Machine Learning II

Junhee Seok, Ph.D.
Associate Professor
School of Electrical Engineering
Korea University, Seoul, Korea

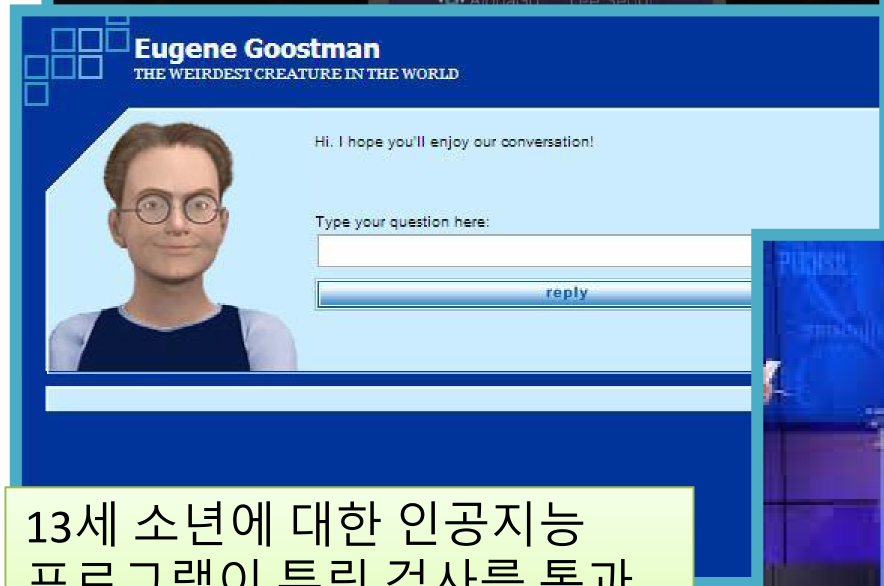
Contents

Overview of Machine Learning

인공지능: 인간처럼 생각하는 기계

알파고와 이세돌 9단의 대국 (2016)

바이두의 초거대 딥러닝 신경망 계획(2015)



13세 소년에 대한 인공지능 프로그램이 튜링 검사를 통과 (2014)

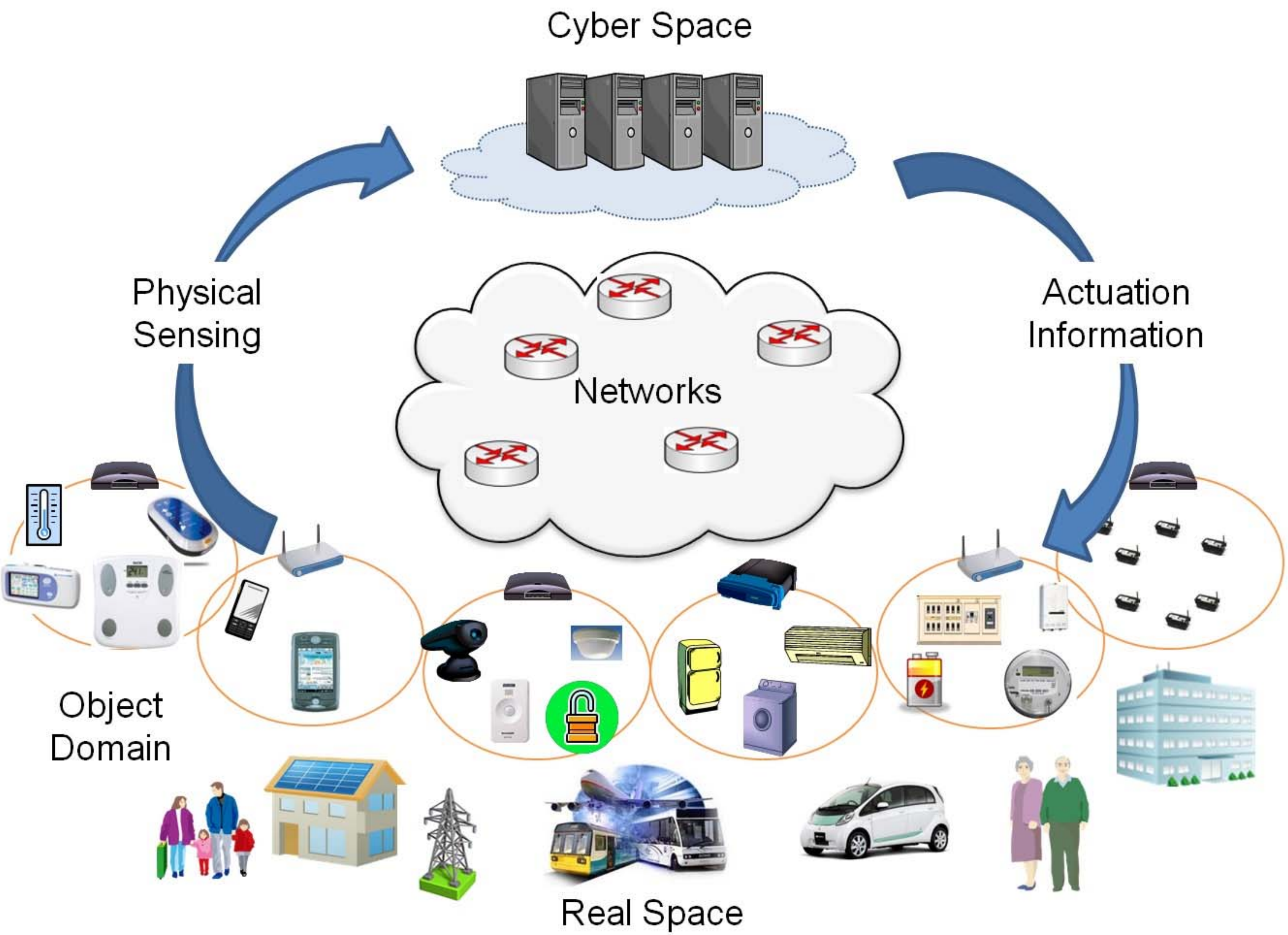
IBM 왓슨이 인간 챔피언과의 퀴즈 대결에서 승리 (2011)

인공지능의 정의

- 인공지능이란?
  - **지능이라는 추상적 개념을 과학적으로 구체화하고, 이를 인공적으로 재현하는 기술**
- 강한 인공지능 (Strong AI; 범용인공지능)
  - 인간처럼 혹은 초인간적인 방법으로 실제 사고를 통해 문제를 해결하는 기술
  - 예: 창조, 감성, 사고
- 약한 인공지능 (Weak AI)
  - 인간의 지능을 모방하여 지적 문제를 해결하는 기술
  - 예: 지식의 발굴, 자료 처리, 상황 판단

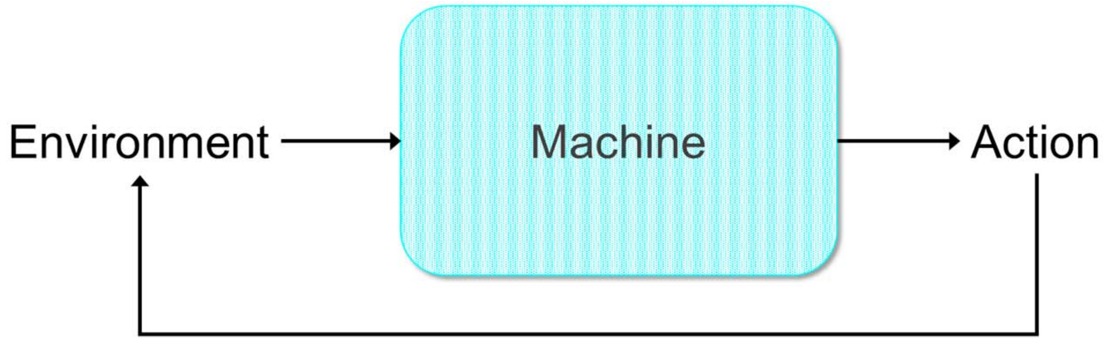
4차 산업혁명과 인공지능

- 사이버물리시스템: 4차 산업 혁명의 기술을 포괄하는 개념



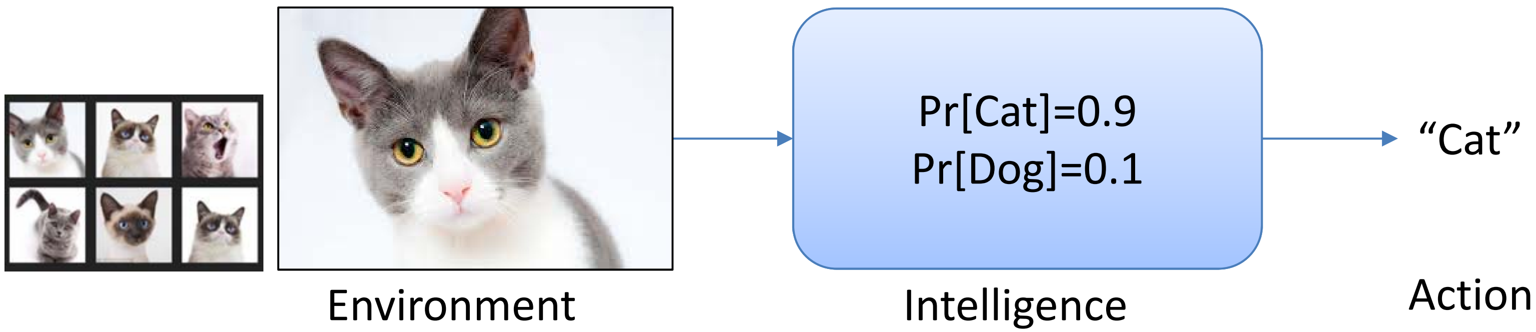
AI vs. Machine Learning

- Intelligence needs to “perceive and decide” for “actions” from “observed data”

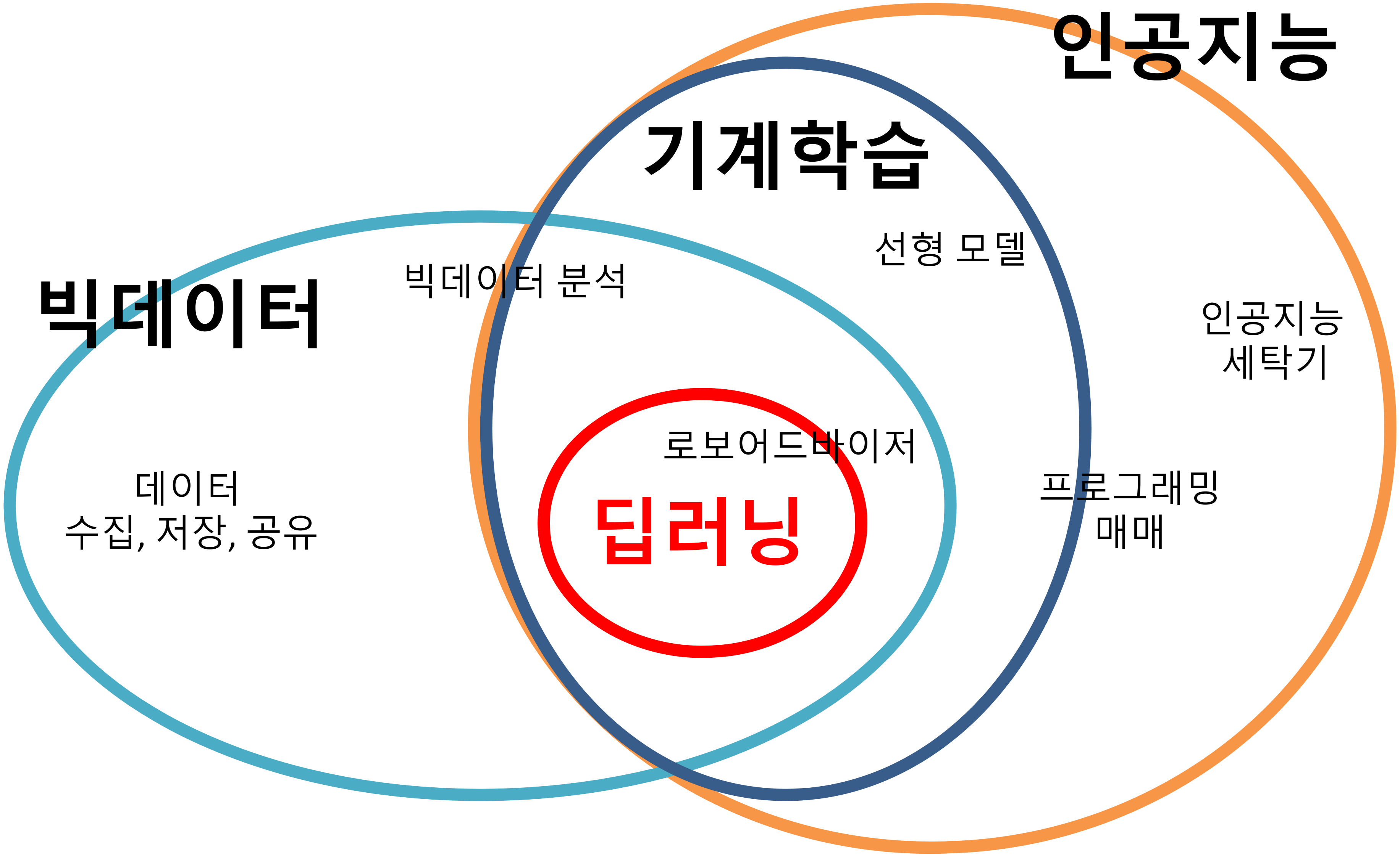


$$\Pr[Y|X]$$

- Environment: random data from a certain distribution (e.g. weather , speech, …)
- Action: often based on probabilistic predictions



인공지능, 빅데이터, 기계학습, 딥러닝??



Python Review

Pandas Module

- An open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis
  - A set of labeled array data structures, the primary of which are Series and DataFrame
  - Index objects enabling both simple axis indexing and multi-level / hierarchical axis indexing
  - An integrated group by engine for aggregating and transforming data sets
  - Date range generation (date_range) and custom date offsets enabling the implementation of customized frequencies
  - Input/Output tools: loading tabular data from flat files (CSV, delimited, Excel 2003), and saving and loading pandas objects from the fast and efficient PyTables/HDF5 format.
  - Memory-efficient “sparse” versions of the standard data structures for storing data that is mostly missing or mostly constant (some fixed value)
  - Moving window statistics (rolling mean, rolling standard deviation, etc.)

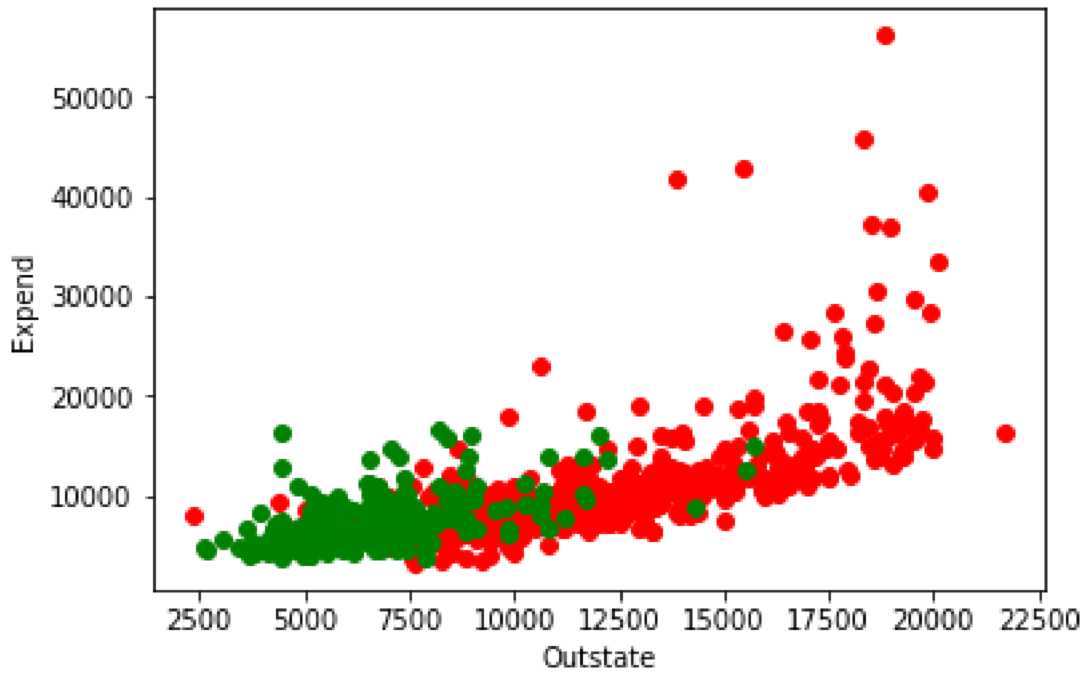
Pandas Module

- Usual Python data structures are for numeric data or string data.
  - Not suitable for data analytics with mixed data types
- Pandas provides a data frame that can include numeric and categorical data

- pandas.read_csv()
- pandas.DataFrame()
- pandas.Series()
- pandas.DataFrame.iloc(), pandas.DataFrame.loc()
- Pandas.DataFrame.plot()

Practice

- Read data02_college.csv and answer the following questions
  - How many colleges? How many private and public?
  - The average expend of private and public colleges?
  - What are the top 10 schools in terms of top 10% of high school class?
  - What are the top 10 schools in terms of acceptance ratio?
  - Plot outstate tuition and expend with different colors for private and public schools.



