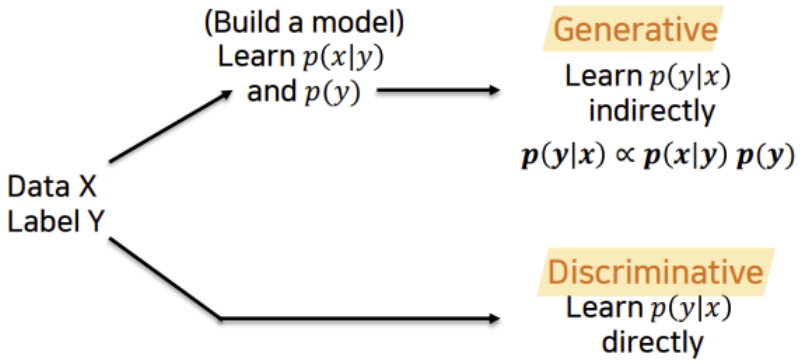
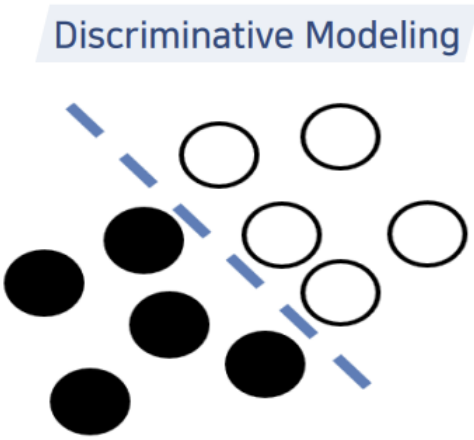
5강: Discriminative vs. Generative(GDA,Naive Bayes)



Learning algorithm은 $P(y=j|X) = E(I(y=j)|X)$를 구하는 방식에 따라

판별모델과 **생성모델**로 구별할 수 있다.

Discriminative Learning Algorithms



Class의 차이에 주목하여 데이터 X가 주어졌을 때 y가 j일 확률, 즉 $P(y=j|X)$(**conditional distribution**)를 **직접 학습**하여 분류를 진행하는 방식을 **discriminative learning algorithms**라고 한다.

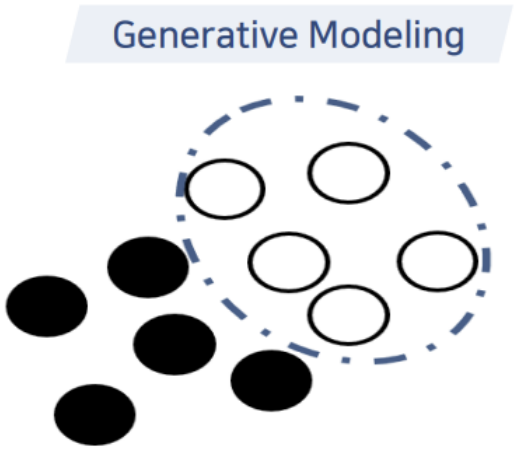
우리가 앞서 1주차 스터디에서 배운 로지스틱 회귀, 퍼셉트론 알고리즘의 경우 discriminative learning algorithm에 해당한다.

예시를 통해 이해해보자.우리가 코끼리(y=1)와 개(y=0)을 모델을 통해 구별해보고자 한다.

training dataset이 주어졌을 때, 로지스틱 회귀, 퍼셉트론 알고리즘은 코끼리와 개를 구분짓는 **결정경계(decision boundary)**를 찾을 것이다. (앞서 배운 경사하강법 st.의 **반복적인 방법**을 이용해서 말이다.)

그리고 새로운 데이터가 주어졌을때, 이를 분류하기 위해서 이 **데이터가 결정경계의 어떤 편에 있는지 보고**, 그에 따라 예측을 진행할것이다. ~~(발번역 미안해요~~ 하하하)~~

Generative Learning Algorithms



다른 접근 방식도 있다!

Class의 분포에 주목하여 $P(y=j|X)$를 $P(y=j)$와 $P(X|y=j)$를 통해 **간접적으로 구하는 방식**을 **generative learning algorithms**라고 한다.

