# CH3: Naive Bayes Classifier

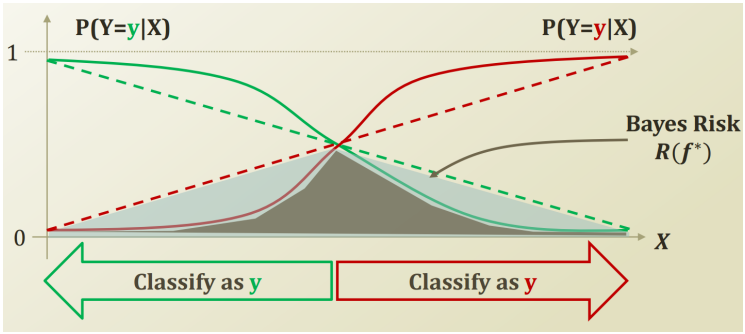


## 3.1 Optimal Classification and Decision Boundary

나이브 베이즈 분류기(Niave Bayes Classifier)는 간단하고 여러 분류 문제에 적용하기 쉬우면서도 뛰어난 성능을 보이는 분류기 중 하나이다. 이런 이름의 유래는, 인스턴스의 레이블을 예측할 때 베이즈 결정 이론(Bayes decision theory)을 사용하기 때문이다. 이제 본격적으로 나이브 베이즈 분류기가 무엇인지 알아보자.

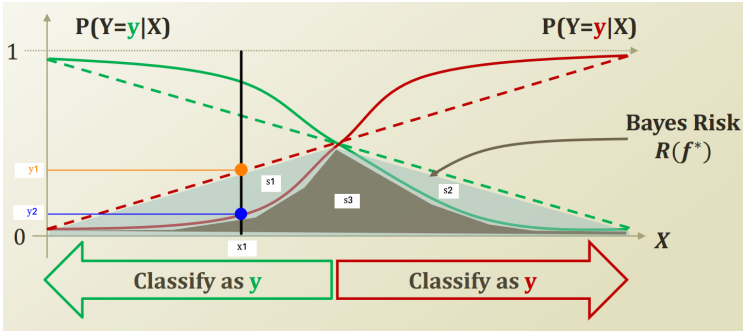
### 3.1.1 Optimal Classification

아래의 그림을 통해 최적의 분류기(optimal classifier)이 무엇인지 알아보자. 그림에는 점선으로 나타난 분류기와 실선으로 나타낸 분류기가 있다.




위 그림에서 x축은 특성 값을 나타내며 각 x에서의 y 값은 클래스를 나타낼 확률을 나타낸다. 두 그래프가 만나는 부분을 결정 경계(Decision Boundary)라고 하며, 이 결정 경계를 기준으로 X가 왼쪽인지 오른쪽인지에 따라 분류기가 예측하는 $\hat{y}$값이 달라진다. 그렇다면 어떤 분류기가 더 좋은 분류기일까?

분류기의 성능은 그림에서도 나타나다시피 베이즈 위험도(Bayes Risk)에 따라 판단할 수 있다. 베이즈 위험이 무엇인지 알아보기 위 그림에 임의의 $x_1$에 해당하는 세로 선을 그려 봤다.



위 그림에서 $x_1$은 결정 경계 왼쪽에 위치하고 있으므로, $x_1$을 입력하게 된다면, 두 분류기 모두 초록색 클래스로 분류될 것이다. 하지만, 이 점에서도 각각의 분류기는 빨간색 클래스일 확률을 가지고 있다. 즉, 초록색으로 분류될 확률이 빨간색으로 분류될 확률이 높을 뿐이다. 점선 분류기는 주황색 점으로 나타나는 위치의 y값인 $y_1$, 실설 분류기는 파란색 점으로 나타나는 위치의 y값인 $y_2$만큼이 특성값 $x_1$에 대한 빨간 클래스로 분류될 확률을 나타낸다.

하지만, 분류기는 이런 확률을 무시하고 초록 클래스를 선택하게 되는데, 이렇게 무시되는 확률을 베이즈 위험이라고 한다. 따라서, 점선 분류기의 베이즈 위험도는 하늘색과 갈색으로 나타나는 부분인 $S_1 + S_2 + S_3$이 된다. 그리고, 실선 분류기의 베이즈 위험은 갈색으로 나타나는 부분인 $S_3$이 된다. 더 좋은 분류기란, 클래스를 선택할 때 해당 클래스가 아닐 확률, 즉, 베이즈 위험을 더 작게 하는 경우이다. 따라서, 위 그림에서는 베이즈 위험도가 더 낮은 실선 분류기가 더 좋은 성능을 가진다고 할 수 있다.