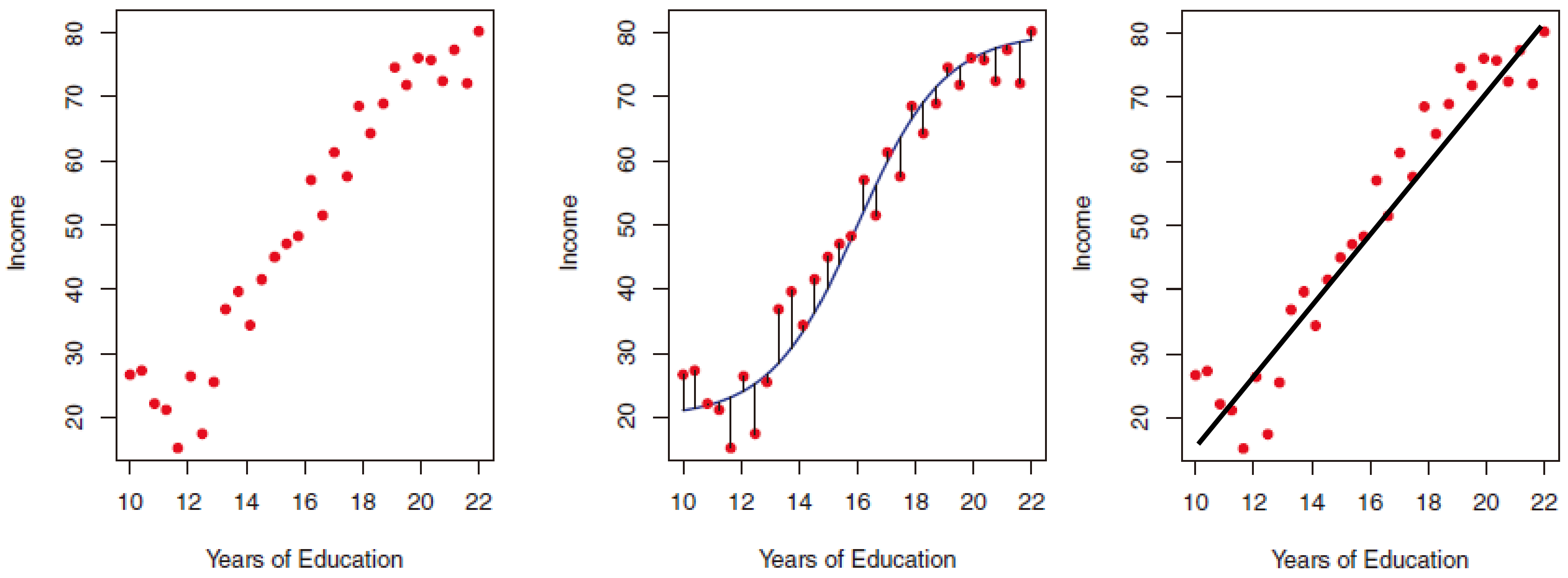
Machine Learning Fundamentals

Machine Learning

- For an output Y and input $X = (X_1, X_2, \ldots, X_p)$, the relation can be generally presented by

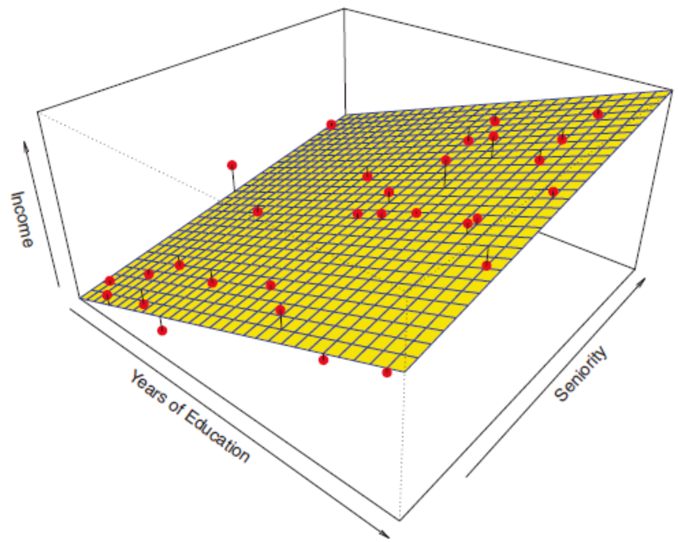
$$Y = f(X) + \epsilon$$

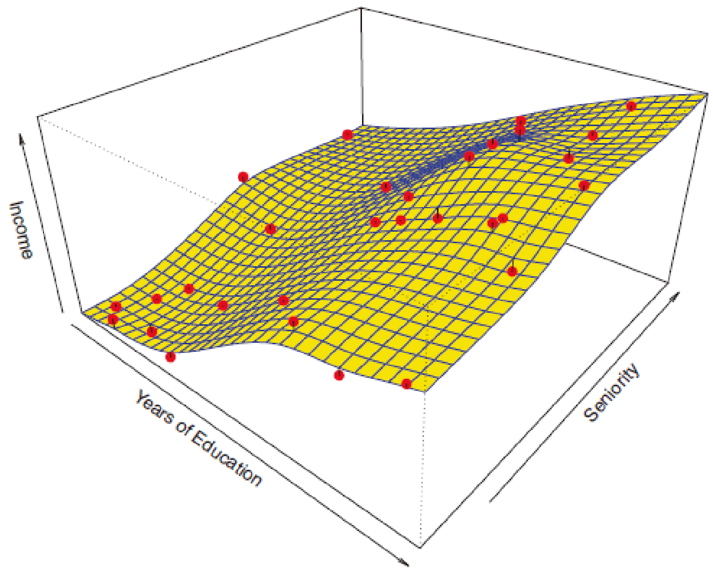
- $f()$ is unknown, can be a simple linear function or complicated non-linear form.
- We want to estimate $f()$ for *prediction* and *inference*.



Estimating *f*()

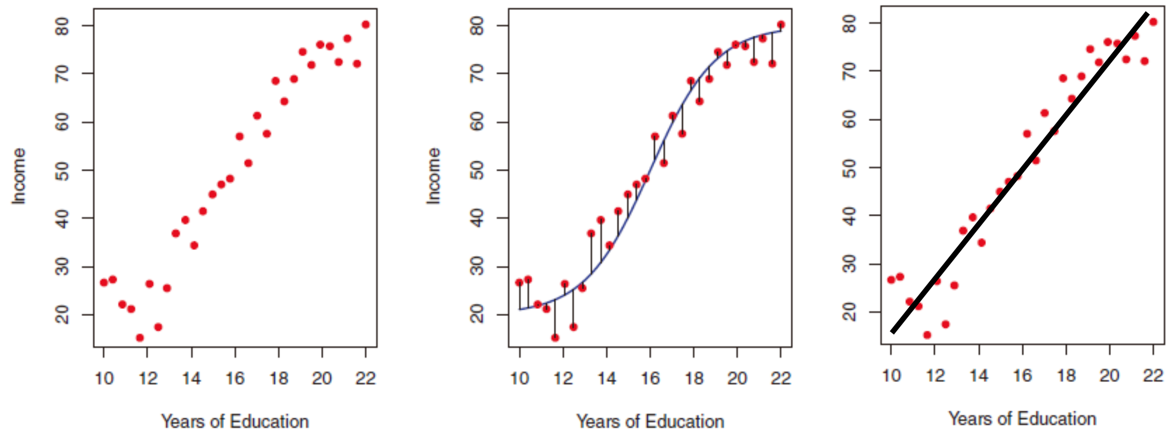
- **Parametric** approaches assumes a certain type of relation (usually linear).
  - e.g. income ~ $\beta_0 + \beta_1$ x education + β_2 x seniority
- **Nonparametric** approaches assumes no prior relation.
  - Fitting *Y* not much roughly and not much wiggly ☺
  - e.g. tree-based methods, splines



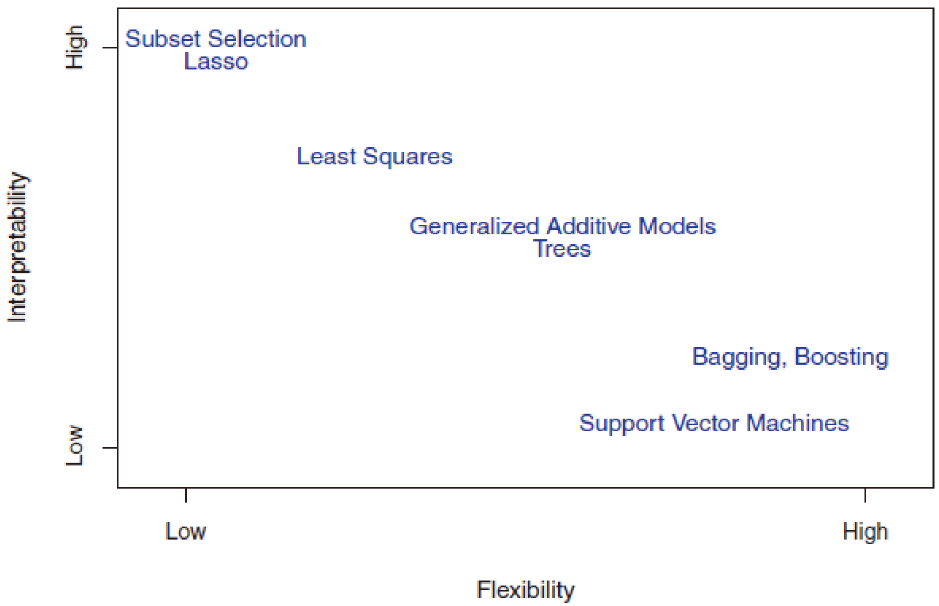


Trade-off between Flexibility and Interpretability

- The spline function is more flexible, but the linear line is more interpretable.

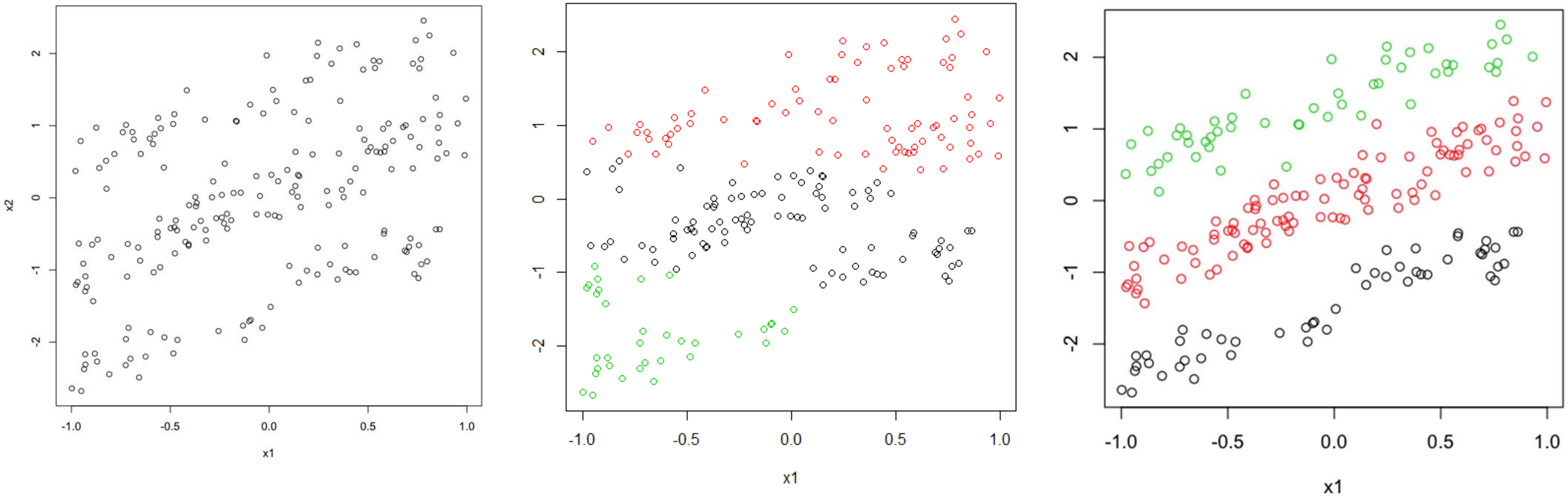


- Trade-off of various methods.



Supervised vs. Unsupervised Learning

- **Supervised learning**: both Y and X are given.
  - E.g. prediction and inference.
- **Unsupervised learning**: Y is unknown or hidden, only X is given.
  - E.g. clustering, which one looks better?



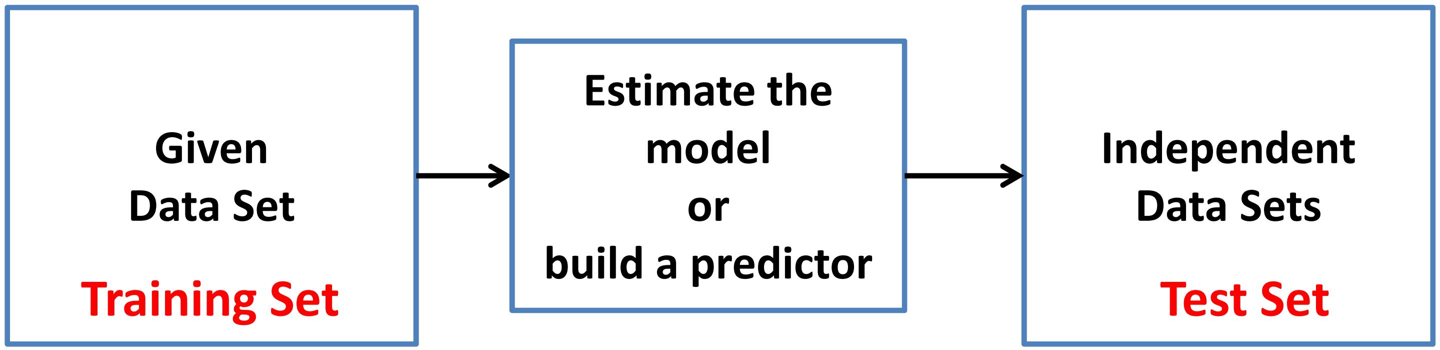
- **Reinforcement Learning**: learning Y by trying X
  - X-Y pairs are not given initially
  - X-Y pairs are collected through trials

Supervised Learning Problem Setting

- In a real problem, the accuracy $E(Y - \hat{f}(X))^2$ is often measured by mean square error (MSE),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

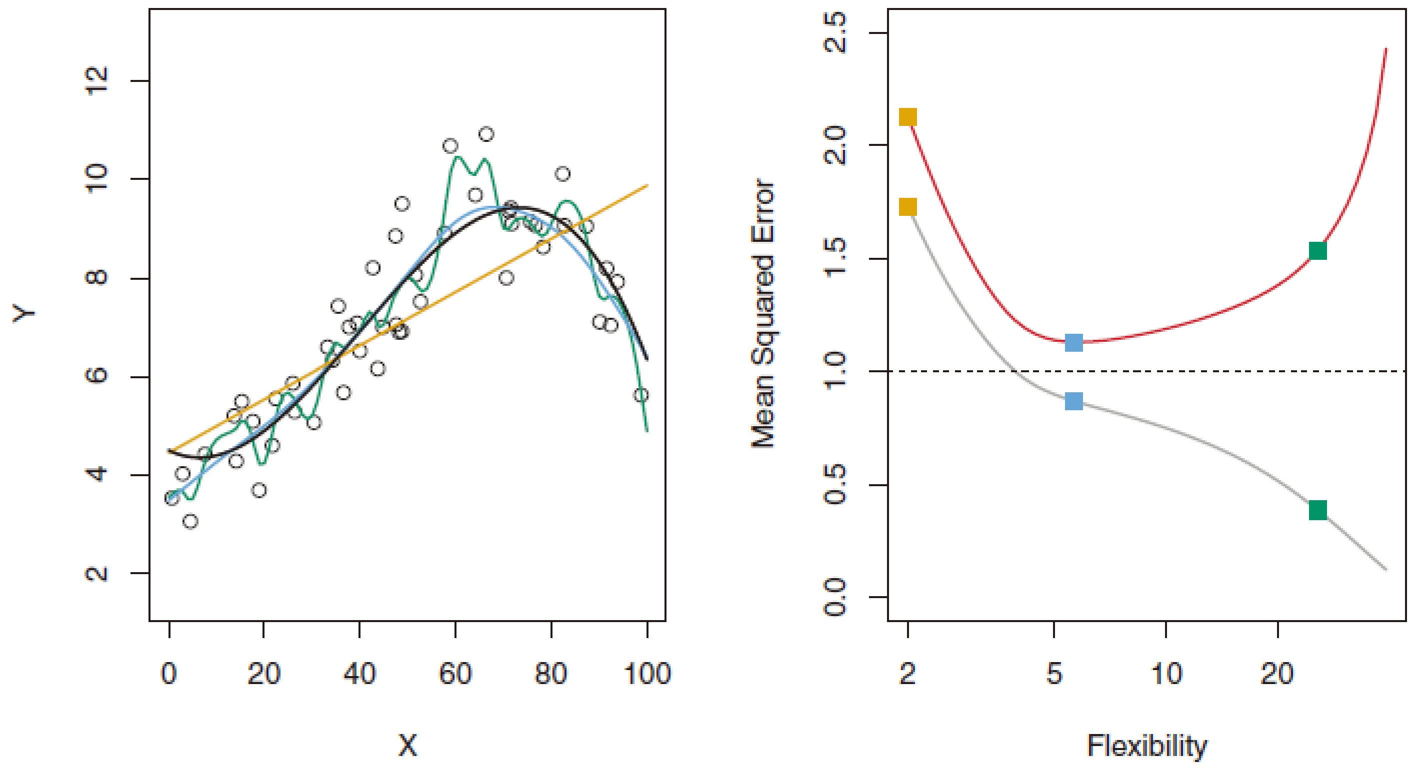
- *Given a data set*, we want to estimate *f*() that will make small MSEs in *other data sets*.
  - Estimating a relation between salary and education using data from Seoul, and confirming it using data from Pusan.
  - Building a predictor of heart attacks with blood pressure using data in 2001~2005, and testing its performance using data in 2006~2010.



Note that we know nothing about the test set when building a predictor.

Training MSE vs. Test MSE as Model Flexibility

- We can reduce the MSE of the training set as much as we want by increasing the model flexibility.
- **Overfitting**: too much flexibility increase the MSE of the test set.
- The model flexibility is often referred by the *degree of freedom*.