앞서 들었던 개&코끼리 분류예시를 다시 들고 와보자면

각각 코끼리의 경우 코끼리가 어떻게 생겼는지/개의 경우 어떻게 생겼는지에 대한 separate한 모델을 build할 수 있다.

새로운 동물을 분류해야할 때는 새로운 동물을 코끼리 모델, 개 모델에 각각 대응 시켜보고 어느 동물에 가까운지 보는 것이다. ~~(히얼 컴즈 더 발번역 어게인~ 이해가 되길 바랍니다)~~

어떻게 $P(y=j|X)$를 $P(y=j)$와 $P(X|y=j)$를 통해 간접적으로 계산할 수 있을까?
당황하지 말고 외쳐 보자 도와줘요
베이즈~~

베이즈 정리와 사후 분포의 계산

베이즈 정리를 통해 사후 분포 $p(y|x)$를 정의하면 다음과 같다.

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

이 때, $p(x) = p(x|y=1)p(y=1) + p(x|y=0)p(y=0)$이고, 우리가 예측을 위해 $p(y|x)$을 계산한다는 것을 고려하면 분모는 우리의 관심사항이 아니라는 금방 알 수 있다. 이를 수식으로 표현해보자면 다음과 같다.

$$\begin{aligned}\arg\max_y\ p(y|x) &= \arg\max_y \frac{p(x|y)p(y)}{p(x)} \\ &= \arg\max_y\ p(x|y)p(y)\end{aligned}$$

$p(y|x)$를 간접적으로 계산하기 위해 모델링하는 $p(y)$,$p(x|y)$를 각각 (class) **priors, features**라고 한다.

> 앤드류 응씨는 생성모델의 예로 , **x값이 continuous한 경우로 GDA,** discrete한 경우 나이브 베이즈를 소개해주셨다.

> “ It’s very **quick to train**, and **non-iterative**.”

Gaussian Discriminant Analysis(GDA)

이 모델의 경우 $p(x|y)$가 **다변량 정규분포**를 따른다고 가정한다.

The multivariate normal distribution

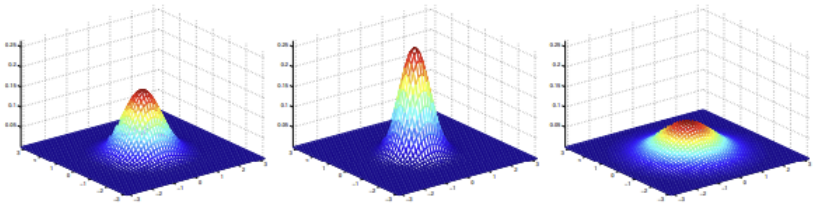
d차원의 다변량 정규분포는 파라미터 값으로 **mean vector** $\mu \in \mathbb{R}^d$ 와 **covariance matrix** $\sum \in \mathbb{R}^{d\times d}$ **(**$\sum \geq 0$ **,** symmetric, positive semi-definite**)**을 가진다.

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$
"$|\Sigma|$" : denotes the determinant of matrix Σ

- $E[X] = \int_x xp(x;\mu,\Sigma)dx = \mu$
- $Cov(Z) = E[(Z-E[Z])(Z-E[Z])^T] = E[ZZ^T] - (E[Z])(E[Z])^T$

$X \sim N(\mu,\Sigma)$라면, $Cov(X) = \Sigma$이다.

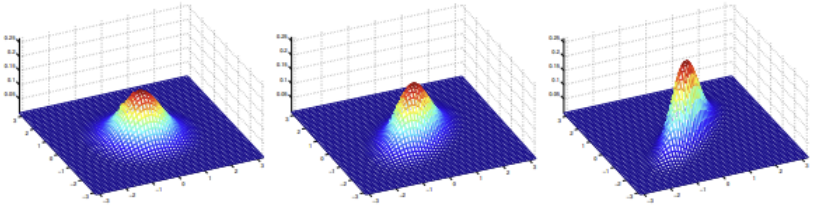
가우시안 분포의 몇가지 예를 살펴보자.



평균은 0이다.

