자료분석 및 실습 - final

Lecture 7: Linear Model Selection and Regularization

7.1. Introduction

-in the regression setting, the standard linear model is $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$

- can be extended to a generalized linear model (GLM)
- can be extended to a non-linear model e.g. splines, GAM

-instead of using least squares, we can use different fitting procedures → better 1) prediction accuracy and 2) model interpretability

-three alternatives:

1. subset selection
2. shrinkage (regularization)
    - reduce variance (bias 늘어나지만 variance 크게 감소)
    - LASSO → can perform variable selection
3. dimension reduction

-same concepts apply to other methods such as the classification models

7.2. Linear Model Selection

Subset Selection

-linear model with *p* predictors → selecting subsets of predictors

- best subset selection
- stepwise model selection

-we use RSS (Residual Sum of Squares), $\sum_i (y_i - \hat{y}_i)^2$ → same as the L2 loss

Best Subset Selection

1. Let M_0 denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation
2. For $k = 1, 2, \ldots, p$:
   a. Fit all $\binom{p}{k}$ models that contain exactlly k predictors
   b. Pick the best among these $\binom{p}{k}$ models, and call it M_k. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model among $M_0, \ldots, M_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

- choose among these p+1 options
- RSS decreases as p increases (low training error)
- but we wish to choose a model that has a *low test error*

Stepwise Selection

-reasons to use stepwise selection

1. for computational reasons, best subset selection cannot be applied with very large p
2. best subset selection may suffer from statistical problems when p is large

- the larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data
- thus an enormous search space can lead to overfitting and high variance of the coefficient estimates

Forward Stepwise Selection

-begins with a model containing no predictors and then adds predictors to the model

-at each step, the variable that gives the greatest additional improvement to the fit is added to the model

-computationally efficient

-however, no guarantee to find the best possible model

1. M_0 denotes the null model
2. For $k = 0, \ldots, p-1$:
    a. Consider all $p-k$ models that augment the predictors in M_k with one additional predictor
    b. Choose the *best* among these $p-k$ models, and call it M_{k+1}. Here *best* is defined as having smallest RSS or highest R^2
3. Select a single best model among $M_0, \ldots, M_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

-consider total $1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models (c.f. best subset selection consider 2^p models)

-선택된 variable이 best subset selection과 다를 수 있음

Backward Stepwise Selection

-begins with the full least squares model containing all *p* predictors

-iteratively removes the least useful predictor

1. M_p denotes the full model
2. For $k = p, p-1, \ldots, 1$:
    a. Consider all k models that contain all but one of the predictors in M_k, for a total of $k-1$ predictors
    b. Choose the *best* among these k models, and call it M_{k-1}. Here *best* is defined as having smallest RSS or highest R^2
3. Select a single best model among $M_0, \ldots, M_p$ using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Hybrid Approaches

-hybrid versions of forward and backward stepwise selection are available

1. add a variable that can improve the model fit (forward selection)
2. try to remove any variables that no longer provide an improvement in the model fit (backward selection)
3. iteratively repeat the above two steps until there is no improvement

-$\simeq p^2$ term으로 증가 → computation 빠름

Choosing the Optimal Model

-to determine the best model: use test error

-two common approaches:

- indirectly estimate test error: making adjustment to the training error to account for the bias due to overfitting
- directly estimate test error: using a cross-validation approach

C_p, AIC, BIC, and Adjusted R^2

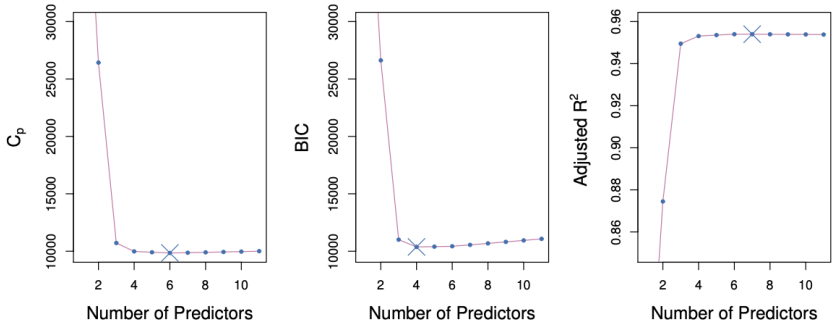


FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

-for a fitted least squares model containing *d* predictors, the C_p estimate of test MSE is:

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

- where $\hat{\sigma}^2$ is an 'estimate' of the variance of the error ϵ
- typically $\hat{\sigma}^2$ is estimated using the full model containing all predictors (error의 variance 가장 작음)
- training error tends to underestimate the test error → adds a penalty of $2d\hat{\sigma}^2$: d가 1 증가하면 RSS $2\hat{\sigma}^2$만큼 감소해야

-AIC criterion is defined for a large class of models fit by maximum likelihood

-in the case of the linear model with Gaussian errors, maximum likelihood and least squares are the same thing

$$AIC = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

-BIC is derived from a Bayesian point of view, but ends up looking similar to C_p

-for the least squares model with *d* predictors, the BIC is given by

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2)$$

-sample size 커질수록 penalty 커짐

-since $\log n > 2$ for any $n > 7$, a heavier penalty on models with many variables

→ result in the selection of smaller models than C_p

-Adjusted R^2 statistic is another popular approach:

$$\text{Adjusted } R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

-unlike C_p, AIC, and BIC, a large value of adjusted R^2 indicates a model with a small test error

-RSS always decreases as *d* increases, but $\frac{RSS}{n-d-1}$ may increase or decrease

Validation and Cross-Validation

-validation set error, cross-validation error: data-driven하게 얻어짐

7.3. Regularization

Basic Ideas

-regularization: technique where constraints are added to a regression model $f_\beta(x) = f(x_{i_1}, ..., x_{i_p}; \beta)$ to prevent overfitting

-model parameter $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ represents the important of the predictors $x = (x_1, \ldots, x_p)$

-consider a loss function $L(\cdot)$ and define the average loss as

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, f_\beta(x_i))$$

-achieve two goals when estimating β:

- minimize the average loss
- avoid overfitting

-overfitting occurs when the model is unnecessarily complex

-in linear models: the number of predictors $\simeq$ model complexity

-first, define the model complexity as

$$\text{complexity}(f_\beta) = \sum_{j=1}^{p} 1(\beta_j \neq 0)$$

-now, we estimate β as follows:

$$\hat{\beta} = \arg\min_{\beta} L(y_i, f_\beta(x_i))$$

- subject to $\sum_{j=1}^{p} 1(\beta_j \neq 0) \leq c$: model complexity가 c보다 작으면서 average loss를 minimize 하도록
- for an integer value c $(< p)$, we can have a fitted model with at most c predictors

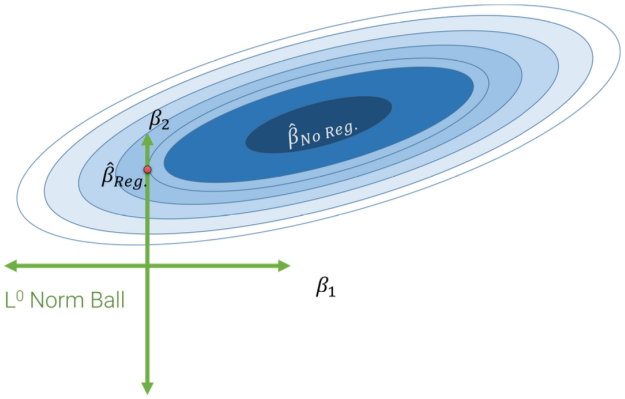
-unfortunately, solving such optimization problem is too hard: NP-hard (including/excluding $x_1, \ldots, x_p$)

-also, the surface of $\sum_{j=1}^{p} 1(\beta_j \neq 0)$ is not convex at all

→ need an approximation for it

L_0 Norm

-restriction: $\sum_{j=1}^{2} 1(\beta_j \neq 0) \leq 1$



-β_1, β_2는 직선 위에서 움직임 → 3번째 원과 만나는 지점에서 β_2 결정, $\beta_1 = 0$

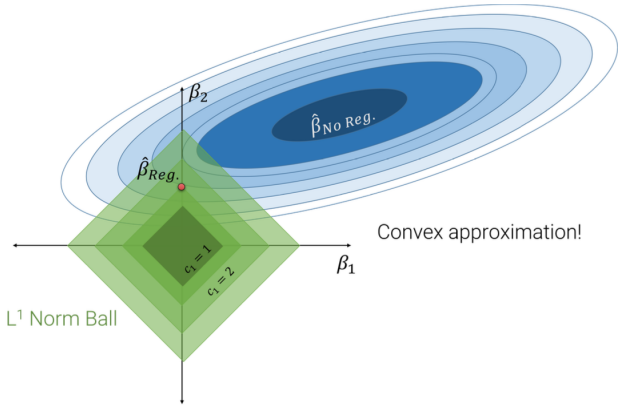
L_1 Norm

-construct a convex approximation?