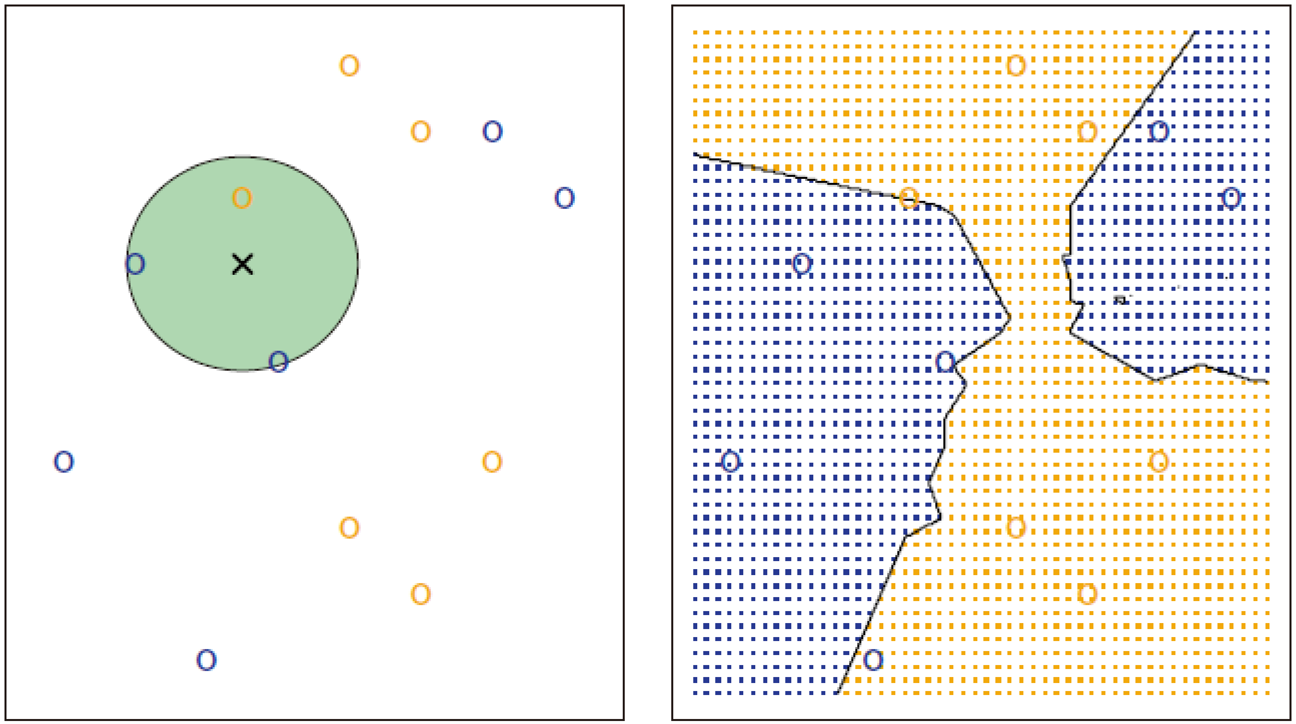


Simple Methods: KNN Method

- **K-nearest neighbor (KNN) method**
  - Determine the outcome of a sample using the k nearest samples of known outcomes.

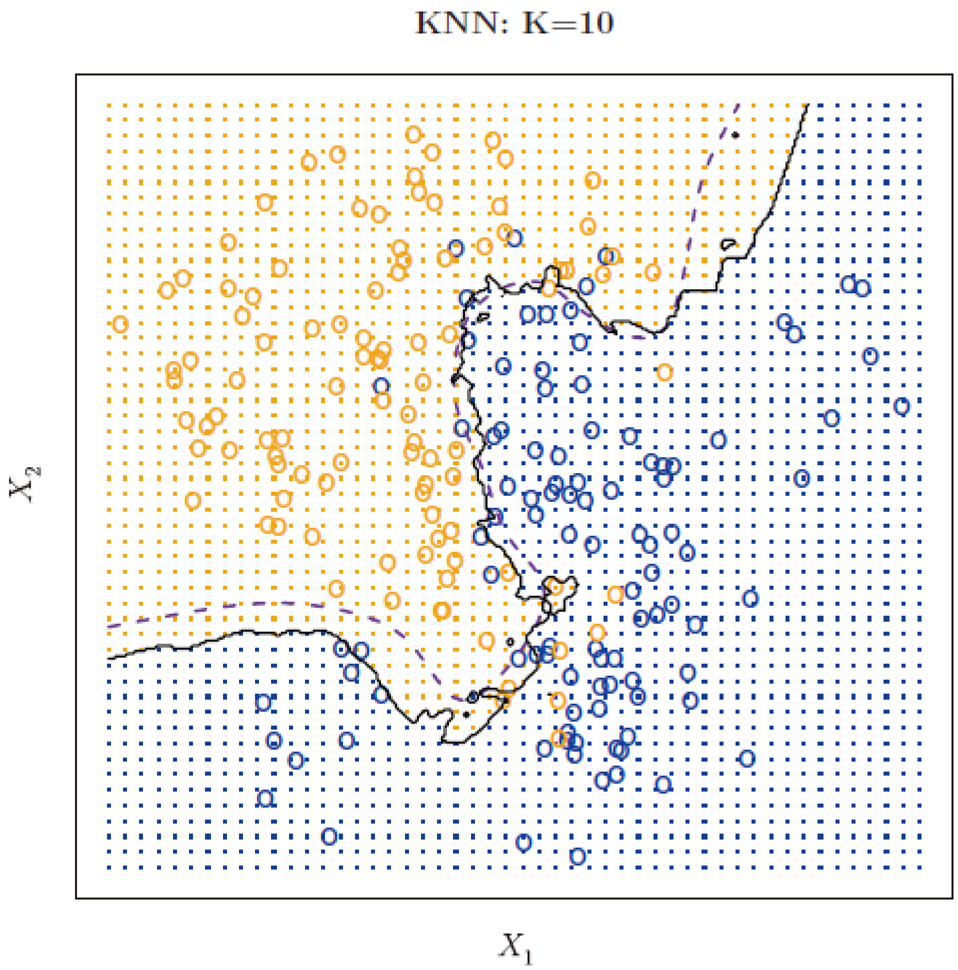
$$\hat{Y} = \frac{1}{K}\sum Y_k$$

  - Totally nonparametric, simple but powerful.



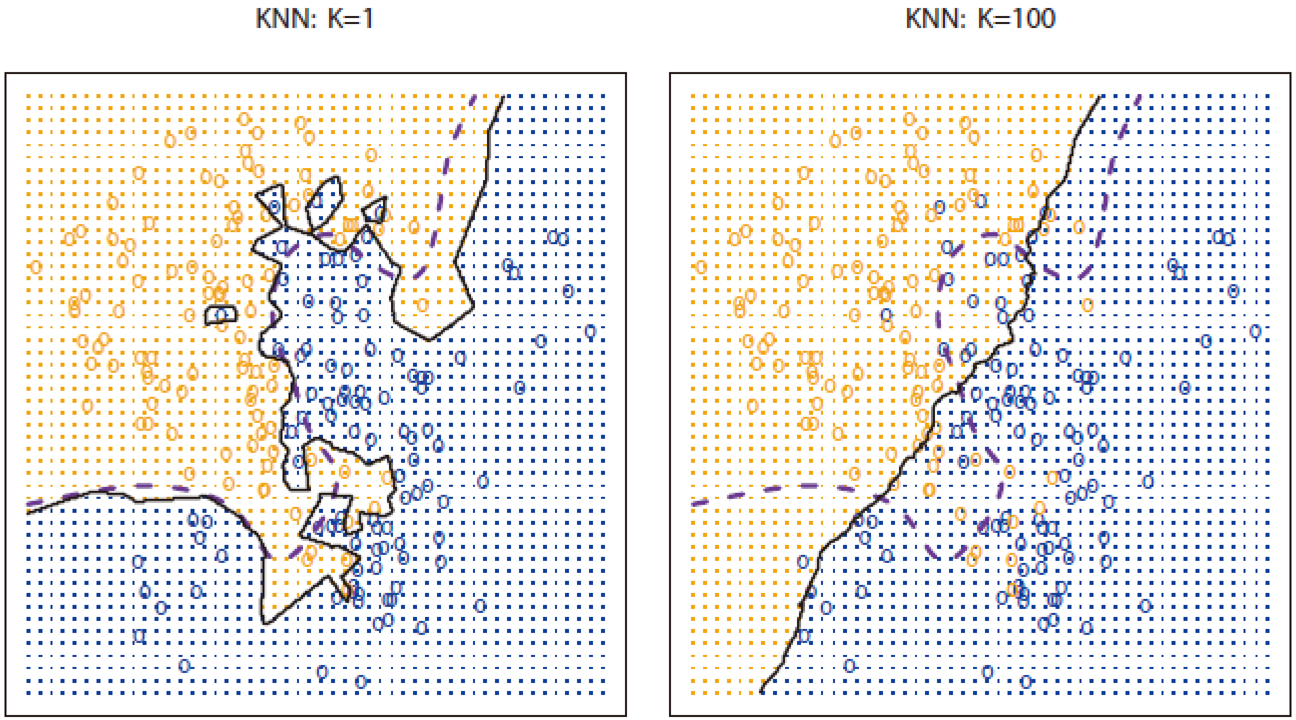
Example: Classification with KNN

- K = 10 seems to have a good fit.



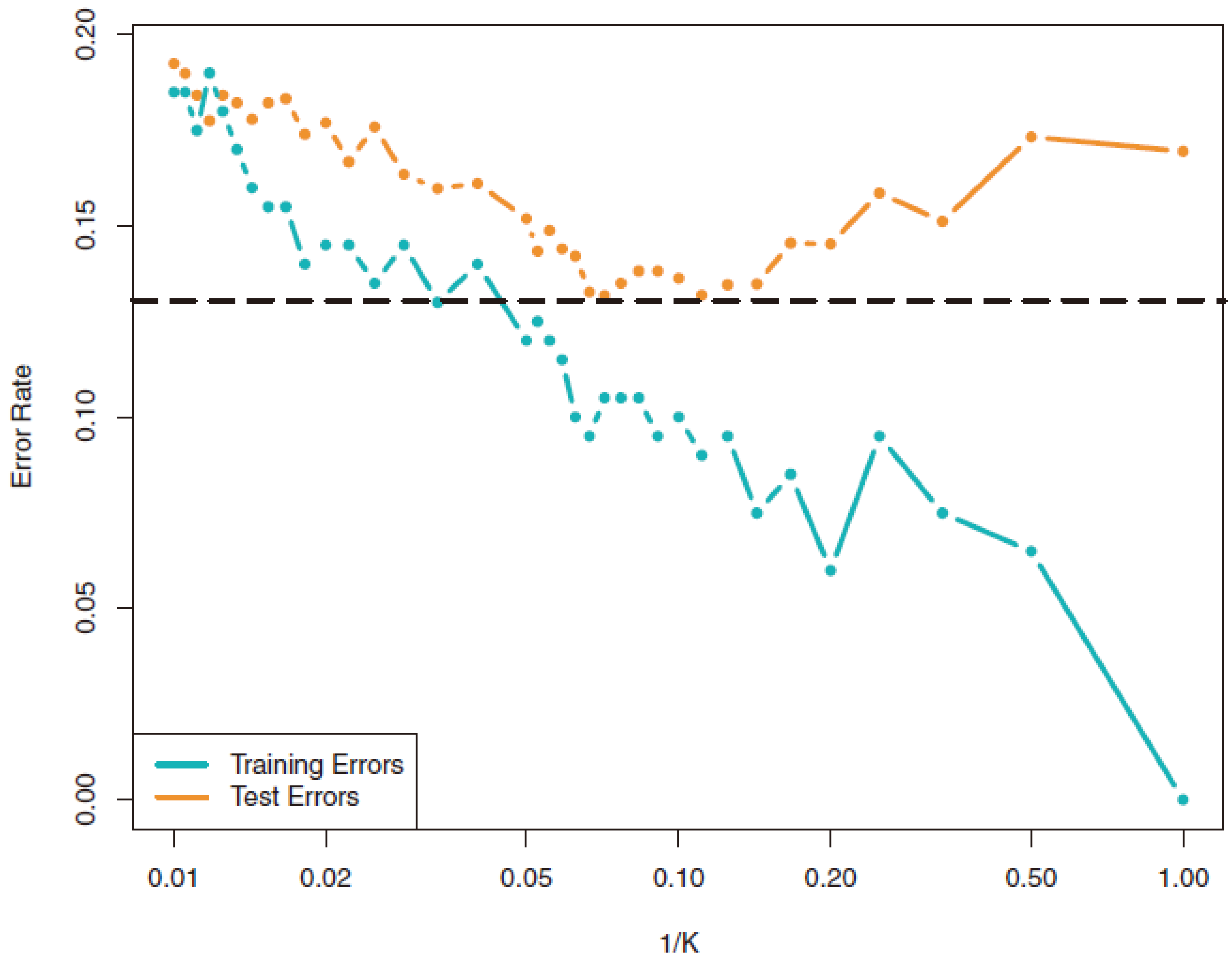
Example: Classification with KNN

- K = 1 is wiggly, and K=10 is rough.
- The model flexibility is controlled by K.



Example: Classification with KNN

- Bias-variance tradeoff



Machine Learning Contents

- Supervised learning

	Regression	Classification
Linear Model	Linear Regression	Logistic Regression
Discriminant Analysis		LDA/QDA
Nonparameteric	KNN	KNN, Naïve Bayesian
Tree	Regression Tree	Classification Tree
Ensemble	Random Forest, Boosted Tree	
Support Vector	Support Vector Regression	Support Vector Machine
Neural Networks	Multi-layer Perceptron and others Deep learning	

- Unsupervised learning
  - Dimension reduction: PCA, ICA, Autoencoder
  - Clustering: K-means, hierarchical
- Model selection
  - Cross-validation
  - Feature selection, penalization

General Procedure of Machine Learning

- Set up a problem → what is Y?
- Data collection → collect X and Y
  - (Traditionally) design a study, perform experiments, and collect data
  - (Now) dig up a database and collect any related data
- Preprocessing
  - Transform data into usable form
- Exploratory data analysis (EDA)
  - See how data looks like
- Data analysis (prediction)
  - Initial data size: $n = 100M$, $p = 10k$
  - Separate training and test set (often by data collection time)
  - Select features via univariate statistical test (t-test, cor-test)
  - (optionally) Dimension reduction (PCA)
  - Select learning methods (often based on intuition)
  - Model selection via cross-validation (methods & parameters)
  - Test performance over the test set
- Validation with a totally new data set (often not existing data when the model is built)

Practices

- sklearn.neighbors.KNeighborClassifier

- Practice
  - college 데이터 셋을 읽어 Private을 다른 변수를 이용하여 KNN 방식으로 예측하시오.
  - 이때, Train와 Test 셋을 나누고 Train 셋을 이용하여 모델을 학습하시오.
  - train_test_split을 이용하여 나누되 test_size=0.4, random_state=0를 이용하시오.
  - K=1, K=10, K=20 에 대하여 train/test accuracy를 구하고
  - 결과를 모델의 유연성과 관련지어 생각해보시오.
  - (추가문제) K를 1부터 20까지 변화시키가며 train/test accuracy의 그래프를 그리시오

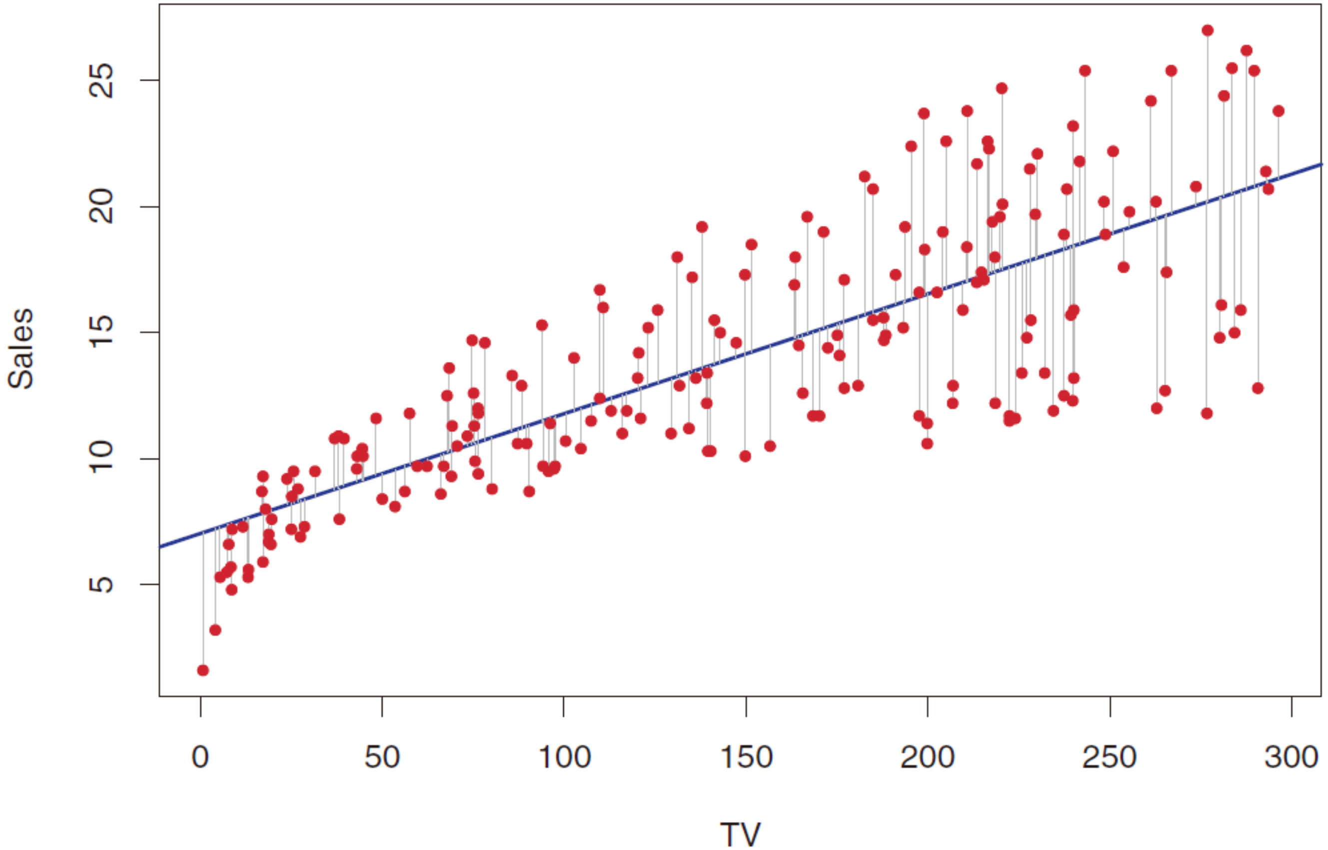
Linear Regression

Simple Linear Regression

- One output variable and one input variable

$$Y \approx \beta_0 + \beta_1 X$$

- The data is modeled by $y_i = \beta_0 + \beta_1 x_i + e_i$ for $i = 1 \cdots n$. We estimate the real β_0 and β_1 by $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.