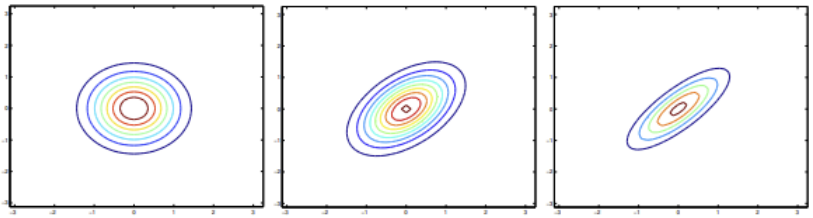
$$\Sigma = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Sigma = 0.6I = \begin{pmatrix} 0.6 & 0 \\ 0 & 0.6 \end{pmatrix} \qquad \Sigma = 2I = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

공분산이 작으면 분포가 압축되고, 커지면 분포가 퍼지는 것을 볼 수 있다.

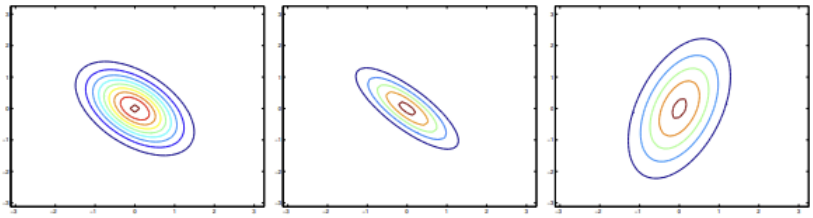


$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

비대각 성분(off-diagonal)이 증가할 수록 45°선을 따라 ($x_1 = x_2$) 밀도가 압축되는 것을 볼 수 있다. 등고선을 통해 확인하면 다음과 같다.



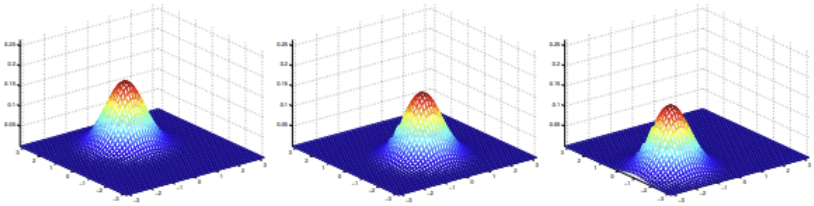
공분산을 다르게 해서 생기는 현상을 보여주는 마지막 예제까지 보자.



$$\Sigma = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}; \Sigma = \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}; \Sigma = \begin{pmatrix} 3 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

가장 왼쪽에서 중간까지 비대각 행렬이 줄어들어서 반대 방향으로 압축되는 모습을 볼 수 있다.

공분산은 $\Sigma = I$로 고정시키고, 평균을 이동시키면 다음 그림과 같다.



$$\mu = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \mu = \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}; \mu = \begin{pmatrix} 3 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

The gaussian discriminant analysis model

모델을 적어보자면 다음과 같다.

$$\begin{aligned} y &\sim Bernoulli(\phi) \\ x|y=0 &\sim N(\mu_0, \Sigma) \\ x|y=1 &\sim N(\mu_1, \Sigma) \end{aligned}$$

$$\begin{aligned} p(y) &= \phi^y(1-\phi)^{1-y} \\ p(x|y=0) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)) \\ p(x|y=1) &= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)) \end{aligned}$$

모델의 파라미터는 $\phi, \Sigma, \mu_0, \mu_1$이다.

(선형 결정 경계를 얻기 위해서 **공통의 covariance matrix** Σ를 가정한다.)

이를 바탕으로 data의 log-likelihood를 적어보자면 다음과 같다.

$$\begin{aligned} l(\phi, \mu_0, \mu_1, \Sigma) &= log \prod_{i=1}^{n} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\ &= log \prod_{i=1}^{n} p(x^{(i)}|y^{(i)};, \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi) \end{aligned}$$

Discriminative learning algorithms들의 경우 log-likelihood function이 $L(\theta) = \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, \theta)$꼴로 conditional function을 곱한 형식인데, Generative learning algorithms은 poesterior를 $p(y|x) \propto p(x|y)p(y)$를 이용해 간접적으로 구하기 때문에 joint probability꼴로 나타난다.