최적의 분류기라면 베이즈 위험을 초소화해야 한다. 클래스를 잘못 선택할 확률을 가장 적게 하는 것이 목적이므로, 최적의 분류기 $f^*$를 다음과 같이 나타낼 수 있다.

$$f^* = \arg\min_f P(f(x) \neq X)$$

이진 분류에 대해서는 다음과 같이 쓸 수 있다.

$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$

이제 최적의 분류기 $f^*$에 대한 식을 어디서 많이 본 모양으로 바꿔보자. 앞서 MLE를 추정했을 때 사용한 방법을 그대로 적용해보면, 다음과 같이 쓸 수 있다.

$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x) = \arg\max_{Y=y} P(X = x|Y = y)P(Y = y)$$

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)} \text{ and } P(x) \text{ is constant}$$

위의 식을 보면, Class conditional density와 class prior를 곱한것을 최대화하는 것을 알 수 있다. 이를 다른 말로 표현하자면, prior와 likelihood값을 구해 이를 최대화 한것이다. 즉, 우리가 앞서 배운 그 내용을 활용한 것이다!

## 3.2 Naive Bayes Classifier

나이브 베이즈 분류기의 나이브는 순진하다는 것을 의미한다. 분류기를 만들 때 가장 문제가 되는 것은 데이터셋 내에 있는 변수/특성/feature 사이에 어떤 관계를 가지고 있는지 모른다. 따라서, 데이터셋 내에 있는 모든 특성이 관계를 맺고 있다면, 클래스가 k개이고, 특성이 d개일 때, Class conditional denstiy를 구하기 위해서는 $(2^d - l)k$개의 파라미터가 필요하다. 그리고 class prior를 구하기 위해 $k - 1$개의 파라미터가 필요하다. 만약 N이 엄청 큰 값이라면, 파라미터의 계수 또한 늘어날 것이며, 차원의 저주, computing power 등 많은 문제가 발생할 것이다. 또한, 나이브 베이즈 분류기는 모든 경우의 class conditional density를 요구하기 때문에, 데이터 셋 자체도 클 뿐만 아니라, 모델의 assumption도 비현실적이기 때문에 어떠한 제약 조건을 걸어야 한다.

이를 해결하기 위해 사용할 예시는 다음과 같다.

| Sky | Temp | Humid | Wind | Water | Forecst | EnjoySpt |
|---|---|---|---|---|---|---|
| Sunny | Warm | Normal | Strong | Warm | Same | Yes |
| Sunny | Warm | High | Strong | Warm | Same | Yes |
| Rainy | Cold | High | Strong | Warm | Change | No |
| Sunny | Warm | High | Strong | Cool | Change | Yes |

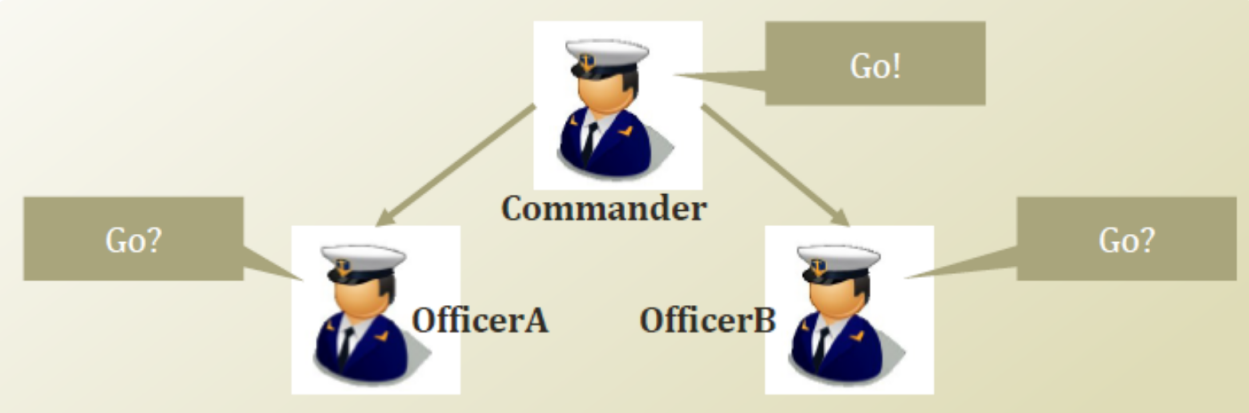
위의 예시의 경우, 클래스의 개수 k는 1이고, 특성의 개수d는 6이므로, $2 \times (2^6 - 1) \times (2 - 1) = 126$개의 파라미터가 필요하다. 즉, 모든 특성이 관계를 맺고 있다면, 간단한 데이터셋을 분류하는데도 많은 파라미터가 필요된다. 하지만, 실제 데이터는 적게는 수십에서 많게는 수백개의 변수를 가지는 경우가 많다. 공식에 따르면, 변수의 개수에 따라 필요한 파라미터의 개수는 기하급수적으로 늘어나므

로, 변수 사이의 관계를 해결하지 않으면, 필요한 파라미터의 개수는 엄청나게 늘어날 것이다. 이 문제를 해결하기 위해, 조건부 독립을 사용해야 한다.

