Statistical Inference and Modeling

2102575

Suwichaya Suwanwimolkul[1]

[1]Electrical Engineering
Chulalongkorn University

Semester II, 2025

Lectures

- Lecture 1: Introduction
- Lecture 2: Model selection
- Lecture 3: Resampling
- Lecture 4: Linear regression
- Lecture 5: Robust regression
- Lecture 6: Classification
- Lecture 7: Logistic regression
- Lecture 8: Neural networks
- Lecture 9: K-nearest neighbor
- Lecture 10: Bayesian decision, LDA, QDA, and naive Bayes
- Lecture 11: Support vector machine
- Lecture 12: Tree-based methods
- Lecture 13: Bagging, random forecast, boosting
- Lecture 14: Unsupervised learning and clustering algorithms
- Lecture 15: Unsupervised learning and PCA
- Lecture 16: Gaussian mixture model clustering
- Lecture 17: DBSCAN clustering

9 ~~11~~. Naive Bayes, LDA, QDA

หน้าตาคล้าย Regression.
ตอน decision f^n

2102575

Suwichaya Suwanwimolkul[1]

[1]Electrical Engineering
Chulalongkorn University

Semester II, 2025

1 Table of Contents

2 Bayesian decision theory

3 Naive Bayes

4 Decision boundary

5 Discriminant function

6 Types of naive Bayes

7 Extend to multiple observations for training

8 Training
    Training of LDA

9 Quadratic discriminant analysis (QDA)

Generative Classifiers

1 Table of Contents

2 Bayesian decision theory

3 Naive Bayes

4 Decision boundary

5 Discriminant function

6 Types of naive Bayes

7 Extend to multiple observations for training

8 Training

9 Quadratic discriminant analysis (QDA)

Terminology in Bayesian theory

- Prior probability $p(x)$ is the knowledge we have *before* looking at an observed x.
- Class likelihood or class-conditional density $p(x|y)$ gives distribution information of features in each class (once we know the response variable belong to the class y)
- Evidence $p(x)$ is the marginal probability that a value x is observed (regardless of the class of Y) —can be computed using total probability.
- Posterior probability $p(y|x)$ is the likelihood of response after x is seen.

Bayes' rule: $$\text{Posterior} = \frac{\text{class likelihood} \times \text{prior}}{\text{evidence}}$$

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

* prior *

เรามักจะ neglect

เป้าหมายเราในการหา Loss function
ของ discriminative task !.

From a discriminative task to generative classifiers (naive Bayes)

We call something is "discriminative" if $\theta = \underset{\theta}{\operatorname{argmax}}\ p(x|\theta)$ where x is observation
unknown
Otherwise, that something is generative.

บทนี้ยังคงเป็น Parametric Model เพราะ Model เรายัง Assume Based on บาง distribution ที่เราตั้ง

A common goal in discriminative task is

$$\hat{c} = \underset{c \in \{1,2,\ldots,C\}}{\textbf{argmax}}\ p(x|y=c) = \text{likelihood}$$

which is to find the class c that has the highest likelihood...

Expectation, Variance ของ X ในแต่ละ $c \in C$

Learning parameters : $\overbrace{\mu^{x}_{y=c},\ \sigma^{2,x}_{y=c}}$; $\forall c \in \{1,2,\ldots,C\}$

Let's turn this problem into a **generative task**:

เรา consider มัน ว่าเป็น distribution ไร.

from → $\dfrac{p(x \mid y=c;\ \mu^{x}_{y=c},\ \sigma^{2,x}_{y=c})}{p(x)}$ เราละอยู่แล้ว. ⇓ generative & learning.

generative tasks.

$$\boxed{\hat{c} = \underset{c \in \{1,2,\ldots,C\}}{\textbf{argmax}}\ p(y=c|x)} = \frac{p(x|y=c)p(y=c)}{\boxed{p(x)}}.$$

สุดท้ายเราจะ neglect.

To solve for c, one needs to calculate the following terms:

- Prior: $p(y=c)$ can be computed using the fraction of each class from the training data.
- Likelihood $p(x|y=c)$ can be computed using the distribution of input features x, which can be partially parameterized by the condition of each class $y=c$ for $c \in \{1,2,\ldots,C\}$.
- This setting we are interested in identifying the class of a given input x.

```
a* = [1000, 2, 10, 1] ; a_old = -inf
for a in a* :
    if a > a_old : a_old = a
```

เรารับ x มาในเทอม likelihood เพื่อมา full-filled information.

Decision boundary

A discriminant function is used to find the decision boundary. The discriminant function can be defined as $p(y = c|x)$ or a function that is proportional to $p(y = c|x)$

$$g_c(x) = p(y = c|x) \quad \text{or} \quad g_c(x) \propto p(y = c|x)$$

Then, the optimization will turn into

เราต้องรู้ distribution ของ $f(x|y=c)$ ด้วย.

$$\hat{c} = \underset{c\in\{1,2,\ldots,C\}}{\textbf{argmax}} \; p(y = c|x) \propto p(x|y = c)p(y = c) \iff \hat{c} = \underset{c\in\{1,2,\ldots,C\}}{\textbf{argmax}} \; g_c(x).$$

The goal is to find c that gives the highest discriminant function which represents the posterior distribution.

Decision rule: suppose that $c \in \{0, 1\}$.

$\underset{c}{\operatorname{argmax}} \; p(x|y=c) \cdot p(y=c)$

- Classifying to $c = 0$

$$p(y = 0|x) > p(y = 1|x) \iff \boxed{\log p(x|y = 0) + \log p(y = 0) > \log p(x|y = 1) + \log p(y = 1)}$$
$$\iff \log g_{c=0}(x) > \log g_{c=1}(x)$$

bias bias

discriminant function

- Classifying to $c = 1$

$$p(y = 1|x) > p(y = 0|x) \iff \boxed{\log p(x|y = 1) + \log p(y = 1) > \log p(x|y = 0) + \log p(y = 0)}$$
$$\iff \log g_{c=1}(x) > \log g_{c=0}(x)$$

bias bias

Decision boundary

Decision rule: suppose that $c \in \{0, 1\}$.

- Classifying to $c = 0$

$$p(y=0|x) > p(y=1|x) \iff \boxed{\log p(x|y=0) + \log p(y=0) > \log p(x|y=1) + \log p(y=1)}$$
$$\iff \log g_{c=0}(x) > \log g_{c=1}(x)$$

discriminant function

- Classifying to $c = 1$

$$p(y=1|x) > p(y=0|x) \iff \boxed{\log p(x|y=1) + \log p(y=1) > \log p(x|y=0) + \log p(y=0)}$$
$$\iff \log g_{c=1}(x) > \log g_{c=0}(x)$$

ในทางคำนวณ เราจะคำนวณ $P(y_j = c) \approx \frac{1}{M} \sum_j I(y_j = c)$; $\forall c \in C$ ซึ่งเรามองว่ามันคือ bias term

แต่ในชีวิตจริง มันมีอยู่ละไอการที่เราไปเก็บข้อมูลมาแล้วมี Class บางกลุ่มที่มีเยอะเป็นพิเศษ.