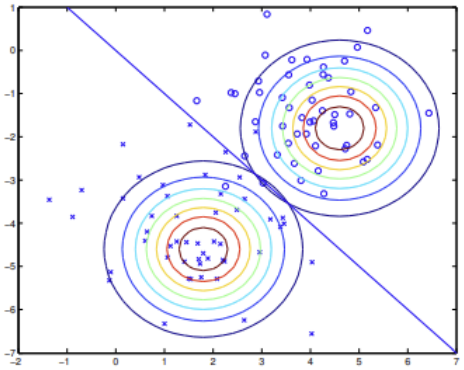
l을 최대화하는 파라미터의 추정값들은 다음과 같다.

$$\begin{aligned} \phi &= \frac{1}{n} \sum_{i=1}^{n} I\{y^{(i)} = 1\} \leftarrow \mathbb{R} \\ \mu_0 &= \frac{\sum_{i=1}^{n} I\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{n} I\{y^{(i)} = 0\}} \leftarrow \mathbb{R}^n \\ \mu_1 &= \frac{\sum_{i=1}^{n} I\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{n} I\{y^{(i)} = 1\}} \leftarrow \mathbb{R}^n \\ \Sigma &= \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T \leftarrow \mathbb{R}^{n \times n} \end{aligned}$$

- $\mu_0 = \frac{\text{sum of feature vectors for all the examples with y=0}}{\text{the number of examples y=0}}$
- $\mu_1 = \frac{\text{sum of feature vectors for all the examples with y=1}}{\text{the number of examples y=1}}$

파라미터를 추정하고 나면, 2개의 다변량 정규분포를 통해 선형의 결정 경계를 얻어낼 수 있다.

아래 그림은 알고리즘을 시각화한 것이다.



GDA and Logistic regression

GDA와 로지스틱 회귀는 흥미로운 관계에 있다~~(사실 흥미롭지 않다)~~

$p(y=1|x;\phi,\Sigma,\mu_0,\mu_1)$을 x에 대한 함수로 본다면 다음과 같은 형식으로 표현할 수 있다.

$$p(y=1|x;\phi,\Sigma,\mu_0,\mu_1)=\frac{1}{1+exp(-\theta^Tx)}$$
$$\text{where } \theta \text{ is some appropriate function of } \phi,\Sigma,\mu_0,\mu_1$$

잉...? 어디서 많이 본 형태인데....

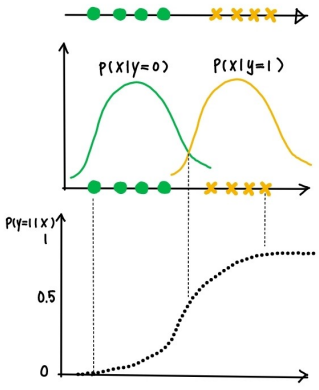
맞다. 로지스틱 회귀의 형태로 GDA 역시 표현됨을 알 수 있다.

~~이거 대마 중간고사 문제였다~~

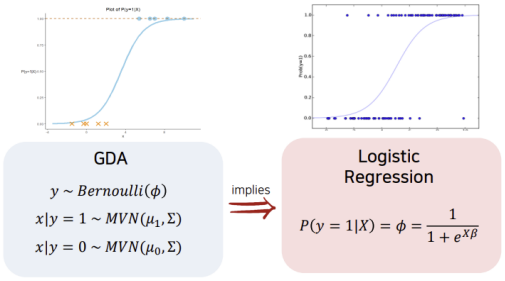
▼ 이에 대한 증명이 궁금하다면....

Appendix 2 Why GDA implies Logistic Regression.pdf

$p(y=1)$와 $p(x|y=1)$를 통해 $p(y=1|x)$를 계산하면 로지스틱 회귀와 같은 방식으로 확률이 계산됨을 확인 가능하다.

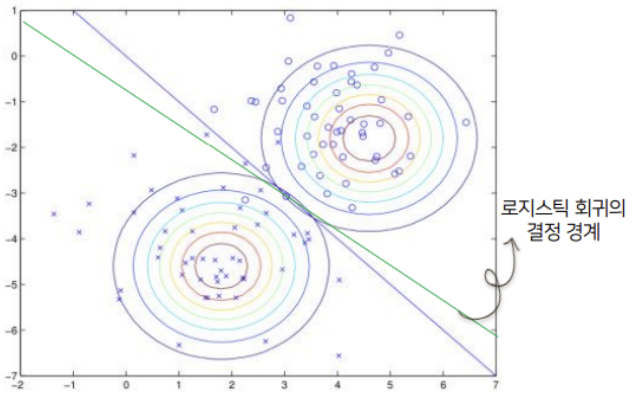


까꿍~ 시그모이드



주의! 역은 성립하지 않는다.