-different complexity definition:

$$\text{complexity}(f_\beta) = \sum_{j=1}^{p} |\beta_j| \leq c_1$$

-call this the L_1-norm ball (or L_1 penalty/regularization)



-c_1을 크게 할수록 restriction을 열어두는 것 (more complex model)

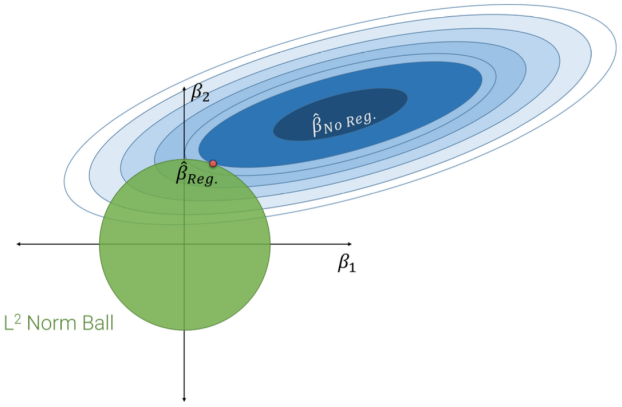
-$c_1 \to 0$: null model or a model that only returns zero (intercept도 0) / $c_1 \to \infty$: ordinary least squares (no effect of restriction)

L_2-Norm Ball

-we can consider a L_2-norm ball instead:

$$\text{complexity}(f_\beta) = \sum_{j=1}^{p} \beta_j^2 \leq c_2$$

-don't be confused: L_2 loss v.s. L_2 penalty → combination of 'L_2 loss' and 'L_1 penalty' is possible



-4분면 어딘가에서 접함 → 0인 estimate 나오지 않음

Elastic Net

-$L_1 + L_2$ Norm (*)

Comparison

-L_0-norm: ideal for **feature selection**, but combinatorically difficult to optimize

-L_1-norm: 많은 경우 꼭짓점에서 solution → feature selection 역할 수행, encourages sparse solutions

-L_2-norm: spreads over features (robust) but doest not encourage spar

-elastic net: compromise between L_1 and L_2 norms

Ridge Regression and LASSO

-two techniques for regularization include ridge regression and the LASSO

- ridge: L_2 penalty, $\sum_{j=1}^{p} \beta_j^2 \leq c_2$
- LASSO: L_1 penalty, $\sum_{j=1}^{p} |\beta_j| \leq c_1$

-these methods are considered part of statistical or machine learning since they automate model selection by shrinking coefficients (for ridge regression) or retaining predictors (for the LASSO) automatically

-such shrinkage may induce **bias** but decrease **variability**

Generic Regularization

-by defining $\text{complexity}(f_\beta) = R(\beta)$,

$$\hat{\beta} = \arg\min_{\beta}[\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \lambda \cdot R(\beta)]$$

- where λ is a regularization hyperparameter or just a tuning parameter (penalty와 loss ftn 중 무엇에 가중치를 더 둘 것인지 결정)

-ridge: minimize $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \lambda \sum_{j=1}^{p} \beta_j^2$

-LASSO: minimize $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \lambda \sum_{j=1}^{p} |\beta_j|$

Shrinkage

-$\lambda \sum_{j=1}^{p} \beta_j^2$ is called a **shrinkage** penlaty

- the penalty is small when β_j's are close to zero
- so, it has the effect of shrinking the estimates of β_j towards zero

-when $\lambda = 0$, the penalty term has no effect, and ridge (or LASSO) regression will produce the least squares estimates

-however, as $\lambda \to \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero

-unlike least squares, ridge regression will produce a different set of coefficient estimates

- say, $\hat{\beta}_\lambda^R$ for each value of λ → $\hat{\beta}$이 λ의 함수가 됨

-selecting a good value for λ is critical

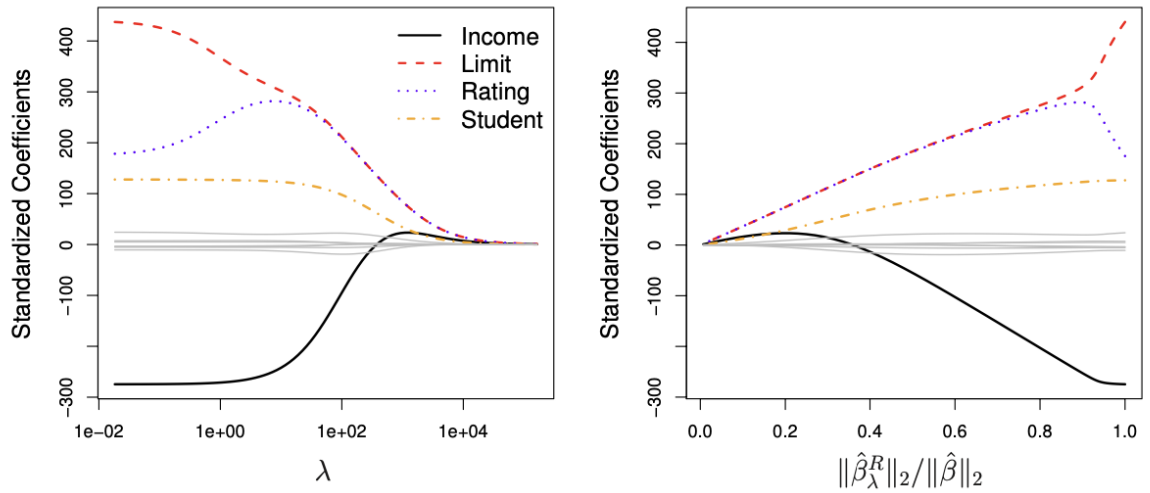


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

-selection of λ: cross-validation

- λ의 값에 따른 모델이 각각 있다고 생각 → 가장 작은 CV error 보이는 것 선택

Why Does Ridge Regression Improve Over Least Squares?

-bias-variance trade-off!

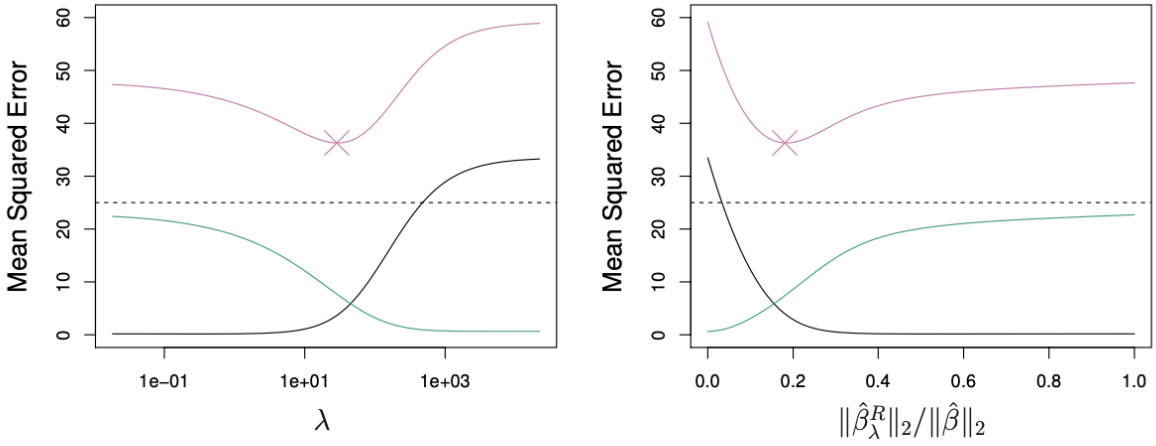


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

-model complexity $\simeq \lambda$: least squares에 비해 bias는 조금 증가하지만 variance 많이 감소

Ridge and LASSO

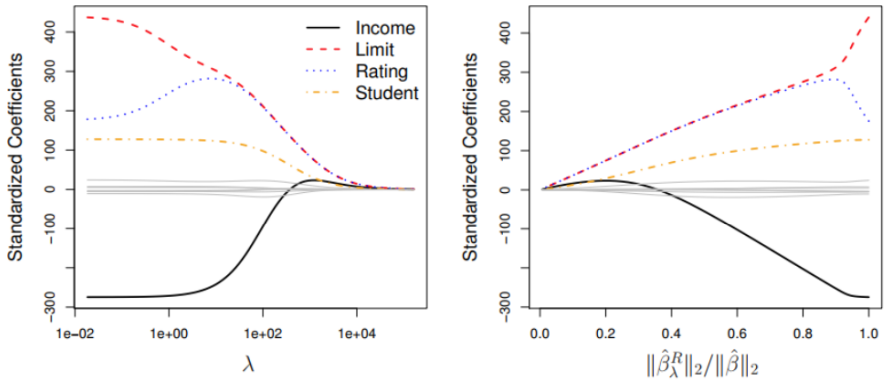


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

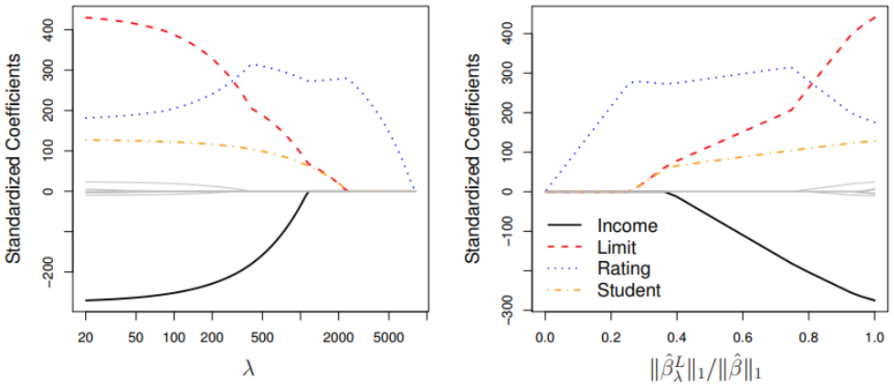


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

-차이점: L_1 penalty에는 (어떤 시점 이후로 계속) 0이 되는 parameter 존재

Simple Special Case for Ridge and LASSO

-consider a case with $n = p$ and X a diagonal matrix with 1's on the diagonal elements

-suppose we consider a model without an intercept

-the usual least squares problem: finding $\beta_1, \ldots, \beta_p$ that minimize $\sum_{j=1}^{p}(y_j - \beta_j)^2$

→ least square solution: $\hat{\beta}_j = y_j$

-the ridge regression estimates are

$$\hat{\beta}_j^{\text{Ridge}} = \frac{y_j}{1+\lambda}$$

- proportionally 0으로 shrinkage 일어남

-the LASSO estimates are

$$\hat{\beta}_j^{\text{LASSO}} = \begin{cases} y_j - \lambda/2 & \text{if} \quad y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if} \quad y_j < -\lambda/2 \\ 0 & \text{if} \quad |y_j| \leq \lambda/2 \end{cases}$$

