Classification Part I

Loss, Bayes Classifier, KNN, Logistic, Evaluation Metrics

송 준

고려대학교
통계학과 / 융합데이터과학 대학원

Review of Regularization

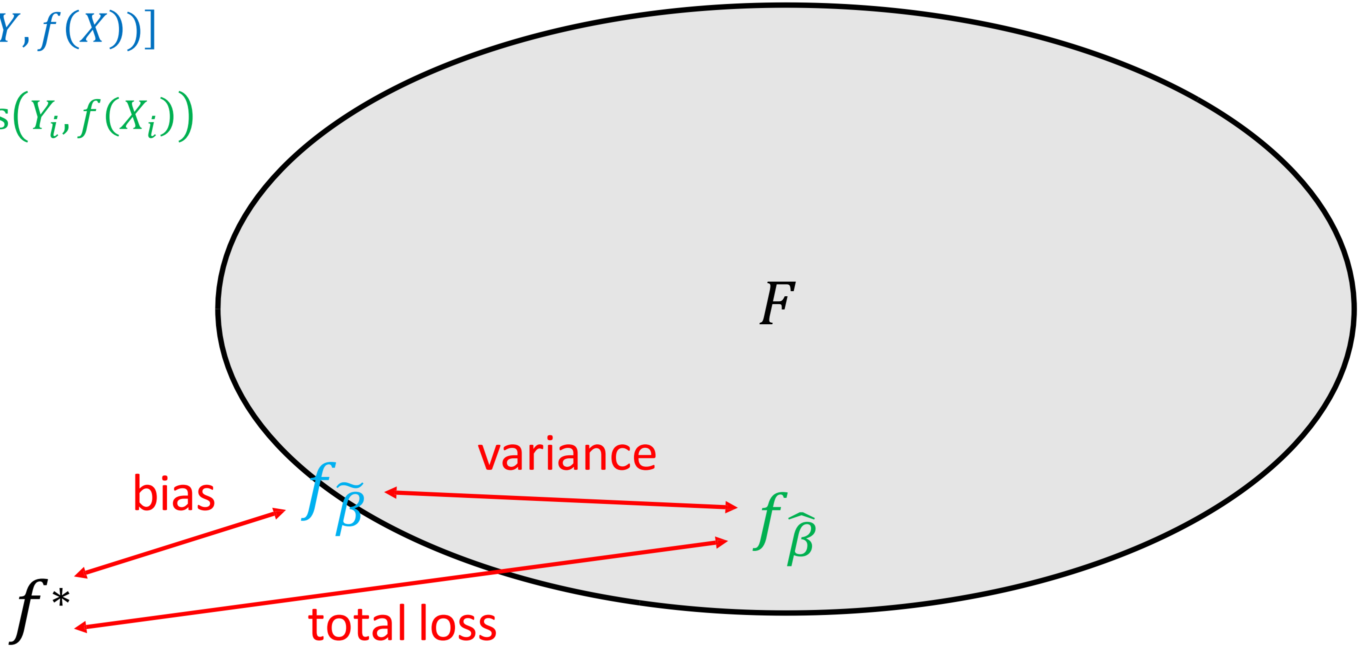
Bias-Variance Tradeoff

Ideal goal: Construct prediction rule $f^*: \mathcal{X} \to \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{\boldsymbol{f}} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f_n}} = \arg\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



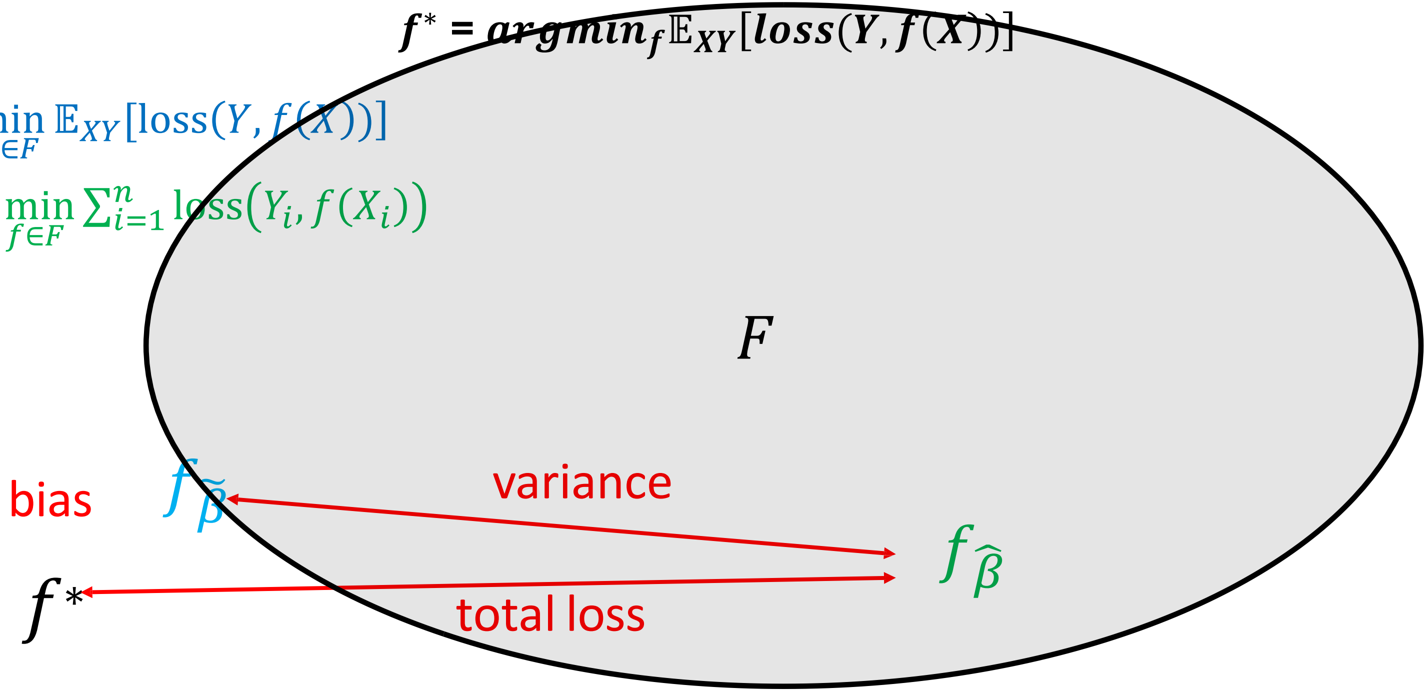
Bias-Variance Tradeoff(Overfitting)

Ideal goal: Construct prediction rule $f^* : \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{\boldsymbol{f}} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f_n}} = \arg\min_{f \in F} \sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



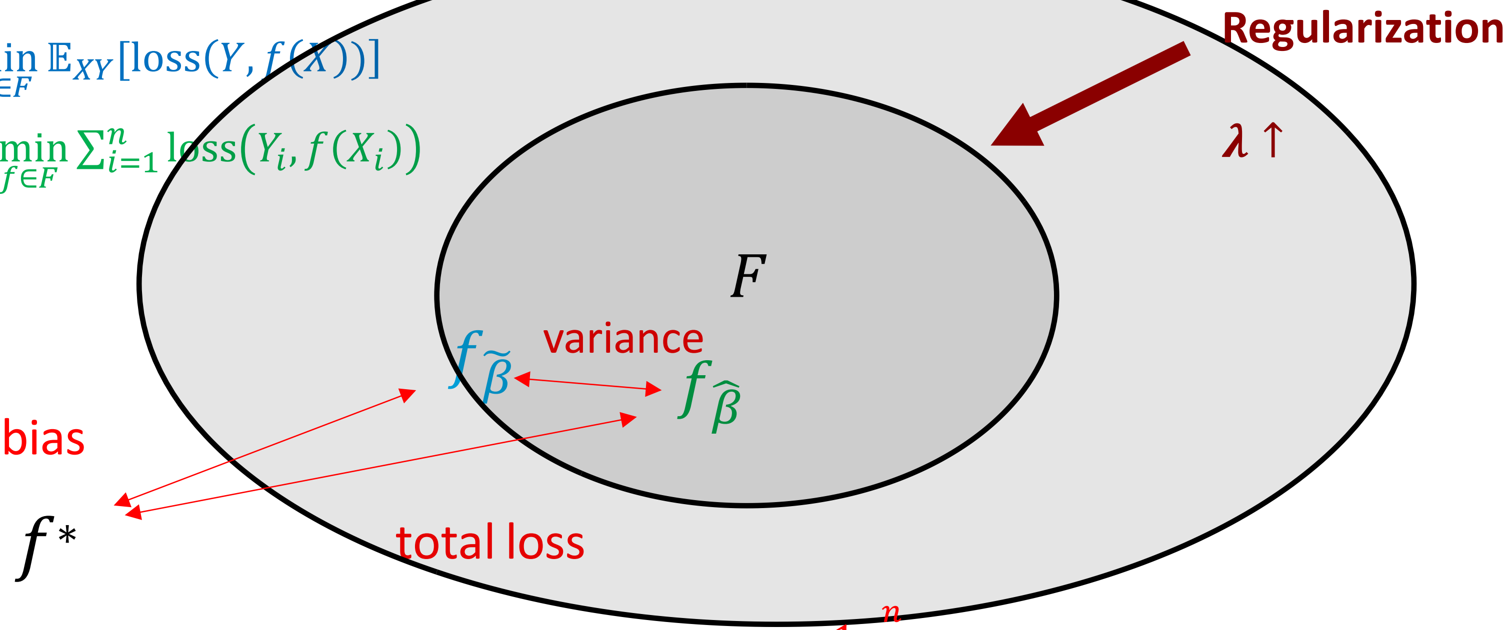
Bias-Variance Tradeoff(Regularization)

Ideal goal: *Construct prediction rule* $f^*: \mathcal{X} \rightarrow \mathcal{Y}$

$$\boldsymbol{f^* = argmin_f \mathbb{E}_{XY}[loss(Y, f(X))]}$$

$$\tilde{\boldsymbol{f}} = \arg\min_{f \in F} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

$$\widehat{\boldsymbol{f}_{\boldsymbol{n}}} = \arg\min_{f \in F} \textstyle\sum_{i=1}^{n} \text{loss}\big(Y_i, f(X_i)\big)$$



$$L(\beta; Z) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta^T x_i)^2 + \lambda \cdot \|\beta\|_p$$