Simple Linear Regression

- **RSS**: residual sum of squares

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)^2$$

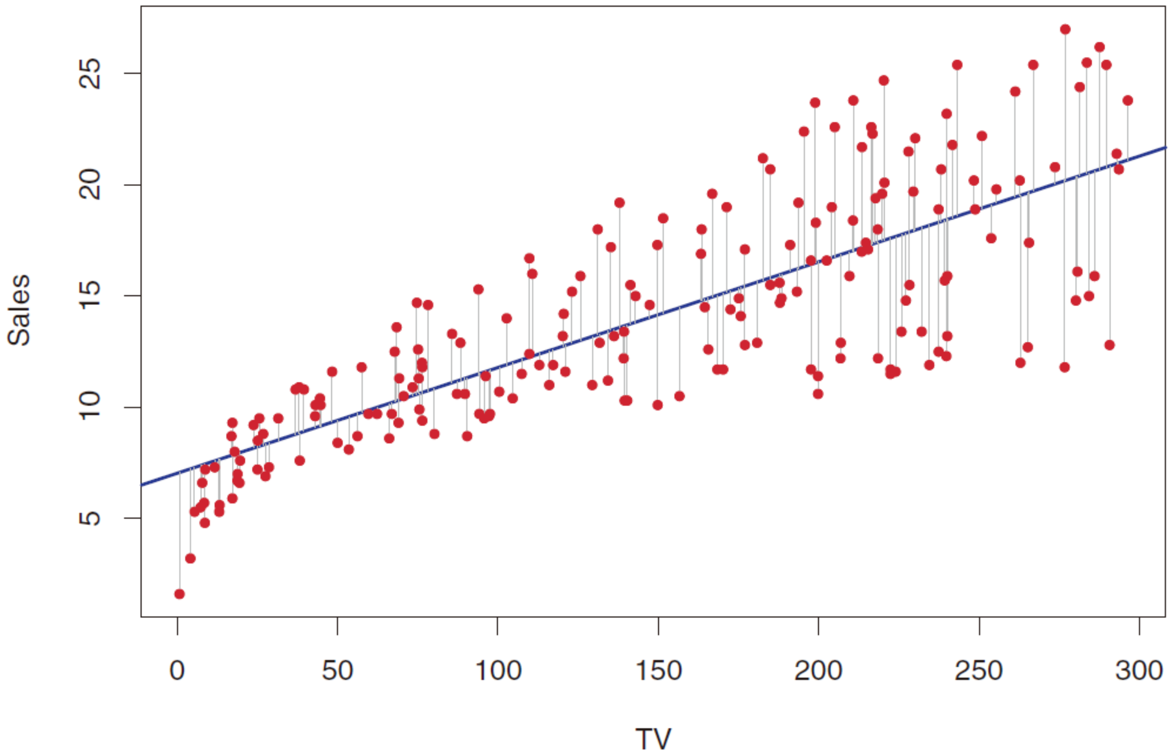
- We estimate coefficients by minimizing RSS.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

Accuracy of the Coefficient Estimates

- Sales vs. TV advertisement



	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

Accuracy of the Model

- Metrics to measure how the model fits the data well.

- RMSE: residual standard error

$$\text{RMSE} = \sqrt{\frac{\text{RSS}}{n}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

- R^2: the proportion of variance of Y explained by X.
  - TSS: total sum of squares, $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$.

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

  - In a simple linear regression setting, $R^2 = \mathrm{Cov}(X,Y)^2$.

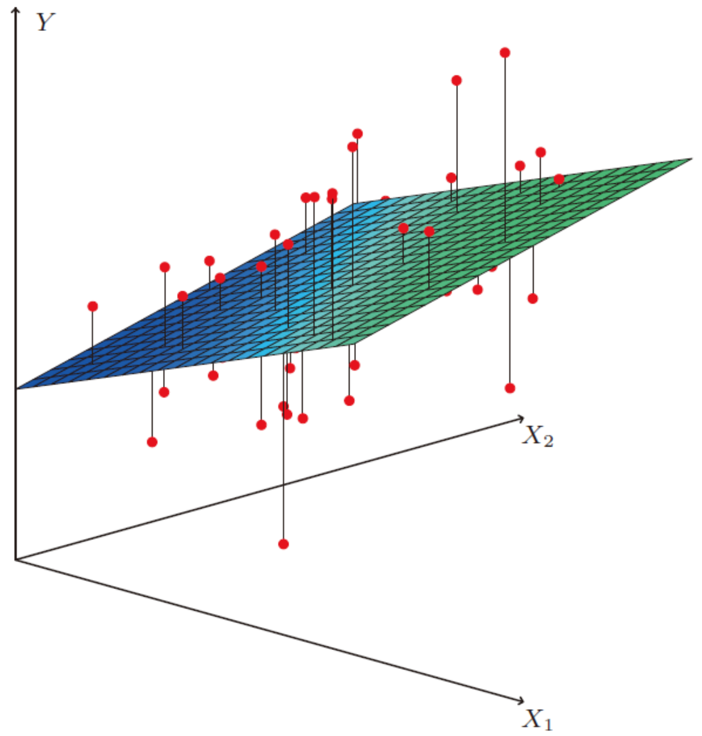
Multiple Linear Regression

- One output variable and several input variables

$$Y \approx \beta_0 + \beta_1 X_1 + \cdots \beta_p X_p$$

- We estimate $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ by minimizing the RSS,

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \cdots \hat{\beta}_p x_{pi}$$



Multiple Linear Regression

- It is often represented by a matrix form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix} \qquad \mathbf{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}$$

- The estimates are $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, and $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

- RSE $\hat{\sigma} = \sqrt{\frac{RSS}{n-p-1}} = \sqrt{\frac{(\mathbf{y}-\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}})}{n-p-1}}$, and SE matrix $\mathrm{SE}\big(\widehat{\boldsymbol{\beta}}\big) = \hat{\sigma}(\mathbf{X}^T\mathbf{X})^{-1}$.
  - $\mathrm{SE}\big(\hat{\beta}_j\big) = [\hat{\sigma}(\mathbf{X}^T\mathbf{X})^{-1}]_{jj}$
- The 95% confidence interval is $\hat{\beta}_j \pm 1.96\mathrm{SE}\big(\hat{\beta}_j\big)$.
- Hypothesis test against $\beta_j = 0$ using t-statistics $t = \frac{\widehat{\beta}_j - 0}{\mathrm{SE}(\widehat{\beta}_j)} \sim T(n-p-1)$.

Simple vs. Multiple Linear Regression

- Multiple: $\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{radio} + \beta_3 \texttt{newspaper} + \varepsilon.$
- Simple: $\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \varepsilon.$

- Which one is more flexible? Has higher d.f.? Smaller RSS? Higher R^2?

Simple vs. Multiple Linear Regression

- Significance of coefficients

	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	−0.001	0.0059	−0.18	0.8599

P-values assuming the other factors have no change.

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	9.312	0.563	16.54	< 0.0001
radio	0.203	0.020	9.92	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	< 0.0001

P-values without considering other factors.

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Variable Selection

- In the `Advertising` example, we see `newspaper` is less important.

- There are some data sets of which $p > n$: high-dimensional data.
  - Typically in genomics, ~100 patients vs. ~20,000 genes.

- In general, **variable selection** guides to finding important variables among many variables.

- Bias-variance problem
  - More variables mean more flexibility.
  - Many variables can reduce RSS in a training set, but may overfit it.

Categorical Predictors

- In a linear model, X's can be categorical variables.

$$Y \approx \beta_0 + \beta_1 X_1 + \cdots \beta_p X_p$$

  - X_1: Age; continuous variable.
  - X_2: Gender; categorical variable with two levels (Male and Female).
  - X_3: Ethnicity; categorical variable with three levels (Asian, Caucasian, AA).

- We introduce dummy indicator variables.

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male,} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

- It is fundamentally the same with

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 - \beta_1 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

Categorical Predictors

- More than two variables

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian,} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian.} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i\text{th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is African American.} \end{cases}$$

- AA is the **baseline**, and β_1 and β_2 are **additional** effects.
- We can apply the same statistics for confidence interval and hypothesis test.

	Coefficient	Std. error	t-statistic	p-value
Intercept	531.00	46.32	11.464	< 0.0001
ethnicity[Asian]	-18.69	65.02	-0.287	0.7740
ethnicity[Caucasian]	-12.50	56.68	-0.221	0.8260

Beyond the Additive Assumption

- Additive model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon.$
- Considering interactions: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon.$
- We consider $X_1 X_2$ as a new variable X_3.

- Example: $\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{radio} + \beta_3 \texttt{TV x radio} + \varepsilon.$

	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

- β_3 is for the interaction between TV and radio.

$$\begin{aligned} \texttt{sales} &= \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times (\texttt{radio} \times \texttt{TV}) + \epsilon \\ &= \beta_0 + (\beta_1 + \beta_3 \times \texttt{radio}) \times \texttt{TV} + \beta_2 \times \texttt{radio} + \epsilon. \end{aligned}$$

  - The effect size of TV is affected by radio.

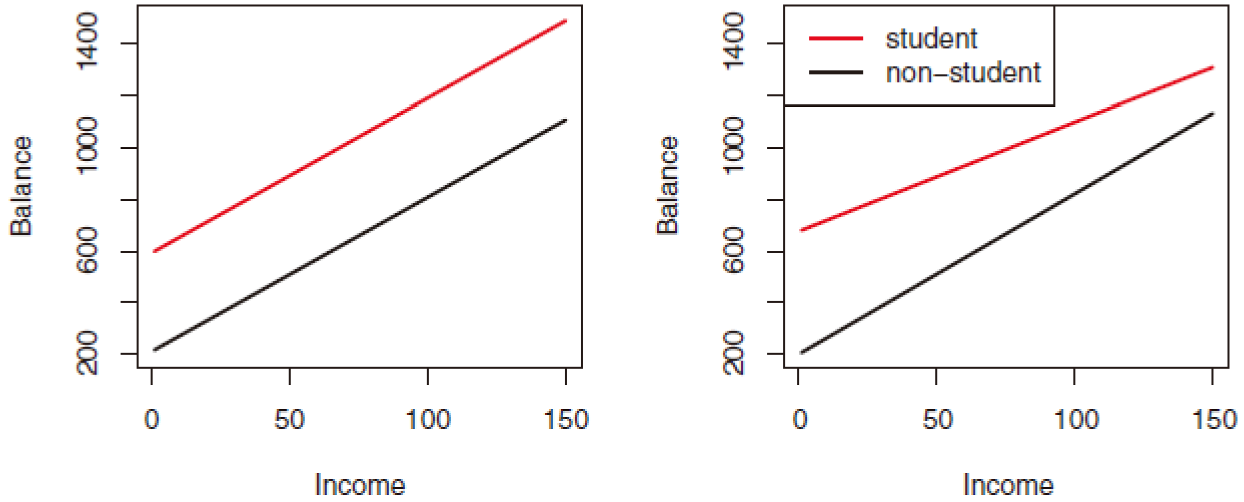
- Hierarchical principle
  - To include an interaction term, we should also include the original variables even though single variables might not be significant.

Beyond the Additive Assumption

- One more example: interaction between continuous and categorical variables.

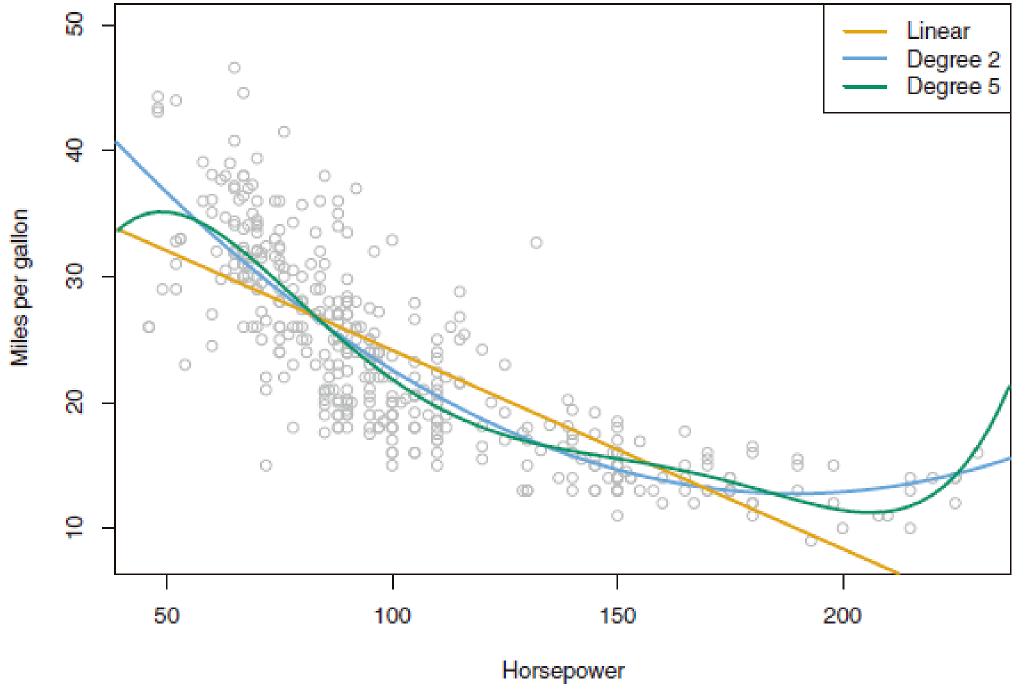
$$\begin{aligned} \texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases} \beta_2 & \text{if } i\text{th person is a student} \\ 0 & \text{if } i\text{th person is not a student} \end{cases} \\ &= \beta_1 \times \texttt{income}_i + \begin{cases} \beta_0 + \beta_2 & \text{if } i\text{th person is a student} \\ \beta_0 & \text{if } i\text{th person is not a student.} \end{cases} \end{aligned}$$

$$\begin{aligned} \texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases} \beta_2 + \beta_3 \times \texttt{income}_i & \text{if student} \\ 0 & \text{if not student} \end{cases} \\ &= \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \texttt{income}_i & \text{if student} \\ \beta_0 + \beta_1 \times \texttt{income}_i & \text{if not student} \end{cases} \end{aligned}$$



Non-linear Relationship

- $Y \approx \beta_0 + \beta_1 X$ vs. $Y \approx \beta_0 + \beta_1 X + \beta_2 X^2$.
- Consider high-order variables as new variables.
- Higher-order? Bias-variable tradeoff.



	Coefficient	Std. error	t-statistic	p-value
`Intercept`	56.9001	1.8004	31.6	< 0.0001
`horsepower`	-0.4662	0.0311	-15.0	< 0.0001
`horsepower`^{2}	0.0012	0.0001	10.1	< 0.0001

Summary

- Linear regression is
  - Easy to calculate coefficients,
  - Easy to test relationship between output and input variables,
  - Easy to interpret the result.

- If the true relationship is not linear, linear regression might not be effective.
  - Using interaction terms
  - Using higher-order terms

- Always, bias-variance tradeoff
  - Including more input variables fits the training data well, but might be harmful.

Practices

- Analysis of Boston data set
  - sklearn.linear_model.LinearRegression
  - fit, predict, score

- data01_iris.csv를 읽으시오. Sepal Width ~ Sepal.Length + Petal.Length + Petal.Width 로 선형 회귀 분석을 수행하시오.
  - (1) R2와 RMSE 값은 얼마인가?
  - (2) 어떤 변수의 제곱항을 추가하였을 때, 가장 높은 R2를 갖는 것은 어느 변수인가?
  - (3) Sepal.Length와 Petal.Length의 interaction 항을 추가하였을 때, R2은 얼마인가?
  - (4) 범주형 변수 Species를 포함시켜 선형 회귀 분석을 수행하시오.

Classification

Regression vs. Classification

- **Regression**
  - Y is continuous, e.g. height, age, sales.
- **Classification**
  - Y is discrete without order, e.g. car type, ethnicity, symptoms.

- Can we use regression for classification?
  - Possible for binary outcomes, but not obvious for outcomes with more than two levels.

$$Y = \begin{cases} 0 & \text{if } \texttt{stroke}; \\ 1 & \text{if } \texttt{drug overdose}. \end{cases} \qquad Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

- We often apply methods dedicated to classification.
- Many classification methods **model the probability of a certain class**.
  - $\Pr[Y = k|X] \sim f(X)$ instead of $Y \sim f(X)$

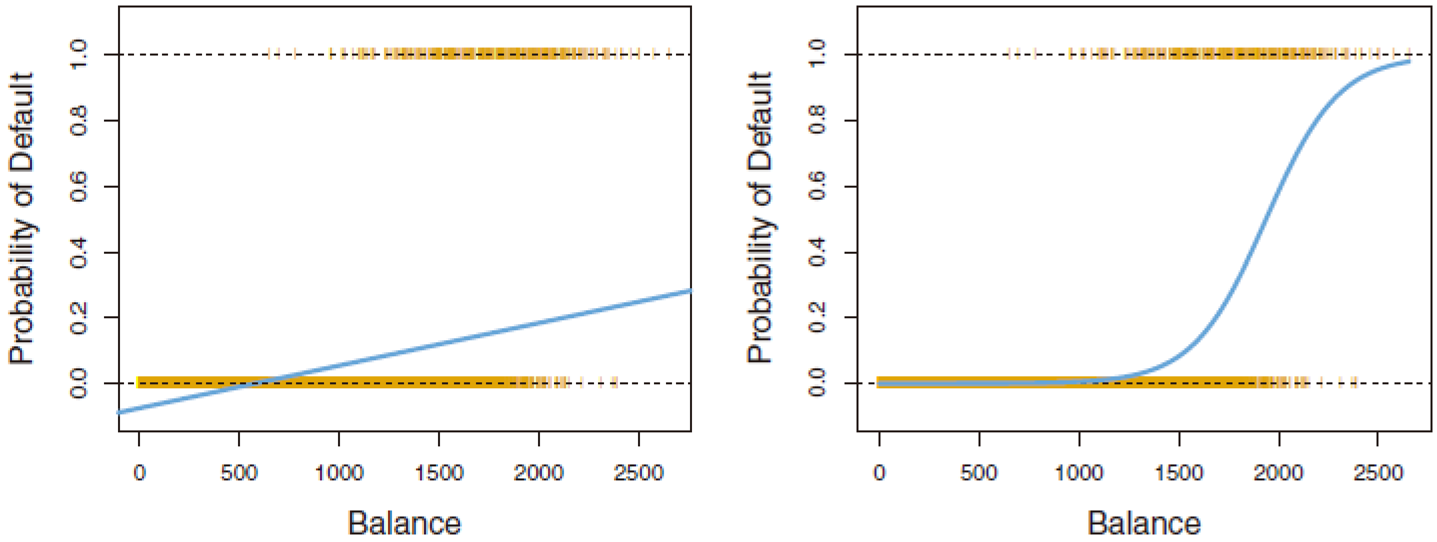
Logistic Regression

- Consider a simple binary output Y (0 or 1) and one input X. Let $p(X) = \Pr[Y{=}1|X]$.