- $\lambda/2$ 빼주는 방향으로 shrinkage 일어남

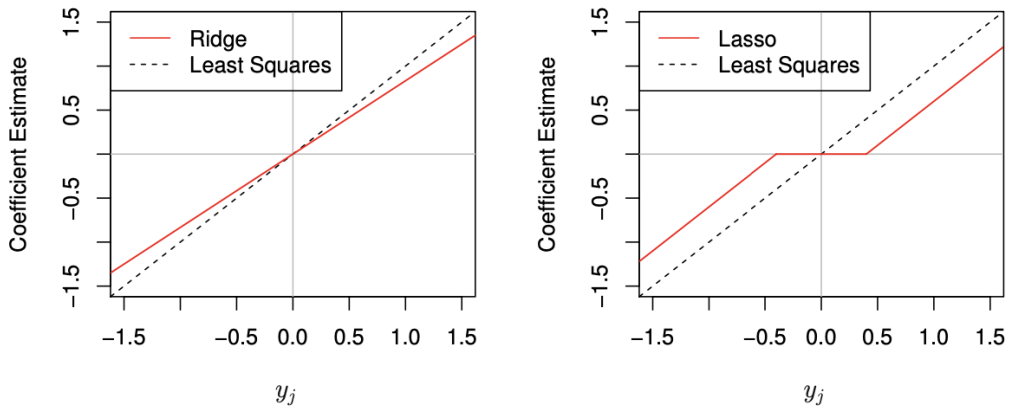


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

Selecting λ

-cross validation! → 오른쪽이 selected λ에 대한 coefficient estimates

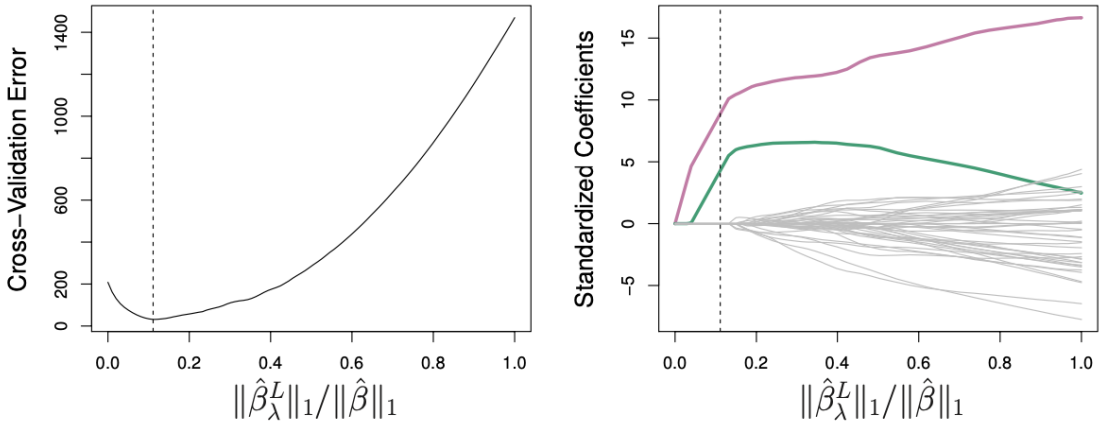


FIGURE 6.13. Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The two signal variables are shown in color, and the noise variables are in gray. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

*언제 MSE, cross-validation을 사용하는지 잘 확인하자.

Lecture 8: Classification 1

8.1. Prediction and Supervised Learning

Introduction

-classification problems are a form of supervised learning

- first train a model on data where the outcome is known
- then apply the model to data where the outcome is not known

Machine Learning

-machine learning algorithms are largely used to predict, classify, or cluster

- supervised learning: prediction, classification → have a response variable
- unsupervised learning: clustering → no response variable

Supervised Learning

1. Prediction of numeric response

-response is `volume` → ‘supervising’ the model fit

-the scatterplot smoother(loess) is predicting `volume`

2. Classification

-classify flowers using `Petal.Length`

-cut point 문제: 겹치는 구간이 발생하면 오류 발생 → 오류를 최소화하는 것이 good classfication

-classification is just a special prediction!

-can classify better when more variables are available

- but boundary 설정할 때 overfitting issue 존재할 수 있음 → boundary가 'simple' 할수록 그럴 가능성 적음

-model-based vs algorithm-based methods

8.2. Classification: Naive Bayes and LDA

Example: High-Earners in the 1994 US Census

-$50,000 이상 = 고소득자

-separate data set into two pieces: 80/20

-model fitting → `parsnip` package / model evaluation → `yardstick` package

Naive Bayes

-uses the probability of observing predictor values given as outcomes

-based on Bayes theorem: $P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)}$

-consider binary response y and want to classify a new observation x^*

- if $P(y=1|x^*) > P(y=0|x^*)$, we have evidence that $y=1$ is a more likely outcome for x^*
- assume we know marginal probabilities, $P(y=1)$ and $P(x=1)$
- $P(x=1|y=1)$ is easy to compute (contingency table)

→ can compute $P(y=1|x=1)$ by using the Bayes theorem

-if there are $x = x_1, \ldots, x_p$: naive Bayes targets $P(x_1, \ldots, x_p|y)$

-naive Bayes makes the (overly) simplistic assumption that $x_1, \ldots, x_p$ are *conditionally independent*

$$p(x_1, \ldots, x_p|y) = p(x_1|y) \cdots p(x_p|y) = \prod_{j=1}^{p} p(x_j|y)$$

-$p(x_j|y)$ is usually easy to compute since y is binary (or categorial)

- model-based: normal assumption
- algorithm-based(nonparametric): 비율/density

-we compute $p(y = k|x_1, \ldots, x_p)$ as

$$\frac{p(x_1, \ldots, x_p|y) \cdot p(y = i)}{p(x_1, \ldots, x_p)} = \frac{(\prod_{j=1}^{p} p(x_j|y = i)) \cdot p(y = i)}{\sum_{k=1}^{K} (\prod_{j=1}^{p} p(x_j|y = k)) \cdot p(y = k)} \quad (1)$$

-note: numeric values $1, \ldots, K$ given to y do not assume any ordering/meaning

-the naive Bayes only uses the assumption of conditional independence

- no regression
- may need to consider a probability model for a numeric predictor x_j

-generally produces good results despite the strong conditional independence assumption

Linear Discriminant Analysis

-most common approach: linear discriminant analysis(LDA)

-motivation: assume $p(x|y)$ is normal or Gaussian

- assume $p(x|y = 0) = f_0(x)$ and $f_0(x) = \phi(x; \mu_0, \sigma^2)$ → normal density with mean μ_0 and variance σ^2
- assume $p(x|y = 1) = f_1(x)$ and $f_1(x) = \phi(x; \mu_1, \sigma^2)$ → different mean μ_1 but the same variance σ^2

-under normal density assumption, $p(y = 1|x)$ can be written as the function of μ_0, μ_1, σ^2 and $p(y = 1)$

- $p(y = 1)$ can be estimated from data, say $\hat{\pi}_k = n_k/n$ $(k = 0, 1)$
- μ_0, μ_1, σ^2 can also be estimated from data (MLE)

-the LDA classifier:

$$\delta_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + log(\hat{\pi}_k)$$

→ can determine whether x is assigned to $y = 1$ or $y = 0$

- if $\delta_1(x) > \delta_0(x)$, then $x \to y = 1$
- i.e. $x > \frac{\hat{\mu}_1 + \hat{\mu}_0}{2}$ if $\mu_1 > \mu_0$ ($\log(\hat{\pi}_k)$는 무시 가능)

LDA with $x_1, \ldots, x_p$

-now, μ_k is the vector of length p

-consider common covariance-variance matrix Σ

-thus, for class k, n is drawn from a multivariate normal distribution $N(\mu_k, \Sigma)$

-the density is

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k))$$

-after taking the log and ignoring the constant terms, we get the term

$$x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

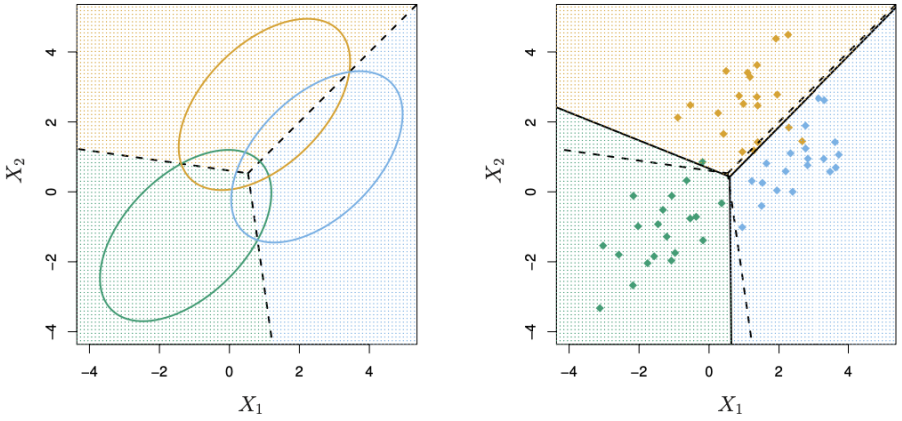
-then, LDA uses $\delta_k(x) = \delta_k(x_1, \ldots, x_p)$,

$$\delta_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k$$

→ LDA classifies x to the class for which $\delta_k(x)$ is the largest

-LDA can be directly extended to categorical responses

- we can always estimate μ_k, $k = 1, \ldots, K$ and Σ
- compute $\delta_k(x)$ and assign x to the class with largest $\delta_k(x)$



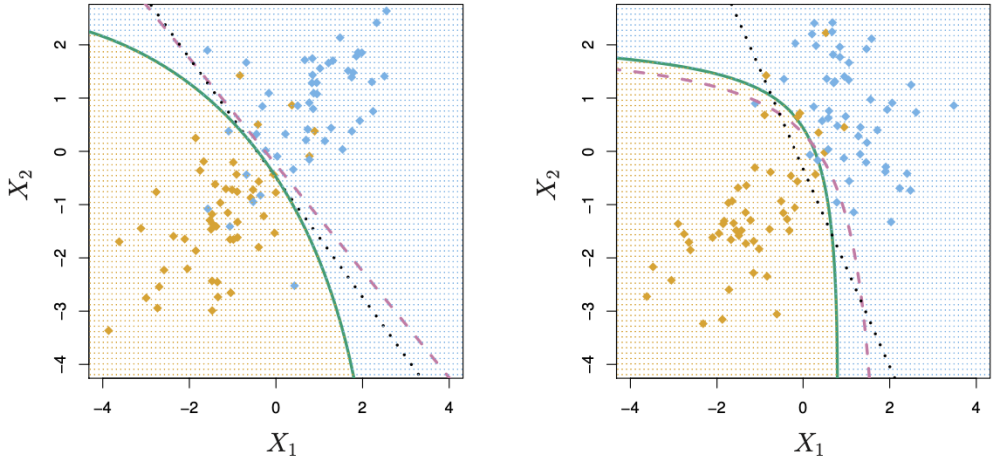
LDA and QDA

-in practice, LDA performs well

-often perform similarly as logistic regression

-LDA uses assumption that $p(x|y=k)$ are normal with the same Σ

- relax the common Σ assumption: specifiying different Σ_k
- this classifier is called quadratic discriminant analysis(QDA): 이차식의 δ_k



