데이터마이닝 기말고사

1. Statistical Learning Overview

1-1. Statistics / ML / AI

- **AI**: the whole person — perception, memory, reasoning, and action.
- **ML**: the brain's learning mechanism — adapting from experience.
- **Statistics**: rational thinking under uncertainty — quantifying evidence and doubt.
- **Data Mining**: curiosity — searching for hidden patterns and structure.

- data mining: 유한한 표본으로부터 미지의 확률모형을 추정하는 것

1-2. Statistical Learning

Supervised Learning: $P(Y \mid X)$

- goals:
  - prediction
  - inference
  - quantify uncertainty
- learning is not just fitting — generalization under uncertainty

Unsupervised Learning: $P(X)$

- goals:
  - structure estimation
  - clustering
  - representation learning
- models structure in $P(X)$

2. Understanding Data

2-1. Data as Probabilistic Evidence

- data are stochastic evidence
- estimators are random because samples are random!

2-2. Assumptions and Identifiability

- data alone do not uniquely determine the target
- we restrict the set of possible worlds (functional form)
- same data can have multiple explanations
- examples of function classes
  - linear models
  - parametric models
  - RKHS models (kernel methods)
  - nonparametric classes

2-3. Data Structures and Their Implications

i.i.d. Assumption

- typical iid settings
  - randomized controlled trials
  - survey responses sampled independently

    - repeated coin tosses
- beyond iid: dependence (e.g. time series data)

High-Dimensional Regime

- $p >> n$ → linear system is underdetermined (OLS is not unique)
- curse of dimensionality
    - number of points needed to cover the space grows multiplicatively
    - high dimension → exponential complexity

Partial Observation

- e.g. survival time

Measurement Error

- observed relationships may be distorted

3. Explanatory Data Analysis

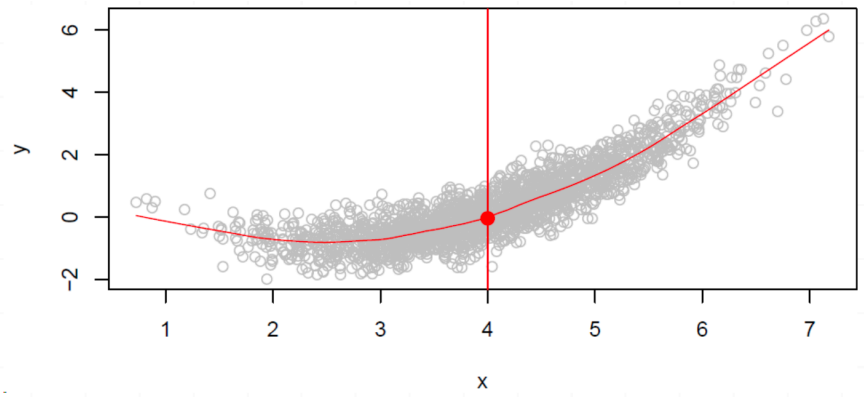
3-1. Statistics

- understanding population
    - functional class를 가정하면 충분통계량만으로 결정 가능
    - e.g. center, dispersion, orientation(correlation), skewness, kurtosis, ...
- summarize data
    - quantiles
    - five-number summary
    - robust statistics: median, IQR, MAD
    - visualization

4. Supervised Learning

4-1. Setup

- statistical model: $Y = f(X) + \epsilon$
    - prediction
    - inference
- in general, the conditional expectation minimizes MSE



- At a given value, say $X = 4$, there may be many possible Y values.
- What is the **best** prediction of Y at $X = 4$?
- In general, the conditional expectation minimizes mean squared error:

$$f(x) = \arg\min_g E[(Y - g(X))^2 \mid X = x] = E(Y \mid X = x),$$

  called the *regression function*.
- Hence, $E(Y \mid X = x)$ is the *optimal predictor* under squared loss.

4-2. Parametric Approach for Estimating f

- 절차
  - specify a functional form for f
  - estimate the parameters from data
- f를 low-dimensional parameter space로 제약
  - 해석 용이하지만 model misspecification 위험

4-3. Non-Parametric Approach for Estimating f

- e.g. KNN, kernel regression, spline smoothing, Gaussian Process regression
- 장단점
  - flexibility 높음
  - 많은 관측치 필요

4-4. Assessing Model Accuracy

- training error(MSE) vs. test error(MSE)
- bias-variance trade-off

    - flexibility vs. interpretability
- test error 외의 기준: penalizes model complexity (number of variables)
    - adjusted R^2
    - AIC
    - BIC
- AIC vs. BIC

	AIC	BIC
Penalty	$2k$	$k \log n$
Goal	Prediction	True model selection
Model size	Larger models	Smaller models
Consistency	No	Yes

- AIC focuses on predictive performance.
- BIC prefers simpler models when n is large.
- In practice:
    - Use AIC for *prediction*.
    - Use BIC when identifying the *true model structure*.