ปล. ในตอนที่เรียน Logistic เราก็มีเหมือนกัน แต่จะอยู่ที่ β_0

Decision boundary

เป็น Classification เหมือนกัน,

ตารางเปรียบเทียบ: Logistic Regression vs Generative Models

หัวข้อเปรียบเทียบ	Logistic Regression	Linear Discriminant Analysis (LDA)	Quadratic Discriminant Analysis (QDA)	Naive Bayes
ประเภทโมเดล	**Discriminative**	**Generative**	**Generative**	**Generative**
เป้าหมายหลัก	หา $P(Y \mid X)$ โดยตรง (หาเส้นแบ่ง)	หา $P(X \mid Y)$ และ $P(Y)$ (จำลองการกระจายตัวของข้อมูล)	หา $P(X \mid Y)$ และ $P(Y)$ (จำลองการกระจายตัวของข้อมูล)	หา $P(X \mid Y)$ และ $P(Y)$ (จำลองการกระจายตัวของข้อมูล)
วิธีการเรียนรู้ (How to Learn)	**Iterative Optimization**	**Closed-form Analytic**	**Closed-form Analytic**	**Closed-form Analytic**
	ใช้ Maximum Likelihood Estimation (MLE) แก้สมการด้วย Gradient Descent หรือ Newton's Method เพื่อหาค่า β ที่ดีที่สุด (ไม่มีสูตรสำเร็จ)	คำนวณค่าสถิติจากข้อมูลโดยตรง: 1. Mean (μ_c) ของแต่ละคลาส 2. Shared Covariance (Σ) ร่วมกันทุกคลาส 3. Prior (π_c) จากสัดส่วนข้อมูล	คำนวณค่าสถิติจากข้อมูลโดยตรง: 1. Mean (μ_c) ของแต่ละคลาส 2. **Separate Covariance (Σ_c)** แยกกันแต่ละคลาส 3. Prior (π_c) จากสัดส่วนข้อมูล	คำนวณความน่าจะเป็นหรือพารามิเตอร์แยกรายฟีเจอร์ (Feature) ตามสมมติฐานว่า **ตัวแปรอิสระต่อกัน (Independent)**
รูปร่าง Decision Boundary	**Linear** (เส้นตรง/ระนาบ)	**Linear** (เส้นตรง/ระนาบ)	**Quadratic** (เส้นโค้ง/วงรี)	อาจเป็น Linear หรือ Non-linear ขึ้นอยู่กับ Distribution ที่เลือกใช้
ความซับซ้อน (Parameters)	ต่ำ (มีแค่ค่า β)	ปานกลาง (ต้องหา Σ หนึ่งก้อน)	สูง (ต้องหา Σ_c หลายก้อนตามจำนวนคลาส)	ต่ำมาก (เพราะคิดแยกฟีเจอร์ ไม่ต้องหา Covariance ระหว่างกัน)

1. Generative classifer
   - ↳ สร้างกราฟ PDF ของแต่ละ Class
   - ↳ สนหน้าตาของข้อมูลทุก Class แล้วค่อยหาแบบที่เหมาะสม.

2. Discriminative classifer
   - ↳ สร้างสมการเส้นแบ่ง (Decision Boundary) หรือ Rule ที่บอกว่า Y อยู่ Class ไหน

Discriminant function

What define $g_c(x)$?

$$\boxed{\begin{aligned} g_c(x) &\propto p(y=c|x) \\ &\propto p(x|y=c)p(y=c) \quad \star \end{aligned}}$$

discrete distribution
ของ Prob class ใดๆ.

1. Prior: $p(y=c)$ can be computed using the fraction of each class from the training data. It can be seen as the weight for each class.
2. Likelihood $p(x|y=c)$ can be computed in the following manners:

→ if we did not assume any distribution on $p(x|y=c)$, then the generative task is just a **Naive Bayes**.

→ if $p(x|y=c) := \begin{cases} q_c & \text{if } x=1 \\ 1-q_c & \text{otherwise} \end{cases}$

$$g_c(x) = \begin{cases} g_c(x) \cdot p(y=c) & ; x=1 \\ (1-g_c(x)) \cdot p(y=c) & ; x=0 \end{cases}$$

where x can only be either 0 or 1, then we call it **Bernoulli Naive Bayes**. Note that q_c is the corresponding event when $x=1$ and can be controlled by $y=c$.

→ if $p(x|y=c) := \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp - \frac{(x-\mu_c)^2}{2\sigma_c^2}$, then we call it **Gaussian Naive Bayes**. Notice that μ_c, σ_c^2 are the parameters associated with the distribution of input features x and can be controlled by $y=c$.

Example: Bernoulli Naive Bayes ☆

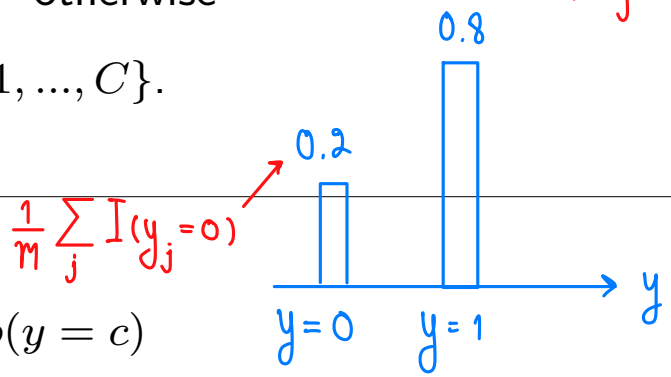
Suppose that $y \sim \mathcal{U}(0, C)$ and $p(x|y=c)$ is a Bernoulli distribution parameterized according to the following:

$f_Y(y) = \frac{1}{C}$

$$p(x|y=c) := \begin{cases} q_c & \text{if } x = 1 \\ 1 - q_c & \text{otherwise} \end{cases}$$

$\frac{1}{m}\sum_j I(y_j = 1)$

where the value of q_c depends on the value of $c \in \{0, 1, ..., C\}$.
How to formulate the discriminant function $g_c(\cdot)$?



$$\begin{aligned} g_c(x) &\propto p(x|y=c)p(y=c) \\ &\propto (q_c)^x (1-q_c)^{1-x} \frac{1}{C} \end{aligned}$$

$\longrightarrow$ discriminant f^n

To simplify the above formulation for computation, we can compute $g_c(x)$ in logarithmic scale:

$$\begin{aligned} g_c(x) &= \log (q_c)^x (1-q_c)^{1-x} \frac{1}{C} \\ &= x \log (q_c) + (1-x) \log (1-q_c) - \log C \\ &= x \left(\log (q_c) - \log (1-q_c)\right) + \log (1-q_c) - \log C \end{aligned}$$

Note $g_c(c)$ is a linear function in x.

Example: Gaussian Naive Bayes ☆ ตัวนี้เป็นฐานการไปหา LDA, QDA.

Suppose that $\boxed{y \sim \mathcal{U}(0, C)}$ and $p(x|y=c)$ is parameterized according to the following:

$f_Y(y) = \frac{1}{C}$

$$p(x|y=c) := \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp - \frac{(x-\mu_c)^2}{2\sigma_c^2}$$

where μ_c and σ_c depends on the value of $c \in \{0, 1, ..., C\}$. How to formulate $g_c(\cdot)$?

$$\begin{aligned} g_c(x) &\propto p(x|y=c)p(y=c) \\ &\propto \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp - \frac{(x-\mu_c)^2}{2\sigma_c^2} \frac{1}{C} \end{aligned}$$

To simplify the above formulation for computation, we can compute $g_c(x)$ in logarithmic scale:

ถ้าเรามองว่า $\sigma_1^2 = \sigma_2^2 = \sigma_c^2$
เวลาใส่ $X_j | y = c_1$, $X_j | y = c_2$
ใน $g_c(x_j)$ มันจะได้ $-\frac{1}{2\sigma_c^2} x_j^2$ เดียวกัน
เลยเหมือนไม่มีผลต่อการเปรียบเทียบ.

$$\begin{aligned} g_c(x) &= \log\left(\frac{1}{\sqrt{2\pi\sigma_c^2}}\right) + \log\left(\exp - \frac{(x-\mu_c)^2}{2\sigma_c^2}\right) + \log\left(\frac{1}{C}\right) \\ &= -\log\sqrt{2\pi\sigma_c^2} - \boxed{\frac{(x-\mu_c)^2}{2\sigma_c^2}} - \log C \\ &= \boxed{-\frac{1}{2\sigma_c^2}x^2 + \frac{\mu_c}{\sigma_c^2}x - \frac{\mu_c^2}{2\sigma_c^2} - \log\sqrt{2\pi\sigma_c^2} - \log C} \end{aligned}$$

LDA : $\beta_1 x + \beta_0$

→ QDA : $\beta_2 x^2 + \beta_1 x + \beta_0$

ไปดูใน proof ด้านหลัง.