-Bayes classifier (best possible, but unknown): purple dashed

-LDA: black dashed

-QDA: green solid

8.2. Classification: Logistic Regression

Introduction

-use logistic regression model for classification

-model: $\text{logit}(q) = \log(\frac{q}{1-q}) = \beta_0 + \cdots + \beta_p x_p = x^T\beta$

- $q = p(y=1|x) = \frac{\exp(x^T\beta)}{1+\exp(x^T\beta)} = \frac{1}{1+\exp(-x^T\beta)} = \text{expit}(x^T\beta)$
- if $\exp(x^T\beta) = 1$, $q = 0.5$ → it means that $p(y=1|x) = p(y=0|x)$

→ $x^T\beta = 0$ is the decision boundary

-high earners example: solve $x^T\beta = \beta_0 + \beta_1 \cdot \text{capital gain} = 0$ → boundary point($-\beta_0/\beta_1$)

Estimating the Coefficient: Pitfalls of Squared Loss

-in logistic regression models, we don't use least squares method to estimate the coefficient

-main issue: the predicted $\hat{y}$ is the probability, but the actual y is binary

→ problematic: MSE is not convex

- adding more variables will not fix this issue
- additional issues
  - model error is always bounded between 0 and 1
  - conceptually questionable

Cross-Entropy Loss

-let y be binary and q be the probability of being 1

-the **cross-entropy loss** is defined as

$$-\{y\log(q) + (1-y)\log(1-q)\}$$

- convex
  - good measure of model error (able to penalize 'off' predictions)

-if $y = 1$, the loss is $-\log(q)$

- $q \to 0$: infinite loss
- $q \to 1$: zero loss

-if $y = 0$, the loss is $-\log(1-q)$

- $q \to 0$: zero loss

- $q \to 1$: infinite loss

Average Cross-Entropy Loss

-for logistic regression, the average cross-entropy loss for n data points is

$$-\frac{1}{n}\sum_{i=1}^{n}(y_i \cdot \log(\frac{\exp(x^T\beta)}{1+\exp(x^T\beta)}) + (1-y_i)\log(\frac{1}{1+\exp(x^T\beta)}))$$

- convex!

Connection to Maximum Likelihood Estimation

-cross-entropy loss is closely related with the maximum likelihood

-for given x_i, we know that y_i is drawn from a Bernoulli distribution with $q_i = p(y_i|x_i)$

-the likelihood is defined as

$$q_i^{y_i} \cdot (1-q_i)^{1-y_i}$$

-the overal likelihood function is

$$\prod_{i=1}^{n} q_i^{y_i}(1-q_i)^{1-y_i}$$

-and the log-likelihood is

$$\sum_{i=1}^{n} y\log(q_i) + (1-y)\log(1-q_i)$$

-since $q_i = \text{expit}(x_i^T\beta)$ in the logistic regression model, minimizing the cross-entropy loss is equivalent to maximizing the log-likelihood function

→ maximum likelihood estimation (MLE)

-many of the ‘model + loss’ combinations can be motivated using MLE

- loss function에 penalty term(L1, L2, …)만 더하면 됨 → x가 많을 때 더 좋은 performance

Logistic Regression for More Than 2 Classes

-multiple-class extensions

- (1) 1 vs 2&3
- (2) 2 vs 3 given not 1

→ not to be used that often: discriminant analysis is more popular

Lecture 9: Classification II

9.1. Logistic Regression for Classification

High-earners Example

-consider the null model without any predictors: $p(y=1|\text{null}) = \text{expit}(\beta_0)$

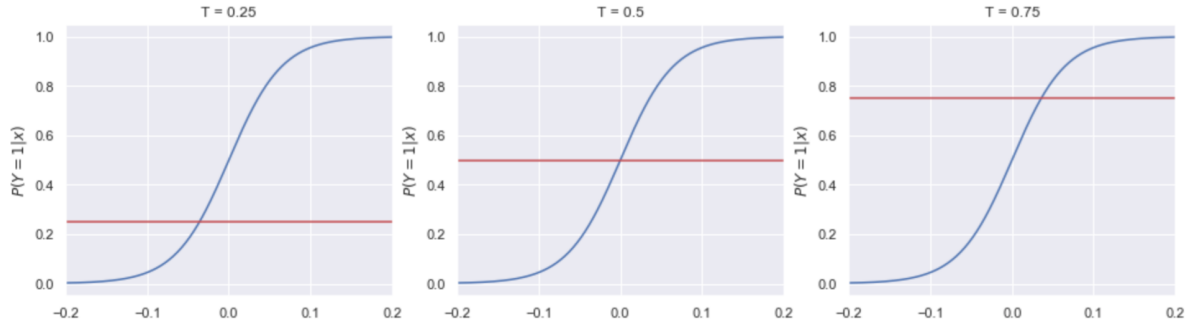
- same predictive probability: $\hat{\theta} = \text{expit}(\beta_0) = 0.241$ (MLE)
- since $p(y=1|\text{null}) = 0.241 < 0.759 = p(y=0|\text{null})$, all observations are classified as $y=0$

Thresholding

-we don’t need to set our threshold to 0.5

-decision rule: depending on the type of errors we want to minimize, we can increase/decrease it

-logistic regression + decision rule = classifier



-observation above the red line: $y=1$ / below the red line: $y=0$

-moving the red line higher means

- less observations are classified as class 1
- few false positives(FP) / true positives(TP)

- more false negatives(FN) / true negatives(TN)

→ how to quantify the performance of models?

Accuracy

-accuracy: how many observations are correctly classified

$$\text{accuracy} = \frac{\text{TP}+\text{TN}}{n}$$

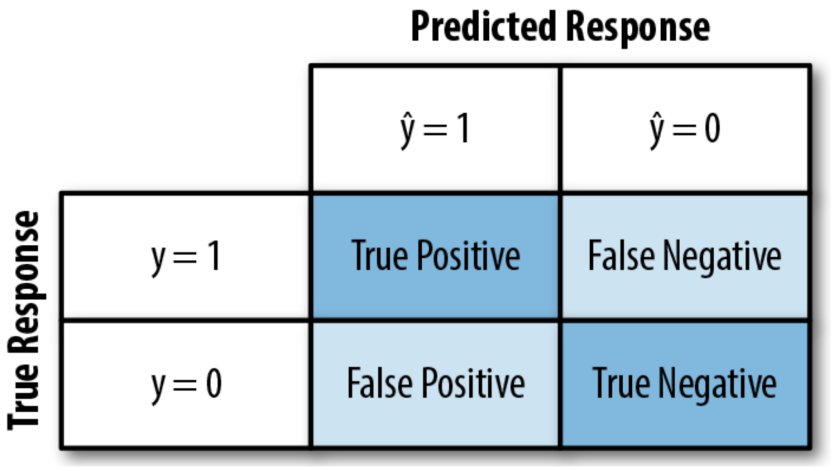
-in the presence of class imbalance, it's not so meaningful

-adding more variables → increase the accuracy (on the training set)

9.2. Evaluating Classifiers

Confusion Matrix & Type of Errors

-class 1 is 'positive' and class 0 is 'negative'



-True Positive(TP): correctly classified class 1

-True Negative(TN): correctly classified class 0

-False Positive(FP): false alarms!

-False Negative(FN): fail to detect

Precision and Recall

-**precision**: of all observations predicted to be 1, what proportion is actually 1?

——

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} = P(y = 1|\hat{y} = 1)$$

-**recall**: of all observations that are actually 1, what proportion did we predict to be 1?

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = P(\hat{y} = 1|y = 1)$$

-trade-off!

- precision penalizes FPs, and recall penalizes FNs
- can achieve 100% recall by making our classifier output “1”, i.e. threshold = 0

  → no FNs, but many FPs → precision would be low

-higher threshold → fewer FP, higher precision

-lower threshold → fewer FN, higher recall

How to Choose Metrics

-depending on the domain

-example 1: predicting whether or not a patient has some disease

- FN is much more dangerous than FP → maximize recall

-example 2: determining whether or not someone should be sentenced to life in prison

- minimize FP (innocent → guilty) → maximize precision

Effects of Thresholds

-accuracy

- if the threshold is too high → many FN
- if the threshold is too low → many FP

-precision

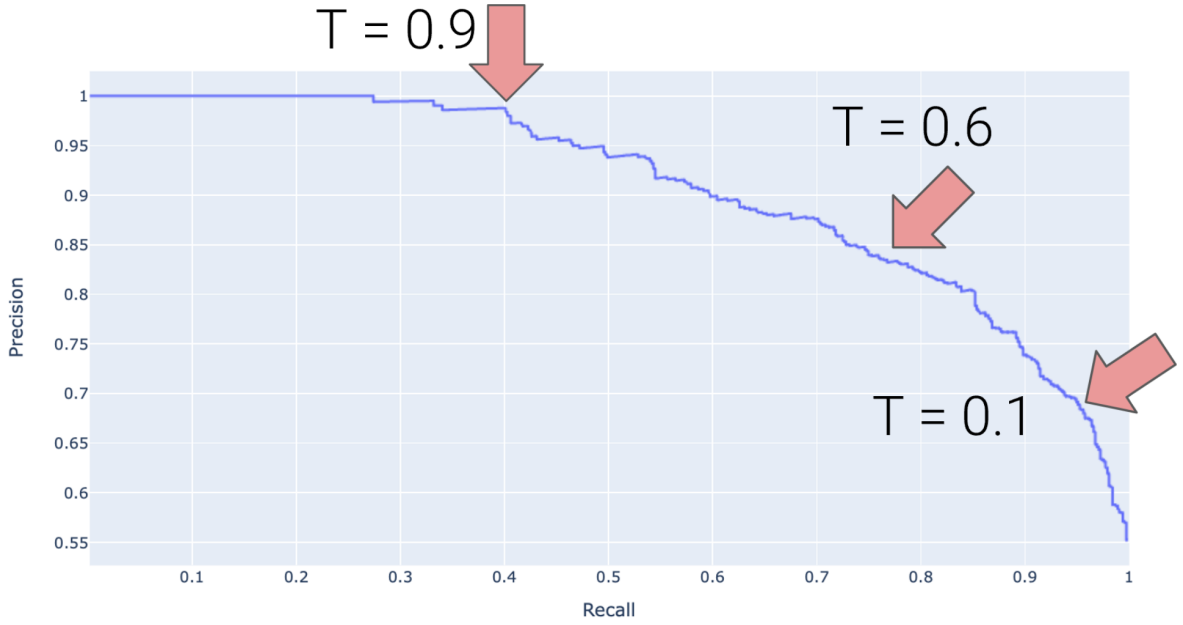
- as the threshold gets higher → fewer FP → precision “tends to” increase