물론 선형경계가 **거의 일치하는 것이지 완전히 같은 것은 아니다**.



$p(x|y)$가 multivariate guassian (with shared Σ)을 따른다면, $p(y|x)$는 logistic function을 따른다.
하지만 역은 성립하지 않는다. 이는
GDA가 로지스틱 회귀보다 더 엄격한 모델 가정을 따르기 때문이다.

When GDA performs better(or efficiently)

- When these modeling assumptions are correct, then GDA will find better fits to the data, and is a better model.
- y와 X|y의 분포 가정만 맞다면 **학습 데이터 수가 적을 때** Logistic Regression보다 훨씬 좋은 성능을 보임
  ;
  data efficient (requires less training data to learn “well”)
- $p(x|y)$~ $N(\mu, \Sigma)$ → GDA asymptotically efficient

When Logistic Regression performs better(or efficiently)

- weaker assumption→ **robust** and less sensitive to incorrect modeling assumptions

예) $x|y=0 \sim Poisson(\lambda_0), x|y=1 \sim Poisson(\lambda_1)$라면
로지스틱 회귀가 더 잘 작동할 것이다.

- When the data is indeed non-Gaussian, then in the limit of large datasets, logistic regression will almost always do better than GDA.

▼ 데이터마이닝 수업을 다시 떠올려보자

1. LDA - linear한 결정경계

It was stated in the text that classifying an observation to the class for which (4.17) is largest is equivalent to classifying an observation to the class for which (4.18) is largest. Prove that this is the case. In other words, under the assumption that the observations in the kth class are drawn from a $N(\mu_k, \sigma^2)$ distribution, the Bayes classifier assigns an observation to the class for which the discriminant function is maximized.

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}. \tag{4.17}$$

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \tag{4.18}$$

Classification : $\log P(Y=K \mid X=x)$를 최대화하는 K

$$P_K(X) = \frac{\pi_K \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(X-\mu_K)^2\right)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(X-\mu_l)^2\right)}$$

$$\log P_K(X) = \log\left(\pi_K \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(X-\mu_K)^2\right)\right) + C_1$$

$$= \log \pi_K - \frac{1}{2\sigma^2}(X^2 - 2X\mu_K + \mu_K^2) + C_2$$

$$= \underbrace{X \cdot \frac{\mu_K}{\sigma^2} - \frac{\mu_K^2}{2\sigma^2} + \log(\pi_K) + C_3}_{\delta_K(X)\text{: Linear discriminant function}}$$

i) $\hat{\pi}_K = \frac{n_K}{n}$

ii) $\hat{\mu}_K = \sum_{y_i=K} x_i / n_K$

iii) $\hat{\Sigma} = \sum_{K=1}^{K} \sum_{y_i=K} (x_i - \hat{\mu}_K)(x_i - \hat{\mu}_K)^T / (n-K)$ * n_K : # of class K observations

2. QDA - 동일한 동분산을 가정하지 않는 경우

This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class-specific mean vector and a class specific covariance matrix. We consider the simple case where $p = 1$; i.e. there is only one feature.

Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, $X \sim N(\mu_k, \sigma_k^2)$. Recall that the density function for the one-dimensional normal distribution is given in (4.16). Prove that in this case, the Bayes classifier is *not* linear. Argue that it is in fact quadratic.

Hint: For this problem, you should follow the arguments laid out in Section 4.4.1, but without making the assumption that $\sigma_1^2 = \ldots = \sigma_K^2$.