- People like a linear model: $p(X) \approx \beta_0 + \beta_1 X$.
- However, the probability should be non-negative: $p(X) \approx e^{\beta_0 + \beta_1 X}$.
- The probability should be $0 \sim 1$.

Logit of p(X) =
Prob. of X / Prob. of Not X

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



Estimating Coefficients

- **Least square (non-linear)**

$$MSE = \sum_{i=1}^{n} \left(1 - \widehat{\Pr}[y_i = k_i | x_i]\right)^2$$

  – Binary logistic regression

$$MSE = \sum_{i:y_i=0} \left(1 - \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}}\right)^2 + \sum_{i:y_i=1} \left(1 - \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}\right)^2$$

- **Maximum likelihood (ML)**
  – Likelihood: probability of observation under a certain model.

$$l = \prod_{i:y_i=0} \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} \prod_{i:y_i=1} \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

  – ML is preferred in non-linear models.

Example

- Output: default (Yes or No)

z-statistic instead of t-statistic
β/SE(β) ~ N(0,1)

- Input: Balance (continuous)

	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

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1{,}000}}{1+e^{-10.6513+0.0055\times 1{,}000}} = 0.00576,$$

- Input: student (Yes or No) using a dummy variable

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−3.5041	0.0707	−49.55	<0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Multiple Logistic Regression

- More than one input variable.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p,$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

- Example

	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

  – Confounding: being a student has a negative effect when considering other variables.

General Logistic Regression

- For responses with more than two classes

$$\log\left(\frac{\Pr[Y = 1|X]}{\Pr[Y = K|X]}\right) = \beta_{10} + \beta_{11}X_1 + \cdots + \beta_{1p}X_p$$

$$\log\left(\frac{\Pr[Y = 2|X]}{\Pr[Y = K|X]}\right) = \beta_{20} + \beta_{21}X_1 + \cdots + \beta_{2p}X_p$$

$$\vdots$$

$$\log\left(\frac{\Pr[Y = K-1|X]}{\Pr[Y = K|X]}\right) = \beta_{K-1,0} + \beta_{K-1,1}X_1 + \cdots + \beta_{K-1,p}X_p$$

$$\sum_{k=1}^{K} \Pr[Y = k|X] = 1$$

General Logistic Regression

- Equivalently,

$$\Pr[Y = 1|X] = \frac{e^{\beta_{10}+\beta_{11}X_1+\cdots+\beta_{1p}X_p}}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}X_1+\cdots+\beta_{jp}X_p}}$$

$$\vdots$$

$$\Pr[Y = K-1|X] = \frac{e^{\beta_{K-1,0}+\beta_{K-1,1}X_1+\cdots+\beta_{K-1p}X_p}}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}X_1+\cdots+\beta_{jp}X_p}}$$

$$\Pr[Y = K|X] = \frac{1}{1 + \sum_{j=1}^{K-1} e^{\beta_{j0}+\beta_{j1}X_1+\cdots+\beta_{jp}X_p}}$$

- For the multi-class classification, LDA is often preferred than logistic regression.

Linear Discriminant Analysis (LDA)

- Assuming that
  - Samples of class *k* are generated from a distribution with pdf f_k.
  - **Prior probability** π_k: some classes *naturally* tend to be observed more.
- Then, the final probability or **posterior probability** is

$$p_k(x) = \Pr[Y = k | X = x] = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- LDA assumes **normal distributions** for *f*'s with **different means** but **the same var**.

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right),$$

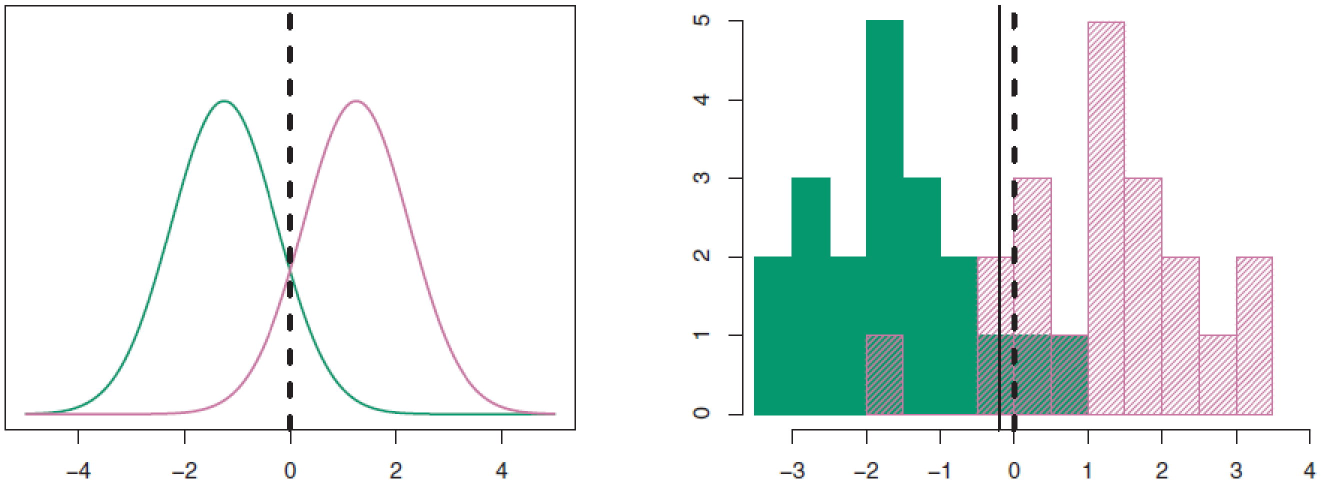
$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}.$$

Linear Discriminant Analysis (LDA)

- LDA assigns the class of which pk(x) is the largest, or the **discriminant function** is the largest.

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

- Example: K=2 and $\pi_1 = \pi_2 = 0.5$.



  – The decision boundary is $(\mu_1+\mu_2)/2$.

Estimation of Parameters

- In LDA, we need to estimate $\pi_1, \ldots, \pi_K$, $\mu_1, \ldots, \mu_K$, and σ.

- We can
  - Minimize MSE: $MSE = \sum_{i=1}^{n} \left(1 - \widehat{\Pr}[y_i = k_i | x_i]\right)^2$.
  - Maximize likelihood: $l = \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \widehat{\Pr}[y_i = k | x_i] I(y_i = k)\right)$.

- In practice, we simply estimate those parameters by

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

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

$$\hat{\pi}_k = n_k / n.$$

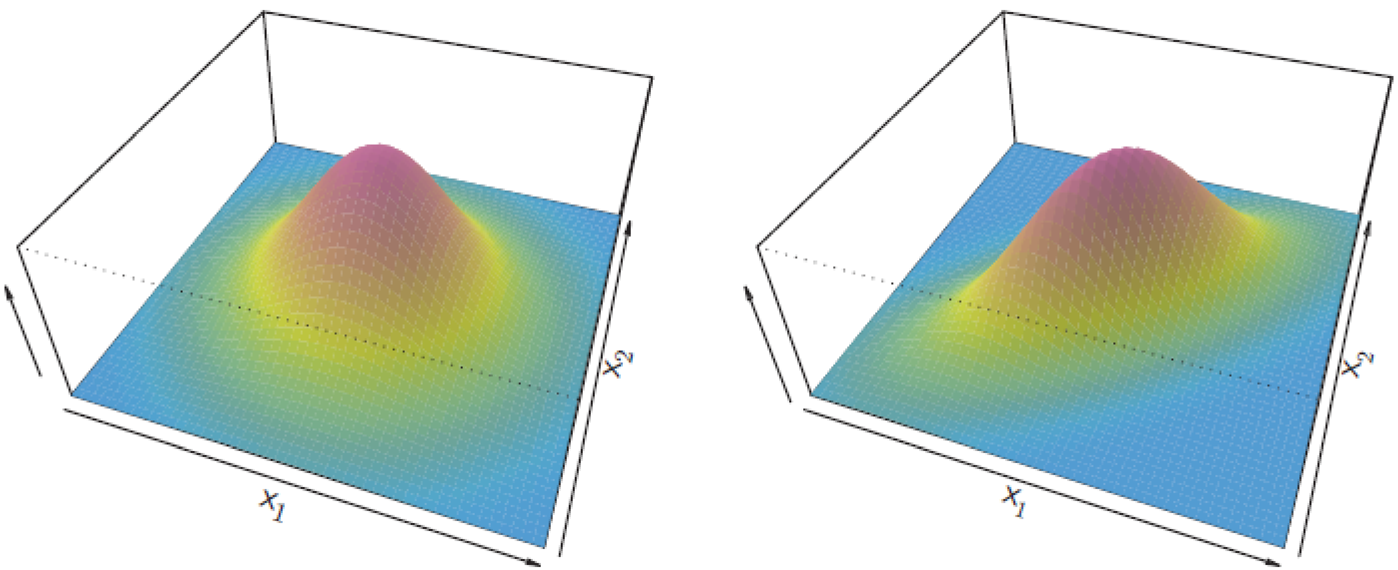
Discriminant function linear to x

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

Multivariate LDA

- For more than one predictor, $X = (X_1, X_2, \ldots, X_p)$ is assumed to be from a joint Gaussian distribution, or $X \sim N(\mu,\Sigma)$.

$$f(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \boldsymbol{\Sigma}^{-1}(x-\mu)\right).$$



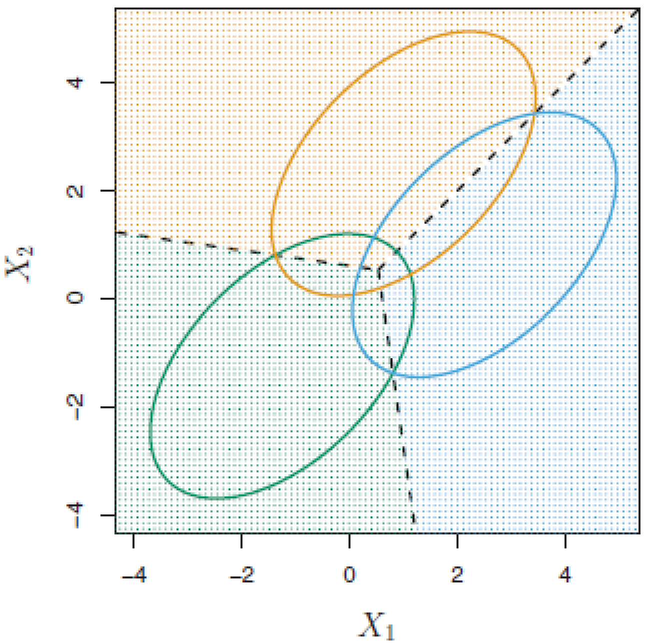
Multivariate LDA

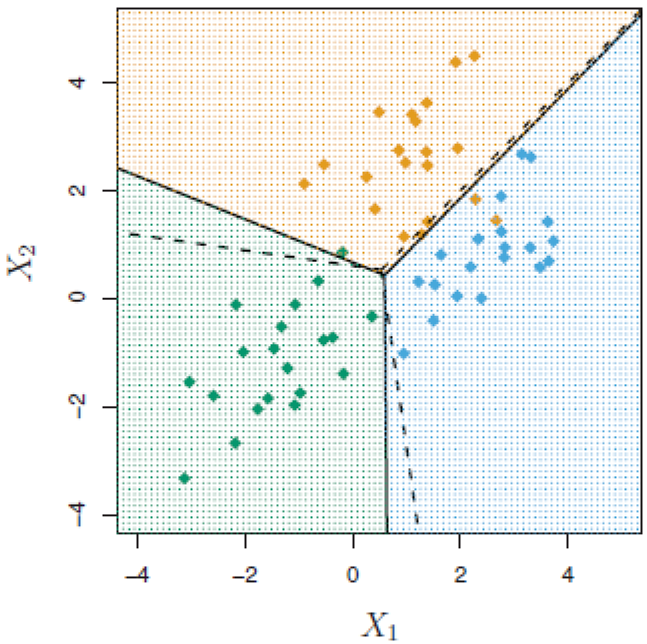
- Discriminant function is given by

$$\delta_k(x) = x^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k$$

- The decision boundary, i.e. $\delta_k(x)=\delta_l(x)$, is linear

$$x^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k = x^T \boldsymbol{\Sigma}^{-1} \mu_l - \frac{1}{2} \mu_l^T \boldsymbol{\Sigma}^{-1} \mu_l$$





Quadratic Discriminant Analysis (QDA)

- LDA assumes the same variance, but QDA allows to have different variances.

$$p_k(x) = \Pr[Y = k | X = x] = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

LDA:

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

$$\hat{\Sigma} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

QDA:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

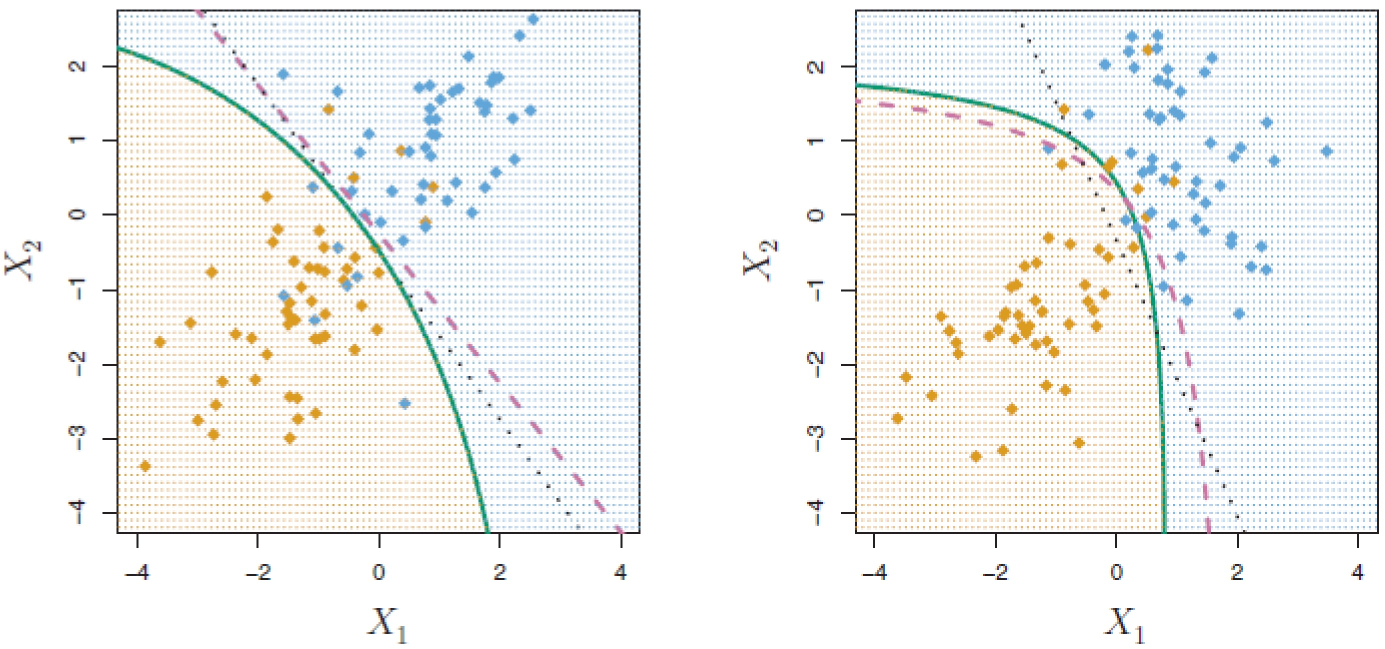
$$\hat{\Sigma}_k = \frac{1}{n-1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

- QDA has more parameters to be estimated, i.e. **more flexible**.

Quadratic Discriminant Analysis (QDA)

- QDA allows a quadratic discriminant function.

$$
\begin{aligned}
\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\boldsymbol{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k \\
&= -\frac{1}{2}x^T\boldsymbol{\Sigma}_k^{-1}x + x^T\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k
\end{aligned}
$$



Logistic Regression vs. LDA

- Consider a simple two-class classification with one predictor.
  - $p_1(x) = \Pr[Y = 1|X = x]$ vs. $p_2(x) = 1 - p_1(x) = \Pr[Y = 2|X = x]$.

- Logistic regression

$$\log\left(\frac{p_1}{1 - p_1}\right) = \beta_0 + \beta_1 x.$$

- LDA

$$\log\left(\frac{p_1(x)}{1 - p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x,$$

- Logistic regression and LDA use the **same linear model**, but the **estimation of parameters is different** as their own assumptions.
  - LDA assumes Gaussian distributions.
  - Logistic regression assumes the logit of the probability is linear.

Assessing Two-Class Classification Performance

- Example: Only 0.1% of babies have down syndrome (Y=1). If we predict all babies are normal (Y=0), then the accuracy is 99.9%. Do you satisfy this prediction?

- Two-class classification is the most general one. Many metrics to measure the performance of two-class classification.