If $\sigma_c^2 = \sigma^2$ for all $c \in \{1, 2, ..., C\}$, then the term $\frac{1}{2\sigma^2}x^2$ can be ignored, and thus, $g_c(x)$ will be a linear function in x. Otherwise, $g_c(x)$ is quadratic function in x.

Types of naive Bayes

A common goal in discriminative classification is

$$\hat{c} = \underset{c=1,2,\ldots,C}{\textbf{argmax}}\ p(y=c|x) = \frac{p(x|y=c)p(y=c)}{p(x)} = g_c(x|\mu_c, \Sigma_c)$$

Inference. The discriminant function $g_l(x|\mu_c, \Sigma_c)$.

- Naive Bayes: $\log g_c(x|\mu_c, \Sigma_c)$ is directly derived from

likelihood f^n

discrete pmf of Y

$$\boxed{g_c(x|\mu_c, \Sigma_c) \propto \prod_{j=1}^{m} p(x_j|y_j)p(y_j)}$$

- LDA : $\log g_c(x|\mu_c, \Sigma_c)$ is a linear discriminant function, *i.e.*,

Special Case ; $\sigma_c^2 = \sigma^2$ $\longrightarrow$ $\log g_c(x|\mu_c, \Sigma_c) \approx \beta_1 x + \beta_0$

- QDA : $\log g_c(x|\mu_c, \Sigma_c)$ is a quadratic discriminant function.

$$\log g_c(x|\mu_c, \Sigma_c) \approx \beta_2 x^2 + \beta_1 x + \beta_0$$

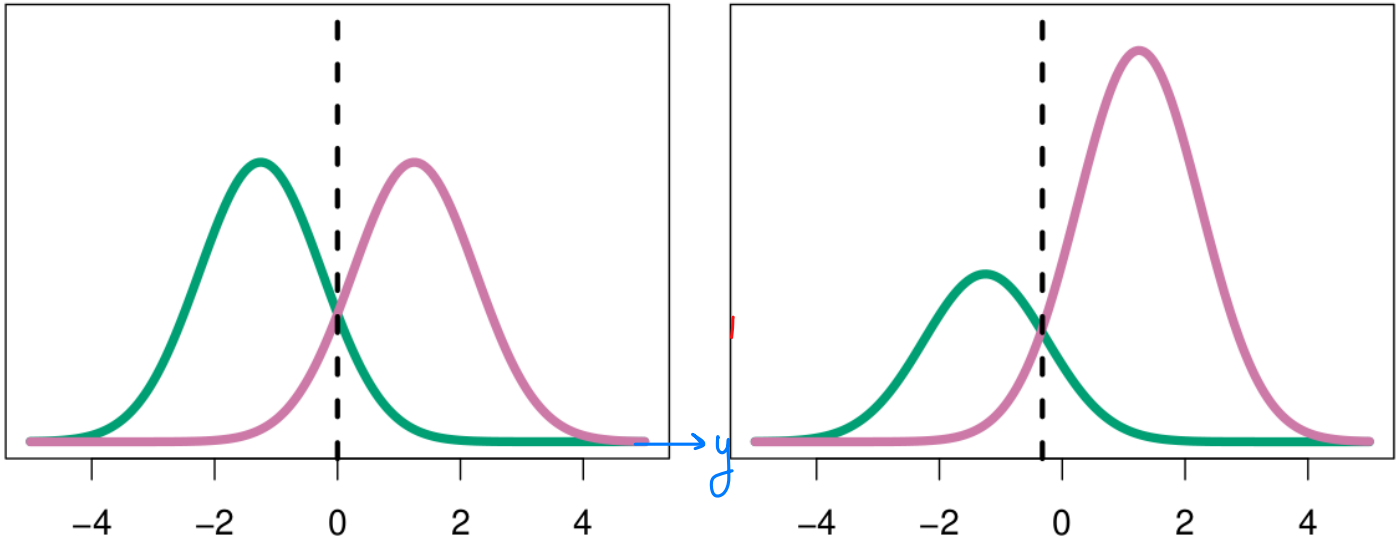
Side notes:

- The parameters associated with $x|y$, *e.g.*, q, μ, Σ, are estimated from sample mean and variance of the training data. → ไม่ต้องใช้ Optimization Problem แบบ Regression
- The estimated mean and variance depends on how x is divided into classes.

Bayes decision boundary: scalar case

An example of posterior densities for two cases of prior probabilities;

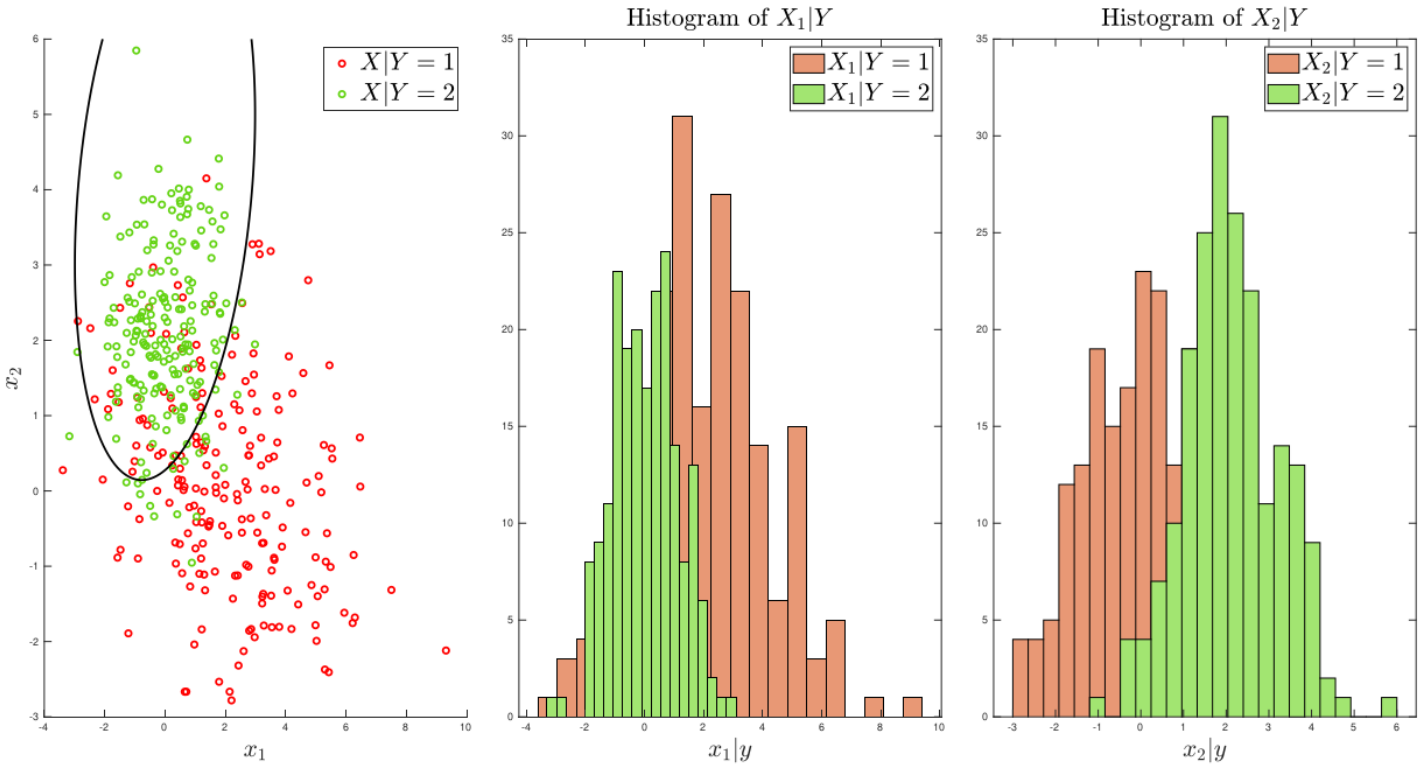
$P(Y = 1) = P(Y = 2) = 0.5$ $\quad$ $P(Y = 1) = 0.3, P(Y = 2) = 0.7$



- For a given test observation x, assign to the class for which the density is highest either class 1 or class 2.
- When the prior of class 2 is higher, the decision boundary is shifted to the left (more favor to class 2). $\hookrightarrow P(Y=2) > P(Y=1)$

Example: conditional 2D Gaussians

$$X|y=1 \sim \mathcal{N}\left(\begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}4 & -1\\-1 & 2\end{bmatrix}\right) \text{ and } X|y=2 \sim \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$



- Bayes decision boundary of Gaussian case leads to a quadratic function.
- What happen to the decision boundary as $p(y=1)$ increases ?

