖고 있다면 w 값을 구하는 데 있어서 큰 도움이 될 것이다.

하지만 별다른 사전 지식이 없더라도 Prior를 반영하는 것은 좋은 경우가 많다.

Output을 제어할 수 있기 때문이다.

예를 들어 모델링한 함수가 키를 줬을 때 몸무게를 잘 맞추게만 하고 싶으면, MLE를 써도 되지만

파라미터의 절대값이 작기를 원한다면, w가 0주변에 분포한다는 Prior를 걸어주면 된다.

Maximum A Posterior

- **Maximum A Posterior**

$$P(w|D) = \frac{P(D|w)P(w)}{\int P(D|w)P(w)dw}$$

Poseterior를 해야 하는 이유는 Likelihood의 경우보다 단순하다.

Posterior는 애초에 w의 확률 분포기 때문에 w가 될 확률이 가장 높은 값으로 정해주는 것이다.

Bayes Rule에 따라서 Posterior는 위의 식으로 구할 수 있다.

분모가 Sigma에서 Integral로 바뀐 이유는 w가 continuous하기 때문이다.

위의 식에서 ∫P(D|w)P(w)dw는 w에 대해서 적분을 하고 있고, D는 주어진 값이기 때문에 결국 상수가 된다.

$\eta=\frac{1}{\int P(D|w)P(w)dw}$ 로 치환하자.

Maximum A Posterior

- **Maximum A Posterior**

$$P(w|D) = \eta P(D|w)P(w)$$

그러면 이제 w의 Prior (P(w))를 정해주어야 한다.
W에 대한 특별한 사전 지식은 갖고 있지 않다고 가정하고, 우리 나름의 제약 조건을 걸어주자.

딥러닝에서 오버피팅을 방지하기 위한 **Weight Decay**라는 방식이 있다.
Loss에 w^2 또는 $|w|$ 등을 추가하여 w 자체의 크기를 줄여 네트워크의 표현력을 감소시키는 방식인데,
이 방식을 MAP를 이용해 유도해보자 !

Maximum A Posterior

▪ Maximum A Posterior

"오버피팅을 방지하기 위해서는 네트워크의 표현력을 감소시켜야 하는데, 그러기 위해서는 w의 절대값이 작아야한다."

위와 같은 Prior를 걸어주자.

위와 같은 Prior를 걸어주려면 w에 0을 평균으로 하는 정규분포를 걸어주면 될 것이다.

$$w \sim N(0, \sigma_w^2)$$

$$p(w) = \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{w^2}{2\sigma_w^2}}$$

그리고 Posterior도 log를 취해주자. 그리고 Likelihood와 같이 그 값을 최대로 하는 w를 찾는 것이 우리의 목표이다.

$$\begin{aligned} w^* &= \text{argmax}_w \{\log p(w|D)\} \\ &= \text{argmax}_w \{\log \eta + \log p(D|w) + \log p(w)\} \end{aligned}$$

Maximum A Posterior

- **Maximum A Posterior**

$$w^* = \text{argmax}_w\{\log p(w|D)\}$$
$$= \text{argmax}_w\{\log \eta + \log p(D|w) + \log p(w)\}$$

여기서 logp(D|w)는 Likelihood이다.
이를 L(w)라고 치환해서 대입하자.

$$w^* = \text{argmax}_w\{\log \eta - L(w) + \log p(w)\}$$

위의 식을 정리하여 상수들을 전부 생략해주면 다음과 같은 식이된다.

$$L(w) + \frac{w^2}{2\sigma_w^2} = \sum_{i=1}^{N}(t_i - y(x_i|w))^2 + \frac{w^2}{2\sigma_w^2}$$

Maximum A Posterior

- **Maximum A Posterior**

$$L(w) + \frac{w^2}{2\sigma_w^2} = \sum_{i=1}^{N}(t_i - y(x_i|w))^2 + \frac{w^2}{2\sigma_w^2}$$

위의 식의 상수를 α 등으로 치환하면 Weight Decay(L2 Regularization) 방식을 적용한 딥러닝의 Loss 함수가 된다.
우리는 정규 분포를 Prior로 준 문제의 **MAP**로부터 Weight Dacay 식을 유도해 낸 것이다!

또한 딥러닝에서 L2 Regularization을 쓴다는 것은 주어진 데이터를 적용함과 동시에 w에 정규분포를 Prior로 걸어 주어 **MAP**를 통해 w를 구하겠다는 것으로 해석할 수 있다 !!

즉 **L2 Regularization을 적용하는 일은 w에 정규 분포를 Prior로 걸어 주는 일**인 것이다.

※ 참고로 Laplacian Distribution을 Prior로 걸어 주면 L1 Regularization을 얻을 수 있다 ※

- End -