-recall

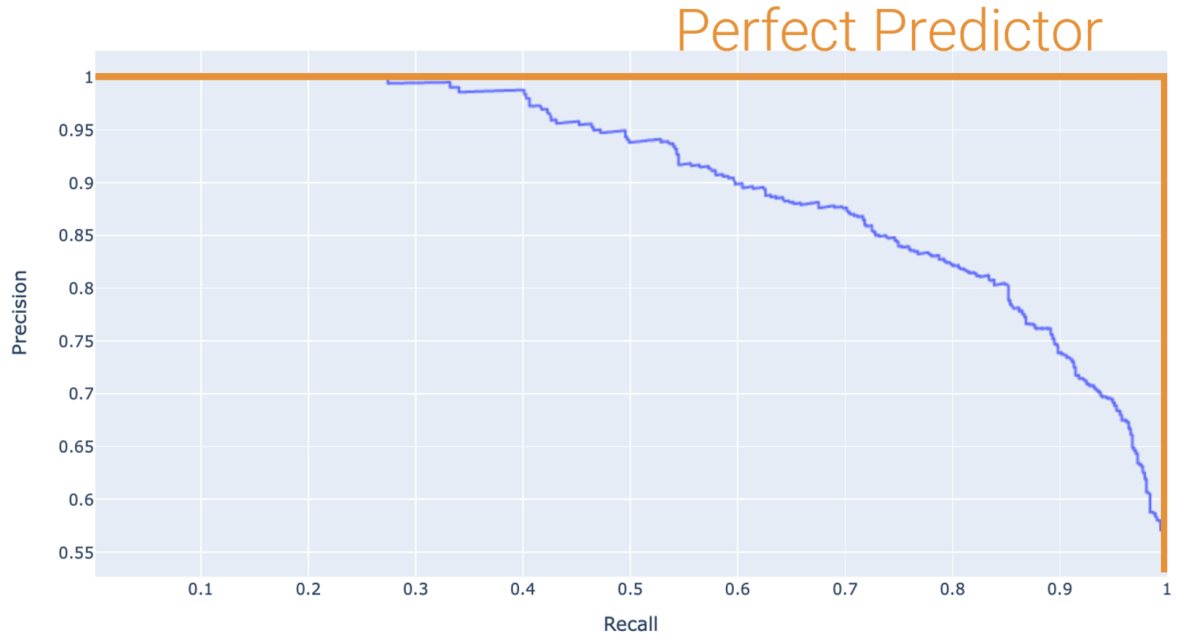
- as the threshold gets higher → fewer FN → recall decreases

Precision-Recall Curve

-relationship between two curves & threshold



-perfect classifier: precision of 1 and recall of 1



Other Metrics

-**sensitivity** = recall = **true positive rate (TPR)**

-**specificity** = **true negative rate (TNR)**

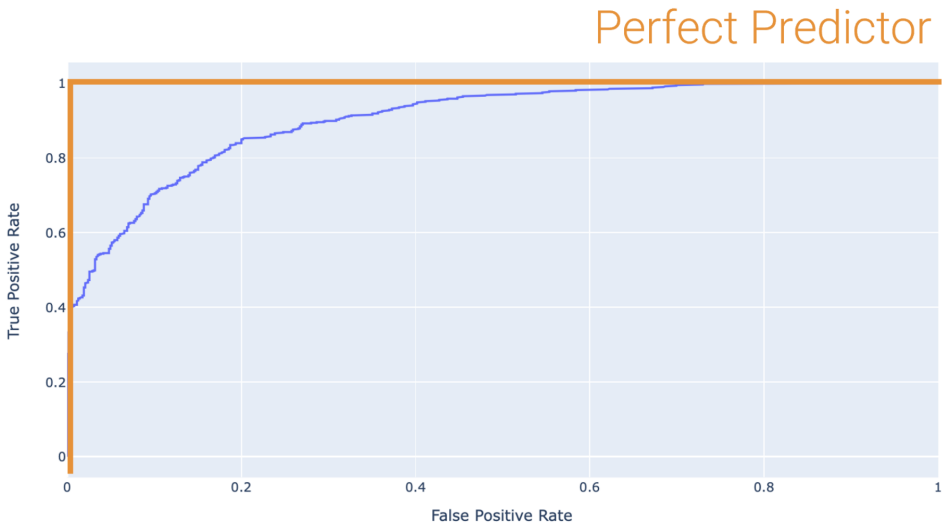
—- -

$$\text{specificity, TNR} = \frac{\text{TN}}{\text{TN+FP}} = P(\hat{y} = 0|y = 0)$$

-$1 - \text{specificity}$ is often called **false positive rate (FPR)**

-**precision** = **positive predictive value (PPV)**

ROC Curve



-instead of the precision-recall curve, create a curve using sensitivity and specificity

-choose 1-specificity or FPR as the x-axis, and sensitivity or TPR as the y-axis

→ 'Receiver Operating Characteristics (ROC)' curve

-both PR and ROC curves are used for evaluation in practice: PR curve is useful in evaluating data with highly unbalanced outcomes

AUC

-we want our ROC curve to be as close to the top left as possible

-compute the area under curve (AUC) of the model

- perfect classifier → AUC = 1
- terrible classifier → AUC = 0.5

-threshold decision: (둘 중 누구를 높여야 할지 모른다면) TPR+FPR이 최대화되는 값 → $y = x + k$ 직선과 접하는 지점

Evaluation Using the Test Set

type	accuracy_train	accuracy_test	sens_test	spec_test
log_all	0.852	0.849	0.598	0.928
log_1	0.800	0.803	0.204	0.991
null	0.759	0.761	0.000	1.000

-each model performs slightly worse on the testing set than it did on the training set

-null model has a sensitivity of 0 and a specificity of 1 (always makes the same prediction)

-full model is slightly less specific than the single explanatory variable, but it is much more sensitive

→ AUC를 기준으로 '모델'을 설정 (이후 원하는 measure를 최대화하는 decision rule 선택)

9.3. Linear Separability, Regularization

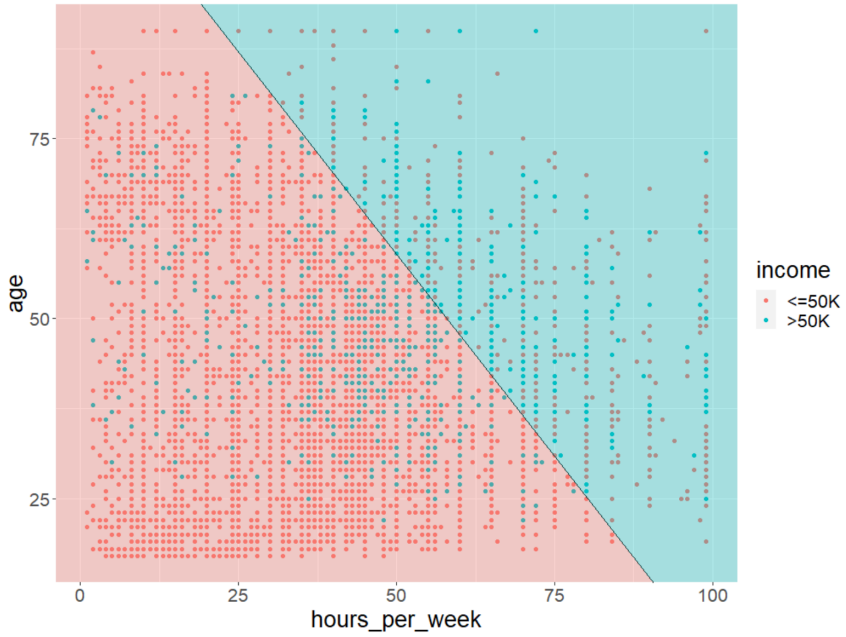
Decision Boundary

-consider a logistic model

- $p(y=1|x) = \text{expit}(\beta_0 + \beta_1 \cdot \text{hours})$
- $\text{logit}(\hat{\pi}) = -3.085 + 0.046 \cdot \text{hours}$
- if we set the threshold to 0.5, then the boundary is $-3.085 + 0.046 \cdot \text{hours} = 0$
  (if `hours_per_week` is greater than 67, then the predicted value is $\hat{y} = 1$)

-if there is one predictor, the boundary is just a point

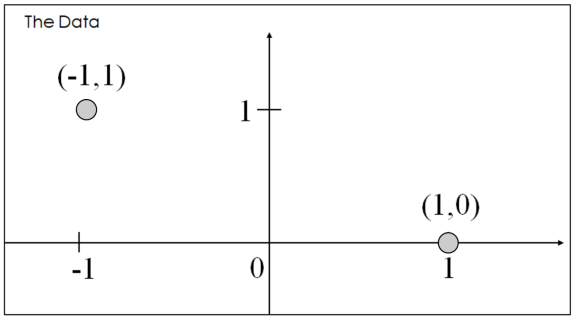
-if there are two predictors, the decision boundary is a line on the 2d surface of two predictors



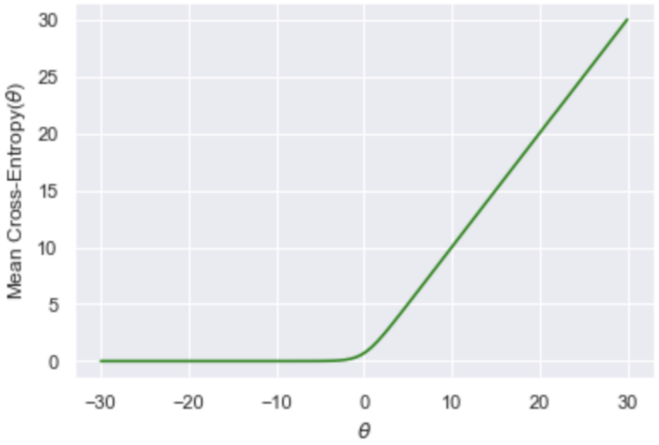
* blue = 1, red = 0

-what if the points are perfectly separated?