Extend to multiple observations for training

- Suppose that our training dataset contains m observation pairs of input features and labels (x_j, y_j) for $j = 1, ..., m$.
- Then, the posterior distribution is

มอง $x_j \in \mathbb{R}^1$ (ตัวเลขปกติ)

$$\boxed{\prod_{j=1}^{m} p(y_j|x_j)} \propto \prod_{j=1}^{m} p(x_j|y_j)p(y_j)$$

Therefore, the discriminant function for a training dataset can be formulated as follows:

$$g_c(x|\mu_c, \Sigma_c) \propto \prod_{j=1}^{m} p(x_j|y_j)p(y_j)$$

- **Multi-dimensional case.**

However, in most cases, x_j is an n-dimensional input features, *i.e.*, $x_j := [x_j^i]_{i\in[n]}$. Thus, we can write $p(x_j|y_j)$ as

$$p(x_j|y_j) = p([x_j^i]_{i\in[n]}|y_j)$$

$[x_j^i] = [x_j^1 \; x_j^2 \cdots x_j^n]$
= data sample j

Therefore, the discriminant function can be formulated as follows:

$$g_c(x|\mu_c, \Sigma_c) \propto \prod_{j=1}^{m} p([x_j^i]_{i\in[n]}|y_j)$$

Training example

- Training of LDA

Training of LDA: problem settings

$$\boxed{\sigma^2_{c=1} = \sigma^2_{c=2} \cdots = \sigma^2_{c} = \Sigma}$$ → แอบ Over simplify → High bias. (Underfit)

Assumption.

LDA applies **a Gaussian assumption with equal covariance** on $p(x|y=c)$.

- $p(x|y=c)$ is a Gaussian with mean μ_c and equal covariance Σ_c for $c = 1, ..., C$
- Let $p_c(x)$ denotes the posterior probability $\triangleq P(y=c|x) \propto \frac{p(x|y=l)p(y=l)}{p(x)}$:

$\pi_c = P(y=c)$

คำนวณทุกๆ คลาส c → $$p_c(x) = \frac{\exp\left(-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right)\pi_c}{\sum_{k=1}^{C}\exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_c^{-1}(x-\mu_k)\right)\pi_k}$$

- **Note:** the denominators are the same, regardless of the density and classes.

Inference.

Take the $\log p_c(x)$ and neglect the term $x^T\Sigma^{-1}x$, we predict the class c for which a discriminant function (defined below) is the highest among $c = 1, 2, ...C$

$$\text{Discriminant function: } g_c(x) = x^T\Sigma^{-1}\mu_c \boxed{- \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \log\pi_c}$$

$$= x^T\boldsymbol{\beta}_1 + \beta_0$$ → รูป form การตัดสินใจจะคล้ายๆ Linear Regression

Decision boundary of LDA: Training

Training.

- The decision boundary between a and b happens when the two classes have equal discriminant values.

$$
\begin{aligned}
& g_a(x) = g_b(x) \iff g_a(x) - g_b(x) = 0 \\
& \iff \left(x^T \Sigma^{-1} \mu_a - \frac{1}{2} \mu_a^T \Sigma^{-1} \mu_a + \log \pi_a\right) - \left(x^T \Sigma^{-1} \mu_b - \frac{1}{2} \mu_b^T \Sigma^{-1} \mu_b + \log \pi_b\right) = 0 \\
& \iff x^T \hat{\Sigma}^{-1} (\hat{\mu}_a - \hat{\mu}_b) - \frac{1}{2} (\hat{\mu}_a^T \hat{\Sigma}^{-1} \hat{\mu}_a - \hat{\mu}_b^T \hat{\Sigma}^{-1} \hat{\mu}_b) + \log \frac{\hat{\pi}_a}{\hat{\pi}_b} = 0
\end{aligned}
$$

- The decision boundary is linear in x.
- When prior densities are equal, the boundary is the midpoint of any two sample means of x associated with each pair of classes: $x = \frac{1}{2}(\hat{\mu}_a + \hat{\mu}_b)$. This can be derived from

$$
\begin{aligned}
x^T \hat{\Sigma}^{-1} (\hat{\mu}_a - \hat{\mu}_b) - \frac{1}{2} (\hat{\mu}_a^T \hat{\Sigma}^{-1} \hat{\mu}_a - \hat{\mu}_b^T \hat{\Sigma}^{-1} \hat{\mu}_b) &= 0 \\
x^T \hat{\Sigma}^{-1} (\hat{\mu}_a - \hat{\mu}_b) &= \frac{1}{2} (\hat{\mu}_a + \hat{\mu}_b)^T \hat{\Sigma}^{-1} (\hat{\mu}_a - \hat{\mu}_b)
\end{aligned}
$$

- This mean that μ_c for x associated with each class can be calculated from

$$\hat{\mu}_c = \text{sample mean of } X \text{ in class } c$$

- Then, the variance $\hat{\Sigma} = \frac{1}{m-C} \sum_{c=1}^{C} \sum_{j \in [m] | y_j = c} (x_j - \hat{\mu}_c)(x_j - \hat{\mu}_c)^T$.

Training phase: compute $p(y=c)$, $\hat{\mu}_c$, and $\hat{\Sigma}$

1. **Partition data**:

   Partition y_j for $j \in [m]$ according to its associated labels y_j: $\mathcal{Y}_c = \{y_j, j \in [m] | y_j = c\}$.
   For example: $\mathcal{Y}_{c=0} = \{y_j, j \in [m] | y_j = 0\}$, $\mathcal{Y}_{c=1} = \{y_j, j \in [m] | y_j = 1\}$, ...,
   $\mathcal{Y}_{c=C} = \{y_j, j \in [m] | y_j = C\}$.

   Partition x_j for $j \in [m]$ according to its associated labels y_j: $\mathcal{X}_c = \{x_j, j \in [m] | y_j = c\}$.
   For example: $\mathcal{X}_{c=0} = \{x_j, j \in [m] | y_j = 0\}$, $\mathcal{X}_{c=1} = \{x_j, j \in [m] | y_j = 1\}$, ...,
   $\mathcal{X}_{c=C} = \{x_j, j \in [m] | y_j = C\}$.
2. **Compute** $p(y=c)$ for all $c \in \{1, 2, ..., C\}$.
   $p(y=0) = \frac{|\mathcal{Y}_{c=0}|}{m}$, $p(y=1) = \frac{|\mathcal{Y}_{c=1}|}{m}$, ..., $p(y=C) = \frac{|\mathcal{Y}_{c=C}|}{m}$
3. **Compute** $\hat{\mu}_c$ for all $c \in \{1, 2, ..., C\}$.
   $\hat{\mu}_{c=0} = \frac{\sum_{x \in \mathcal{X}_{c=0}} x}{|\mathcal{X}_{c=0}|}$, $\hat{\mu}_{c=1} = \frac{\sum_{x \in \mathcal{X}_{c=1}} x}{|\mathcal{X}_{c=1}|}$, ..., $\hat{\mu}_{c=C} = \frac{\sum_{x \in \mathcal{X}_{c=C}} x}{|\mathcal{X}_{c=C}|}$
4. **Compute** $\hat{\Sigma}$...