5. Resampling

5-1. Cross-Validation

- test error estimation 하는 방식
- validation set approach
    - test error를 overestimate 할 수 있음 (어떻게 split 되는지에 따라)
- LOOCV
    - less bias (대부분의 data set 이용)
    - computationally intensive (n번 모델 적합)
- K-fold CV
    - randomly divide the data into K equal-sized parts
    - typical choice is $K = 5 \text{ or } 10$

    - LOOCV에서는 fold끼리 highly correlated → average가 높은 분산
    - 변수선택 + 적합을 같이해야 함: fold k를 validation으로 뺀 다음 나머지 데이터만으로 변수 선택 → 그 변수로 모형 fit → fold k에서 error 계산 → 이를 K번 반복

$$CV_{(k)} = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k$$

- CV는 무엇을 추정하는가?
    - prediction risk $R(\hat{f}) = E[(Y - \hat{f}(X))^2]$
    - LOOCV is asymptotically equivalent to AIC
- limitation
    - true model을 식별하지는 않음 (이를 위해서는 BIC)
    - LOOCV는 overly complex 한 모델을 선호할 수 있음
- CV → fold 별 MSE 구할 수 있음 + SE

5-2. Bootstrap

- estimator의 불확실성 추정
    - standard error

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \overline{\hat{\alpha}}^{*}\right)^2}$$

    - confidence intervals (bootstrap percentile)
- prediction error estimation으로는 쓸 수 없음 (each bootstrap → overlap with the original data(training sample) → underestimate true prediction error)

6. Regression

6-1. Regression

Single Predictor X

- least squares approach → minimize RSS (residual sum of squares)
- standard error: how estimator varies under repeated sampling → CI 계산

$$\hat{\beta}_1 \pm t_{1-\alpha/2,df} \cdot \mathrm{SE}(\hat{\beta}_1)$$

- hypothesis testing: compute t-statistic → compute p-value

$$t_{\text{stat}} = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)} \sim t_{n-2}$$

- overall accuracy of the model
  - RSE (ϵ의 standard deviation의 추정량)
  - R^2

$$\text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} \quad R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}}$$

Multiple Linear Regression

- 회귀계수의 해석
  - predictor 간의 상관이 문제를 일으킬 수 있음
  - e.g. β_1, β_2 → $\hat{\beta}_2$는 β_1이 설명하는 영역 제외하고 얼마나 설명하는가?

Some Important Questions

- Is at least one predictor useful? → F-statistic (대립가설: 적어도 하나의 $\beta_j \neq 0$)
- deciding on the important variables → variable selection
- how well does the model fit the data? → RSE, R^2
- how accurate is our prediction → 세 종류의 불확실성
  - reducible error: 계수의 정확성 (β와 $\hat{\beta}$의 차이)
  - model bias: 선형모형 $f(X)$는 언제나 근사 (실제는 비선형일수도)
  - irreducible error: ϵ의 존재 → $\hat{y}$가 아무리 좋아도 y와 다를 수밖에 없음

$$
\begin{aligned}
E\left[(y_0 - \hat{f}(x_0))^2\right] &= E\left[(f(x_0) + \epsilon - \hat{f}(x_0))^2\right] \\
&= E\left[(f(x_0) - \hat{f}(x_0))^2\right] + 2E[\epsilon] \cdot E\left[f(x_0) - \hat{f}(x_0)\right] + E[\epsilon^2] \\
&= E\left[(f(x_0) - \hat{f}(x_0))^2\right] + \mathrm{Var}(\epsilon) \qquad (\because E[\epsilon] = 0)
\end{aligned}
$$

Let $\mu = E[\hat{f}(x_0)]$. Then:

$$
\begin{aligned}
E\left[(f(x_0) - \hat{f}(x_0))^2\right] &= E\left[(f(x_0) - \mu + \mu - \hat{f}(x_0))^2\right] \\
&= (f(x_0) - \mu)^2 + 2(f(x_0) - \mu)\underbrace{E[\mu - \hat{f}(x_0)]}_{=0} + E\left[(\hat{f}(x_0) - \mu)^2\right] \\
&= \mathrm{Bias}(\hat{f}(x_0))^2 + \mathrm{Var}(\hat{f}(x_0))
\end{aligned}
$$

$$
\therefore \quad E\left[(y_0 - \hat{f}(x_0))^2\right] = \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\text{variance}} + \underbrace{\mathrm{Bias}(\hat{f}(x_0))^2}_{\text{bias}^2} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{irreducible error}}
$$

Qualitative Predictors

- categorical → dummy variable

- Then both of these variables can be used in the regression equation, in order to obtain the model

$$
y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i \text{ th is from the South} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i \text{ th is from the West} \\ \beta_0 + \epsilon_i & \text{if } i \text{ th is from the East} \end{cases}
$$

Gauss-Markov Theorem

Gauss–Markov Theorem

Suppose

$$y = X\beta + \varepsilon, \qquad E(\varepsilon) = 0, \qquad \mathrm{Var}(\varepsilon) = \sigma^2 I,$$

and X has full column rank. For any fixed vector c, consider

$$\psi = c^\top \beta.$$

Then among all *linear unbiased* estimators of ψ, the estimator

$$\hat{\psi} = c^\top \hat{\beta}, \qquad \hat{\beta} = (X^\top X)^{-1} X^\top y,$$

has the *minimum variance*. It is unique.

- least squares is the BLUE (Best Linear Unbiased Estimator)

Extensions of the Linear Model

- interactions
- non-linear (polynomials) → residual plot으로 체크

6-2. Regression: Potential Problems

Common Issues

Data Problems	• outliers • measurement error • missing data
Model Problems	• omitted variable bias • model misspecification • multicollinearity
Error Problems	• normality • heteroskedasticity • autocorrelation

Non-Gaussianity

- if the errors are not normal, $\hat{\beta}$ is not exactly normal
- but by CLT, $\hat{\beta} \simeq N(\beta, \sigma^2 (X^\top X)^{-1})$ for large n
  - t-검정과 신뢰구간 근사적으로 valid
  - 혹은 bootstrap inference 이용 (resampling → $\hat{\beta}^*$ 계산 → empirical distribution)

Outliers

- robust regression (via alternative objectives)
  - least absolute deviation
  - Huber regression (절충)
  - least trimmed squares (가장 작은 h개 잔차만 이용)

    - **Least Absolute Deviation (LAD)**

      $$\hat{\beta} = \arg\min_{\beta} \sum |y_i - x_i^\top \beta|$$

    - **Huber Regression**

      $$\hat{\beta} = \arg\min_{\beta} \sum \rho(r_i), \qquad r_i = y_i - x_i^\top \beta$$

      where

      $$\rho(r) = \begin{cases} \frac{1}{2} r^2 & |r| \le c \\ c|r| - \frac{1}{2} c^2 & |r| > c \end{cases}$$

    - **Least Trimmed Squares (LTS)**

      $$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{h} r_{(i)}^2$$

      using only the smallest squared residuals.