Tuning λ

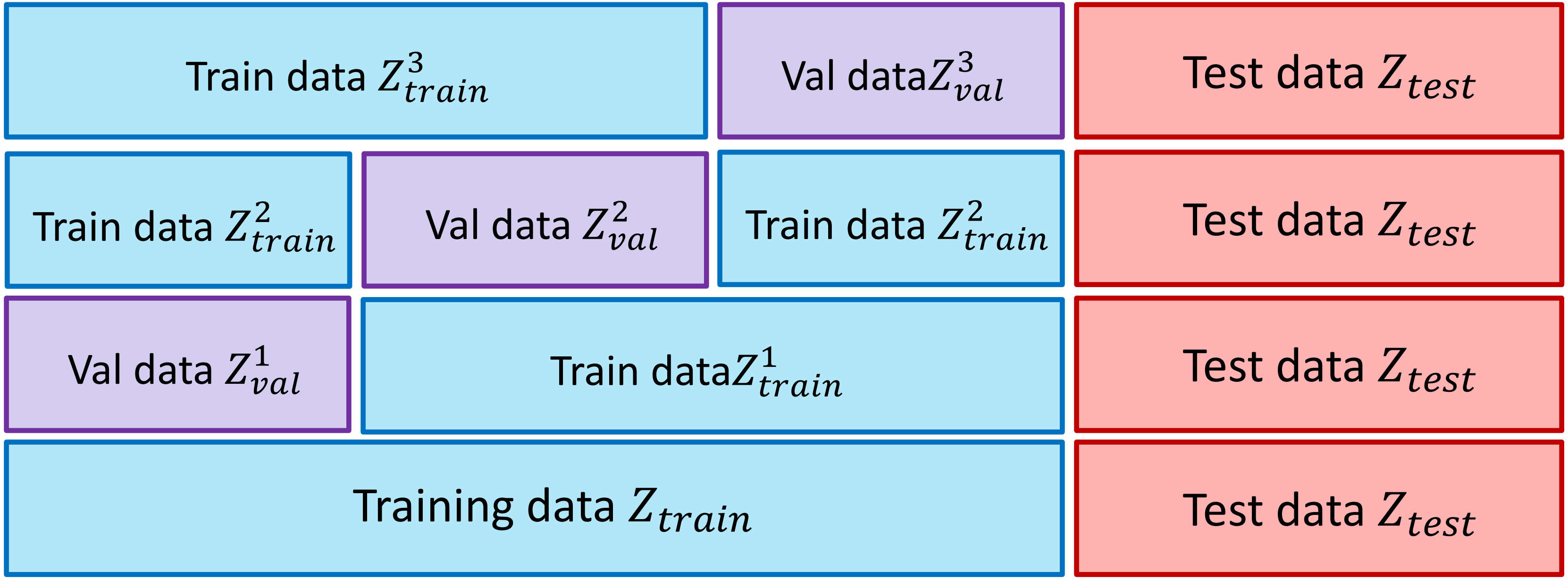
For each grid point of potential $\lambda > 0$

- **Step 1:** *Split Z into Z_{train}, Z_{val}, and Z_{test}*

- **Step 2:** *For $t \in \{1, \dots, h\}$:*
  - **Step 2a:** *Run linear regression with Z_{train} and λ_t to obtain $\hat{\beta}(Z_{train}, \lambda_t)$*
  - **Step 2b:** Evaluate validation loss $L_{val}^{t} = L\big(\hat{\beta}(Z_{train}, \lambda_t); Z_{val}\big)$
- **Step 3:** *Use best λ_t*
  - *Choose $t' = \arg\min L_{val}^{t}$ with lowest validation loss*
  - Re-run linear regression with Z_{train} and $\lambda_{t'}$ to obtain $\hat{\beta}(Z_{train}, \lambda_{t'})$

Tuning λ

For each grid point of potential $\lambda > 0$, we can replace Step 2 with the below K-fold CV



- **Step 3:** *Use the best λ_t*
  - *Choose $t' = \arg\min L^t_{val}$ with lowest validation loss*
  - Re-run linear regression with Z_{train} and $\lambda_{t'}$ to obtain

Test Loss

$$L\left(\hat{\beta}(Z_{train}, \lambda_{t'}); Z_{test}\right)$$

5-minute Quiz

Regularization 방법을 활용하여 적절한 lambda를 찾아보고자 시도.
M1~M4 는 penalty를 바꿔보며 적합함. $\lambda = 0,\ \lambda = 1,\ \lambda = 5,\ \lambda = 10.$

Model	Coefficients	Training Error	CV 1 Error	CV 2 Error	CV 3 Error	Average CV Error	Test Error
M1	$0.5h_1(x) + 0.3h_2(x) - 0.1h_3(x)$	0.06	0.09	0.07	0.08	0.08	0.07
M2	$0.1h_1(x) + 0h_2(x) + 0h_3(x)$	0.15	0.16	0.12	0.14	0.14	0.16
M3	$0.35h_1(x) + 0.1h_2(x) + 0h_3(x)$	0.10	0.04	0.08	0.15	0.09	0.05
M4	$0.45h_1(x) + 0.2h_2(x) - 0.05h_3(x)$	0.07	0.08	0.07	0.06	0.07	0.06

1. 각 모델별 lambda 값은?
2. Ridge? LASSO?
3. 어떤 모델을 선택할지?

Submit the quiz! (출석 대체)

Code-Review

See Regression 2 notebook file.

Classification Problems

Classification Examples: Prediction 관점

- Fraudulent Transaction Detection
    - Transaction이 발생했을 때 위치, IP Address, 사용자의 기존 거래정보를 기반으로 실시간 이상 감지

- 대출
    - 고객의 소득, 카드사용량, 신분 등의 정보를 기반으로 대출을 해줬을 때 상환을 잘 할 것인가에 대한 예측

- 유기동물 분양 예측
    - 유기동물의 나이, 종, 색깔 등의 상태, 업로드 된 사진 수 등을 기반으로 추후 입양할지 여부의 예측

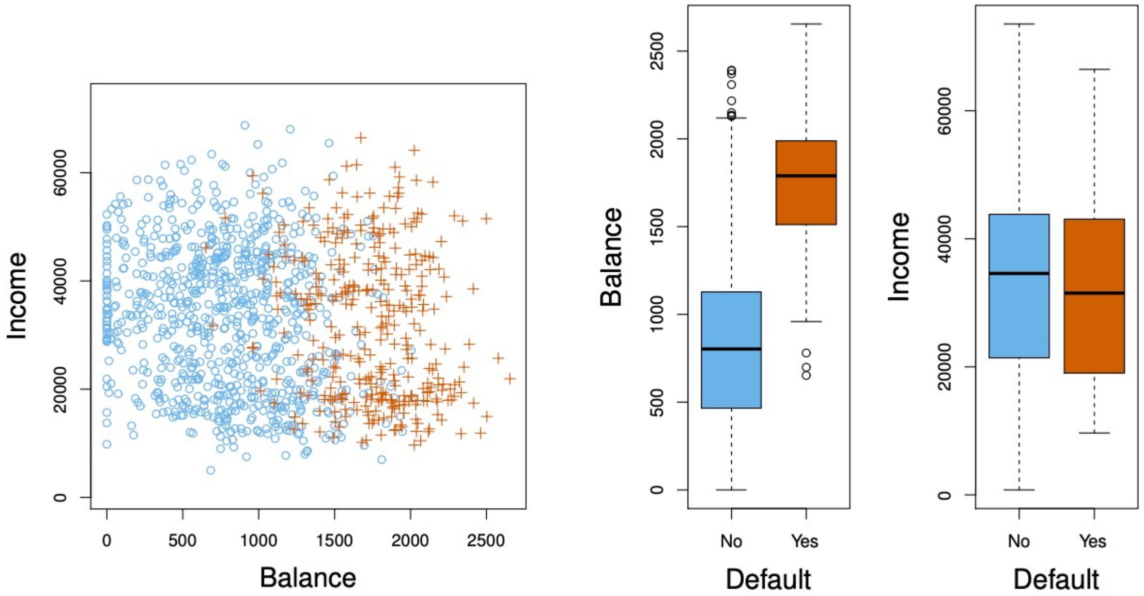
Classification Examples: Further Inference

- Fraudulent Transaction Detection
  - Transaction이 발생했을 때 위치, IP Address, 사용자의 기존 거래정보를 기반으로 실시간 이상 감지

- 대출
  - 고객의 소득, 카드사용량, 신분 등의 정보를 기반으로 대출을 해줬을 때 상환을 잘 할 것인가에 대한 예측. 어떤 정보가 대출상환 가능성에 큰 영향을 미치는가?

- 유기동물 분양 예측
  - 유기동물의 나이, 종, 색깔 등의 상태, 업로드 된 사진 수 등을 기반으로 추후 입양할지 여부의 예측. 유기동물이 좋은 보호자에게 더 잘 입양되도록 하기 위해 보호소는 어떤 일을 더 할 수 있을까?

Classification Example



- 소득, 카드사용량, 신분 등의 정보를 기반으로 향후 파산 여부

Classification Procedure

- *Data :* $(x_1, y_1), \ldots, (x_n, y_n)$
- $x_i : predictor\ information\ from\ i_{th} object.$

  $e.g.\ ith\ person's\ image\ (vectorized\ pixel\_values), ith\ person's\ financial\ information,$
- $y_i :\ label\ for\ ith\ object.$

  $e.g.\ ith\ person's\ full\ name\ or\ id, ith\ person's\ default\ status$

Goal: *find* $\hat{f}$

$$x \xrightarrow{\hat{f}} y$$

- *The range of* $\hat{f}$ *is* **a set of** *labels.* $\{ c_1, c_2, \ldots, c_k \}$

How did we do in regression?

At the population level,

- *Loss function:* $\ell(Y, f(X))$ = $(y - f(x))^2$
- *Risk function:* = $\mathrm{R}(f) = \mathbb{E}[(y - f(x))^2]$ *(***test MSE***, not known)*
- *Goal:* $f^* = argmin_{f \epsilon \mathcal{F}} \mathbb{E}[(y - f(x))^2]$

How did we do in regression?

At the population level, let f be a classifier.

- *Loss function:* $\ell(Y, f(X))$ = $(y - f(x))^2$
- *Risk function:* = $\mathrm{R}(f) = \mathbb{E}[(y - f(x))^2]$ = $\mathbb{E}[\ell(Y, f(X))]$
- *Goal:* $f^* = argmin_{f \epsilon \mathcal{F}}\ \mathbb{E}[(y - f(x))^2]$ = $argmin_{f \epsilon \mathcal{F}}\ \mathbb{E}[\ell(Y, f(X))]$

Risk 를 최소화하는 f 를 찾는다.

Risk 는 loss 에 대응하는 값.

Classification 에 적합한 Loss?