- consider a simple logistic model $p(y=1|x) = \text{expit}(\beta_0 + \beta_1 x)$
- two data points are the following



- best estimate of β_1 (minimizes mean cross-entropy loss) = $-\infty$



Issues with Large β_1

-suppose we have a new observation (0.5, 1)

- our model predicts $p(y = 1|x = 0.5)$ to be 0
- the cross-entropy loss is infinite! → ‘overly confident’

Linear Separability

-points are linearly separable if we can correctly separate the classes with a line

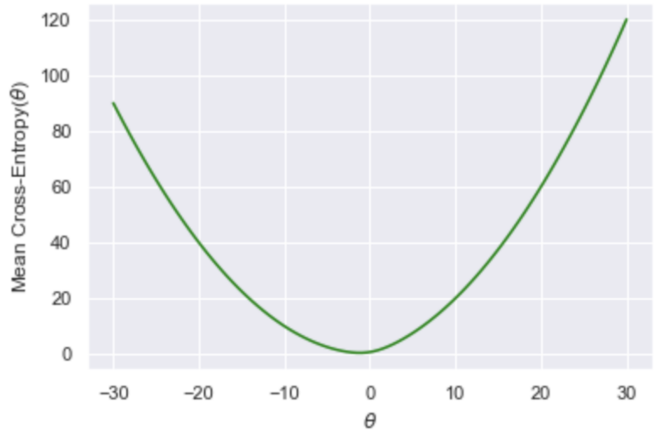
-more formally, a set of d-dimensional points is linearly separable if we can draw a degree $d - 1$ hyperplane(line) that separates the point perfectly

Regularized Logistic Regression

-if our training set is linearly separable, some of the parameters will diverge to infinity

→ use regularization to avoid such large values

-loss surface is now changed (convex)



* we need to standardize our features before applying regularization

Lecture 10. Splines and GAM

10.1. Introduction

-linear models have two advantages

1. simple to describe and implement
2. easy to interpret and make inference

-linear assumption may be not realistic, but can be a good approximation

-what can we do more to relax the linearity assumption while still attempting to maintain as much interpretability as possible?

10.2. Polynomial Regression and Step Functions

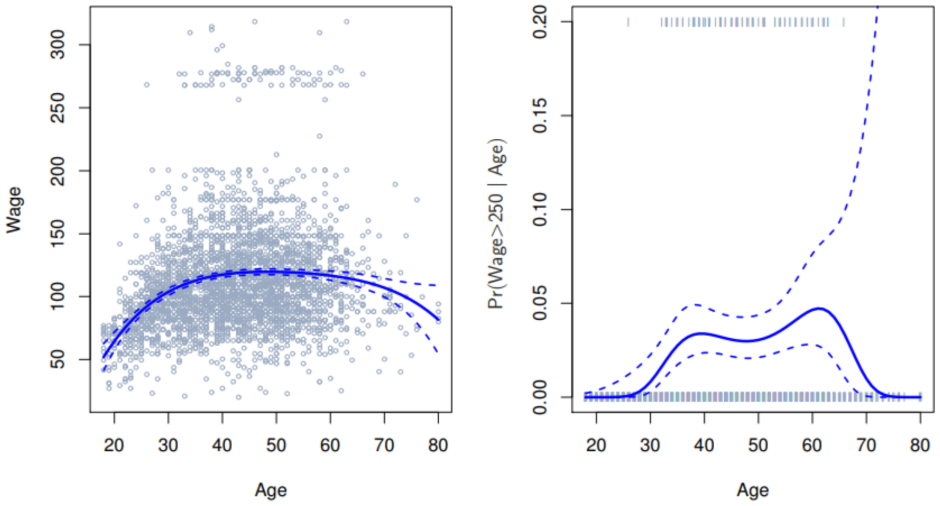
Polynomial Regression

-if the relationship is non-linear → $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon_i$

-d개의 covariate 있다고 생각

- coefficients β_j can be easily estimated using least squares
- for large d, the fitted curve can be extremely non-linear → unusual to use $d > 4$
- polynomial curve with $d > 4$ can be come overly flexible, have very strange shapes especially near the boundary of X

Example: Age and Wage



$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4$$

-the relationship between age and wage is non-linear

-the coefficients β_j are not of interest → need to look at the entire function

-also, we need to consider two dashed curves that represent the pointwise confidence intervals

$$\Pr(y_i > 250 \mid x_i) = \frac{\exp(\beta_0 + \cdots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \cdots + \beta_d x_i^d)}$$

-logistic regression can be used

-confidence intervals are wide on the right-hand side: 0으로 추정, but sample in zero(?)인지 알 수 없음

Step Functions

-the polynomial regression imposes a global structure on the non-linear function of X

→ to avoid imposing such global structure, use step functions

-simple approach: break the range of X into bins → fit a different constant

- create cutpoints $c_1, \ldots, c_K$ to construct $K + 1$ new variables

$$C_0(X) = I(X < c_1)$$
$$\vdots$$
$$C_{K-1}(X) = I(c_{K-1} \leq X < c_K)$$
$$C_K(X) = I(c_K \leq X)$$

- notice that for any value of X, $C_0(X) + C_1(X) + \cdots + C_K(X) = 1$

-fit linear model: $y_i = \beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$

- for a given value of X, at most one of $C_1, \ldots, C_K$ can be non-zero
- when $X < c_1$, β_0 → mean value of Y for $X < c_1$
- when $c_j \leq X < c_{j+1}$, $\beta_0 + \beta_j$ → β_j represents the average increase in the response for X in $c_j \leq X < c_{j+1}$ relative to $X < c_1$

10.3. Regression Splines

Basis Functions

-polynomial and piecewise-constant regression models are in fact special cases of a **basis function** approach

-fit the model: $y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$ for some fuctions/transformation $b_j(X)$

-note that basis functions are fixed and known

-coefficient estimates are not of interest

Regression Splines: Piecewise (Cubic) Polynomials

-common choice for a basis function: **regression splines**

-more flexible than piecewise constant regression

-piecewise cubic with no knots = standard cubic polynomial

-piecewise cubic with single knot at a point c:

$$y_i = \begin{cases} \beta_{01} + \beta_{11} + \beta_{21} + \beta_{31} + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12} + \beta_{22} + \beta_{32} + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

- fit two different polynomial functions
- two sets of coefficients
- what can be wrong?: two curves do not meet

Constraints and Splines

-need the constraint that the fitted curve is continuous (at c)

-can add two additional constraints that the first and second derivatives are continuous at c

-for the left part ($x < c$): need to use 4 degrees of freedom

-for the right part:

- without constraints → 4
- continuous constraints → 3
- smooth constraints(cubic spline) → 1

-a cubic spline with K knots uses a total of $K + 4$ degrees of freedom

The Splines Basis Representation

-fitting a piecewise degree-d polynomial under the constraint

→ start off with a basis for a cubic polynomial, add a truncated power basis function per knot

-a truncated power basis function: (where ξ is the knot)

$$h(x, \xi) = (x - \xi)_+^3 = \begin{cases} (x - \xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$$

→ e.g. of two knots:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - c_1)^3 + \beta_5 (x - c_2)^3 + \epsilon_i$$

-discontinuity in only the third derivative at ξ

-fitting a cubic spline with K knots: $K + 4$ regression coefficients in total

Natural Spline

-splines can have high variance at the outer range of X

-the confidence bands in the boundary region appear wild

-**natural spline**: a regression spline with additional boundary constraints

- function is required to be linear at the boundary

Choosing the Number and Locations of the Knots

-where should we place knots?

- where we feel the function might vary most rapidly
- common approach: a uniform fashion (like quantiles)

-how many knots? → bias-variance trade-off와 연관

- try different numbers..
- cross-validation! (which gives the smallest RSS?)

10.4. Smoothing Splines

An Overview

-want $RSS = \sum_{i=1}^{n}(y_i - g(x_i))^2$ to be small

-if we don't put any constraints on $g(x_i)$ → can always make RSS zero → overfitting!

-want a function that makes RSS small, but that is also smooth

→ natural approach: **smoothing spline** $\hat{g}$ that minimizes the objective function

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

Regularization: Loss + Penalty

-the objective function$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$ takes the 'loss + penalty' formulation

-the term $g''(t)$ measures the roughness of g

- if g is very smooth, then $g'(t)$ will be close to constant → small $\int g''(t)^2 dt$
- if g is jumpy, then $g'(t)$ will vary significantly → large $\int g''(t)^2 dt$

→ the larger the value of λ, the smoother the g will be

-when $\lambda = 0$, the penalty term has no effect

- g will be very jumpy
- g will exactly interpolate the (training) observations

-when $\lambda \to \infty$,

- g will be perfectly smooth (straight line!)