	Truly Positive	Truly Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

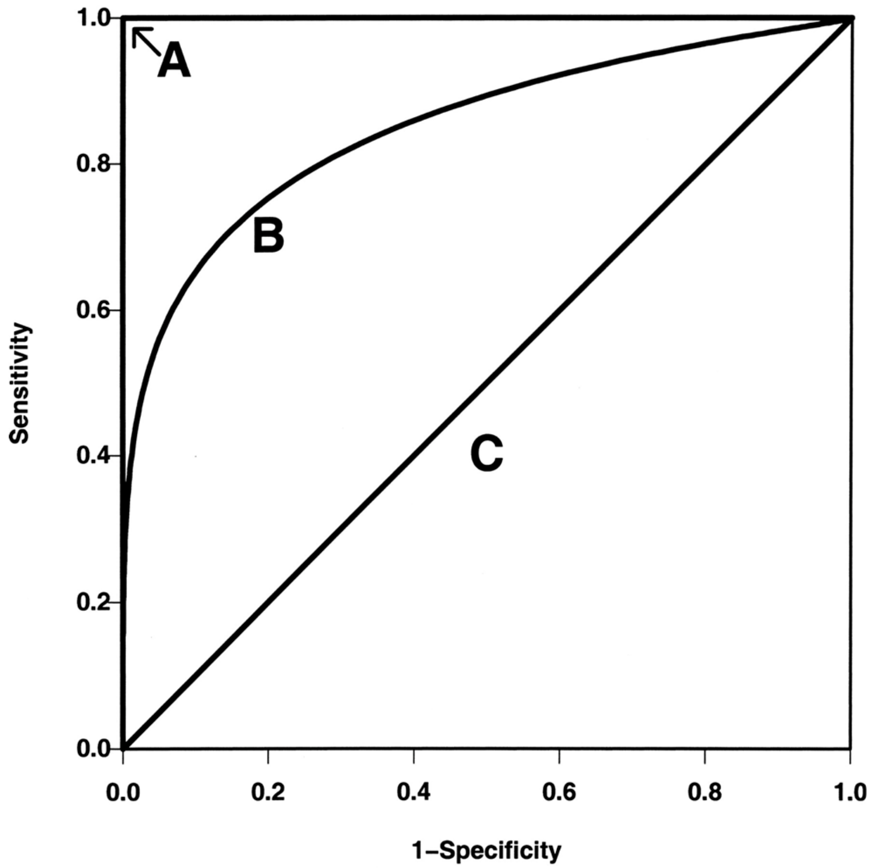
  - **True Positive Rate** or **Sensitivity**: TPR = Sensitivity = TP/(TP+FN).
  - **False Positive Rate** or (1-**Specificity**): FPR = 1-Specificity = FP/(FP+TN).
  - **False Discovery Rate** or (1-**Positive Predictive Value**): FDR=1-PPV=FP/(TP+FP).
  - **Accuracy**: ACC = (TP+TN)/(TP+FP+FN+TN).
  - **Recall** = TPR = Sensitivity = TP/(TP+FN).
  - **Precision** = PPV = 1-FDR = TP/(TP+FP).

- Good performance often means **high TPR** and **low FPR**.
  - Accuracy sometimes does not provide a good metric for the performance.
  - Cancer diagnosis vs. pregnancy test.

Receiver Operating Characteristic (ROC) Curve

- Calling positive if $p_1(x)>0.5$, but sometimes we may want to use more strict threshold, i.e. $p_1(x)>0.8$.
- The performance of classification varies according to the decision threshold.

- **ROC Curve**: plot TPR and FPR by varying the decision threshold.



Area Under Curve (AUC): area under a ROC curve (0~1).

A: perfect classifier (AUC=1)
C: random classifier (AUC=0.5)

Discussion
1. Good predictor?
2. How to determine the decision threshold?

ROC and AUC is the final report of your prediction!!!

Practices

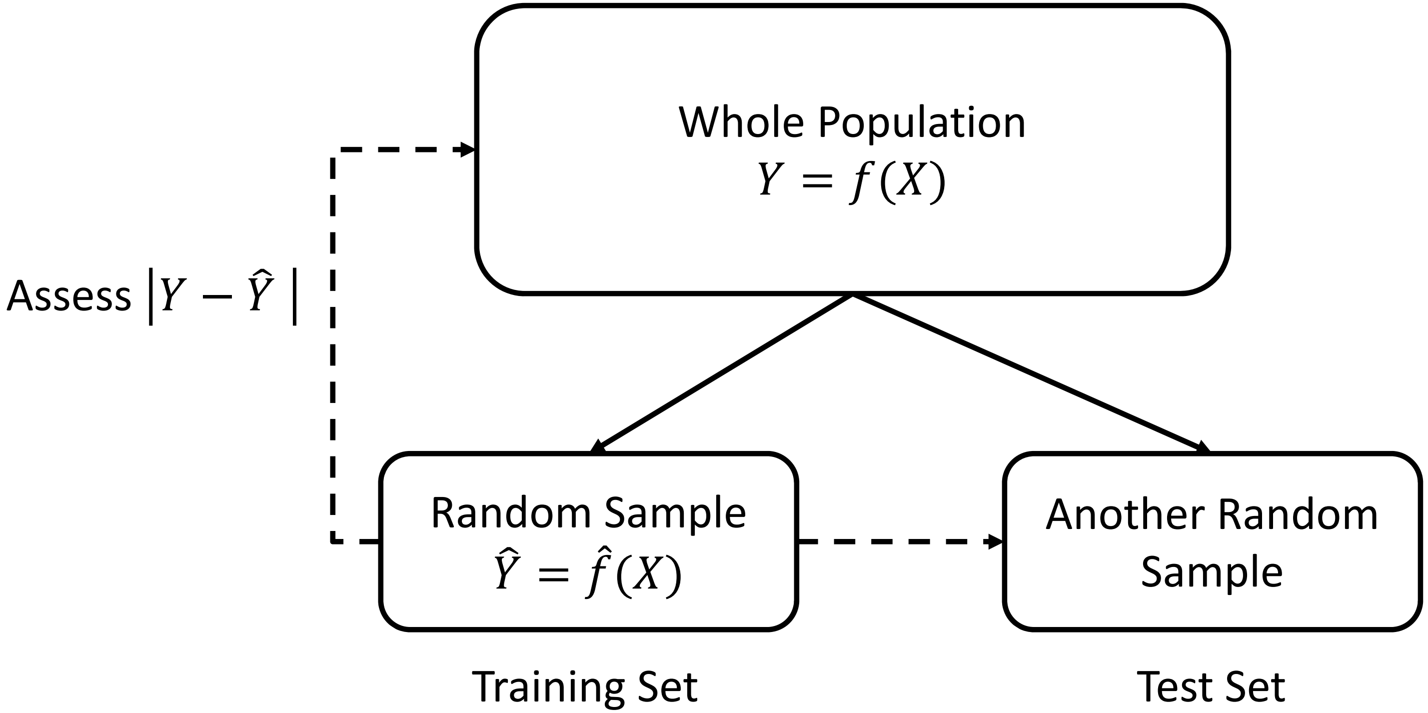
- Traditional classification methods
  - sklearn.linear_model.LogisticRegression
  - sklearn.discriminant_function.LinearDiscriminantAnalysis
  - sklearn.neighbors.KNeighborsClassifier
  - sklearn.metrics.roc_curve

- Practice
  - ‘data02_college.csv’를 읽고, Private을 예측하는 모델을 만드시오. 전체데이터는 train_test_split을 이용하여 임의로 training과 test 셋으로 나누시오. (test_size=0.4, random_state=0)
  - 테스트 셋의 accuracy를 구하고, ROC 커브를 그리시오. AUC는 얼마인가?

Cross-Validation

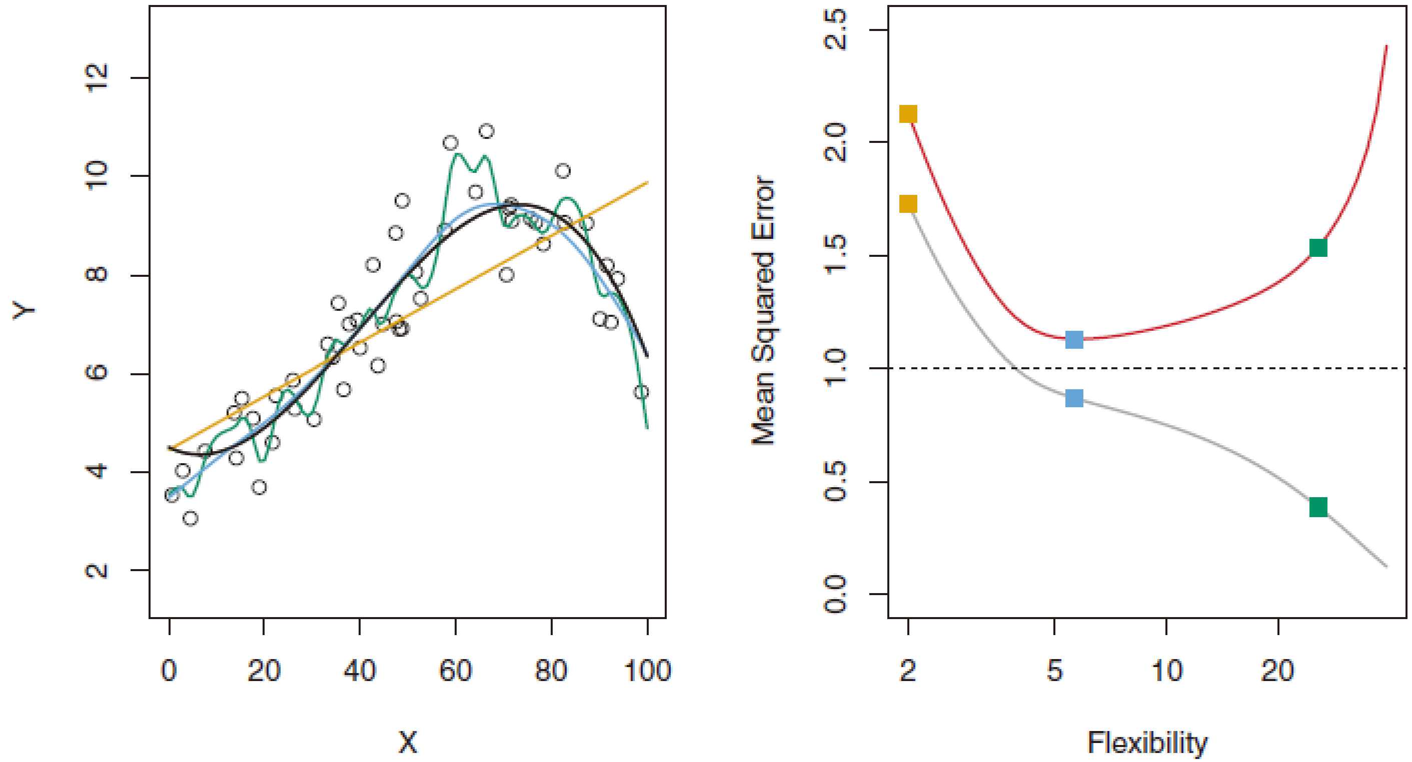
The Goal of Statistical Learning

- Estimating true *f*(X) from random samples.



The Goal of Statistical Learning

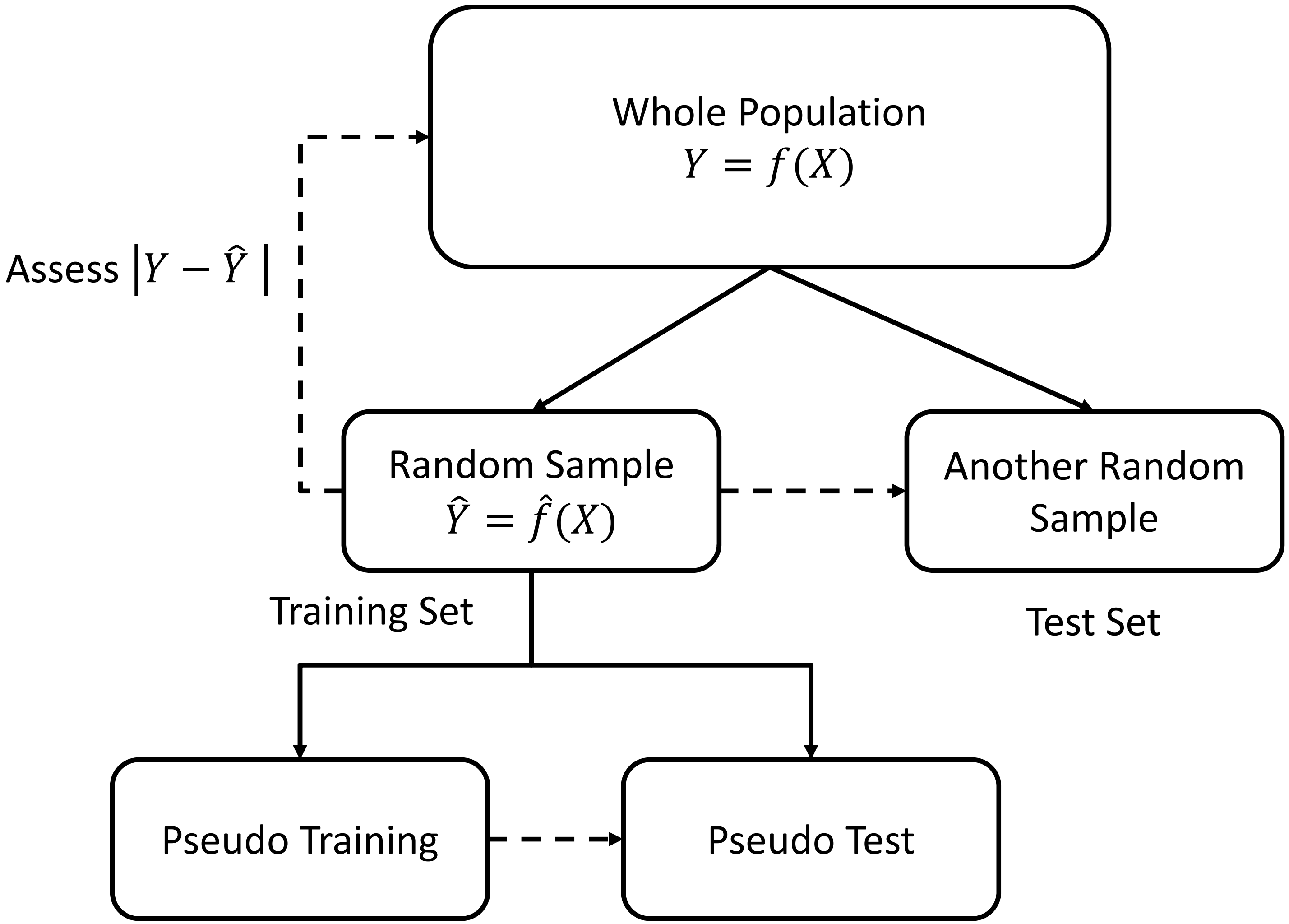
- Training errors and test errors are different.



- We know nothing about the test set when we estimate *f*(*X*).
- **How to estimate the test errors without looking at the test set?**

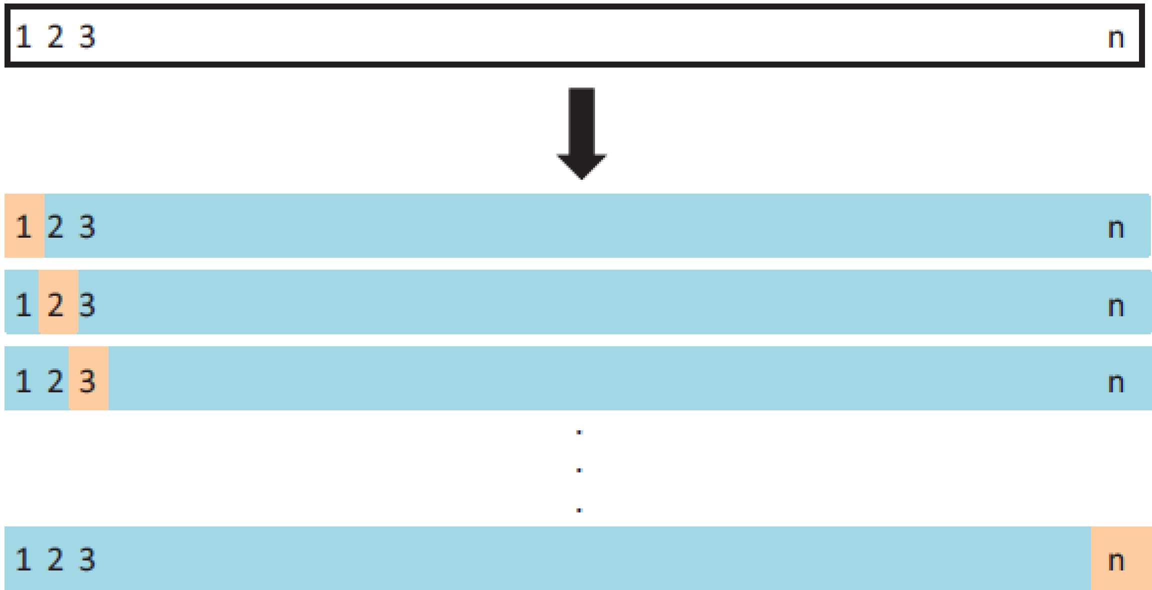
Cross-Validation

- Simulating training-test sets within the real training set.



Leave-One-Out Cross Validation (LOOCV)

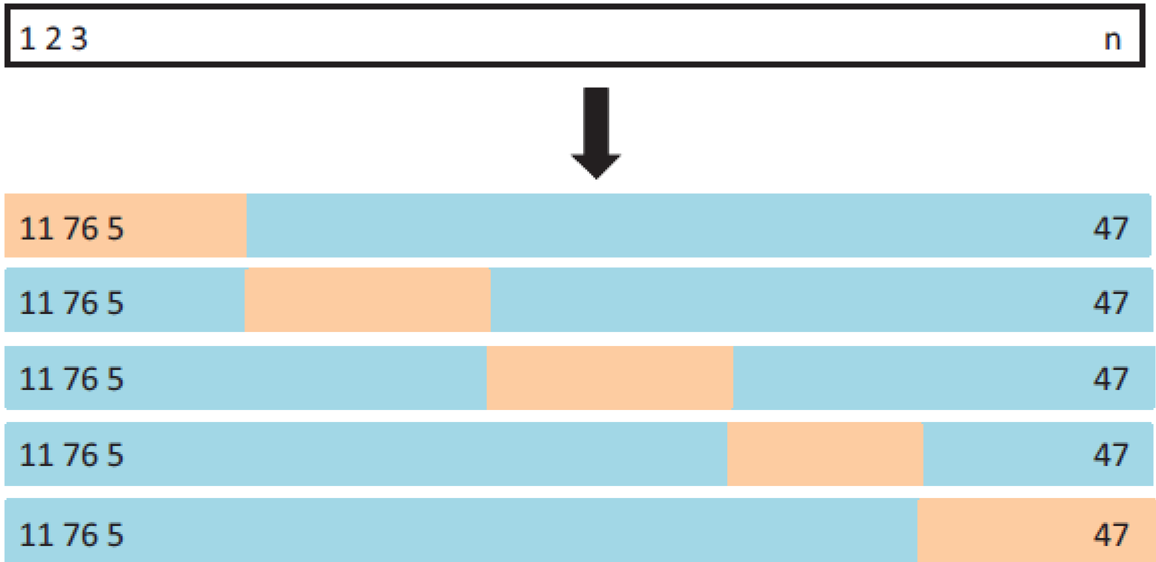
- Consider one sample (x_j,y_j) as a test set, and the rest (x_i,y_i) $i \neq j$ as a training set. Repeat it for all samples.