Recap: Performance Measures

Performance of supervised learning:

$$\text{Risk } R(f) \equiv \mathbb{E}_{XY}[loss(Y, f(X))]$$

	Classification	**Regression**
$loss(Y, f(X))$	$\mathbb{1}_{\{f(X) \neq Y\}}$	$(f(X) - Y)^2$
Risk $R(f)$	$P(f(X) \neq Y)$	$\mathbb{E}[(f(X) - Y)^2]$

Classification with 0-1 Loss

At the population level, let f be a classifier.

- Loss function: $\ell(y, f(x)) = \mathrm{I}(y \neq f(x))$

Correct classification 0-class, incorrect classification 1-loss.

- Risk function: R(f) = $\mathbb{E}[I(Y \neq f(X))] = \mathrm{P}(\mathrm{Y} \neq f(X))$ (**test error rate**)

- Goal: $f^* = argmin_{f \epsilon \mathcal{F}}\ \mathbb{E}[I(Y \neq f(X))] = argmin_{f \epsilon \mathcal{F}}\ \mathrm{P}(\mathrm{Y} \neq f(X))$

데이터를 기반으로 한 Loss 계산은?

Classification with 0-1 Loss

At the population level, let f be a classifier.

- Loss function: $\ell(y, f(x)) = \mathrm{I}(y \neq f(x))$

Correct classification 0-class, incorrect classification 1-loss.

- Risk function: $\mathrm{R}(f) = \mathbb{E}[I(Y \neq f(X))] = \mathrm{P}(\mathrm{Y} \neq f(X))$ (**test error rate**)

- Goal: $f^* = argmin_{f \epsilon \mathcal{F}}\ \mathbb{E}[I(Y \neq f(X))] = argmin_{f \epsilon \mathcal{F}}\ \mathrm{P}(\mathrm{Y} \neq f(X))$

데이터를 기반으로 한 Loss 계산은?

At the sample level, using the empirical distribution of (X, Y),

- $f^* = argmin_{f \epsilon \mathcal{F}}\ \mathbb{E}_n[I(Y \neq f(X))]$

$$\mathbb{E}_n[I(Y \neq f(X))] = \frac{1}{n} \sum_{i=1}^{n} I(Y_i \neq f(X_i))$$

Training error rate(misclassification rate)

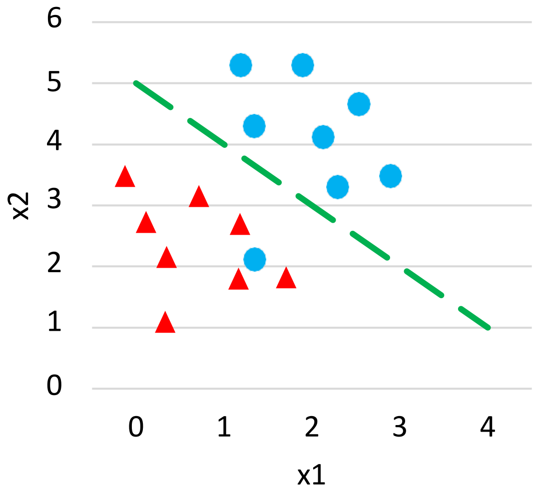
Decision Boundary for Linear Classifiers

$f(x) = 1\ or\ 0$ (sometimes, -1 or 1)
f is **linear**: the decision boundary is linear

- **(In)accuracy:**

$$L(\beta; Z) = \frac{1}{n}\sum_{i=1}^{n} 1\big(y_i \neq f_\beta(x_i)\big)$$

- **Classification:**
  - Labels $y_i \in \{0, 1\}$
  - Predict $y_i \approx 1\ \ (\beta^\top x_i \geq 0)$
  - $1(\,C)$ equals 1 if C is true and 0 if C is false



$$L(\beta; Z) = \frac{1}{16}$$

Bayes Classifier

The Bayes Classifier

- **Setting**:
  - Input: X
  - Output: Y $\in C = \{1,2,3,\ldots,K\}$

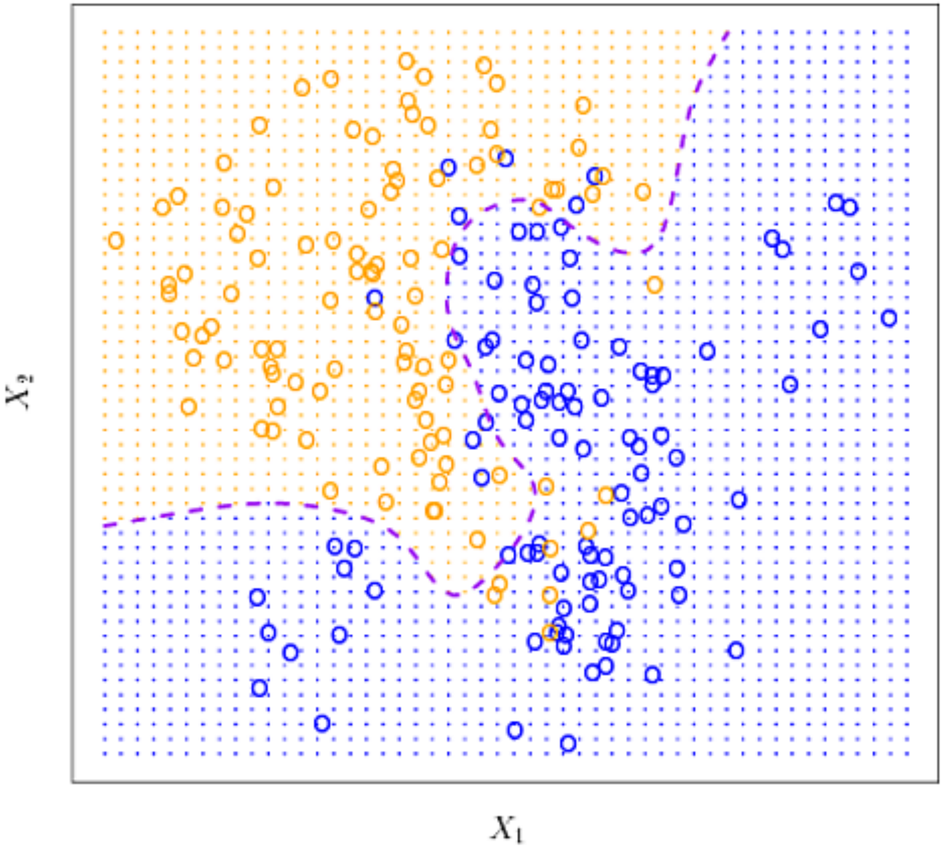
- **Goal**: Observe a new data point, $X = x_0$, predict its class Y

- **Population Level Prediction:** (X,Y) 의분포를 다 알고있을 경우,

1. Compute $P(Y = j|X = x_0)$ for $j = 1,\ldots,K$.
2. Assign Y as

$$f(x_0) = argmax_{j \in C} P(Y = j|X = x_0)$$

The Bayes Classifier Decision Boundary



- Setting
  - Input: X: 2차원
  - Output: Y (orange or blue)

- 그림
  - grid 포인트마다 Y assign

    $argmax_{j \in C} P(Y = j | X = x_0)$

  - dashed line: 확률이 같은 지점

The Bayes Classifier: Why?

- 분포를 다 알고 있다는 가정 하에 Bayes classifier 는 (0-1 loss 기반) test error rate 을 최소화 시키는 방법!
- 많은 경우에 Bayes Classifier를 gold standard 로 둠.

- Challenges
  - 이론상으로만 존재
  - we do not know the conditional distribution $Y|X=x$

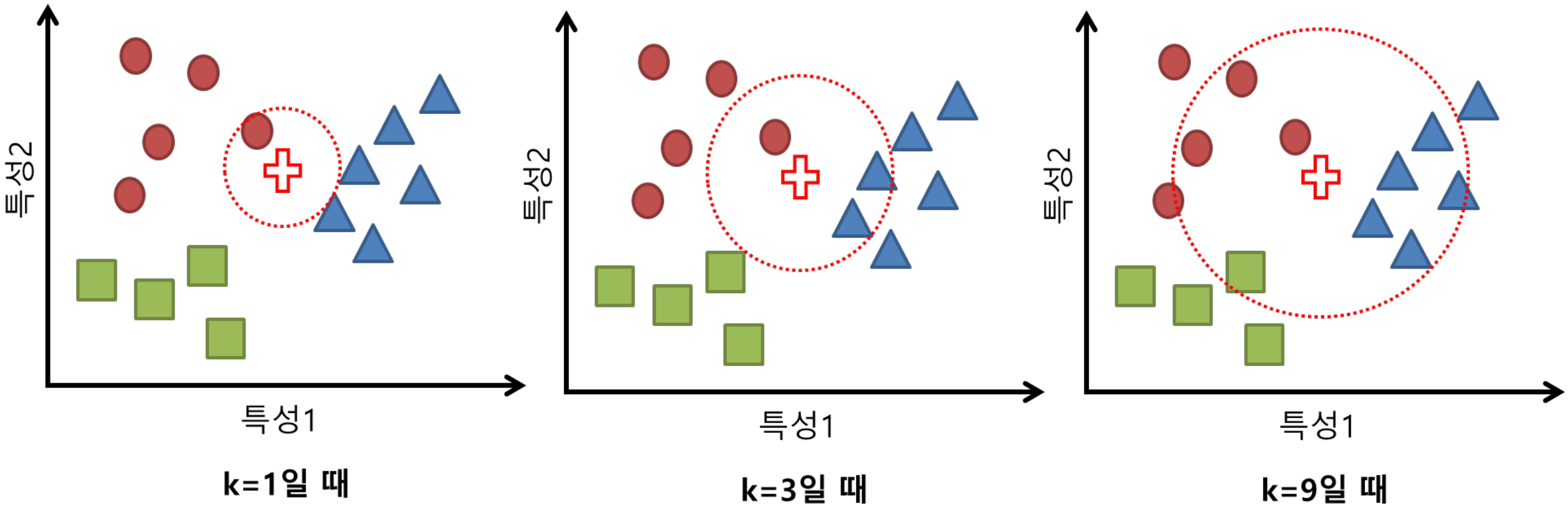
- Estimate!