- breakdown point: measures the robustness of an estimator (오염의 최소 비율)
  - high-breakdown estimators can tolerate nearly 50% contamination

- It is the smallest fraction of contamination that can make the estimator arbitrarily large.

Method	Breakdown Point
OLS	0
LAD	0
Huber M-estimator	0
Least Median of Squares	0.5
Least Trimmed Squares	≈ 0.5

Measurement Errors

- true model: $Y = \beta_0 + \beta_1 X^* + \epsilon$
- predictor is measured with error: $X = X^* + u$
- attenuation bias: slope is attenuated toward zero

$$E(\hat{\beta}_1) = \beta_1 \frac{Var(X^*)}{Var(X^*) + Var(u)}$$

- 대처 방법
  - 도구변수: X^*와는 상관되어 있지만 u와는 독립인 변수 이용
  - repeated measurements: 같은 대상 여러 번 측정해서 평균
  - erros-in-variables models: measurement process를 모델링
  - SIMEX (simulation-extrapolation): 인위적으로 노이즈 추가 후 노이즈=0 일 때로 외삽
  - validation data: X^*를 직접 관측할 수 있는 부분집합 이용

6-3. Explainable Model

Subset Selection

- best subset selection → 2^p
- stepwise selection (using CV prediction error, C_p, AIC, BIC, adjusted R^2 - complexity 보정)
  - forward selection → $\frac{p(p+1)}{2}$
  - backward elimination → $1 + \frac{p(p+1)}{2}$
    - $n < p$인 상황에서는 사용하기 어려울 수도

Shrinkage

- ridge regression
  - tuning parameter λ: CV로 선택 (클수록 penalty 증가)
  - standardize 한 뒤 적용하는 것이 필요
  - ridge > OLS인 이유: bias-variance trade-off
  - $n = p$인 경우 불가능
  - β^*가 dense 하지만 ℓ_2-norm이 작을 때 유용

$$\hat{\beta}_R = (X^\top X + \lambda I)^{-1} X^\top y$$

- LASSO
  - variable selection
  - tends to produce sparse models → response가 적은 설명변수로 결정되면 LASSO가 우위

- theoretical view
  - 특정한 조건(restricted eigenvalue 등) 하에서 LASSO는 consistent estimation 가능
  - exact support recovery 하기 위해서는 강한 조건 필요 (e.g. irrepresentable condition, beta-min condition)

Dimension Reduction

- key idea
  - construnct $M < p$ transformed predictors and fit the linear model
  - 각각의 Z_m은 feature space의 direction을 대표
  - β의 차원 축소 → 분산 감소, 예측 성능 향상

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j$$

- PCR
  - $Cov(X)$ → eigendecomposition → largest eigenvalue의 eigenvector 선택
  - 각각의 principal component는 직교
  - PC 생성 전 standardization 필요
  - limitations
    - predictor를 잘 설명하는 방향 ≠ response 예측을 위한 최적의 방향
    - 숫자의 의미 해석 어려움
    - prediction accuracy 높지 않을 수 있음 (X_1, X_2 겹치는 부분 있지만 Y 설명하지 못하는 경우)
- PLS
  - new set of predictors를 기존 변수의 선형결합으로 생성
  - PCR과 달리 새 설명변수를 supervised way로 식별 (Y를 이용) → response/predictor 모두를 잘 설명하는 방향 찾는 것
  - assumption: β^* lies in a low-dimensional structure
  - mechanism: iterative projection

- The first PLS direction is obtained by solving

$$w_1 = \arg\max_{\|w\|=1} \mathrm{Cov}(Xw, Y)^2.$$

- The first component is then

$$Z_1 = Xw_1.$$

- Subsequent directions are obtained sequentially by removing the variation explained by previous components.
- Hence, PLS finds directions that maximize *covariance with the response*.

Algorithm (Sequential Construction)

- Standardize X and Y
- For $m = 1, \cdots, M$:
  - Compute direction

$$w_m \propto X^{(m-1)T} Y^{(m-1)}$$

  - Form component

$$Z_m = X^{(m-1)} w_m$$

  - Regress and update residuals:

$$Y^{(m)} = Y^{(m-1)} - \hat{\beta}_m Z_m$$

  - Deflate predictors:

$$X^{(m)} = X^{(m-1)} - Z_m p_m^T$$

PLS 이해

6-4. Non-Linear Regression

Basis Function Based Regression

- polynomial regression
  - d → cross-validation으로 정함

- step-function (piecewise-constant regression)

  - Another way of creating transformations of a variable - cut the variable into distinct regions.

$$C_0(X) = I(X < c_1),\ C_1(X) = I(c_1 \le X < c_2), ...,\ C_K(X) = I(c_K \le X),$$

  - For any value of X, $C_0(X) + C_1(X) + ... + C_K(X) = 1$, since X must be in exactly one of the $K + 1$ intervals

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + ... + \beta_K C_K(x_i) + \epsilon_i$$

- basis function-based
  - 기타: fourier series, wavelets, ...