### 3.2.1 Conditional Independence

조건부 독립(Conditional Independence)는 임의의 관계에 있는 두 사건이 주어진 조건하에서는 독립이 되는 경우를 뜻한다. 조건부 예시를 몇 가지 들어보자.

아래의 그림은 지휘과과 지휘관의 명령을 따르는 직원 A,B를 나타내고 있다.




먼저 A가 지휘관을 명령을 듣지 못하는 상태, 즉 지휘관의 명령을 알 수 없는 상태라고 가정해보자. 이 경우에는 B가 앞으고 가는 것을 보았을 때와 그렇지 않을 때 A가 앞으로 갈 확률은 다를 것이다. A가 B가 앞으로 가는 것을 본다면, '지휘관이 B에게 명령을 내렸겠군!'이라고 생각하고 앞으로 가게 될 것이다. 수식적으로는 다음과 같이 나라낼 수 있다.

$$P(\text{OfficerA} = \text{Go}|\text{OfficerB} = \text{Go}) \neq P(\text{OfficerA} = \text{Go})$$

이번에는 A가 지휘관의 명령을 들을 수 있는 상태라고 가정하고 같은 상황에 대한 확률을 구해보자. 지휘관의 명령을 들을 수 있다면 B가 앞으로 가든 말든 지휘관의 명량만 따라 움직이면 되기 때문에, A가 앞으로 갈 확률에는 차이가 없다.

$$P(\text{OfficerA} = \text{Go}|\text{OfficerB} = \text{Go}, \text{Commander} = \text{Go}) = P(\text{OfficerA} = \text{Go}|\text{Commander} = \text{Go})$$

이렇게 독립이 아닌 두 사건에 대해 특정 조건이 개입했을 때, 조건부 확률이 되는 관계를 조건부 독립이라고 한다. 조건 y에 대하여 조건부 독립인 사건 $x_1, ..., x_n$은 다음과 같이 표현한다.

$$(\forall x_1, x_2, y)$$

$$P(x_1|x_2, y) = P(x_1|y)$$

$$\therefore P(x_1, x_2|y) = P(x_1|y)P(x_2|y)$$

위의 공식을 사용하여 class conditional density의 항을 다음과 같이 변형할 수 있다. 이렇게 변경하면, 계산이 가능해지며, 쉬워질 것이다.

$$P(X = \{x_1, ..., x_d\}|Y = y) \rightarrow \prod_{i=1}^{d} P(X_i = x_i|Y = y)$$

### 3.2.2 Naive Bayes Classifier

이제 베이즈 분류기를 수식으로 나타낸 $f^*$에 변형된 수식을 적용해보면 다음과 같다.

$$f^* = \arg\max_{Y=y} P(X=x|Y=y)P(Y=y) \approx \arg\max_{Y=y} P(Y=y) \prod_{1 \le i \le d} P(X=x|Y=y)$$

특성 $x_1, ..., x_d$가 모두 관계를 맺고 있을 때 class conditional density를 구하기 위해서 필요한 파라미터의 개수는 $(2^d - 1)k$였다. 반면, 변형한 식에서 class conditional density를 구하기 위해 필요한 파라미터의 개수는 $d \times k$개이다. 위에서 했던 것 처럼, 특성이 6개, 클래스가 2개인 분류 문제를 위해 필요한 파라미터의 개수를 구해보면 (class prior를 구하기 위한 파라미터 개수는 $k-1$로 동일) $6 \times 2 \times 1 = 12$개가 된다.

하지만 현실세계에서는 특성들이 서로 조건부독립을 만족하지 않는 경우가 대부분이다. 파라미터의 개수를 줄이기 위한 가정이 말 그대로 너무 순진한(Naive, 나이브)한 가정이기 때문에 이 가정을 사용한 베이즈 분류기를 나이브 베이즈 분류기(Naive Bayes Classifier)라고 부른다. 나이브 베이즈 분류기는 비록 특수한 가정을 사용하였지만 분야에서는 다른 복잡한 모델보다도 더 좋은 성능을 내기도 한다.

## 3.3 Text Mining Application: Simple Sentiment Classification

예시 코드와 출력 결과가 아래의 코드에 잘 나와있으니 참고 부탁드립니다

나이브 베이즈 분류기



https://ratsgo.github.io/machine%20learning/2017/05/18/naive/