KNN Classifier

prototype method

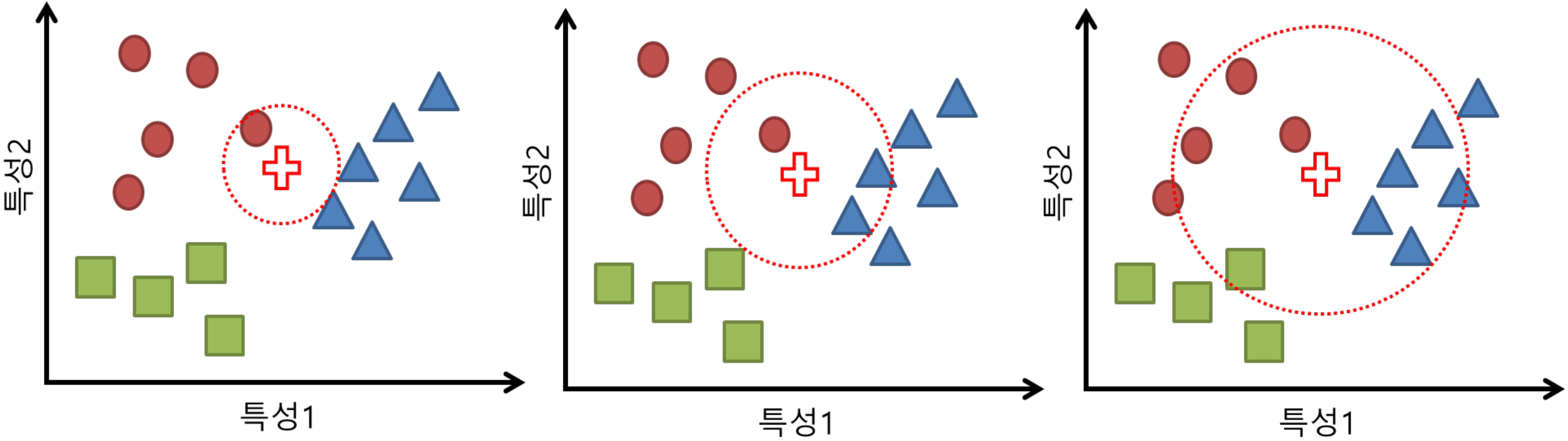
K-Nearest Neighbors (KNN) Classifier

Point: datasets, ✚ : new data observed
Question: which class would you assign for ✚ ?



K-Nearest Neighbors (KNN) Classifier

Point: datasets, : new data observed
Question: which class would you assign for ?

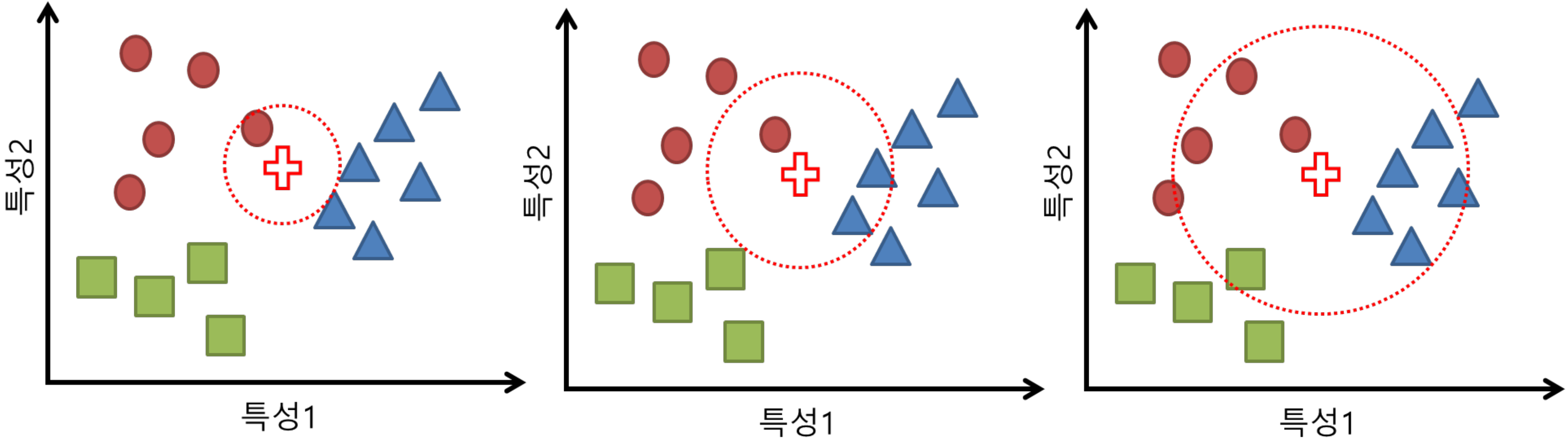


The KNN assigns

k=1일 때 **k=3일 때** **k=9일 때**

K-Nearest Neighbors (KNN) Classifier

Point: datasets, : new data observed
Question: which class would you assign for ?



The KNN assigns **k=1일 때** **k=3일 때** **k=9일 때**

K-Nearest Neighbors (KNN) Classifier

When a new X=x is observed, to predict Y

1. Choose

   x와 가장 **가까운** K개의 data points in the **training data**

2. Assign Y by using K개의 클래스 중 Majority

Relation to the Bayes Classifier

1. Estimate the conditional probability for each class j as

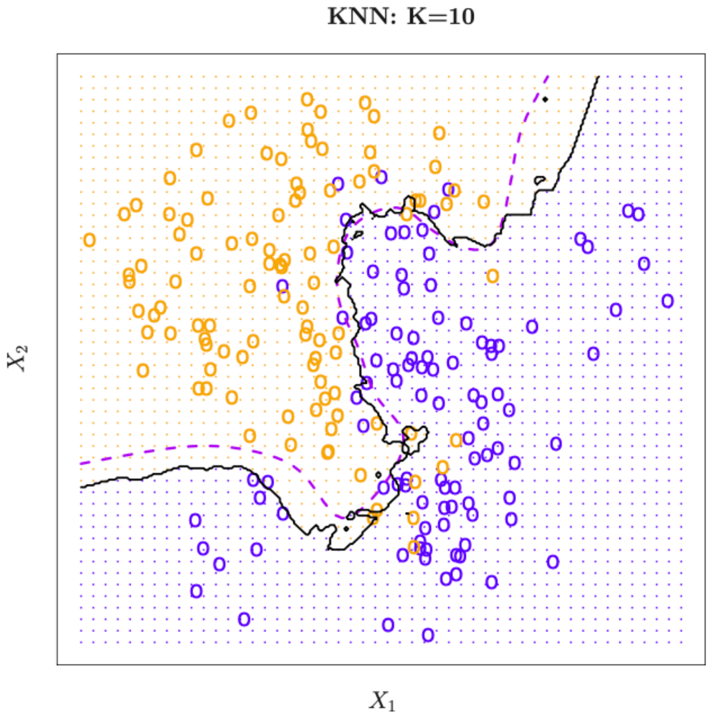
$$P(Y = j|X = x_0) = \frac{1}{K}\sum_{i \in \mathcal{N}_0} I(y_i = j)\,, proportion\ of\ class\ in\ \mathcal{N}_0$$

$$\mathcal{N}_0 = \{K - nearst\ points\ in\ the\ training\ data\ that\ are\ closest\ to\ x_0\}$$

2. KNN assigns the class such that the class with the largest probability.

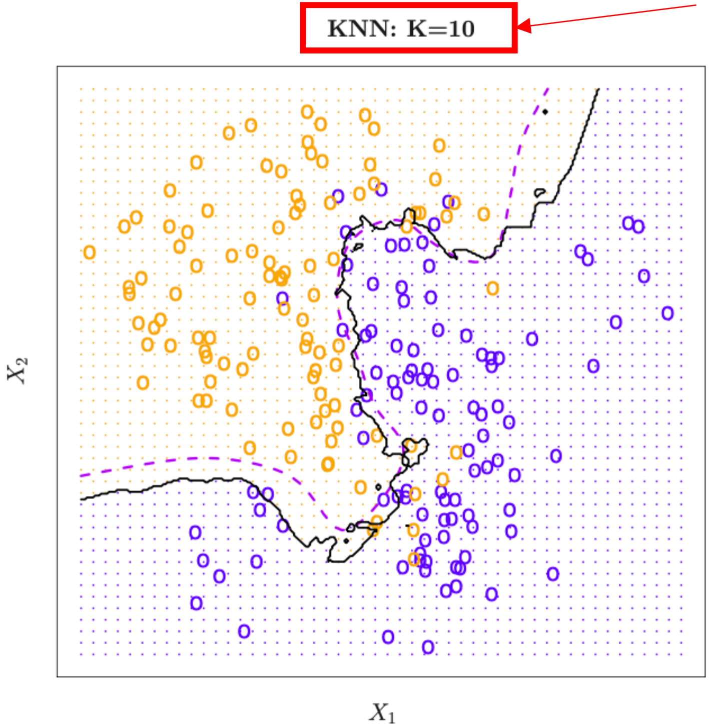
K-Nearest Neighbors (KNN) Classifier

Despite the fact that it is very simple approach, KNN can often produce classifiers that are surprisingly close to the optimal Bayes classifier.



K-Nearest Neighbors (KNN) Classifier

Despite the fact that it is very simple approach, KNN can often produce classifiers that are surprisingly close to the optimal Bayes classifier.



Choice of K?

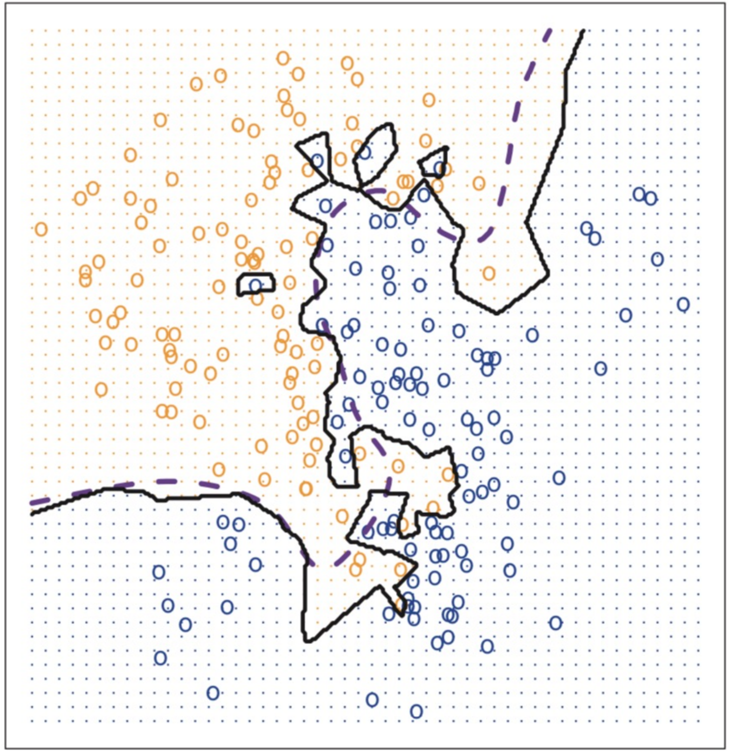
What is the role of K?