  - Basis function-based regression:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_p b_p(x_i) + \epsilon_i$$

  - Polynomial and piecewise-constant regressions are just a particular kind of a *"basis function"* approach.
  - Instead of writing a polynomial regression as

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_p x_i^p + \epsilon_i$$

  - We can write it as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_p b_p(x_i) + \epsilon_i$$

    where

$$b_j(x_i) = x_i^j$$

  - Similarly, for piecewise-constant regression,

$$b_j(x_i) = I(c_j \le x_i < c_{j+1})$$

Regression Splines

- piecewise polynomials
- cubic splines

- *A cubic spline with knots at $\xi_k, k = 1, ..., K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.*
- We can represent this model with truncated power basis functions

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i,$$

$$\begin{aligned} b_1(x_i) &= x_i, & b_2(x_i) &= x_i^2 \\ b_3(x_i) &= x_i^3 & b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, .., K \end{aligned}$$

where

$$(x_i - \xi_k)_+^3 = \begin{cases} (x_i - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$

- natural cubic splines → extrapolates linearly beyond the boundary knots
  - determination of K (number of knots): CV, heuristic

Smoothing Splines

Consider this criterion for fitting a smooth function $g(x)$ to some data:

$$\underbrace{\underset{g \in S}{\text{minimize}} \sum_{i=1}^{n} (y_i - g(x_i))^2}_{\text{RSS}} + \underbrace{\lambda \int g''(t)^2 dt}_{\text{roughness penalty}}$$

- The first term tries to make $g(x)$ match the data at each x_i.
- The second term controls how wiggly $g(x)$ is. It is modulated by the *tuning parameter* $\lambda \geq 0$.
  - The smaller λ, the more wiggly the function, eventually interpolating y_i when λ=0.
  - As $\lambda \to \infty$, the function $g(x)$ becomes linear.
  - λ can be determined using k-fold CV.

Local Regression

Algorithm 7.1 *Local Regression At* $X = x_0$

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.
2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.
3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2. \tag{7.14}$$

4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

GAM

GAM allows for flexible nonlinearities in several variables, but retains the additive structure of linear models.

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i.$$

- pros
  - non-linear fit
  - additive → predictor 각각의 효과 볼 수 있음
  - more accurate predictions
- cons
  - interactions can be missed

7. Classification

7-1. Classification Errors

Classification Problem

- regression → MSE; classification → error rate $\frac{1}{n}\sum I(y_i \neq \hat{y}_i)$
- Bayes optimal classifier
  - $\arg\min_C$ misclassification error $P(C(x) \neq y)$
  - $\arg\max_k$ conditional class probabilities $P(Y = k \mid X = x)$
- Bayes error rate: lowest possible error rate that could be achieved
- SVM은 $C(x)$에 대한 모델 만들고, LR/GAM은 조건부확률 모델링

K-NN

- estimation

$$\hat{f}(x) = \frac{1}{K} \sum_{x_i \in N_K(x)} y_i$$

- bias-variance trade-off: k가 작을수록 flexibility/complexity가 높아진다
- ROC-AUC: false positive rate과 true positive rate을 plot으로 나타냄
- cross-validation: data를 K개의 fold로 나눈 뒤, $CV_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$ where $\mathrm{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$

7-2. Discriminative Methods for Classification

Generative vs. Discriminative

Discriminative : directly estimate $P(Y = c_k \mid X = x)$	Generative : (1) Build a model then (2) estimate
• KNN • logistic regression • classification tree (CART) • ensenble methods • support vector machine	• LDA • QDA • Naive Bayes

Logistic Regression

- GLM: Y의 종류에 따라
  - continuous
  - binary/categorical → logistic (μ)
  - count → Poisson (p)
  - skewed-continuous → Gamma / Exp ($\frac{1}{\lambda}$)
- Logit transformation (two-class case: c_1, c_2)

$$\log \frac{P(Y=c_1 \mid X)}{P(Y=c_2 \mid X)} = \log \frac{P(x_i)}{1-P(x_i)} = \beta_0 + \beta_1 X$$

$$P(Y=c_1 \mid X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}, \quad P(Y=c_2 \mid X) = \frac{1}{1+e^{\beta_0+\beta_1 X}}$$

- limitation of linear regression: class 간 간격 동일하다고 간주 → multiclass logistic regression 혹은 discriminant analysis
- estimation: maximum likelihood (= 이 파라미터에서 관측된 0, 1이 나올 확률)
  - partial derivative → $p+1$개의 식
  - nonlinear in β → Newton-Raphson Algorithm (e.g. iteratively reweighted least squares, IRLS)

$$\begin{aligned}\ell(\beta_0, \beta) &= \log \left\{ \prod_{i:\, y_i=1} p(x_i) \prod_{j:\, y_j=0} \big(1-p(x_j)\big) \right\} \\ &= \sum_i y_i(\beta_0+\beta_1 x_i) - \log(1+e^{\beta_0+\beta_1 x_i})\end{aligned}$$

- 해석: 1 unit (X) increase → increase in the log odds by ___ units
- confounding: 변수 추가 시 계수의 부호 바뀜 → multiple logistic regression은 confounding을 통제
- 확장: polynomial / piecewise constant / GAM / regularization (minimize $-\ell(\beta_0, \beta) + \lambda||\beta||_i$)

Multi-Class Classification Problem

- multiclass (K classes) logistic regression = multinomial regression

$$\log \frac{P(Y=k \mid X)}{P(Y=K \mid X)} = \beta_{k0} + \beta_k X$$

7-3. Generative Models for Classification

Discriminant Analysis

- model the distribution of X
  - LDA: $X \mid Y=c_k \sim N(\mu_k, \sigma^2)$
  - QDA: $X \mid Y=c_k \sim N(\mu_k, \sigma_k^2)$
- Bayes Theorem

$$\Pr(Y=k \mid X=x) = \frac{\Pr(X=x \mid Y=k) \cdot \Pr(Y=k)}{\Pr(X=x)}$$

- discriminant analysis (π_k = marginal / prior probability)

$$\Pr(Y=k \mid X=x) = \frac{\pi_k \, f_k(x)}{\sum_{l=1}^{K} \pi_l \, f_l(x)} \propto \pi_k \, f_k(x)$$