① Assumption: Same as LDA except $\Sigma_k \neq \Sigma$

② $$P_k(X) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp\left(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2\right)}$$

$$\log P_k(X) = \log\left(\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)\right) + C_1$$

$$= \log \pi_k - \log \sigma_k - \frac{1}{2\sigma_k^2}(x^2 - 2x\mu_k + \mu_k^2) + C_2$$

$$= \log \pi_k - \log \sigma_k - \frac{\mu_k^2}{2\sigma_k^2} + x \cdot \frac{\mu_k}{\sigma_k^2} - x^2 \cdot \frac{1}{2\sigma_k^2} + C_2$$

③ Estimation of π_k, μ_k, Σ_k

i) $\hat{\pi}_k = n_k/n$, $\hat{\mu}_k = \sum_{y_i=k} x_i / n_k$

ii) $\hat{\Sigma}_k = \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (n_k - 1)$

3. LDA 활용문제

Suppose that we have a binary output variable Y (= 0 (Group 0) or 1 (Group1)) and a single discrete input variable X and we consider the linear discriminant analysis (i.e., we want to classify observations into either Group 0 or Group 1 based on X). For Group 0, X has the Poisson distribution with a parameter $\lambda = 10$ and X for Group 1 has the Poisson distribution with $\lambda = 20$. As prior probabilities, $P(Y = 0) = 0.4$ and $P(Y = 1) = 0.6$. Consider the linear discriminant analysis and answer the following questions:

(1) Find specific discriminant functions for Group 0 and 1, $\delta_k(x)$, $k = 0, 1$, respectively.

(2) Find a decision boundary point for Group 0 and 1 [**NOTE:** Since X has one-dimensional space, the decision boundary in this case is a point].

(1) Since $X|Y=k \sim Poisson(\lambda_k)$, $k=0,1$, $f_k(x)=\frac{\lambda_k^x e^{-\lambda_k}}{x!}$. Also, $P(Y=0)=\pi_0=0.4$ & $P(Y=1)=\pi_1=0.6$. Thus,

$$\begin{aligned}
\log P(Y=k|X=x) &= \log \frac{f_k(x)\pi_k}{\sum_{j=0}^{1} f_j(x)\pi_j} \\
&= \log(f_k(x)\pi_k) + C_1 \\
&= \log \pi_k + x\log\lambda_k - \lambda_k - \log x! + C_1 \\
&= \log \pi_k - \lambda_k + x\log\lambda_k + C_2 \\
&= \delta_k(x) + C_2.
\end{aligned}$$

Thus,

$$\begin{aligned}
\delta_0(x) &= \log \pi_0 - \lambda_0 + x\log\lambda_0 \\
&= \log(0.4) - 10 + x\log(10) \\
&= 2.303x - 10.916.
\end{aligned}$$

$$\begin{aligned}
\delta_1(x) &= \log \pi_1 - \lambda_1 + x\log\lambda_1 \\
&= \log(0.6) - 20 + x\log(20) \\
&= 2.996x - 20.511.
\end{aligned}$$

(2) To obtain the decision boundary point, we need to find x satisfying $\delta_0(x)=\delta_1(x)$.

$$\begin{aligned}
&\delta_0(x)=\delta_1(x) \\
\Rightarrow\ & 2.303x - 10.916 = 2.996x - 20.511 \\
\Rightarrow\ & (2.996-2.303)x = 20.511 - 10.916 \\
\Rightarrow\ & x = \frac{9.595}{0.693} \\
\Rightarrow\ & x = 13.846.
\end{aligned}$$

Thus, the decision boundary point is 13.846.

4. Comparison of LDA & Logistic regression

LDA	Logistic
MVN assumption	No assumption ⇒ better, robust
Qualitative inputs are not available (Theoretically)	Qualitative inputs are available

▼ ISLR에서 GDA

```
> library ( MASS )
> lda . fit <- lda ( Direction ~ Lag1 + Lag2 , data = Smarket ,
subset = train )
> lda . fit
Call :
lda ( Direction ~ Lag1 + Lag2 , data = Smarket , subset = train )
Prior probabilities of groups :
Down Up
0.492 0.508
Group means :
Lag1 Lag2
Down 0.0428 0.0339
Up -0.0395 -0.0313
Coefficients of linear discriminants :
LD1
Lag1 -0.642
Lag2 -0.514
> plot ( lda . fit )
```

```
> lda . pred <- predict ( lda . fit , Smarket .2005)
> names ( lda . pred )
[1] " class " " posterior " " x"
```

```
> lda . class <- lda . pred $ class
> table ( lda . class , Direction .2005)
Direction .2005
lda . pred Down Up
Down 35 35
Up 76 106
> mean ( lda . class == Direction .2005)
[1] 0.56
```

Naive Bayes

- 베이즈 정리를 바탕으로,
  각 변수들 간의
  조건부 독립을 가정하여 베이즈 정리의 복잡한 조건을 완화한 생성 모델
- 주로 **텍스트 데이터 분류**에 많이 사용됨

스팸메일을 받으면 이를 자동으로 분류하는 모델을 만든다고 가정하자.