K-Nearest Neighbors (KNN) Classifier

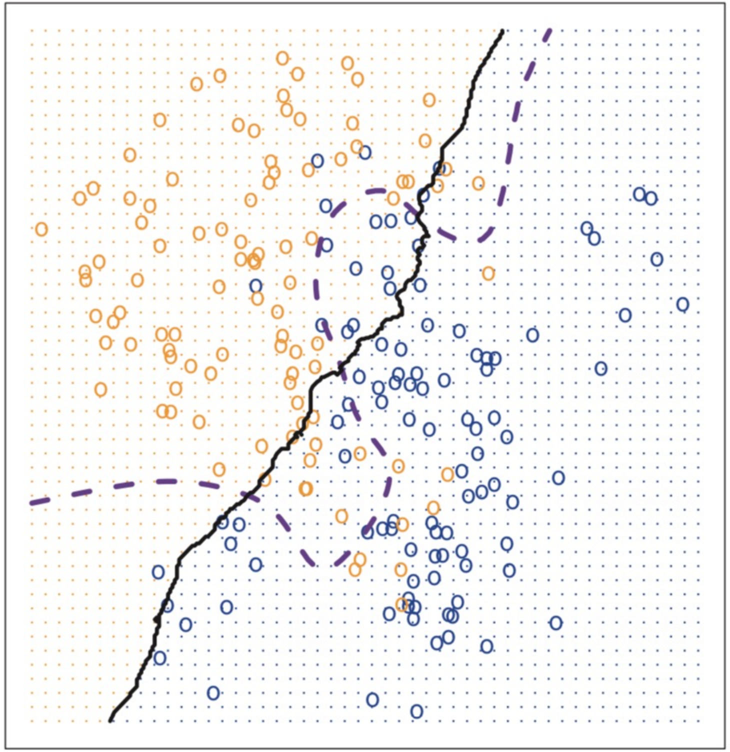
- K: 주어진 점과 가장 가까운 neighbor 의 수
- K=1: the decision boundary is overly **flexible** -> high variance
  (소수의 데이터 변동만 있어도 추정한 classifier의 변동이 심함)
- Large K: the decision boundary is not sufficiently flexible -> low variance

K-Nearest Neighbors (KNN) Classifier

KNN: K=1

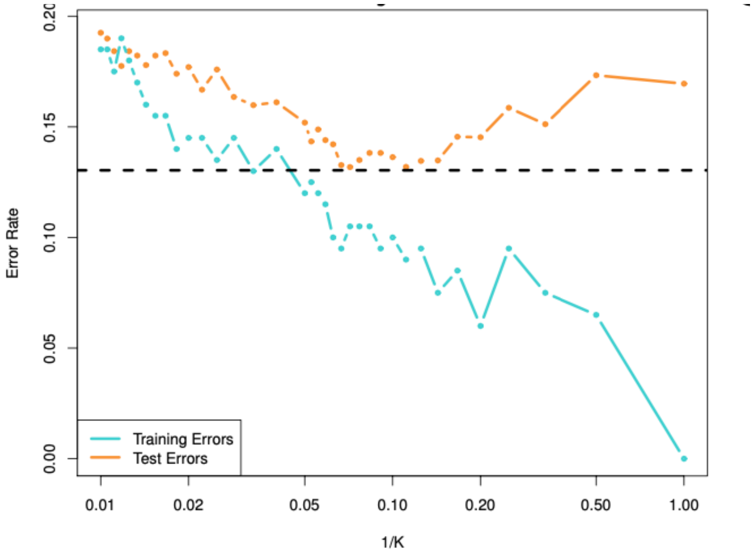


KNN: K=100



K-Nearest Neighbors (KNN) Classifier

- Plot the test and training errors as a function of $1/K$.
- As $1/K$ increases, the method becomes more flexible.
- In both the regression and classification settings, choosing the correct level of flexibility is critical to the success of any statistical learning method.



KNN Classifier: Limitation

- Scalability with Large Datasets
  - Train 과정이 따로 없음. Test 데이터가 주어졌을 때 모든 training data와의 거리 계산을 다 해야 함. (large data로 미리 학습 불가능)
  - 모든 feature 들이 동일한 contribution (거리계산 고려해보기)

- Curse of Dimensionality
  - 차원이 커질 수록 Data point간 거리에 대한 개념이 명확하지 않음

- K-Selection
  - 적절한 K 값의 선택이 매우 중요함.

- Inference
  - There is no model interpretation

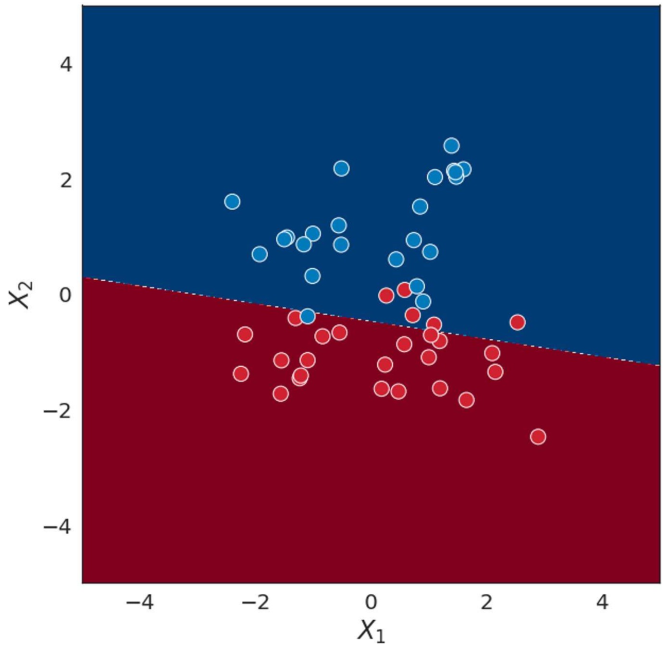
Logistic Regression

Generalized Linear Models (GLM)

Linear Functions for (Binary) Classification

- **Input:** Dataset $Z = \{(x_1, y_1),(x_2, y_2)\ldots,(x_n, y_n)\}$

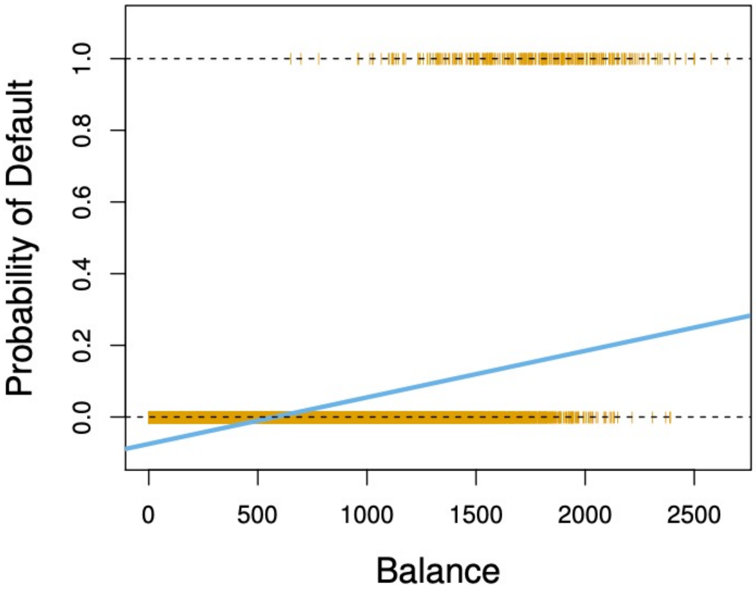
- **Classification:**
  - Labels $y_i \in \{0, 1\}$
  - Predict $y_i \approx 1\ (\beta^\top x_i \geq 0)$
  - $1(C)$ equals 1 if C is true and 0 if C is false
  - How to learn β?

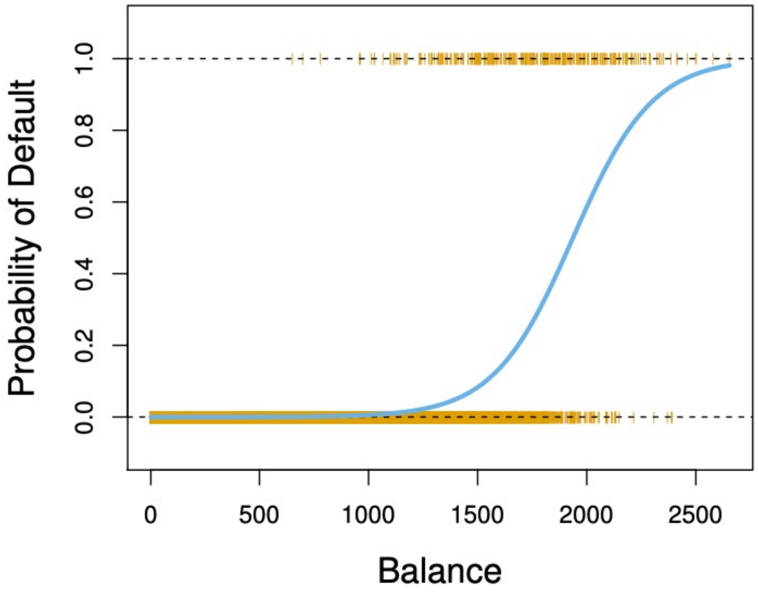


Can we use Linear Regression?

- Y is already encoded as 0 (No default) and 1(Default).
- Can we fit a linear regression?
    - For a new X = x, and predict its class as Default if $\hat{Y} > 0.5$ and No Default if $\hat{Y} \leq 0.5$
- No. The value might be less than zero or above 1.
- We need to estimate the probability! P(Y=1|X=x), instead of Y|X directly.

Can we use Linear Regression?





Linear Regression
whatever the Balance is,
all the predicted values are 0

Logistic Regression

Logistic Regression

- Let's use notation $p(x) = P(Y = 1 \,|X = x)$ for short. Logistic regression uses the form

$$P(Y = 1 \,|X = x) \approx \hat{f}(X).$$

- We are no longer able use $f(X) = \beta_0 + \beta_1 X$ since $0 \leq p(X) \leq 1$.
- To use the linear regression, we want the range of Y to be any real number, (-∞, ∞).