- classify a new point according to which probability is highest

$$\hat{C}(x) = \arg\max_k \Pr(Y=k \mid X=x) = \arg\max_k \pi_k \cdot f_k(x)$$

- advantages
  - 클래스가 '잘 나뉘어 있는 경우', logistic regression의 추정량은 불안정
  - n이 작고 클래스별 X의 분포가 근사적으로 정규분포일 때
  - response classes가 2개 이상일 때

LDA

- assume $\sigma_k = \sigma$

$$p_k(x) = \Pr(Y=k \mid X=x) = \frac{\pi_k \dfrac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\displaystyle\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

- discriminant score: ($\log p_k(x)$에서 k에 관한 항만)

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

- decision rule: $\hat{C}(x) = \arg\max_k \delta_k(x)$
  - note that $\delta_k(x)$ is a linear function of x
- estimation:

$$\hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} x_i, \quad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$$

- $p > 1$
  - $f(x) = \dfrac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu))$
  - $\delta_k(x) = x^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log(\pi_k)$
  - linear decision boundary
- prior를 조정 vs. varying the threshold (로지스틱)

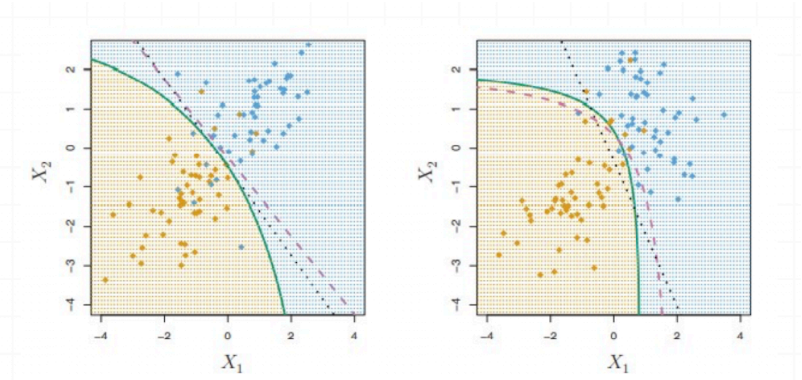
QDA

- Σ_k's are allowed to be different

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k = \underbrace{x^T W_k x + x^T w_k + b_k}_{\text{quadratic form}}$$

- decision boundary:

$$\{x : x(W_k - W_{k'})x + x(w_k - w_{k'}) + (b_k - b_{k'}) = 0\}$$



- The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries

Naive Bayes

- assume independence among input variance when class is given

$$f(x_1, \ldots, x_p \mid y = c_k) = f(x_1 \mid c_k) \cdots f(x_p \mid c_k)$$

- Gaussian NB assumes each Σ_k is diagonal

$$\delta_k(x) \propto \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right] = -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log\sigma_{kj}^2\right] + \log\pi_k$$

- useful when p is large: p-dimensional density function 추정의 어려움 해소

7-4. Comparison of Classification Methods

LR vs. LDA

- For LDA, the log-posterior odds between class k and class K is linear

$$\begin{aligned}\log\left(\frac{\Pr(y=c_k \mid x)}{\Pr(y=c_K \mid x)}\right) &= x^T\Sigma^{-1}(\mu_k-\mu_K)+\log\frac{\pi_k}{\pi_K} \\ &\quad -\frac{1}{2}(\mu_k+\mu_K)^T\Sigma^{-1}(\mu_k-\mu_K) \\ &= \alpha_{k0}+\alpha_k^T x\end{aligned}$$

- Logistic model has linear logits by construction

$$\log\frac{\Pr(y=c_k \mid x)}{\Pr(y=c_K \mid x)} = \beta_{k0}+\beta_k^T x$$

- *The same form.* Are they the same estimator?

- linearity
  - LDA: Gaussian assumption + common variance assumption
  - LR: by construction
- common component

  - The joint density of (X, Y) is

$$\Pr(X=x, Y=c_k) = \Pr(X=x)\cdot\Pr(Y=c_k \mid x)$$

    where $\Pr(x)$ is the marginal density of the input x.

  - For both LDA and logistic regression, the second term $\Pr(Y=c_k \mid x)$ has the same logit linear form

$$\Pr(Y=c_k \mid x) = \frac{\exp(\theta_{k0}+\theta_k^T x)}{1+\sum_{k'=1}^{K}\exp(\theta_{k'0}+\theta_{k'}^T x)}$$

- different assumption about $\Pr(X)$
  - LR: marginal density of X unspecified → fewer assumptions, more general
  - LDA: Gaussian mixture density 가정
- parameter estimation
  - LR: maximizing the conditional likelihood $\Pr(Y=c_k \mid x)$, marginal density $\Pr(X)$ is ignored
  - LDA: maximizing the full likelihood $\Pr(X=x, Y=c_k)$, marginal density plays a role

- remarks
  - LDA 계산 더 쉬움
  - true $f_k(x)$가 Gaussian → LDA가 더 나음
  - robustness: LR (decision boundary에서 먼 포인트 down-weight); LDA는 모든 data points 이용

QDA vs (LDA, LR, KNN)