$$\hat{\Sigma} = \frac{1}{m-C} \sum_{c=1}^{C} \sum_{x \in \mathcal{X}_c} (x - \hat{\mu}_c)(x - \hat{\mu}_c)^T$$

Inference phase

Given the input x, **discriminant function** is computed for all $c \in \{0, 1, 2, ..., C\}$:

$$g_c(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_c - \frac{1}{2} \hat{\mu}_c^T \hat{\Sigma}^{-1} \hat{\mu}_c + \log p(y=c)$$

Training of LDA: problem settings

$$\boxed{g_c(x) = x^T \Sigma^{-1} \mu_c - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c + \log \pi_c} \quad \text{— } (*)$$

1. การเรียนรู้ (Learning Phase) = "การหาค่าสถิติ 3 ตัว"

ในขั้นตอนการเรียนรู้ LDA จะทำหน้าที่เหมือน "นักสถิติ" ที่กวาดตาดูข้อมูลทั้งหมด แล้วจดบันทึกค่าสำคัญ 3 อย่างเก็บไว้เป็น "ความรู้" (Model Parameters) ครับ:

1. **หาความน่าจะเป็นตั้งต้น (Prior $\hat{\pi}_k$):**
   - **ทำยังไง:** นับจำนวนสมาชิกในแต่ละคลาส หารด้วยจำนวนข้อมูลทั้งหมด
   - **ตัวอย่าง:** มีข้อมูล 100 ตัว เป็นคลาส A 30 ตัว, คลาส B 70 ตัว
   - **ค่าที่จำ:** $\pi_A = 0.3, \pi_B = 0.7$
2. **หาค่าเฉลี่ยของแต่ละคลาส (Mean $\hat{\mu}_k$):**
   - **ทำยังไง:** เอาข้อมูล x ทั้งหมดที่เป็นคลาส A มาหาจุดกึ่งกลาง (Average) และทำแบบเดียวกันกับคลาส B
   - **ค่าที่จำ:** จุดศูนย์กลางของกลุ่ม A (μ_A) และจุดศูนย์กลางของกลุ่ม B (μ_B)
3. **หาความแปรปรวนร่วม (Pooled Covariance $\hat{\Sigma}$):**
   - **ทำยังไง:** คำนวณว่าข้อมูลมีการกระจายตัวกว้างแค่ไหนและมีรูปร่างรีหรือกลม
   - **จุดสำคัญของ LDA:** LDA จะบังคับว่า **"ทุกคลาสต้องใช้ทรงเดียวกัน"** ดังนั้นมันจะคำนวณ Covariance ของแต่ละคลาส แล้วเอามา **"เฉลี่ยรวมกัน"** (Weighted Average) ให้เป็นก้อนเดียวใช้ร่วมกันทั้งโมเดล
   - **ค่าที่จำ:** เมทริกซ์ Σ ก้อนเดียว

สรุปการ Learn: พอคำนวณ 3 ตัวนี้เสร็จ (π, μ, Σ) ถือว่า **"เทรนเสร็จแล้ว"** ครับ (เร็วกว่า Logistic Regression มากเพราะแค่บวกลบคูณหารจบเลย)

2. การบอกผลลัพธ์ (Output Phase) = "การเทียบคะแนน"

เมื่อมีข้อมูลใหม่ x เข้ามา LDA จะบอก Output ผ่านการคำนวณ **"Discriminant Score"** ($g_k(x)$) ครับ

วิธีการคำนวณ Output:

LDA จะเอา x ไปเข้าสมการเส้นตรง (Linear Equation) ของแต่ละคลาส เพื่อดูว่าคะแนนของคลาสไหนเยอะกว่ากัน

สมการคะแนนของคลาส k ($\delta_k(x)$):

$$\delta_k(x) = \underbrace{x^T \Sigma^{-1} \mu_k}_{\text{เหมือน Weight } \beta} - \underbrace{\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k}_{\text{เหมือน Bias/Intercept}}$$

$= g_c(x)$

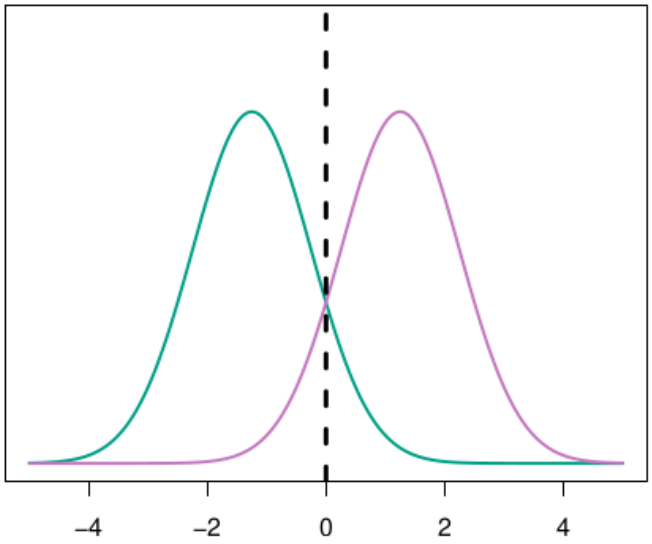
รูปแบบของ Output ที่เราจะได้:

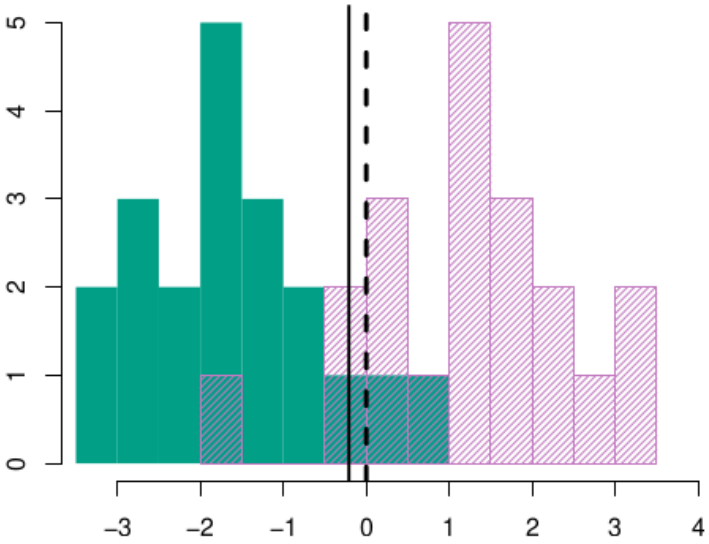
1. **Discriminant Score (คะแนนดิบ):**
   - จะได้ค่าตัวเลขออกมา 2 ค่า (ถ้ามี 2 คลาส) เช่น
     - Score(Class A) = 5.2
     - Score(Class B) = 3.8
2. **Predicted Class (คลาสที่ทำนาย):**
   - เลือกคลาสที่คะแนนสูงสุด → **ตอบ Class A**

$$\boxed{\hat{C} = \max_{c \in \mathbb{C}} \text{Score}}$$

3. **Posterior Probability (ความน่าจะเป็น):**
   - เราสามารถแปลง Score กลับเป็น % ได้โดยใช้ Softmax function
   - เช่น $P(y = A|x) = 0.8, P(y = B|x) = 0.2$