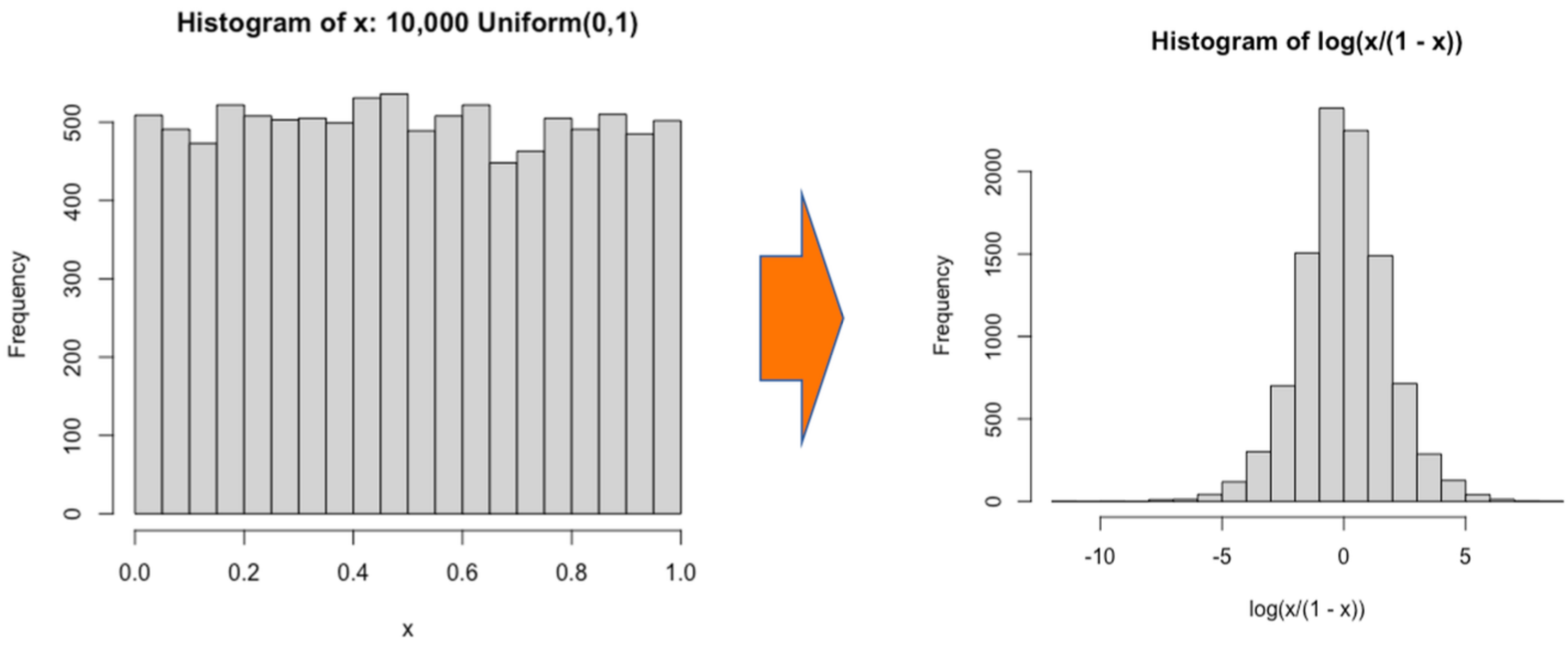
Logistic Regression: logit transformation

1. Odds Ratio: $\frac{p}{1-p}$, $p : success\ of\ probability$
   - $p \in (0,1) \rightarrow (0, \infty)$
   - $p \in (0,0.05) \rightarrow (0, 1)$
   - $p \in (0.5,1) \rightarrow (1, \infty)$
2. Log Transformation: $\log(\text{odds ratio}) = \log(\frac{p}{1-p})$
   - $(0,1) \rightarrow (-\infty, 0)$
   - $(1, \infty) \rightarrow (0, \infty)$
   - Remark: log-transformation is frequently used for counting data as well (e.g., poisson)

The combination of these two steps is called logit transformation

If $p \sim Unif(0,1)$, what will be the distribution $logit(p)$?

Logistic Regression: logit transformation



Remark: the logit function is derived based on the canonical link function of the exponential family

Logistic Regression Model

- Now, our model is

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

- The left-hand side is called the log-odds or logit.
- Hence, the logistic regression has a logit that linear in X.

Sigmoid function

- In terms of $p(X)$, with a little algebra,

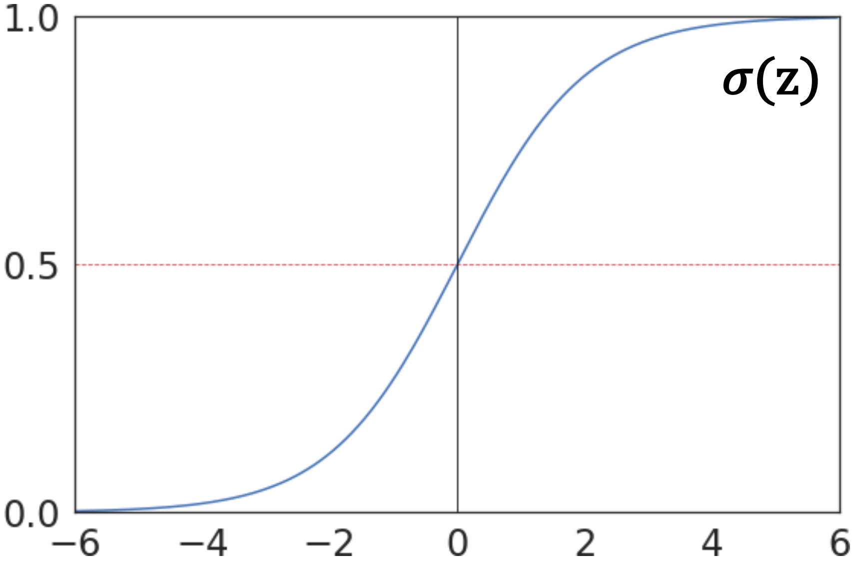
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$z = \beta_0 + \beta_1 X$$

- Remark: $p(X)$ is an increasing function of $\beta_0 + \beta_1 X$

Logistic/Sigmoid Function



Sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$z = \beta_0 + \beta_1 X$$

Logistic Regression: Estimation

- Given an X value, x, we need to estimate $p(x) = P(Y = 1|X = x)$.

(note that $P(Y = 0|X = x) = 1 - P(Y = 1|x = x) = 1 - p(x)$)

- Y takes a value 1 or 0. i.e., it is a Bernoulli random variable.

- **Maximum Likelihood Estimation (MLE)**: 데이터가 주어졌을 때 가장 **가능도**가 높은 값으로 parameter 추정.

$$L(\beta_0, \beta_1, \mathcal{D}) = \prod_{y_i=1} p(x_i) \prod_{y_i=0} (1 - p(x_i)) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

Logistic Regression: Estimation

- Given an X value, x, we need to estimate $p(x) = P(Y = 1|X = x)$.

(note that $P(Y = 0|X = x) = 1 - P(Y = 1|x = x) = 1 - p(x)$)

- Y takes a value 1 or 0. i.e., it is a Bernoulli random variable.

- **Maximum Likelihood Estimation (MLE):** 데이터가 주어졌을 때 가장 **가능도**가 높은 값으로 parameter 추정.

$$L(\beta_0, \beta_1, \mathcal{D}) = \prod_{y_i=1} p(x_i) \prod_{y_i=0} (1 - p(x_i)) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

Where $\mathcal{D}$ stands for the dataset.

Then find the maximizer $\widehat{\beta_0}$, $\widehat{\beta_1}$.

For computational convenience, we use log-likelihood,

$$l(\beta_0, \beta_1; \mathcal{D}) = \sum_i [\, y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))]$$

Logistic Regression: Estimation

- Compute the likelihood function based on β_0, β_1.

$$L(\beta_0, \beta_1, \mathcal{D}) = \prod_{y_i=1} p(x_i) \prod_{y_i=0} (1 - p(x_i)) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

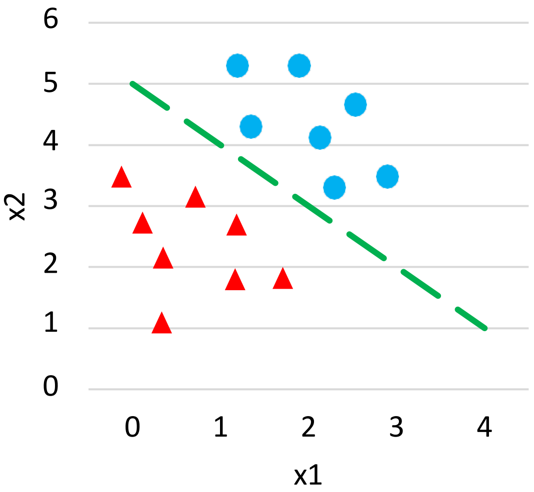
Where $\mathcal{D}$ stands for the dataset.

- Then find the maximizer $\widehat{\beta_0}$, $\widehat{\beta_1}$.
- For computational convenience, we use log-likelihood,

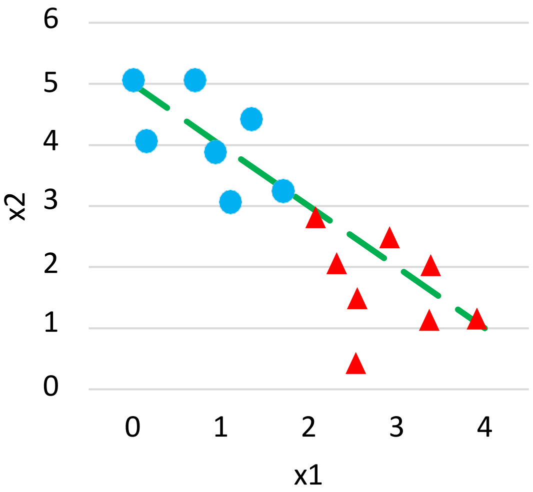
$$l(\beta_0, \beta_1; \mathcal{D}) = \sum_i [\, y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))]$$

$$\hat{\beta} = \operatorname{argmax} l(\beta_0, \beta_1; D) = \operatorname{argmax} \sum (y_i \log p(x_i) + (1 - y_i) \log\big(1 - p(x_i)\big)$$
$$\hat{\beta} = \operatorname{argmin} - l(\beta_0, \beta_1; D) = \operatorname{argmin} \sum (-y_i \log p(x_i) - (1 - y_i) \log\big(1 - p(x_i)\big)$$
$$\textit{where } p(x_i) = \sigma(\beta_0 + \beta_1 x_i)$$

Intuition on the Likelihood



High likelihood (Low NLL)



Low likelihood (High NLL)

Logistic Regression: Classification!