-λ controls the bias-variacne trade-off of the smoothing line

-remark. a smoothing spline is a natural cubic spline with knots at $x_1, \ldots, x_n$

Choosing the Smoothing Parameter λ

-smoothing spline = natural cubic spline with knots at every unique value of x_i → so many degrees of freedom (n + …)

-but the tuning parameter λ controls the roughness

- *effective degrees of freedom*: df_λ
- as λ increases from 0 to ∞, df_λ decreases from n to 2 ($\simeq$ constant function)

→ df_λ is a measure of the flexibility of the smoothing spline

Effective Degrees of Freedom

-we can write $\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y}$

- $\hat{\mathbf{g}}_\lambda$ is the solution to the objective function for a particular choice of λ

-the effective degrees of freedom is defined to be the sum of the diagonal elements of the matrix $\mathbf{S}_\lambda$

$$df_\lambda = \sum_{i=1}^{n} \{\mathbf{S}_\lambda\}_{ii}$$

-in fitting a smoothing spline, we do not need to select the number or location of the knots

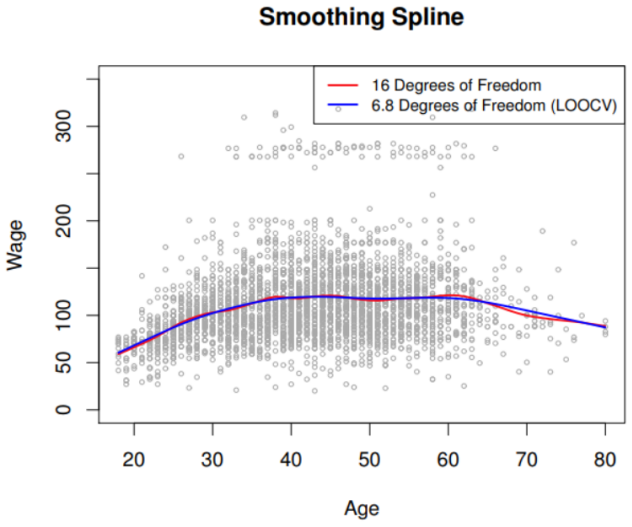
(there will be a knot at each training observation $x_1, \ldots, x_n$)

-another problem: choosing λ? → cross-validation!

→ LOOCV can be computed very efficiently

$$\mathrm{RSS}_{CV}(\lambda) = \sum_{i=1}^{n} (y_i - \hat{g}_\lambda^{(-1)}(x_i))^2 = \sum_{i=1}^{n} [\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{\mathbf{S}_\lambda\}_{ii}}]^2$$

- $\hat{g}_\lambda^{(-1)}(x_i)$: fitted value at x_i using the training observations except for the ith observation (x_i, y_i)
- $\hat{g}_\lambda(x_i)$: fitted value at x_i using all the training observations



Local Regression

-similar to LOESS

-span이 커질수록 smoother (lower df)

10.5. Generalized Additive Models (GAM)

Introduction

-GAMs provide a general framework for extending a standard linear model by allowing non-linear functions of each of the variables

- maintain additivity
- can be applied with both quantitative and qualitative responses

GAM for Regression Problems

-a natural way to extend the multiple linear regression model $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$

→ replace each linear component $\beta_j x_{ij}$ with a (smooth) nonlinear function $f_j(x_{ij})$

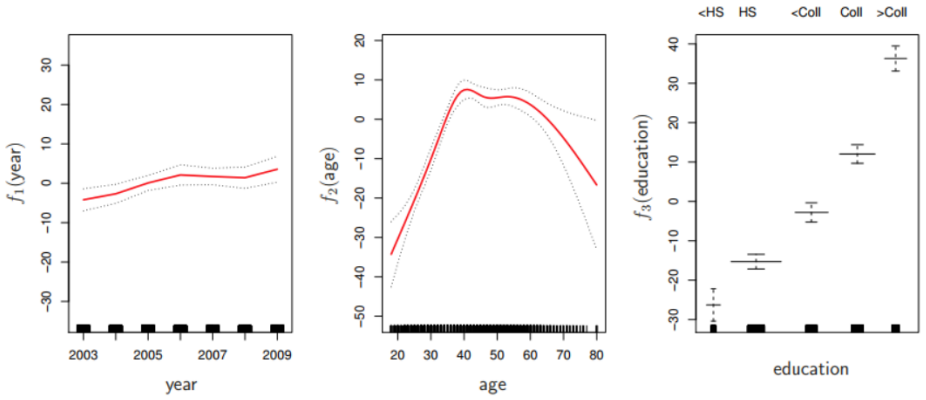
$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i$$
$$= \beta_0 + f_1(x_{i1}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

-'additive': calculate a separate f_j for each X_j, and then add together all of their contributions

-we can use many methods as building blocks for fitting an additive model (e.g. natural splines)

Example

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$



-how can we fit the GAMs?

- entire model is just a big regression onto (natural) spline basis functions and dummy variables
- but, smoothing splines 택할 경우 one big regression으로 생각할 수 없음

-how can we interpret the GAMs?

- look at each function f_j
- 다른 변수를 fixed한 상태의 변동

Pros and Cons of GAMs

-pros

- allow us to fit non-linear f_j to each X_j → can model non-linear relationships (need not try different transformations on each variable individually)
- can make more accurate predictions
- can examine the effect of each X_j on Y individually while holding all of the other variables fixed ($\because$ additive)
- smoothness of the function f_j for X_j can be summarized via degrees of freedom

-cons

- restricted to be additive → interactions can be missed
    - but can manually add interaction terms (fitted using two-dimensional smoothers…)

-for fully general models: more flexible approaches such as random forests and boosting

GAM for Classification Problems

-logistic regression GAM:

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + f_1(X_1) + \cdots + f_p(X_p)$$

Lecture 11. Causal Inference

11.1. Introduction

-many questions are causal e.g. Does delivery at high-level NICUs lower the risk of BPD as it lowers the risk of death?

-confounder X: a variable that affects both the treatment Z and the outcome Y

-effect of smoking on lung cancer → observational studies

- matched 36,975 heavy smokers to non-smokers
- 122 pairs → 12 pairs (non-smokers died) / 110 pairs (smokers died)
- if smoking has no effect: expect to observe, on average, 61 pairs where the non-smokers died
- p-value is less than 0.0001 since $P(X \geq 110) < 0.0001$ if $X \sim B(122, 0.5)$

→ unmeasured confounder?

11.2. Potential Outcome Framework and Causal Assumptions

Potential Outcome Framework

-for unit i ($i = 1, \ldots, N$):

- X_i: (observable) covariates

- $Z_i \in \{0, 1\}$: treatment
- $Y_i(1)$: potential outcome if unit i receives a treatment
- $Y_i(0)$: potential outcome if unit i doesn't receive a treatment
- Y_i: observed outcome → $Y_i = Y_i(Z_i) = Z_i Y_i(1) + (1 - Z_i) Y_i(0)$

-observed: $Z_i = 1$, $Y_i = 1$

- $Y_i = Y_i(1)$ → $Y_i(1)$ is the factual outcome
- $Y_i(0)$ is the quantity that we can hypothetically imagine → $Y_i(0)$ is the counterfactual outcome

-other notation: Z_i represents a random variable, z_i represents a particular value of a random variable

Underlying Assumptions in Notations

-notation $Y_i(0), Y_i(1)$ → potential outcome can be represented by $Y_i(Z_i)$

1. no interference assumption: $Y_i(Z_i)$ cannot depend on other units' treatments Z_j (+ outcomes $Y_j(1), Y_j(0)$)
2. consistency: $Y_i(Z_i) = Y_i(1)$ if $Z_i = 1$
3. no multiple versions of treatment: if $Z_i = 1, Z_j = 1$, units i, j must have the same treatment

Measures of Causal Effect

-individual-level treatment effect is usually defined as

$$\tau_i = Y_i(1) - Y_i(0)$$

-the average treatment effect (ATE) is defined as

$$\tau = E[Y_i(1) - Y_i(0)] = E[Y_i(1)] - E[Y_i(0)]$$

-alternatives:

- risk ratio: $E[Y_i(1)] / E[Y_i(0)]$
- odds ratio (for binary outcomes, $E[Y_i(1)] = \Pr(Y_i(1) = 1)$): $\frac{\Pr(Y_i(1)=1)\Pr(Y_i(0)=0)}{\Pr(Y_i(1)=0)\Pr(Y_i(0)=1)}$

Defining Causal Effects

-the ATE can be considered in two ways

- sample average treatment effect (SATE): about a certain sample

$$SATE = \frac{1}{N}\sum_{i=1}^{N}\{Y_i(1) - Y_i(0)\}$$

- population average treatment effect (PATE): about the population where the sample is drawn from

$$PATE = E[Y_i(1) - Y_i(0)]$$

→ $SATE \to PATE$ as $n \to \infty$

Fundamental Problem of Causal Inference

Unit	Treated	Earnings in $ in 1978			Black	Education (yrs)
i	Z_i	$Y_i(1)$	$Y_i(0)$	Y_i	X_{i1}	X_{i2}
1	1	2000	?	2000	1	9
2	0	?	3000	3000	0	10
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	6000	?	6000	0	12

-treatment effect τ_i cannot be observed because only one of $Y_i(1)$ and $Y_i(0)$ is observed

-the individual treatment effect τ_i cannot be estimated!

→ by considering **causal assumptions**, we can estimate the average treatment effect (ATE) $E[\tau_i] = E[Y_i(1)] - E[(Y_i(0)]$

1. random treatment assignment
2. unconfoundedness or strong ignorability

Other Causal Assumptions

-stable unit treatment value assumption (SUTVA): potential outcomes for each unit are not affected by the treatment status of other units

-overlap: each unit has chances to receive and not to receive a treatment → $0 < \Pr(Z = 1 \mid X) < 1$

11.3. Randomized Experiments and Observational Studies

Randomized Experiments

-when a treatment is randomly assigned, treated and control groups tend to be similar in terms of **both** observed and unobserved covariates

-why randomization is important?

1. randomization yields an unbiased estimator
2. randomization is the ‘reason basis for inference’

The Role of Randomization for Statistical Control

$\leftrightarrow (Y(1), Y(0)) \perp\!\!\!\perp Z$

-consider a new cholesterol drug (treatment) that can reduce a cholesterol level (outcome)

- treatment assignment = tossing a fair coin → 50% chance to receive the drug
- cholesterol levels (potential outcomes) are nothing to do with treatment assignments!