- QDA: non-parametric KNN과 linear LDA, LR의 절충
- true decision boundary가
  - linear → LDA, logistic
  - moderately non-linear → QDA
  - more complicated → KNN

8. Tree-Based Models

8-1. Tree

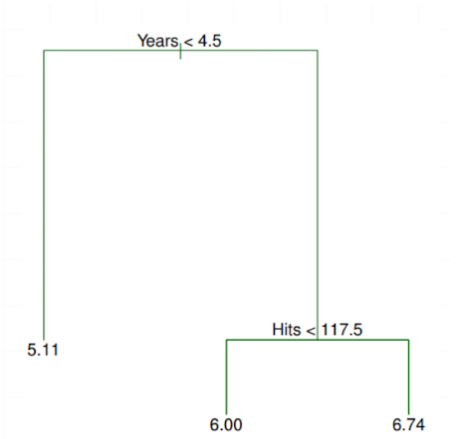
Decision-Tree Methods

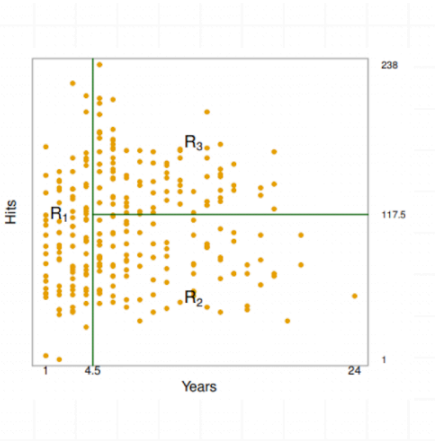
- predictor space를 stratifying 혹은 segmenting
- splitting rule을 tree로 요약
- 장단점
  - pros: simple, useful for 해석 (prediction accuracy는 상대적으로 낮음)
  - cons: 해석이 복잡

8-2. Tree for a Regression Problem

Example

- log(salary) ~ years_played + hits





- leaf의 숫자 = mean of the response
- R_1, R_2, R_3: terminal nodes
- points along the tree: internal nodes (e.g. `years` < 4.5, `hits` < 117.5)
- 해석
  - `years` 가 가장 중요한 변수
  - 저연차 선수는 `hits` 가 상대적으로 적은 영향

Tree-Building Process

- goal: find boxes $R_1, R_2, \ldots, R_J$ that minimizes the RSS

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean response for the training observations within the j-th box.

- 모든 partition을 고려하는 것은 불가능
  - top-down search
  - greedy search

Tree-Pruning

- overfit 방지
- RSS 감소 폭을 이용한 방법 → 지금 worthless해 보이는 split이 나중에는 RSS를 크게 줄일 수도 있음 (short-sighted)
- 대신 매우 큰 tree 만든 다음 prune 하여 subtree 만드는 방식

- Consider a sequence of trees indexed by a nonnegative tuning parameter α. For each value of α, there corresponds a subtree $T \subset T_0$ such that

$$\min \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

  - $|T|$ indicates the number of terminal nodes of the tree T
  - R_m is the rectangle corresponding to the m-th terminal node
  - $\hat{y}_{R_m}$ is the mean of the training observations in R_m.

- tuning parameter α
  - trade-off: α가 증가할수록 subtree 작아짐
  - cross-validation을 통해 정함
  - 다시 full data에서 정한 α에 맞는 subtree 찾음

8-3. Tree for a Classification Problem

Classification Tree

- prediction → most commonly occurring class in the region
- RSS 대신 classification error rate 이용
  - not sufficiently sensitive for tree-growing

$$E = 1 - \max_k (\hat{p}_{mk}).$$

Here, $\hat{p}_{mk}$ represents the proportion of training observations in the m-th region that are from the k-th class.

- Gini Index: measure of total variance across the K classes

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Cross-Entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

Advantages and Disadvantages of Trees

- tree vs. linear model
  - linear model도 basis 잘 고르면 되지만 그러기 어렵다.
- advantages
  - explain, display 하기 쉬움
  - 인간의 decision-making 절차와 더 유사
  - qualitative predictor 이용 가능 (dummy 화 할 필요 없음)
- disadvantages
  - predictive accuracy 떨어짐
  - robustness 낮음
- aggregation of many decision trees → 예측 성능 향상

8-4. Ensemble Classifier

Bagging

- general-purpose procedure for reducing the variance of a statistical learning method
  - $Z_1, \ldots, Z_n$ → $\bar{Z}$의 분산이 1/n
  - multiple training set을 bootstrap을 이용해 확보
- B different bootstrapped training data sets
  - train our method on the b-th set → $\hat{f}^{*b}(x)$
  - then average all the predictors: $\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{i=1}^{B}\hat{f}^{*b}(x)$
- classification tree에서
  - B개의 tree에서 prediction 기록
  - 가장 많이 나온 class를 최종 예측값으로
  - 예측 성능은 향상되지만(more flexible decision boundaries) 해석은 더 어려워짐
- Out-of-Bag(OOB) error estimation
  - test error 추정 방법
  - 평균적으로 각각의 bagged tree는 sample의 2/3을 이용
  - 나머지 1/3 → OOB sample
  - estimation: i번째 관측치가 OOB인 tree ($B/3$개) 평균 → LOOCV error for bagging if B is large

Random Forest

- bagging과 다르게 tree를 decorrelate 시킴 → reduces the variance when we average the trees
  - split 할 때 m개의 predictor를 random selection (그 중 하나 선택)

    - 매 split 마다 새로운 조합 이용
    - 주로 $m \simeq \sqrt{p}$
- RF ($m < p$) vs. Bagging ($m = p$) → RF가 더 나음
- 충분히 큰 B → overfitting 방지