beta값 추정 이후 Classification은? 아래 조건부확률을 최대화시키는 Class로 추정

$$f_\beta(x) = \arg\max_y p_\beta(y \mid x)$$

$$= \arg\max_y \begin{cases} \sigma(\beta^\top x) & \text{if } y = 1 \\ 1 - \sigma(\beta^\top x) & \text{if } y = 0 \end{cases}$$

$$= \begin{cases} 1 & \text{if } \sigma(\beta^\top x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$p_\beta(y|x)$

$$\hat{\beta} = \operatorname{argmax} \boxed{l(\beta_0, \beta_1; D)} = \operatorname{argmax} \sum (y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)))$$

$$\hat{\beta} = \operatorname{argmin} - l(\beta_0, \beta_1; D) = \operatorname{argmin} \sum (-y_i \log p(x_i) - (1 - y_i) \log(1 - p(x_i)))$$

$$\textit{where } p(x_i) = \sigma(\beta_0 + \beta_1 x_i)$$

Logistic Regression: Classification!

beta값 추정 이후 Classification은? 아래 조건부확률을 최대화시키는 Class로 추정

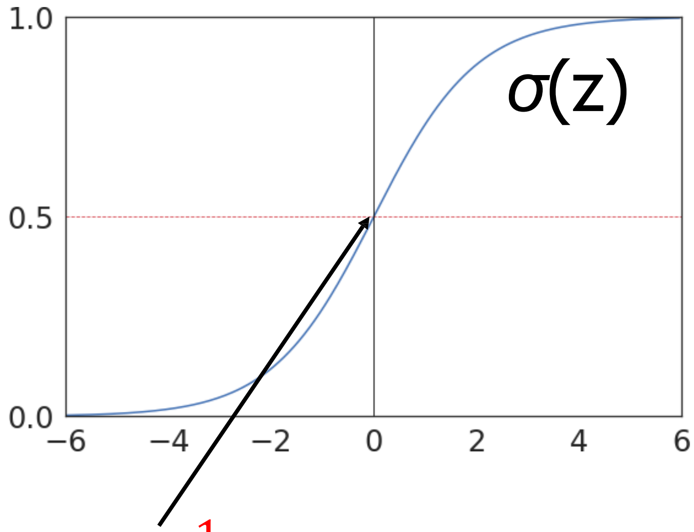
$$f_\beta(x) = \arg\max_y p_\beta(y \mid x)$$

$$= \arg\max_y \begin{cases} \sigma(\beta^\top x) & \text{if } y = 1 \\ 1 - \sigma(\beta^\top x) & \text{if } y = 0 \end{cases}$$

$$= \begin{cases} 1 & \text{if } \sigma(\beta^\top x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1 & \text{if } \beta^\top x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= 1(\beta^\top x \geq 0)$$



$$\sigma(0) = \frac{1}{2}$$

- **Recovers linear classifiers!**

(i.e., Decision Boundary is Linear

Logistic Regression: Interpretation

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \text{ , } p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

- Interpreting what β_1 means isn't very easy with logistic regression, because we are prediction $p(X) = P(Y = 1 \mid X)$ and not Y
- If $\beta_1 = 0$, this means that there is no (linear) relationship between Y and X. In other words, when we predict the class of Y, X doesn't matter.
- If $\beta_1 > 0$, this means when X gets larger the probability, $P(Y = 1 \mid X)$ gets larger.
- If $\beta_1 < 0$, this means when X gets larger the probability, $P(Y = 1 \mid X)$ gets smaller.

Logistic Regression: Interpretation

- 기사 예제
  - *xx*가 *1* 증가할 때 암발생/미발생 비율이 *y*배 증가한다.
  - 남자가 여자보다 암발생/미발생 비율이 *YY*배 증가한다 *(X*가 *categorical*일때*)*

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \, , \; p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

  - *The odds are*

$$Odds = \frac{p(X)}{1-p(X)} = \exp(\beta_0 + \beta_1 X) = e^{\beta_0}(e^{\beta_1})^x$$

  - *As* x *increases by one unit, the odds is multiplied by* e^{β_1}

  - *When* $\beta = 0$, $e^{\beta_1} = 1$ *so that no changes in odds.*

Logistic Regression: Inference

$$H_0 : \beta_1 = 0 \qquad v.s. \qquad H_1 : \beta_1 \neq 0$$

- Likelihood-ratio test

$$LR = -2\log\left(\frac{l_0}{l_1}\right) = -2(L_0 - L_1) \sim \chi_1^2$$

Here l_0 and l_1 are maximum likelihoods under $H_0 \cup H_1$ respectively

- If $|z| \geq z_{\frac{\alpha}{2}}$, $z^2 \geq \chi_1^2(\alpha)$, or $LR \geq \chi_1^2(\alpha)$ (p-value < α=0.05), then reject H_0

Logistic Regression: Inference

Below the p-value for balance is very small, and β_1 is positive, so if the balance increases, then the probability of default will increase as well.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Logistic Regression: Inference

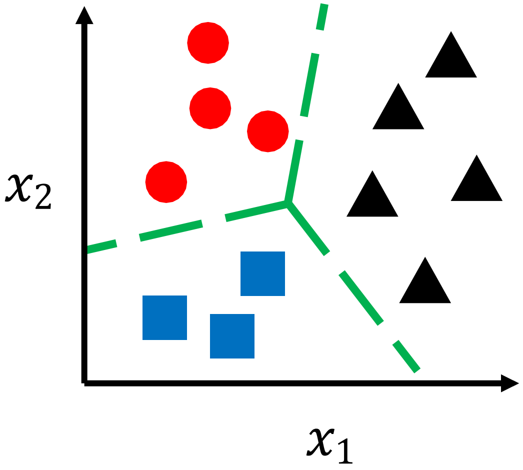
- Logistic Regression 은 **Generalized Linear Models (GLM)**의 한 종류
- Predictor 가 Quantitative, Qualitative 변수들이 mix돼있어도 잘 작동

▶ Predictors: Balance (quantitative), Income (quantitative) and Student (qualitative)

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[yes]	-0.6468	0.2362	-2.74	0.0062

Multi-Class Classification

- What about more than two classes?
  - **Disease diagnosis:** healthy, cold, flu, pneumonia
  - **Object classification:** desk, chair, monitor, bookcase
  - In general, consider a finite space of labels C={1,…,K}



Multi-Class Classification

- **Naïve Strategy 1:** One-vs-One classification

1. Develop $\binom{k}{2}$ classifiers.
    1. For each pair (i, j) of the classes, $f_{ij}(x)$ classifies if x is either of class i or j
    2. Then we have $f_{12}(\cdot), f_{13}(\cdot), \dots, f_{K-1,K}(\cdot)$.
2. Given a new observation x^*, assign the class by the majority of the classifiers.

예: K=3, C={1,2,3}, $f_{12}(x^*) = 1$, $f_{13}(x^*) = 3$, $f_{23}(x^*) = 3$. Then the final classifier assigns 3 for the class of x^*.

Multi-Class Classification

- **Naïve Strategy 2:** One-vs-rest (or One-vs-All) classification

1. Develop K classifiers

   1. $f_1(x)$ classifies if x is either of class 1 or the other.

   i.e., $f_1(x) = 1$ if it is classified as 1, or $f_1(x) = -1$ if it is classified as the other

   .

   .

   K. $f_K(x)$ classifies if x is either of class K or the other.

2. Given a new observation x^*, assign the class by

$$\underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} f_k(x^*)$$

Multi-Class Logistic Regression

- **Strategy:** *Include separate* β_y *for each label* $y \in \mathcal{Y} = \{1, \ldots, K\}$
- *Let* $p_\beta(y \mid x) \propto e^{\beta_y^\top x}$, *i.e.*

$$p_\beta(y|x) = \frac{e^{\beta_y^T x}}{\sum_{y' \in \mathcal{Y}} e^{\beta_{y'}^\top x}}$$

- We define softmax $(z_1, \ldots, z_K) = \left[\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \ldots \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}},\right]$
- Then, $P_\beta(y|x) = softmax(\beta_1^T x, \ldots, \beta_k^T x)_y$
  - Thus, sometimes called softmax regression

Performance Measure for Classifiers

Evaluation of Classifiers

Mis-classification rate(Error rate) (overall error rate)

$$\frac{\#\ of\ incorrectly\ classified\ objects}{total\ \#\ of\ objects}$$

Let $\hat{\delta}(X)$ be a classifier computes using the training set.

Training error rate

$$\frac{\#\ of\ incorrectly\ classified\ objects\ of\ \hat{\delta}(X.train)}{total\ \#\ of\ objects\ in\ training\ set.}$$

Test error rate

$$\frac{\#\ of\ incorrectly\ classified\ objects\ of\ \hat{\delta}(X.test)}{total\ \#\ of\ objects\ in\ test\ set.}$$

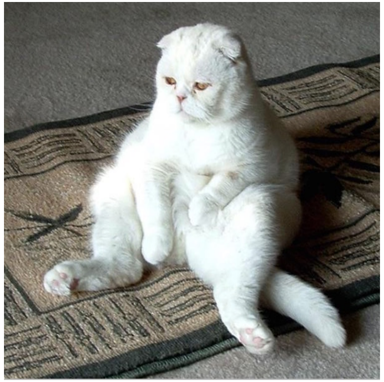
- ✓ General
- ✓ 0-1 loss 기반

Loss가 동일?

0-1 Loss

- 동물 종 *Classifier*를 만들었다고 하자. *Loss* 계산을 할 때
  - 페르시안 고양이를 스코티시 고양이로 분류했을 때의 *Loss*
  - 페르시안 고양이를 포메라니안 강아지로 분류했을 때의 *Loss*

f_1(이미지)



f_2(이미지)

$$L(f_1, data_i) = L(f_2, data_i)??$$

0-1 로스의 경우 둘 다 1

Errors in Binary Classification

- 피검사를 통한 암진단 키트를 새로 개발했다고 가정하자.
  - 암이 없는 환자를 암이 있다고 예측 *(Test Positive)*
  - 암이 있는 환자를 암이 없다고 예측 *(Test Negative)*

  - 두 오류의 *Cost* 가 같은가?

Errors in Binary Classification

- 법정에서의 판결 과정: 유죄? 무죄?
  - 실제 범죄를 저지른 피고인을 무죄로 판정
  - 범죄를 저지르지 않은 피고인을 유죄로 판정

- 두 오류가 같은가?