이를 위해 어떤 단어들이 등장할 때, 그 메일이 스팸메일일 확률을 계산해보자.

인덱싱 된 단어 10000개

m개의 이메일

	스팸여부(y)	강의(x_1)	광고(x_2)	⋯	보험(x_{5714})	⋯
이메일 1	1(스팸)	0	1	⋯	1	⋯
이메일 2	0(정상)	1	0	⋯	1	⋯
⋮	⋮	⋮	⋮	⋯	⋮	⋮
이메일 m	0	1	1	⋯	0	⋯

이를 위해 나이브 모델은 m개의 이메일에서 10000개 단어의 출연 여부를 학습하게 된다.

로지스틱 회귀로 분류한다면, 설명변수가 많아 예측을 위해 많은 파라미터를 필요로 하고,
효과적으로 분류를 진행하기 어렵다.

$$\arg\max_y p(y|x) = \arg\max_y p(y)p(x|y) = \arg\max_y p(x,y)$$

Generative하게 접근한다면 좀 더 쉽고 효과적으로 분류할 수 있다!

$$P(X,y) = P(y)P(x_1|y)P(x_2|y,x_1)\cdots P(x_{10000}|x_{9999},..,x_2,x_1,y)$$

네...? 이게 어떻게 효과적인거죠? 너무 복잡한데요...

통계 이 XX 가만 안둬

단순히 y가 주어졌을 때, x_j끼리는 **독립이라는 조건부 독립을 가정**

$$P(X,y) = P(y)P(x_1|y)P(x_2|y)\cdots P(x_{10000}|y) = P(y)\prod_{i=1}^{n} P(x_j|y)$$

→ 베이즈 법칙의 조건을 완화하여 기존의 매우 복잡한 연산을 비교적 간단하게 바꿀 수 있음

Naive-Bayes의 분포가정

- $\phi_y = P(y=1) \rightarrow y \sim Bernoulli(\phi_y)$
- $\phi_{j|y=1} = P(x_j=1|y=1) = 1 \rightarrow (x_j|y=1) \sim Bernoulli(\phi_{j|y=1})$
- $\phi_{j|y=0} = P(x_j=1|y=0) = 1 \rightarrow (x_j|y=0) \sim Bernoulli(\phi_{j|y=0})$

$P(y)$와 $P(x_j|y)$를 계산하기 위해서 분포를 가정할 때,

y는 스팸메일 여부, $x_j|y$는 스펨메일 여부가 주어졌을 때 단어의 출연 여부이므로

y와 $x_j|y$의 분포 모두 **베르누이 분포**를 가정함

Likelihood를 최대화하는 방향으로 학습하면 다음과 같은 **MLE** 추정량을 얻어낼 수 있다.

- $\hat{\phi}_y^{MLE} = \frac{\sum_{i=1}^{m} I(y^{(i)}=1)}{m}$
- $\hat{\phi}_{j|y=1}^{MLE} = \frac{\sum_{i=1}^{m} I(x^{(i)}=1, y^{(i)}=1)}{\sum_{i=1}^{m} I(y^{(i)}=1)}$
- $\hat{\phi}_{j|y=0}^{MLE} = \frac{\sum_{i=1}^{m} I(x^{(i)}=1, y^{(i)}=0)}{\sum_{i=1}^{m} I(y^{(i)}=0)}$