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

k-Fold Cross Validation

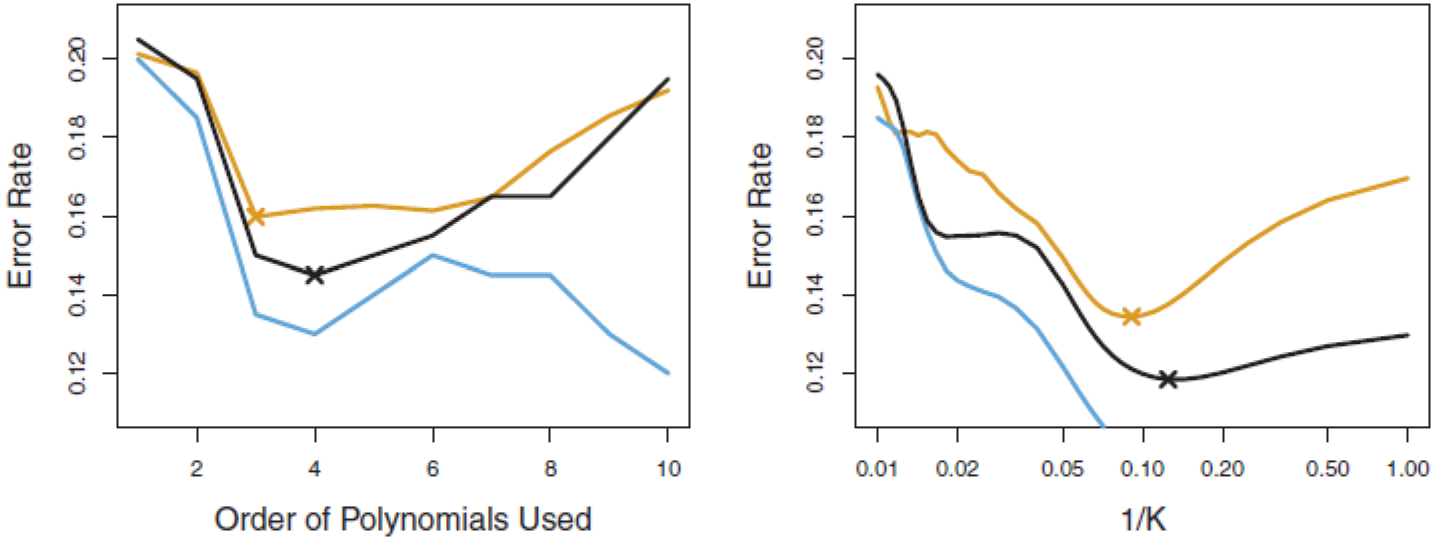
- Randomly divide the whole training data into *k* bins. Consider one bin is a test set and the rest bins are a training set. Repeat it for all *k* bins.
  - LOOCV is *n*-fold CV.



$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i.$$

Examples

- Classification



  – Orange: true test error; Blue: training error; Black: CV error.

- CV error reflects the pattern of the true test errors well
  – **Useful for model assessment.**

LOO vs. *k*-Fold Cross Validation

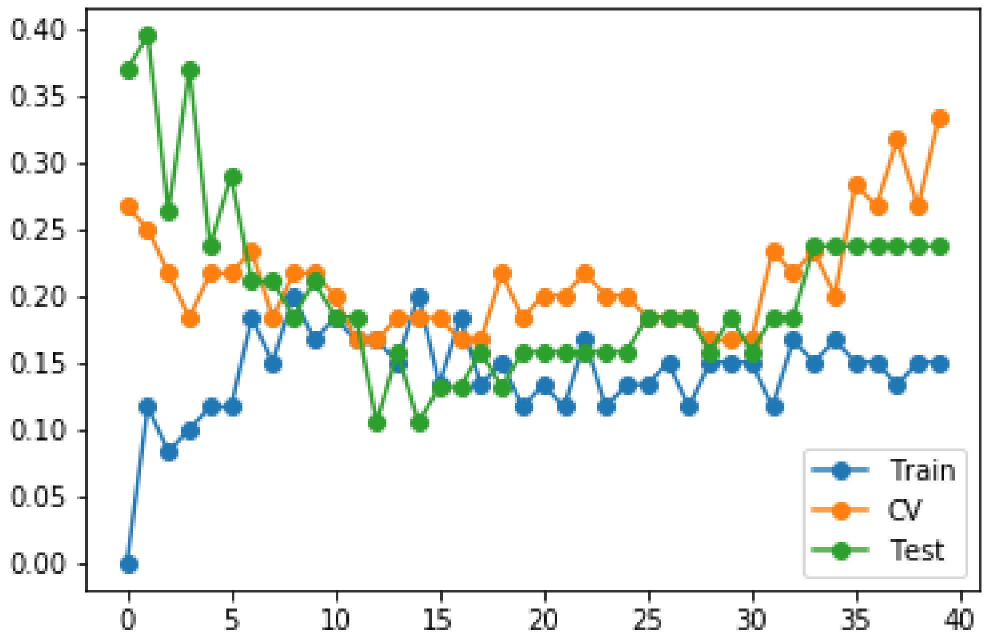
- LOOCV
  - Almost unbiased estimation for the true test errors because it uses n–1 samples for training: <u>less bias</u>.
  - The n fitted models are similar to each other and highly stick to the training data: <u>high variance.</u>
  - Computationally intensive: n model fittings.

- k-Fold CV (extremely k=2)
  - Underestimated the true test errors because it uses n/2 samples: <u>high bias</u>.
  - The k fitted models can be different and less stick to the original training set: <u>low variance.</u>
  - Computationally less intensive: k model fitting.

Practices

- Cross-validation
  - sklearn.model_selection.LeaveOneOut
  - sklearn.model_selection.Kfold
  - sklearn.model_selection.cross_val_score

- Practice
  - Read ‘data07_iris.cvs’ and plot train, cv, and test errors using KNN method by changing K from 1 to 40



Feature Selection

Best Subset Selection

- Selecting *k* best predictors among *p* predictors.
  - "Best" often means the lowest MSE.

- Algorithm

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ 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 exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

  - We will see later about AIC, BIC and adjusted R^2.

Stepwise Selection

- **Forward stepwise selection**: starting from a null model, and adding the best variables one-by-one.
  - In total, 1+*p*(*p*+1)/2 models are fitted.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

- Example

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income` `student`, `limit`	`rating`, `income`, `student`, `limit`

Stepwise Selection

- **Backward stepwise selection**: starting from a full model, and removing the worst variables one-by-one.
  - In total, $1+p(p+1)/2$ models are fitted.

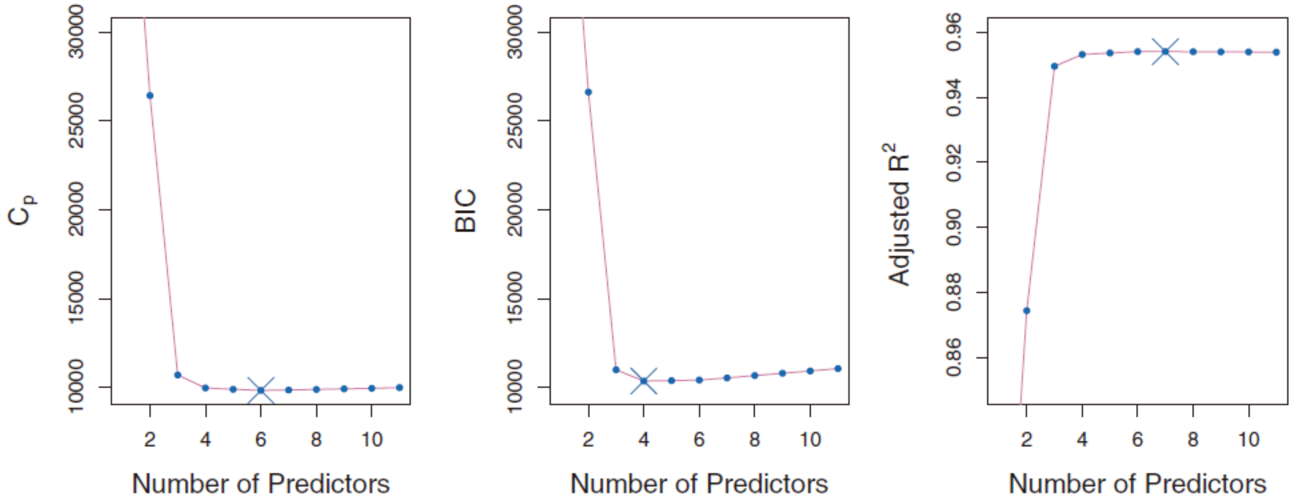
Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

- Forward stepwise selection if $p > n$.
- Backward stepwise selection if $n > p$.

Model Selection Criteria

- More predictors always decrease the training error.
- Adjusting the training error by accounting the number of predictors.



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

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

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

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

Practices

- Practice 1
  - Implement backward feature selection by yourself
  - Apply it to the diabetes data set

- Practice 2
  - Read ‘data02_college.csv’, calculate the acceptance rate from the data, and predict the acceptance rate using feature selection methods
  - What is your score on the test set?

Penalization

Ridge Regression

- A usual least squares fitting finds β's that minimize

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2.$$

- **Ridge regression** finds β's that minimizes

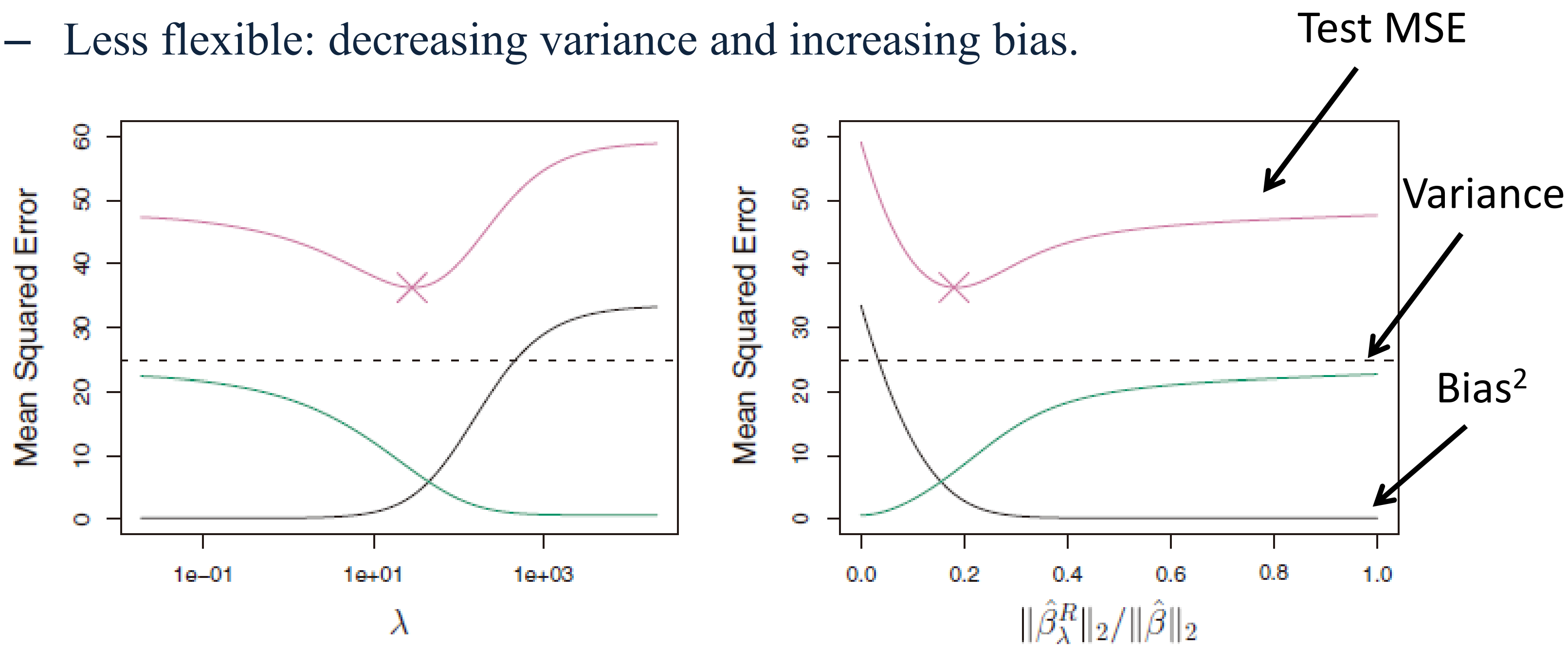
Shrinkage penalty

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

- λ is a tuning parameter that determines the amount of shrinkage.
  – Determined separately, often from cross-validation.

Ridge Regression

- Bias-variance tradeoff
  - Ridge regression regulates the variability of coefficients.
  - Less flexible: decreasing variance and increasing bias.



- Computationally easy
  - It has an analytic solution: $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X} + \boldsymbol{\lambda}^2\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$.
  - It can be solved by one common matrix inversion for any λ.

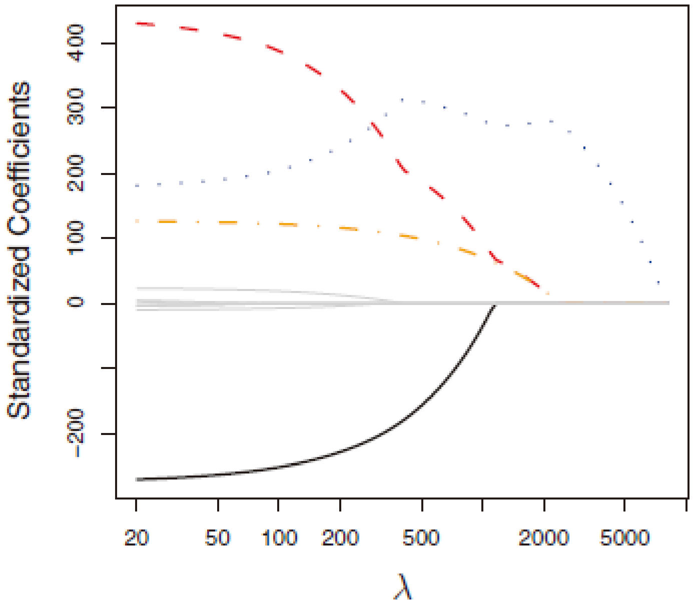
Lasso

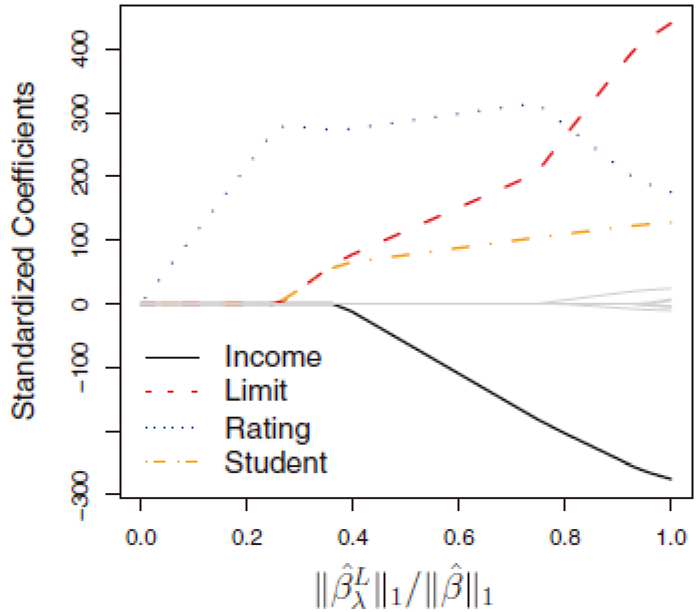
- **Lasso** finds β's that minimizes

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

  – Lasso selects variables because it makes β's zero (ridge regression doesn't).

- Example





Elastic Net

- Ridge

$$\beta^* = \operatorname{argmin}\left(RSS + \lambda \sum \beta_i^2 \right)$$

- Lasso

$$\beta^* = \operatorname{argmin}\left(RSS + \lambda \sum |\beta_i| \right)$$

- Elastic Net

$$\beta^* = \operatorname{argmin}\left(RSS + \lambda_1 \sum |\beta_i| + \lambda_2 \sum \beta_i^2 \right)$$

Practices

- Cross-validation of linear model
  - sklearn.linear_model.Ridge
  - sklearn.linear_model.Lasso

- Pracice 1
  - By applying 5-fold cross-validation for lasso and elastic net, find the best model. What is you test score?

- Practice
  - Read ‘data02_college.csv’, calculate the acceptance rate from the data, and predict the acceptance rate using elastic net.
  - What is your score on the test set?

Principal Component Regression & Partial Least Squares

Dimension Reduction

- **Coordinate transformation**: the original data $X_1, X_2, \ldots, X_p$ can be transformed into a new coordinate system of $Z_1, Z_2, \ldots, Z_p$ using linear combinations.

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

- **Dimension reduction in regression**: fitting y using M Z's ($M<p$) instead of p X's.

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n,$$

  - By carefully choosing the coordinate transformation, we can outperform the ordinary regression.