Errors in Binary Classification

- 법정에서의 판결 과정: 유죄? 무죄?
  - 실제 범죄를 저지른 피고인을 무죄로 판정
    - 검사가 충분한 범죄를 입증 할만한 근거를 수집하지 못함
  - **범죄를 저지르지 않은 피고인을 유죄로 판정**
    - **이 경우 더 큰 문제로 인식. 무죄추정의 원칙**

- 통계적 가설 검정: 모델이 유의한가? 유의하지 않은가?
  - 적절한 통계량이 기준 수치 도달하지 않음 (*p-value* 기준수치 이상)
    - 모델이 유의미하다는 충분한 근거가 없음
  - 적절한 통계량이 기준 수치 이상 (*p-value* 기준수치(5%) 이하): 모델 유의

- 위 두 상황 모두, 기본 가정을 두고, 충분한 근거가 있을 때만 다른 선택

Errors in Binary Classification

- 통계적 가설 검정: 모델이 유의한가? 유의하지 않은가?
  - 적절한 통계량이 기준 수치 도달하지 않음 *(p-value* 기준수치 이상*)*
    - 모델이 유의미하다는 충분한 근거가 없음
  - 적절한 통계량이 기준 수치 이상 *(p-value* 기준수치*(5%)* 이하*)*: 모델 유의

- 모델이 유의미하지 않은데*(*귀무가설이 참인데*)* 귀무가설 *Reject* 할 확률

  Type I error (5%)
- 모델이 유의미한데*(*귀무가설 *Reject*해야 하는데*)* 귀무가설 *Reject* 못할 확률

  Type II error (1 – Power)

통계적 가설 검정은, 모델이 유의미하지 않은데 유의미하다고 판단하는 오류에 *Focus*

Errors in Binary Classification

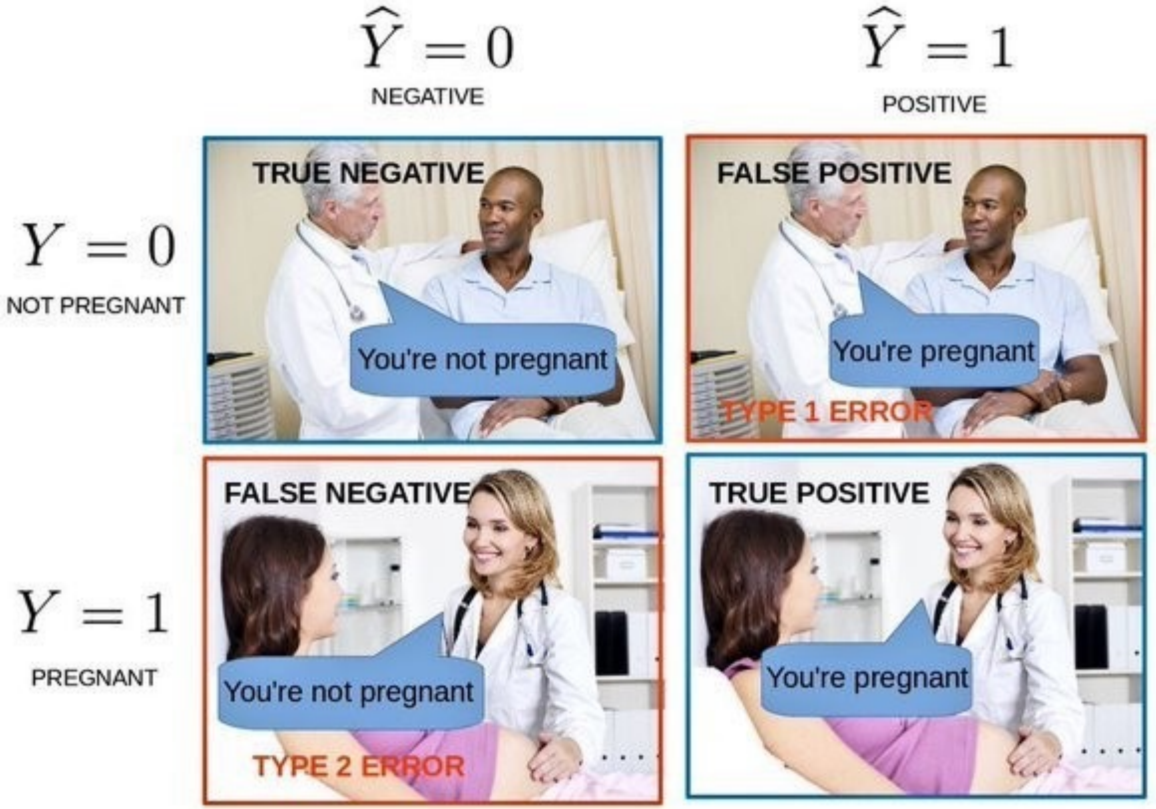
Binary Classification: Class 가 두개인 경우

- Positive: 측정값이 일정 수치 넘었을 경우 (Reject H_0)
- Negative: 기본 상태 (Do not reject H_0)

Errors in Binary Classification

Prediction

TRUE Status



Classification Metrics

- **Classify test examples as follows:**
  - **True positive (TP):** *Actually positive, predictive positive*
  - **False negative (FN):** *Actually positive, predicted negative*
  - **True negative (TN):** *Actually negative, predicted negative*
  - **False positive (FP):** *Actually negative, predicted positive*

- *Many metrics expressed in terms of these; for example:*

$$\mathit{accuracy} = \frac{TP+TN}{n} \qquad \mathit{error} = 1 - \mathit{accuracy} = \frac{FP+FN}{n}$$

Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

Accuracy = 0.8

Sensitivity & Specificity

- **Sensitivity(민감도):** What fraction of **actual positives** are **predicted positive**?
  - **Good sensitivity:** If you have the disease, the test correctly detects it
  - Also called **true positive rate** **(Power =1 – Type II error)**

- **Specificity(특이도):** What fraction of **actual negatives** are **predicted negative**?
  - **Good specificity:** If you do not have the disease, the test says so
  - Also called **true negative rate**
  - **1-Speicificity: False positive rate**

- Commonly used in medicine

Sensitivity & Specificity

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

$$\text{specificty} = \frac{TN}{TN + FP}$$

Sensitivity & Specificity

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

$$\text{specificity} = \frac{TN}{TN + FP}$$

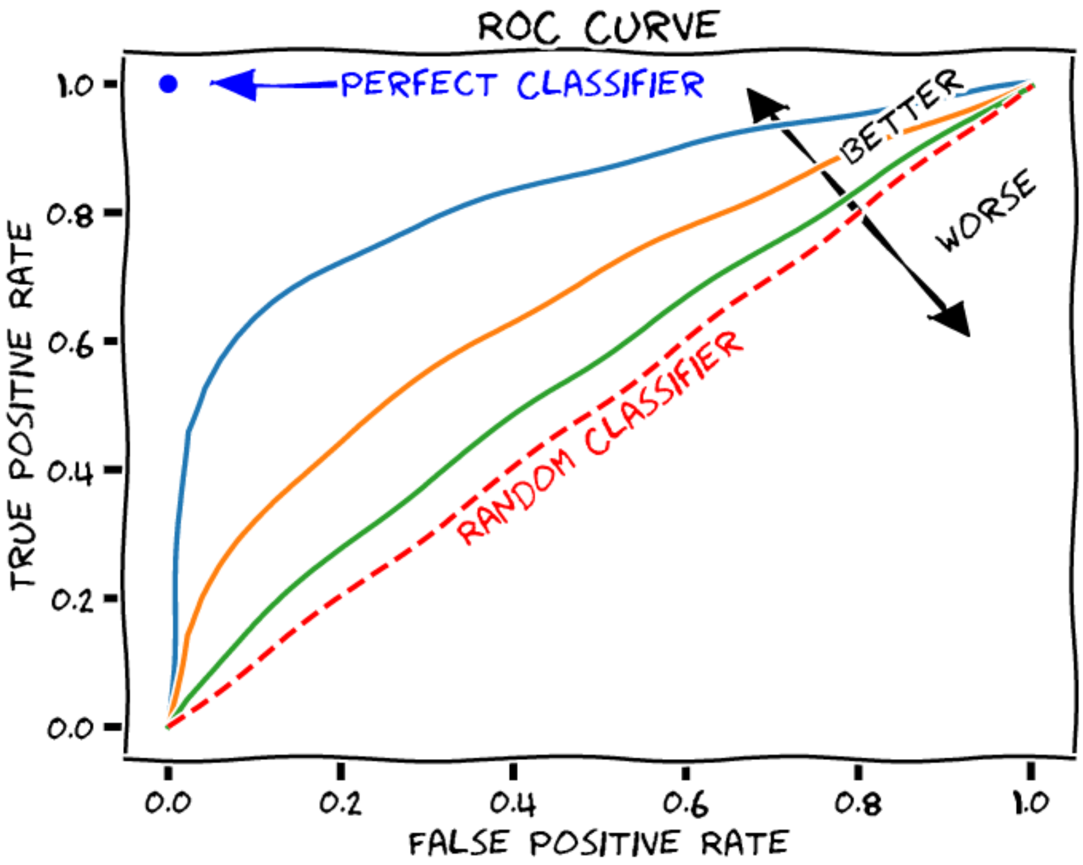
Sensitivity & Specificity

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37 TN

$\text{sensitivity} = 3/7$

$\text{specificity} = 37/43$

Sensitivity & Specificity: ROC Curve



Measure of Perforance:

- AUC: Area Under the Curve
- P(Y=1)>threshold -> y=1
- 이 임계값의 변화에 따라 TPR, FPR이 바뀜

- TPR: Sensitivity
- FPR: 1-Specificity

Sensitivity & Specificity

- 희귀질병에 대한 진단을 한다고 하자. 발병율 *0.1%*
- 진단키트 개발시 모두가 병이 없다고 했을 경우, *Accuracy =?*

Sensitivity & Specificity

- 희귀질병에 대한 진단을 한다고 하자. 발병율 *0.1%*
- 진단키트 개발시 모두가 병이 없다고 했을 경우, *Accuracy* = **99.9%**

		Predicted Class		
		Yes	No	
Actual Class	Yes	0 TP	1 FN	$\text{sensitivity} = 0$
	No	0 FP	999 TN	$\text{specificity} = 100$

Precision & Recall

- **Recall(재현율):** What fraction of **actual positives** are **predicted positive**?
  - **Good recall:** If you have the disease, the test correctly detects it
  - Also called the **true positive rate** (and sensitivity)

- **Precision(정밀도):** What fraction of **predicted positives** are **actual positives**?
  - **Good precision:** If the test says you have the disease, then you have it
  - Also called **positive predictive value**

- Used in information retrieval, NLP

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37TN

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

Precision & Recall

		Predicted Class	
		Yes	No
Actual Class	Yes	3 TP	4 FN
	No	6 FP	37TN

recall $= 3/7$

precision $= 3/9$

Classification Metrics

- F1-score: Precision과 Recall의 weighted sum
- Top-k Accuracy: multi-class classification 문제에서 주로 사용
- etc