-we have $E[Y(1)] = E[Y(1) \mid Z = 1] = E[Y(1) \mid Z = 0]$

- $Y(1)$ is observable when $Z = 1$
- the equality doesn’t hold in general

-therefore,

$$\begin{aligned} E[Y(1) - Y(0)] &= E[Y(1)] - E[Y(0)] \\ &= E[Y(1) \mid Z = 1] - E[Y(0) \mid Z = 0] \\ &= E[Y \mid Z = 1] - E[Y \mid Z = 0] \\ &= \text{Mean for the treated - Mean for the control} \\ &= \bar{Y}_t - \bar{Y}_c \end{aligned}$$

-$(Y(1), Y(0)) \perp\!\!\!\perp Z$ is the key causal assumption

→ **identification assumption**: no assumption needed for the distributions of $Y(1), Y(0)$

When $(Y(1), Y(0)) \not\perp\!\!\!\perp Z$?

-what if a doctor decided to give the drug to people who would expect high cholesterol levels without the drug?

- higher $Y_i(0)$ → higher chance to receive the drug ($Z_i = 1$)
- this leads to $E[Y(0) \mid Z = 1] > E[Y(0) \mid Z = 0]$
- the doctors decision may be based on previous medical history X → confounder!

Observational Studies

-in observational study, assignment is not at random

-treated and control groups tend to differ systematically in their covariates

→ naive comparison is biased

1. overt bias: from observed covariates
    - controlled by using adjustment methods
    - we can use **regression adjustment, matching**, and **weighting**
2. hidden bias: from unobserved covariates
    - controlled by some designs
    - examined by **sensitivity analysis**

Identification Assumptions (in Observation Studies)

-to estimate the ATE from data, we need the following assumption

$$Y(1), Y(0) \perp\!\!\!\perp Z \mid X$$

→ called the assumption **unconfoundedness**, **ignorability**

-if we have enough information about X → given X, the potential outcomes are independent of Z

-for the same X, receiveing a treatment looks like random

-we want to estimate ATE: $E[Y(1) - Y(0)]$

$$E[Y(1)] = E[E[Y(1) \mid X]] = E[E[Y(1) \mid X, Z = 1]] = E[E[Y \mid X, Z = 1]]$$
$$E[Y(0)] = E[E[Y(0) \mid X]] = E[E[Y(0) \mid X, Z = 0]] = E[E[Y \mid X, Z = 0]]$$

-use regression → $\mu_z = E[Y(z) \mid X]$

- fit a model on the treated subject data → $\hat{\mu}_1(X)$
- fit another model on the control subject data → $\hat{\mu}_0(X)$
- then, we can have the ATE estimate as

$$\hat{\tau}^{ATE} = \frac{1}{n}\sum_{i=1}^{n}((\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i))$$

Causation-Association Difference

-$E[Y(1)] \neq E[Y \mid Z = 1]$!

- causation: $E[Y(1)]$
- association: $E[Y \mid Z = 1]$

Connected to Regression Models

-the classical approach to control for X is using a regression model like $E(Y \mid X) = \beta_0 + \beta_1 Z + \beta_2 X$

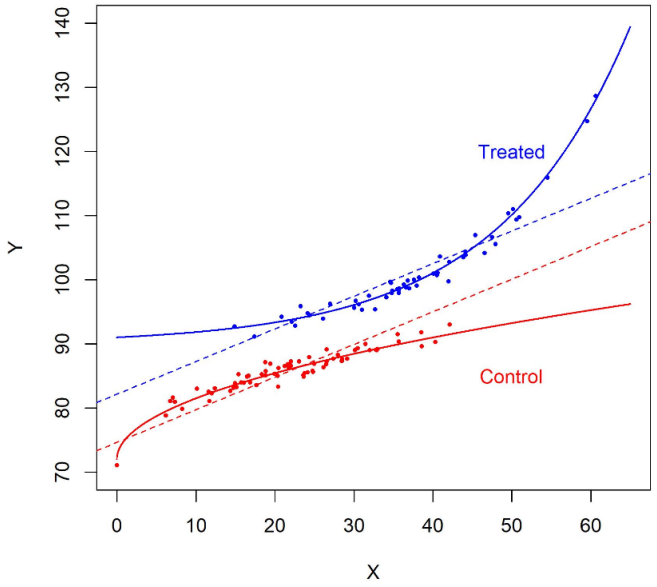
-corresponding counterfactual model:

- $E[Y(1) \mid X] = \beta_0 + \beta_1 + \beta_2 X$
- $E[Y(0) \mid X] = \beta_0 + \beta_2 X$

→ $ATE = E[Y(1) - Y(0)] = E[E[Y(1) - Y(0) \mid X]] = \beta_1$

-model specification can be very different between $E[Y(1) \mid X]$ and $E[Y(0) \mid X]$

-$E[Y(1) \mid X] - E[Y(0) \mid X]$ is a function of X, say $\tau(X)$



Concern with Regression Approaches

1. overlap issue
    - extraploation 일어나야 함
2. in standard practice, covariate adjustments are done with looking at outcomes
    - can introduce another source of bias
    - cherry-picking

11.4. Propensity Score Based Methods

Propensity Score

-for covariates X_i, we consider the conditional probability of receiving treatment,

$$e(x) = \Pr(Z_i = 1 \mid X_i = x)$$

-we call this probability the propensity score

- x_i can be multi-dimensional
- $e(x_i)$ is one-dimensional quantity (dimension reduction한 것)

-key result: $(Y(1), Y(0)) \perp\!\!\!\perp Z \mid e(x)$

- given $e(x)$, it looks like Z is random
- finding people who share $e(x)$ is enough

Estimate Propensity Score (PS)

-however, the propensity score $e(X_i)$ is unknown → has to be estimated from the data

-since Z_i is binary, we can consider a logistic regression

- $\text{logit}\{\Pr(Z_i = 1 \mid X_i)\} = \beta_x^T X_i$
- estimate β_x as $\hat{\beta}_x$
- compute $\hat{e}(X_i) = \text{expit}(\beta_x^T X_i)$

-PS based methods: PS stratification, inverse probability weighting, PS matching

PS Stratification

-if we make strata that contain units with the same $e(X_i)$, within each stratum, the unconfoundedness assumption holds

-since $e(X_i)$ is not discreate in general, it is difficult to make strata based on $e(X_i)$

→ if we make each stratum narrow enough, units are comparable within each stratum

-procedure:

1. fit a model and estimate $\hat{e}(X_i)$
2. discretize $\hat{e}(X_i)$ into K strata (quantiles, equi-spaced intervals)
3. by assuming each stratum k obtained from a randomized experiment, we construct the estimator
    - $\bar{Y}_{k1}$ and $\bar{Y}_{k1}$ are the means for the treated and control groups in stratum k
    - n_k is the size of stratum k

$$\hat{\tau} = \sum_{i=1}^{K} (\bar{Y}_{k1} - \bar{Y}_{k0}) \cdot \frac{n_k}{n}$$

-gives different weights for observations

- treated individuals in stratum k → weight = $\frac{1}{n_k} \frac{n_k}{n}$
- control individuals in stratum k → weight = $\frac{1}{n_{k0}} \frac{n_k}{n}$

→ simply fit a regression model $Y = \alpha + \beta Z + \epsilon$ with weights defined above

Matching

-suppose # of treated subjects < # of control subjects

-procedure:

1. for each treated subject, select one control subject with the same X or $e(X)$ and create a matched pair
    - with or without replacement → bias-variance trade-off와 연관
2. examine covariate balance between treated and matched control groups (X가 비슷한지?)
3. discard unmatched control subjects
4. make inferences using matched pairs
    - estimation: compute the treated-minus-control outcome differences and take the average
        - alternatively, fit a regression model $Y = \alpha + \beta Z + \epsilon$ on 'matched' data
        - possible to add X to the model, but not recommended
    - conduct a hypothesis test

Inverse Probability Weighting (IPW)

-consider a toy example

-gold standard: randomized controlled trials (RCT)

- measured (and even unmeasured) covariates would be balanced between the groups
- $P(Z = 1 \mid \text{고소득} = 1) = \frac{1}{2} = P(Z = 1 \mid \text{고소득} = 0)$

-in an observational study,

- there are systematic differences in covariates between the groups
- $P(Z = 1 \mid \text{고소득} = 1) > P(Z = 1 \mid \text{고소득} = 0)$

> intuition: suppose one has $e(X) = \frac{1}{5}$ → Among five clones of him, one is treated and four are control on average. Thus, one treated subject is a representative of the five clones

-weights for the treated subjects are $\frac{1}{P(Z=1|X)} = \frac{1}{e(X)}$

-weights for the control subjects are $\frac{1}{P(Z=0|X)} = \frac{1}{1-e(X)}$

→ $(Y_i(1), Y_i(0)) \perp\!\!\!\perp Z_i$ for the pseudo-population

-the IPW method can identify the ATE as

$$\tau = E[\frac{Z_i Y_i}{e(X_i)}] - E[\frac{(1-Z_i)Y_i}{1-e(X_i)}]$$

-using IPW, we can estimate the ATE

- $\tau^{ATE} = E[Y(1) - Y(0)]$
- since the true $e(X_i)$ is not known, use the PS estimates $\hat{e}(X_i) = \hat{e}_i$
- $\hat{\tau}^{ATE} = \frac{1}{n}\sum_{i=1}^{n} \frac{Z_i Y_i}{\hat{e}_i} - \frac{1}{n}\sum_{i=1}^{n} \frac{(1-Z_i)Y_i}{1-\hat{e}_i}$
- we can fit a regression model with weights $w_i = Z_i \cdot \frac{1}{\hat{e}_i} + (1 - Z_i) \cdot \frac{1}{1-\hat{e}_i}$
- remark. when $\hat{e}_i \approx 0$ or $\hat{e}_i \approx 1$ (i.e., large w_i), $\hat{\tau}^{ATE}$ can be unstable

Different Target Population

-we can estimate different target estimates such as the average treatment effect on treated (ATT)

$$\tau^{ATT} = E[Y(1) - Y(0) \mid Z = 1]$$

-this is what 1:1 matching targets

-in weighting, we can use weights 1 for treated and $\frac{\hat{e}_i}{1-\hat{e}_i}$ for control

-ATT = ”How much have treated subjects already gained or benefited from the treatment?”

-c.f. $\tau^{ATC} = E[Y(1) - Y(0) \mid Z = 0]$ → ATE는 ATC와 ATT의 weighted average

Full Matching

-1-1 matching의 확장

- 1:1 matching → variance 줄어들지만 사람들이 sample을 버리려 하지 않음..