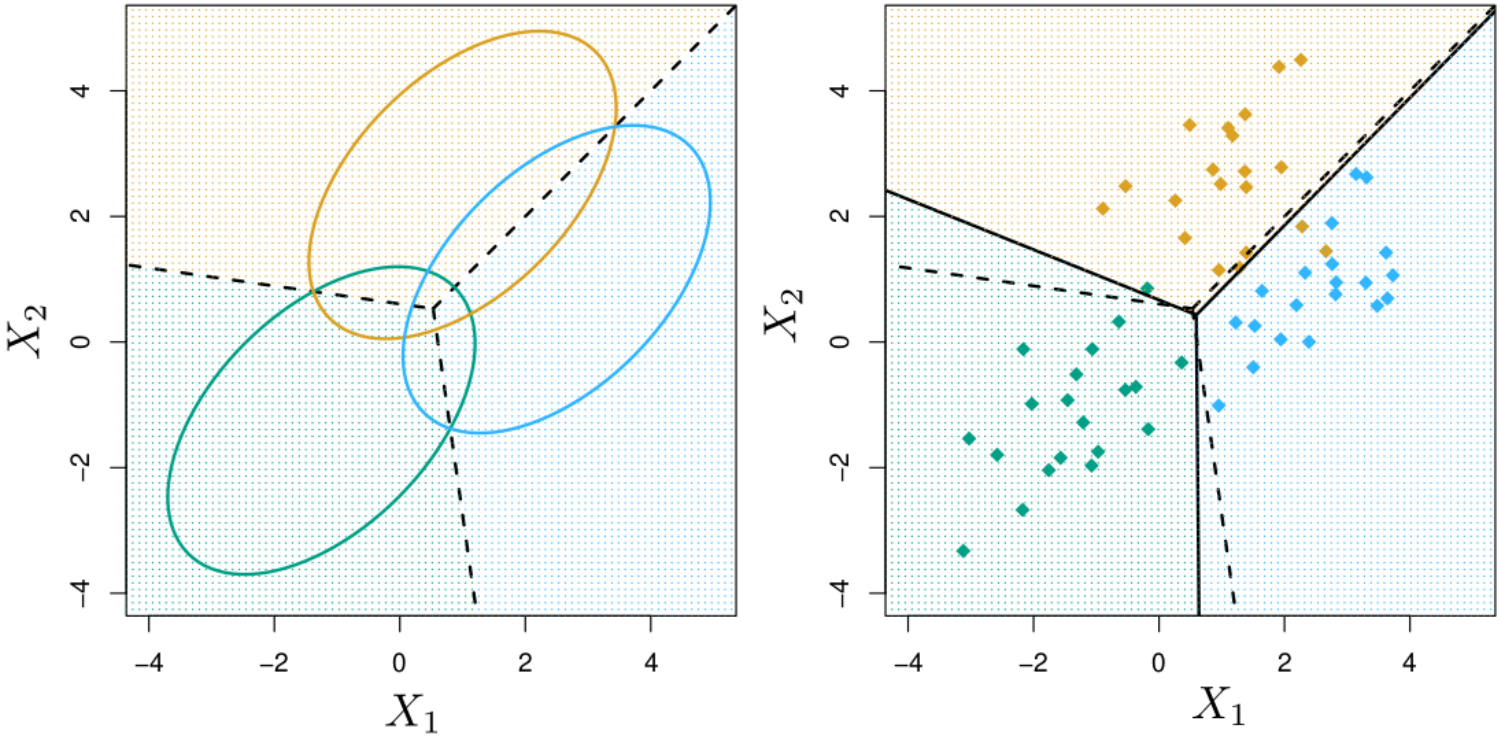
LDA decision boundaries for scalar x





- **Setting:** Let the true densities be Gaussian distributions: $P(x|y=0)$ with $\mu_1 = -1.25$ and $P(x|y=1)$ with $\mu_2 = +1.25$ and unit variances for both. Draw 20 observations from each class.
- One can use Bayes decision (left) is then $x = (\mu_1 + \mu_2)/2 = 0$, which is unknown in real-life.
- The area for which two densities overlapped is the classification error
- LDA decision boundary (solid line) is a little off from Bayes since it uses the sample mean computed from the training data.

LDA decision boundaries for a vector of x



- **Setting:** 3-class, true distribution is Gaussian with equal Σ, 20 samples for each class
- Bayes decision boundary and 95% confidence ellipsoid (use true μ_c, Σ).
- Solid lines are LDA decision boundaries $(\hat{\mu}_c, \hat{\Sigma})$. Dashed lines are Bayes.

Quadratic discriminant analysis (QDA)

$X \mid y=c \sim \mathcal{N}(\mu_c, \Sigma_c)$

Assumption.
QDA assumes observations in each class are Gaussian but each has its own covariance matrices — the posterior probability $p_c(x) \triangleq P(y=c|x)$ is

จุดแตกต่างหลักกับ LDA

$$p_c(x) = \left[\underbrace{\frac{\exp\left[-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1}(x-\mu_c)\right]}{(2\pi)^{n/2}(\det \Sigma_c)^{1/2}} \cdot \pi_c}_{P(X \mid y=c) \cdot P(y=C)} \right] / p(x)$$

Inference.
QDA follows Bayes classifier to perform the prediction, choosing the class l where

$$\text{Discriminant func.: } g_c(x) = -\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1}(x-\mu_c) - \frac{1}{2}\log\det \Sigma_c + \log \pi_c$$

is highest among $c = 1, 2, ..., C$; however, it uses estimates of μ_c, Σ_c.

- the decision boundary between class c and k is given by the set of x that $g_c(x) = g_k(x)$ — becoming quadratic function in x.
- QDA has more number of parameters to estimate — leading the chance of getting less bias but also higher variance than LDA.

Comparison between LDA and QDA

feature dimension = 2 → $x \in R^2$

Class 1 $\{(x_1, x_2)^{j \in M_1}\} \to (\mu_{x_1}^{c=1}, \mu_{x_2}^{c=1})$
Class 2 $\{(x_1, x_2)^{j \in M_2}\} \to (\mu_{x_1}^{c=2}, \mu_{x_2}^{c=2})$
Class 10 $\{(x_1, x_2)^{j = M_{10}}\} \to (\mu_{x_1}^{c=10}, \mu_{x_2}^{c=10})$
$\times 10$

$\pi_{c=1}$
$P_{c=2}$
$\pi_{c=10}$ $n^2 - n$.

Assume n feature dimension and C class

Prior $\pi_c \in R$ Mean $\mu_c \in R^n$

$O(n^3)$

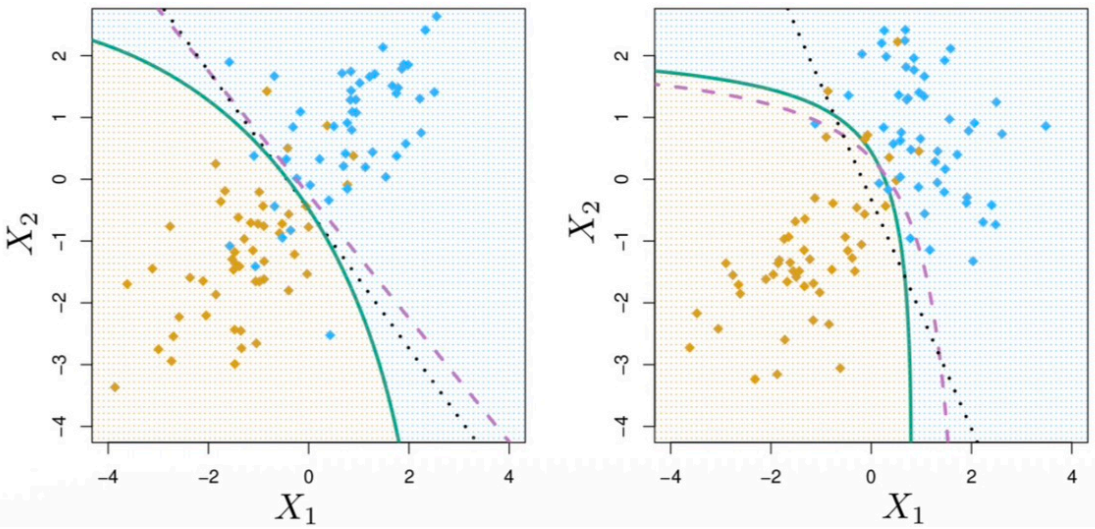
method	number of parameters	model property
LDA	~~nC~~ $C(1 + n)$	low flexibility, low variance
QDA	$n(n+1)C/2$	high flexibility, high variance

$= C\left(1 + n + \frac{(n^2 - n)}{2} + n\right)$

Comparison: Gaussian with *left*: common and *right*: different covariances.

prior π_c mean $\mu_c \in R^n$ Covarian → Symmetry matrix $\Sigma_c \in R^{n \times n}$

$\frac{n^2 - n}{2} + n$ — cov.