Boosting

- bagging과 유사하지만 tree가 sequentially grown: 이전 tree의 정보(residual) 이용
- learns slowly
- tree의 개수: B
    - bagging이나 RF와 달리 B가 너무 커지면 overfit
- 각 tree의 split 횟수 (complexity): d
- shrinkage parameter λ: slows the process down even further (더 많고 다양한 tree 이용되도록 함)
    - 주로 0.01 혹은 0.001
    - λ 작을수록 B 커져야 성능 좋음
- tuning parameter: cross-validation

Boosting - Toy Example

AdaBoost (Adaptive Boosting)

- 학습에 이용된 개별 분류기에 서로 다른 가중치 (샘플링 확률 조정)
    - 틀린 샘플이 학습 데이터에 더 많이 등장하게 함

(i) 반복추출로 데이터 추출 D_b
(ii) b-번째 모델 h_b을 D_b을 기반으로 학습
(iii) h_b의 오분류율(ϵ_b)과 가중치(α_b)를 계산

$$\alpha_b = \frac{1}{2} \log \frac{1 - \epsilon_b}{\epsilon_b}$$

(iv) 데이터 분포 업데이트:

$$D_{b+1} = \frac{D_b \exp(-\alpha_b y h_b(x))}{Z_b}$$

단, Z_b는 정규화 상수

- 최종 예측치:

$$H(x) = \text{sign}\left(\sum_{t=1}^{B} \alpha_b h_b(x) \right)$$

Gradient Boosting Machine (GBM)

- GD를 이용하여 손실함수를 최소화하는 방향으로 학습

  GBM에서 손실함수란?

(i) b-번째 모델 f_b을 D_b을 기반으로 학습

(ii) 잔차(R_b)를 계산

$$y = f_b(x) + R_b \iff R_b = y - f_b(x)$$

(iii) 잔차를 기반으로 y 업데이트, $y = R_b$

- 최종 예측치:

$$\widehat{y} = \sum_{b=1}^{B} \hat{f}_b(x) + R_B$$

- standard GBM의 한계
  - overfit 가능성 (tree 개수 너무 많은 경우)
  - tree complexity에 대한 직접적 정규화 없음
  - sequential fitting이 computationally inefficient

Extreme Gradient Boosting (XGBoost)

- regularized version of GBM
  - tree의 depth에 대해 penalty 부여

Regularized Objective

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{b=1}^{B} \Omega(f_b),$$

where

$$\Omega(f_b) = \gamma T_b + \frac{\lambda}{2} \sum_{j=1}^{T_b} w_{bj}^2.$$

Here, T_b is the number of leaves in tree f_b, and w_{bj} is the score assigned to leaf j.

- GBM에 비해 optimization + computation 좋아짐
  - **regularization**으로 overly complex tree에 대해 penalty → reduce overfitting
  - **second-order optimization**
  - system optimization

LightGBM

- efficient implementation of GBM (단점: accuracy는 낮아짐)
  - **histogram-based learning**: bins continuous features

    - **leaf-wise tree growth**
    - efficient computation
- LightGBM은 scalability에 초점
    - histogram-based splitting으로 faster training
    - leaf-wise growth → better accuracy (reduce loss 더 공격적으로)
    - efficient for large datasets / high-dimensional features

CatBoost

- designed to handle categorical data efficiently
    - **native categorical handling**: 수동 전처리 필요 없음
    - **ordered boosting**: reduces target leakage and prediction shift

      Ordered Boosting이란?

    - robust training: less sensitive to overfitting
- 장점
    - reduced bias (ordered boosting으로 정보 누수 방지)
    - better categorical modeling (target statistic 이용)
    - improved robustness

- 추가 장점: XGBoost, LightGBM은 결측치 처리에 유리

8-5. Importance Measure

Feature Importance in Tree-Based Methods

- each split → reduces a loss function (e.g. RSS, Gini)
- 중요한 변수 = split에서 자주 등장 + 큰 improvements
- ensemble에서 importance는 모든 tree에 대해 합산
- higher importance ⇒ stronger influence on prediction (causal 아님)

Importance Measure

$$\text{Imp}(X_j) = \sum_{\text{splits on } X_j} \text{reduction in loss.}$$

- Regression: reduction in RSS
- Classification: reduction in Gini or cross-entropy

$$\text{Final importance} = \frac{1}{B}\sum_{b=1}^{B} \text{Imp}_b(X_j)$$

Limitations of Tree-Based Importance

- fast, but biased toward variables with many possible splits
- depends on model structure → not comparable across models

Permutation Importance

- measures how prediction error changes when a feature is randomly shuffled
  - based on prediction error

Idea

Break the relationship between X_j and Y, and observe the increase in prediction error.

$$\text{Imp}(X_j) = \text{Error}_{\text{perm}(j)} - \text{Error}_{\text{original}}$$

SHAP (Shapley Additive Explanation)

- explains predictions by fairly distributing contributions among features

Key Idea

Each feature contributes to prediction based on its marginal contribution across all subsets.

$$f(x) = \phi_0 + \sum_{j=1}^{p} \phi_j$$

- ϕ_j: contribution of feature X_j
- Provides **local explanations** (per observation)
- Consistent and theoretically justified

9. Support Vector Machine

9-1. Support Vector Classifier

Motivation

- two classes, two predictors
  - two classes가 linearly separable 하다면?
  - 두 class를 가장 크게 분할하는 line 찾는 것이 자연스러움 (the points are as far from the line as possible)
- hyperplane in p dimensions: flat affine subspace of dimension $p-1$
- equation for a hyperplane: $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0$
- 아이디어의 확장
  - soften what we mean by “separates”
  - enrich, enlarge the feature space