각 파라미터의 추정치가 단순히

① 전체 이메일 수: m

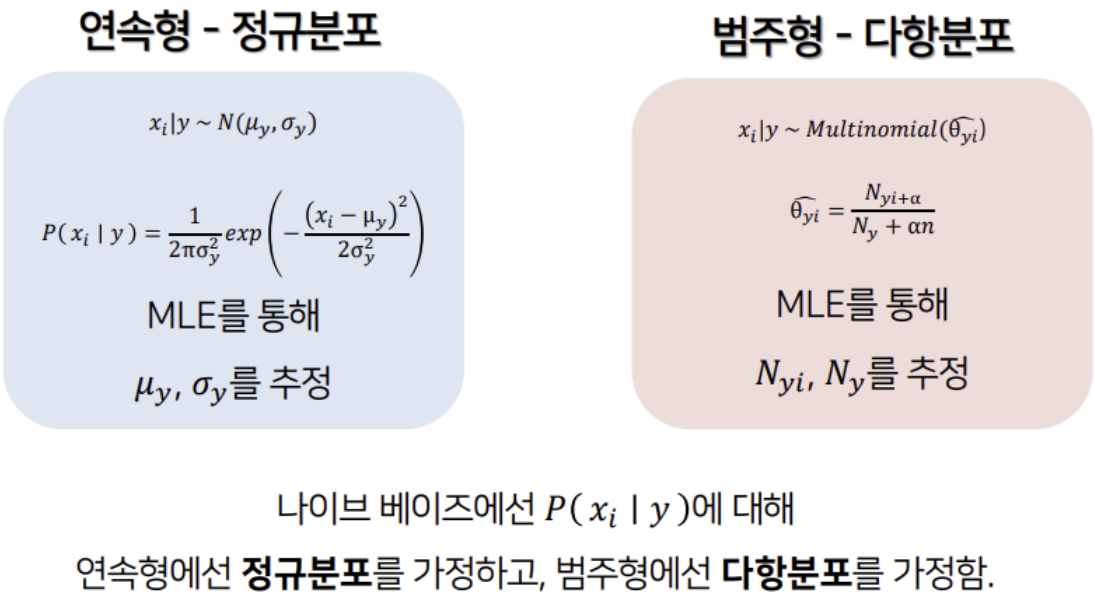
② 스팸메일 수 : $\sum_{i=1}^{m} I(y^{(i)} = 1)$

③ 스팸메일에서 해당 단어 j가 등장한 이메일 수 : $\sum_{i=1}^{m} I(x^{(i)} = 1, y^{(i)} = 1)$

④ 정상메일에서 해당 단어 j가 등장한 이메일 수 : $\sum_{i=1}^{m} I(x^{(i)} = 1, y^{(i)} = 0)$

이는 단순히 숫자만 세어주면 되기 때문에 **학습속도도 빠르고, 적용하기도 쉽다**는 장점을 가지게됨

▼ **Mixed Naive Bayes**



나이브 베이즈에선 $P(x_i \mid y)$에 대해

연속형에선 **정규분포**를 가정하고, 범주형에선 **다항분포**를 가정함.

Naive Bayes Classifier의 장점

장점 ①

MLE 추정량이 **단순 등장 빈도** 내지는 **확률 계산**으로 이루어짐

→ 단순히 데이터에서 특정 값이 나타난 횟수만 세어주거나
확률 계산만 하면 되어 모델이 **가볍고 학습·예측 속도가 빠름**

장점 ②

설명변수의 수가 많고 **이산형 변수**가 많을수록 **효과적**임

→ 데이터 셋의 설명변수 대부분을 차지하는 이산형 변수들을
최대한 활용하여 분류를 진행할 수 있게 됨.

Naive Bayes Classifier의 단점

한계 ①

변수들 간의 조건부 독립이라는 가정을 **만족하지 못하면**
모델의 **성능이 저하**됨.

→ 그러나 데이터 간 상관관계가 잘 나타나지 않아
충분히 모델이 성능을 발휘할 수 있을 것으로 기대됨

한계 ②

학습데이터에 **없는 값**이 들어왔을 때,
확률값이 0이 되어 분류가 **제대로 진행되지 않을 수**도 있음.

→ 라플라스 평활법(Laplace Smoothing)을
적용하여 이를 방지할 수 있음

연속형 변수와 이산형 변수가 섞여 있을 때

① 연속형 변수는 **정규분포**를 따른다고 가정

② 이산형 변수는 **다항분포**를 따른다고 가정

③ 이후 설명변수의 값이 주어졌을 때의 특정 라벨이 나타날 확률을 ①과 ②를 계산하고 곱해주어 **가장 나타날 확률이 높은 라벨로 예측**