- *Left:* Bayes decision boundary is linear and accurately approx. LDA
- *Right:* now Bayes decision boundary is quadratic, and QDA is more accurate.

Comparison between LDA and QDA

จำนวน $\hat{\theta}_c = \{\hat{\underline{\mu}}_c, \hat{\Sigma}_c, \hat{\pi}_c\}$

① . หา $\hat{\mu}_c = [\hat{\mu}_{1c} \ \hat{\mu}_{2c} \ \cdots \ \hat{\mu}_{nc}]^T$

② . หา $\hat{\Sigma}_c = \begin{bmatrix} \sigma_1^{2,c} & \sigma_{12}^{c} & \sigma_{13}^{c} & \cdots & \sigma_{1n}^{c} \\ \vdots & \sigma_2^{2,c} & \vdots & & \vdots \\ \vdots & & \sigma_3^{2,c} & & \vdots \\ \sigma_{n1}^{2,c} & \cdots & & \ddots & \sigma_n^{2,c} \end{bmatrix}_{n \times n}$

③ . หา $\hat{\pi}_c$ (bias term)

เมื่อ C เพิ่มขึ้น จะทำให้ # parameters ใน QDA >> LDA ส่งผลให้มัน Lead ไปสู่การ "Overfit" ได้ง่าย.

LDA	QDA.
$C \cdot n$ ← # μ ในแต่ละ Class	$C \cdot n$
$1 + \binom{n}{2}$	$C\left(n + \binom{n}{2}\right)$
$\sigma_{ij} \mid i \neq j$	$\frac{n(n-1)}{2}$
C	C.

บางทีก็จะ C-1 เพราะหาแค่ C-1 class อีก class จะ $1 - \sum_{j=1}^{c-1} \hat{\pi}_j$

Augmented features in LDA vs QDA

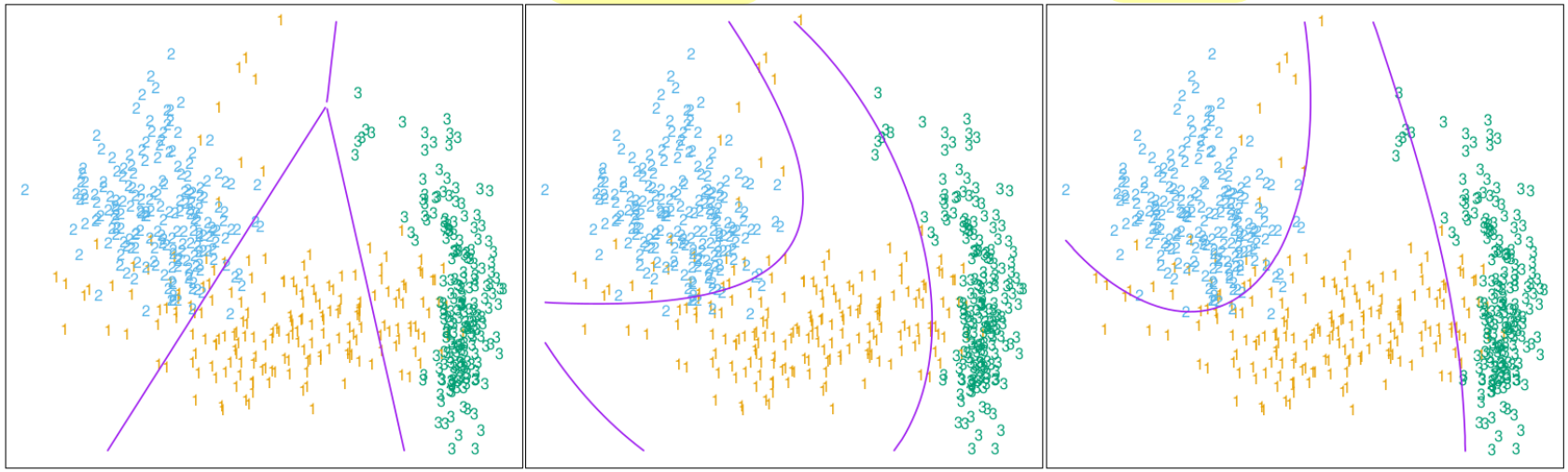
เป้าหมายคือลด Time Consume, Significantly Seperate

เราอาจแปลง X เป็นอย่างอื่นได้ เช่น Laplace, Fourier, ..., Polynomial

Feature spaces: original and its mapping by a quadratic function.

Original: $\{X_1, X_2, ..., X_n\}$, Augmented: $\{X_1, X_2, ..., X_n, X_1^2, X_2^2, ..., X_n^2\}$

LDA original feature space | LDA augmented feature space | QDA original feature space



- *Middle*: linear functions of LDA performed on the augemented space results in quadratic functions in the original space.
- While preferring QDA, the differences between QDA and LDA (middle) are small.

Comparing LDA, QDA, and Naïve Bayes

These classifiers have a common goal, *i.e.*, to assign an observation to the class that maximizes $P(y = c|x)$.

Let's set C as a *baseline* class; equivalently, we aim at assigning an observation that maximize the followings, for $c = 1, ..., \mathsf{K}$,

$$\log\left(\frac{P(y = c|x)}{P(y = C|x)}\right).$$

Comparing LDA, QDA, and Naïve Bayes: LDA

LDA: predictors within each class are drawn from a multivariate normal density with class-specific mean and **shared** co-variance matrix.

$$
\begin{aligned}
\log\left(\frac{P(y=c|x)}{P(y=C|x)}\right) &= \log\left(\frac{\pi_c f_c(x)}{\pi_C f_C(x)}\right) \\
&= \log\left(\frac{\pi_c \exp\left(-\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c)\right)}{\pi_C \exp\left(-\frac{1}{2}(x-\mu_C)^T\Sigma^{-1}(x-\mu_C)\right)}\right) \\
&= \log\left(\frac{\pi_c}{\pi_C}\right) - \frac{1}{2}(\mu_c+\mu_C)^T\Sigma^{-1}(\mu_c-\mu_C) \\
&\quad + x^T\Sigma^{-1}(\mu_c-\mu_C) \\
&= a_c + \sum_{i=1}^{n} b_{ci}x_i
\end{aligned}
$$

where $a_c = \log\left(\frac{\pi_c}{\pi_C}\right) - \frac{1}{2}(\mu_c+\mu_C)^T\Sigma^{-1}(\mu_c-\mu_C)$ and b_{ci} is the ith component of $\Sigma^{-1}(\mu_c-\mu_C)$. Hence LDA, like ~~logistic~~ linear regression, has the log odds of the posterior probabilities is linear in x.

Comparing LDA, QDA, and Naïve Bayes: QDA

QDA has the log odds of the posterior probabilities is quadratic in x.

$$\log\left(\frac{P(y=c|x)}{P(y=C|x)}\right) = a_c + \sum_{i=1}^{n} b_{ci}x_i + \sum_{p=1}^{n}\sum_{i=1}^{n} c_{cip}x_i x_p$$

where a_c, b_{ci}, c_{cip} are functions of $\pi_c, \pi_C, \mu_c, \mu_C$ and Σ.