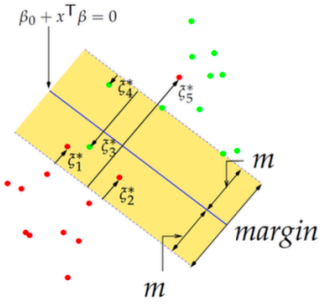
Maximal Margin Classifier

- m: minimum perpendicular distance between each point and the separating line
- find the line which maximizes m → optimal separating hyperplane
- constrained optimization problem
  - maximize m, subject to $\frac{1}{||\beta||} y_i(\beta_0 + x_i\beta) \geq m$
- 문제
  - non-separable data ($N < p$ 아니라면 대부분..)
  - noisy data

Support Vector Classifier

- small subset of the observations are on the wrong side of the margin

- Let ζ_i represent the amount that the i-th point is on the wrong side of the margin.



$$\max_{\beta_0,\beta_1} \quad m$$

$$\text{subject to} \quad \frac{y_i}{\|\beta\|}(\beta_0 + x_i\beta) \geq m(1-\zeta_i)$$

$$\zeta_i \geq 0, \quad \sum_i \zeta_i \leq C$$

- ζ_i: slack variable
- C: tuning prameter

- slack variable(ξ_i) and tuning parameter (C)

- $\zeta_1, \zeta_2, \cdots, \zeta_n$ allow individual observations to be on the wrong side of the margin or the hyperplane
  - If $\zeta_i = 0$, then the i-th observation is on the correct side of the margin
  - If $1 \geq \zeta_i > 0$, then the i-th observation is on the wrong side of the margin
  - If $\zeta_i > 1$, then the i-th observation is on the wrong side of the hyperplane
- C can be viewed as a budget for the amount that the margin can be violated by the n observations.
- As the budget C increases, we become more tolerant of violations to the margin, and so the margin will widen. Conversely, as C decreases, we become less tolerant of violations to the margin and so the margin narrows.

- when C is large → high tolerance for obs being on the wrong side of the margin → margin will be large
  - CV로 설정

9-2. Support Vector Machines

Non-Linear Decision Boundary

- feature expansion → SV classifier in the enlarged space
- e.g. cubic polynomials, kernels (SVM)

Inner Products and Kernel

- inner product between vectors: $\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} \cdot x_{i'j}$
- linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

- $n + 1$ parameters → estimation을 위해서는 모든 관측치 쌍에 대한 $\binom{n}{2}$개의 inner products $\langle x_i, x_{i'} \rangle$ 가 필요
- 대부분의 $\hat{\alpha}_i$는 0이 될 수 있음 → S = support set of indices i s.t. $\hat{\alpha}_i \neq 0$

$$f(x) = \hat{\beta}_0 + \sum_{i \in S} \hat{\alpha}_i \langle x, x_i \rangle$$

- e.g. polynomial kernel, radial kernel

- Polynomial kernel of degree d,

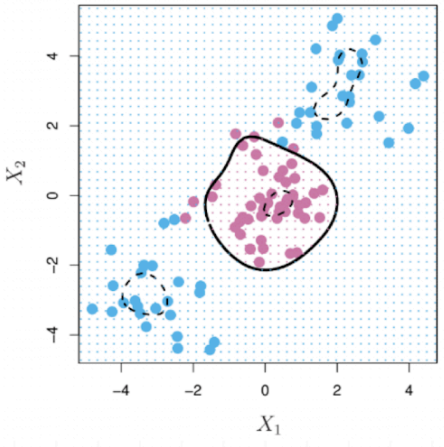
$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

computes the inner-products needed for d dimensional polynomials

- The solution has the form

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(x, x_i).$$

$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right)$ where γ is a positive constant



- Implicit feature space; very high dimensional.
- Controls variance by squashing down most dimensions severely.

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i \, K(x, x_i)$$

Kernel vs. Enlarged Feature Space

- advantages of using a kernel
  - computational complexity 감소
  - some kernel은 enlarged features로 표현 불가능

9-3. SVM with More than Two Classes

SVM with $K > 2$

- OVA (one versus all) vs. OVO (one versus one)
  - if K is not too large, use OVO

- Two most popular solutions: (1) *OVA* and (2) *OVO*

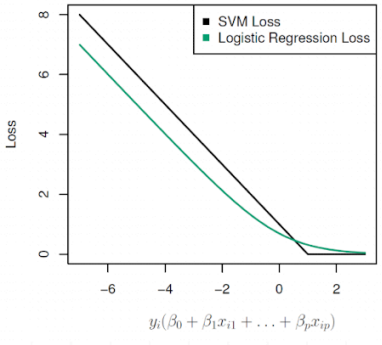
OVA One versus All. Fit K different 2-class SVM classifiers $\hat{f}_k(x),\ k = 1, \ldots K$; each class versus the rest. Classify x^* to the class for which $\hat{f}_k(x^*)$ is largest.

OVO One versus One. Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_{kl}(x)$. Classify x^* to the class that wins the most pairwise competitions.

SVM vs. Logistic Regression

- SV classifier with $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ can be

$$\underset{\beta}{\text{minimize}}\Big\{ \underbrace{\sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)]}_{\text{hinge loss}} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{\text{penalty}} \Big\}$$



- This has the form *loss plus penalty*.
- The loss is known as the *hinge loss*.
- Very similar to "loss" in logistic regression (negative log-likelihood).

- which to use: SVM or LR
  - class가 (nearly) separable → SVM (+ LDA) > LR

- not separable → LR (with ridge penalty) ≃ SVM
- wish to estimate probabilities → LR
- nonlinear boundaries → kernel SVM (kernel LR, LDA는 계산 복잡도 높음)

SVM 관련