Comparing LDA, QDA, and Naïve Bayes: Naïve Bayes

Naïve Bayes: assume conditional independency among features given a label k.

$$
\begin{aligned}
\log\left(\frac{P(y=c|x)}{P(y=C|x)}\right) &= \log\left(\frac{\pi_c f_c(x)}{\pi_C f_C(x)}\right) \\
&= \log\left(\frac{\pi_c \prod_{i=1}^{n} f_{ci}(x_i)}{\pi_C \prod_{i=1}^{n} f_{Cj}(x_i)}\right) \\
&= \log\left(\frac{\pi_c}{\pi_C}\right) + \sum_{i=1}^{n} \log\left(\frac{f_{ci}(x_i)}{f_{Ci}(x_i)}\right) \\
&= a_c + \sum_{i=1}^{p} g_{ci}(x_i)
\end{aligned}
$$

Independent.

where $a_c = \log\left(\frac{\pi_c}{\pi_C}\right)$ and $g_{ci}(x_i) = \log\left(\frac{f_{ci}(x_i)}{f_{Ci}(x_i)}\right)$

Softwares

Methods	Python (scikit-learn)
Logistic regression	`linear-model.LogisticRegression`
Naïve Bayes	`naive-bayes`
kNN	`neighbors.KNeighborsClassifier`
LDA, QDA	`discriminant-analysis`

References I

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[2] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

[3] Jitkomut Songsiri. "Statistical Learning". In: in EE575 Random Processes for EE. Chulalongkorn University, 2022. Chap. 8. Introduction to Classification. URL: http://jitkomut.eng.chula.ac.th/ee575.html.

Example: conditional 2D Gaussians $(n=2)$

Proof Decision Boundary ของ 2D Gaussians!

covariance matrix of X

ให้ $X \mid y=c \sim \text{Normal}(\underline{\mu}_c, \Sigma_c)$ โดย $\underline{\mu}_c = \begin{bmatrix} \mu_{1c} \\ \vdots \\ \mu_{nc} \end{bmatrix}_{n\times 1}$ และ $\Sigma_c = \begin{bmatrix} \sigma_1^{2,c} & \sigma_{21}^{c} & \cdots & \sigma_{n1}^{c} \\ \sigma_{12}^{c} & \sigma_2^{2,c} & & \vdots \\ \vdots & \vdots & \ddots & \\ \sigma_{1n}^{c} & \cdots & & \sigma_n^{2,c} \end{bmatrix}_{n\times n}$

โดย $\mu_{ic} = \dfrac{\text{ค่า } X \text{ แยกตาม features } i, \text{ Class } c}{\text{จำนวนค่า } X \text{ features } i, \text{ Class } c}$; $\forall i, c$

$\sigma_i^{2,c}, \sigma_{ij\,(i\neq j)}^{c}$ ก็คำนวณตามปกติ โดยแยกตาม c

จาก $g_c(X) = \log[P(X \mid Y=c)] + \log \pi_c$

$$= \log\left\{\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_c|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\underline{X}-\mu_c)^T\Sigma_c^{-1}(\underline{X}-\mu_c)\right]\right. + \log \pi_c$$

$$= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log(\det\Sigma_c) - \frac{1}{2}(\underline{X}-\mu_c)^T\Sigma_c^{-1}(\underline{X}-\mu_c) + \log \pi_c$$

$$= -\frac{1}{2}(\underline{X}-\mu_c)^T\Sigma_c^{-1}(\underline{X}-\mu_c) - \frac{1}{2}\log(\det\Sigma_c) - \frac{n}{2}\log(2\pi) + \log \pi_c$$

Example: conditional 2D Gaussians

$$g_c(X) = -\frac{1}{2}(\underline{X}-\mu_c)^T \Sigma_c^{-1}(\underline{X}-\mu_c) - \frac{1}{2}\log(\det \Sigma_c) - \frac{n}{2}\log(2\pi) + \log \pi_c$$

$$= -\frac{1}{2}\left(\underline{X}^T \Sigma_c^{-1} \underline{X} + 2\mu_c^T \Sigma_c^{-1} \underline{X} + \mu_c^T \Sigma_c^{-1} \mu_c\right) - \frac{1}{2}\log(\det \Sigma) - \frac{n}{2}\log(2\pi) + \log \pi_c$$

$$\boxed{g_c(X) = -\frac{1}{2}\underline{X}^T \Sigma_c^{-1} \underline{X} \underbrace{- \mu_c^T \Sigma_c^{-1} \underline{X}}_{\text{linear term}} \underbrace{- \frac{1}{2}\mu_c^T \Sigma_c^{-1} \mu_c - \frac{1}{2}\log(\det \Sigma_c) - \frac{n}{2}\log(2\pi)}_{\text{constant}} + \log \pi_c}$$

quadratic $A = -\frac{1}{2}\Sigma_c^{-1}$

linear term $B = -\mu_c^T \Sigma_c^{-1}$

constant C

Base on pmf.

→ Decision Rule : $g_1(\underline{X}) = g_2(\underline{X})$

$$\boxed{\underline{X}^T(A_1 - A_2)X + (B_1 - B_2)\underline{X} + (C_1 - C_2) = 0}$$

$$\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}$$

ถ้า $\underbrace{\Sigma_1 = \Sigma_2 = \dots = \Sigma_c = \Sigma}$

Class ทุก Class มี covariance matrix เท่ากัน

⇒ เทอมนี้จะถูกตัดออก ⇒ LDA

Example: conditional 2D Gaussians

<u>กรณีอื่น</u> $\Sigma_c = \sigma_c^2 \cdot I_n$ (ทุก features มัน Independent หมด)

$$\rightarrow (\sigma_1^2 - \sigma_2^2)\underline{X}^T\underline{X} - \left(\frac{\mu_1^T}{\sigma_1^2} - \frac{\mu_2^T}{\sigma_2^2}\right)\underline{X} - \frac{1}{2}\left(\frac{\mu_1^T\mu_1}{\sigma_1^2} - \frac{\mu_2^T\mu_2^2}{\sigma_2^2}\right)$$

$$- \frac{1}{2}\log\left(\sigma_1^2/\sigma_2^2\right) - \frac{n}{2}\log(2\pi) + \log\left(\pi_1/\pi_c\right) = 0.$$

$$g_c(x|\mu_c, \Sigma_c) \propto \prod_{j=1}^{m} p([x_j^i]_{i\in[n]}|y_j)$$

หน้าที่เราคือเราทำ MLE เพื่อหา $\hat{\theta}$ ที่เหมาะสม.

X_i

$$= \sum_{j=1}^{m} \log[P(X_j^1, X_j^2, \ldots, X_j^n | y_j)]$$

$$= \sum_{j=1}^{m}\left(\sum_{i=1}^{n} \log[p(x_j^i | y_j)]\right)$$

Conditional Independent.

จากนี้เราก็ไปหา jPDF ของแต่ละ feature มา.

~~Example: conditional 2D Gaussians~~ แนวข้อสอบ Midterm.

Part 1. 16th Feb

1. Model selection
2. Validation techniques
3. Linear regression
4. Outlier analysis

AIC, BIC

ถ้าเห็นการทดลองนี้คิดว่าเป็น Hold, K-Fold,... ข้อดี-ข้อเสีย

→ Derive Linear Regression แปลกๆ (แบบที่เรียนเราจะ correlate $(w, \hat{y})$)

ดู β เป็นไง.
การ Normalize X

→ แยกออกมั้ยเป็น cross-validate แบบไหน

→ Concept Base, อ่าน P-value ได้มั้ย, Non Para vs Para?

สมมติว่าให้ H-matrix มา
บอกได้มั้ยว่า High-Lev, outlier.

Part 2. 18th Feb

1. Evaluation for binary classification
2. Evaluation for multiclass classification
3. Generative classifiers: LDA/QDA/Naive Bayes.

→ Confusion Matrix Micro, Macro ยังไง

→ Derive LDA/QDA/Naive ⇝ discriminant f^n

⇓

Gaussian, Naive Bayes เป็นไงทำได้มั้ย.

→ Calculate เยอะ.

You can bring

1. One handwritten-A4 cheat sheet for each part.
2. You will need to write your answer with pen. So, don't forget to also bring the liquid eraser.
3. A calculator.