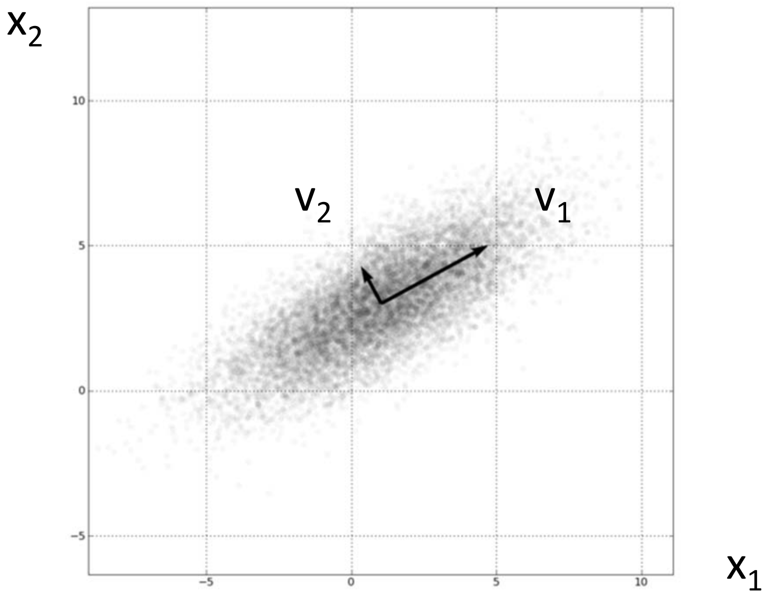
- It is different from subset selection because all p X's are used.

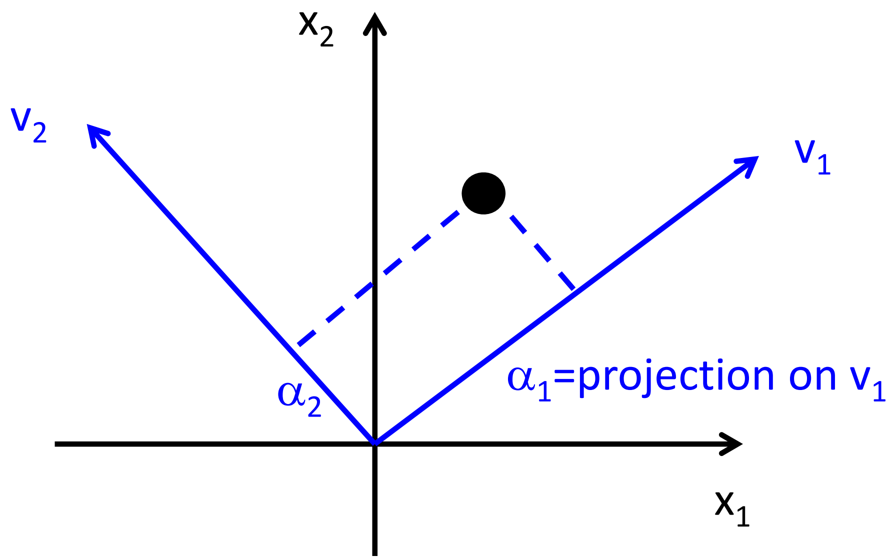
$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij},$$

Principal Component Regression

- **Principal components**
  - Uncorrelated or **orthogonal** variables that explain the data.
  - Transform the coordinates: finding a direction corresponding to **the maximal variance,** or explaining the largest variance.

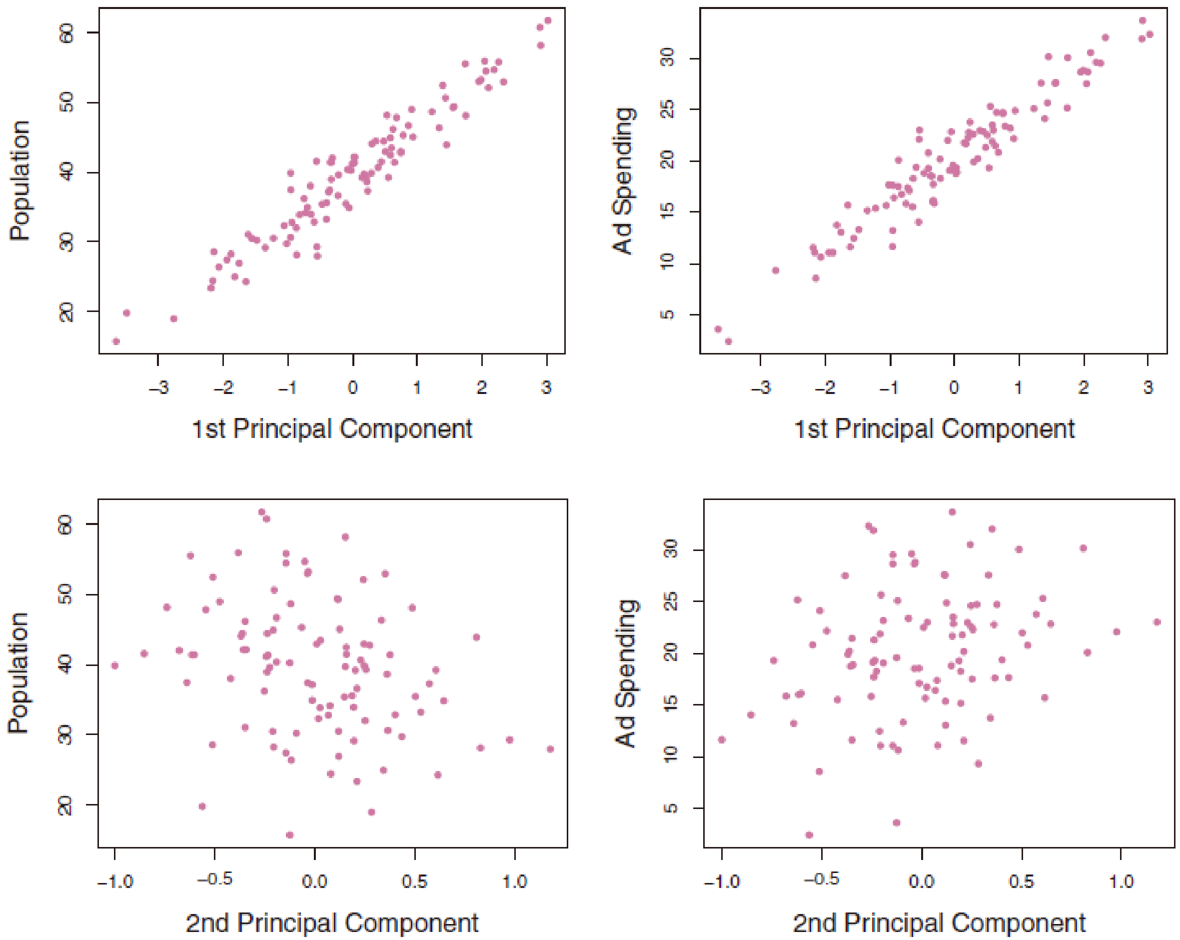
$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha_1 \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} + \alpha_2 \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$$





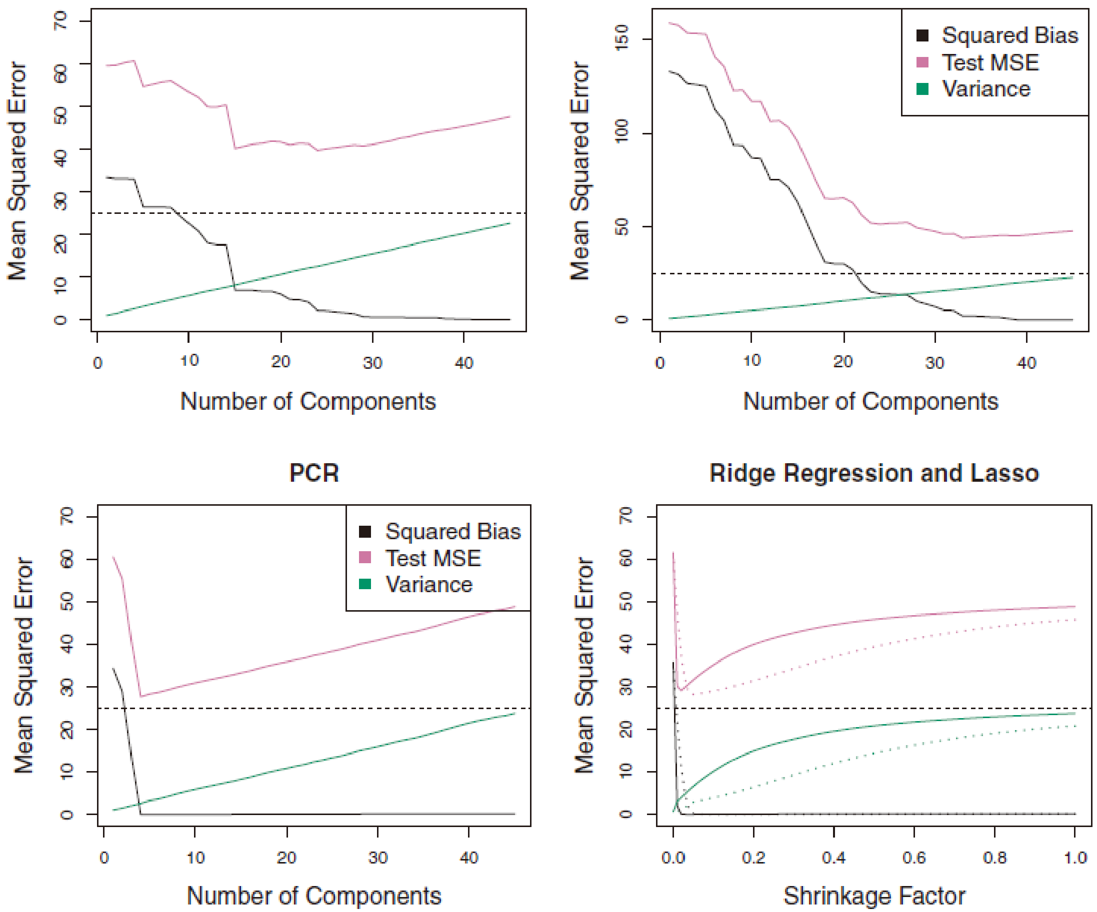
Principal Component Regression

- Example: PCs from pop and ad.



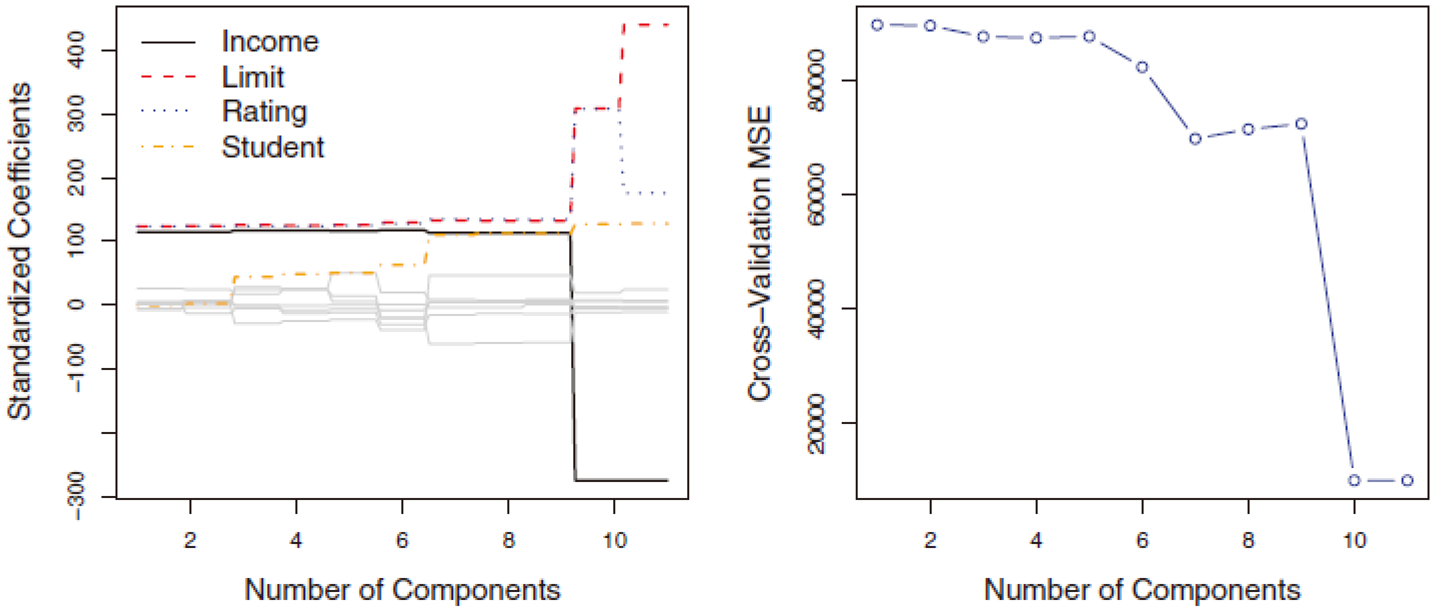
Principal Component Regression

- **Principal component regression** (PCR): regression with the first a few PCs.
  - PCs are derived in a unsupervised way.
  - PCR assumes that the directions in which inputs are the most variable are the direction that are associated with the output.
- Example



Principal Component Regression

- # of PCs are a tuning parameter, which can be selected by cross-validation.

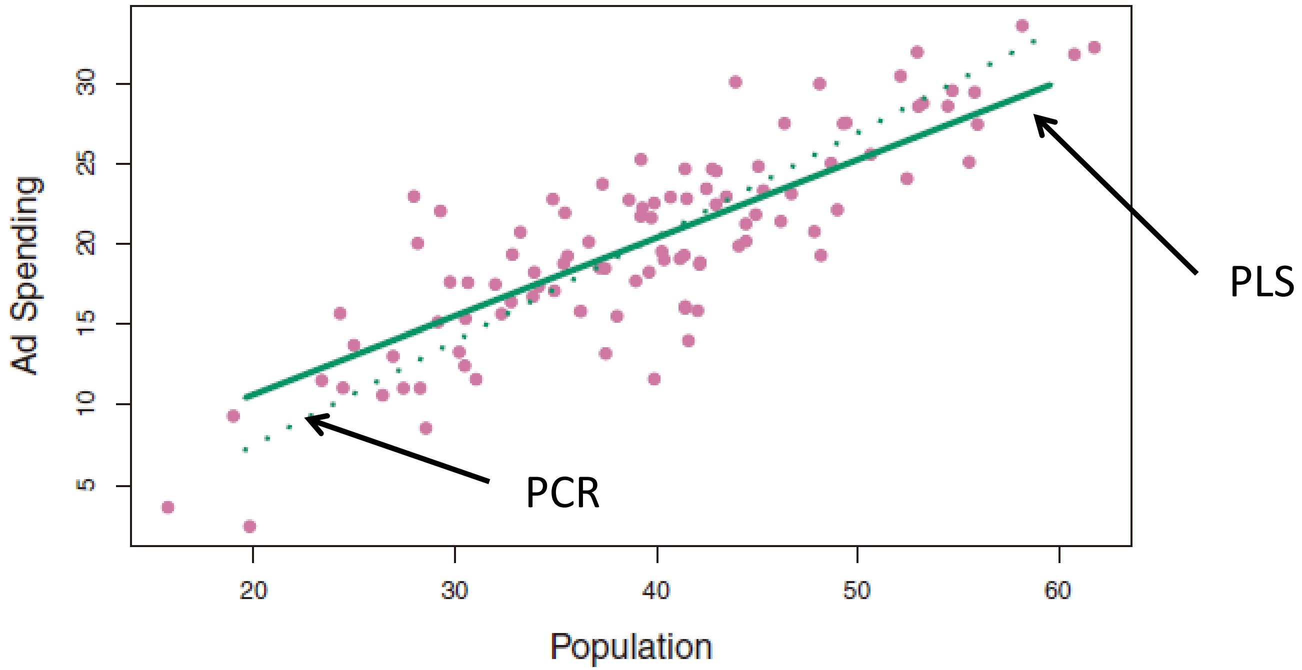


Partial Least Squares (PLS)

- In the coordinate transformation

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

- PCR finds Z's corresponding to the maximum variance of X: *unsupervised*.
- PLS finds Z's corresponding to the maximum variance of both X and Y: *supervised*.



Partial Least Squares (PLS)

- Calculation of PLS directions
  - The first direction (ϕ_{j1}) is calculated as the coefficients of a simple regression between Y and X_j.
  - X's with higher correlation with Y has larger coefficients, or ϕ_{j1}.
  - The first component $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$, higher weights on highly correlated X.
  - Calculating the residual by regressing Y with Z_1.
  - Calculating the second direction by repeating the above steps for the residual.

- Regressing Y with M Z's.
  - M is the number of components, which can determined through cross-validation.

Practices

- Traditional classification methods
  - sklearn.linear_model.LogisticRegression
  - sklearn.discriminant_function.LinearDiscriminantAnalysis
  - sklearn.neighbors.KNeighborsClassifier
  - sklearn.metrics.roc_curve

- Practice
  - diabetes.csv를 읽고 Y를 나머지 변수로 linear regression을 수행하시오. 이때 R2는 얼마인가?
  - 위의 데이터에 대하여 PCR을 수행하시오. 2개의 component를 사용했을 때, R2는 얼마인가?
  - 위의 데이터에 대해여 PLS를 수행하시오. 2개의 component를 사용했을 때 R2는 얼마인가?

Appendix

References

- Probability and Stochastic Processes: A Friendly Introduction to Electrical and Computer Engineers (3rd edition), Yates and Goodman, Wiley
- Probability, Statistics, and Random Processes for Electrical Engineering (3rd edition), Leon-Garcia, Pearson International Edition.
- An Introduction to Statistical Learning with Applications in R, James, Witten, Hastie, Tibshirani, Springer
- Pattern Recognition and Machine Learning, Bishop, Springer

About the Lecturer

- **Junhee Seok, PhD**
  - Assistant Professor, Electrical Engineering, Korea University
  - Director of Mirae Asset AI Fintech Research Center
  - Education
    - BS, Electrical Engineering, KAIST, 2001
    - PhD, Electrical Engineering, Stanford University, 2011
  - Professional Experiences
    - Postdoctoral Fellow, Statistics, Stanford University
    - Assistant Professor, HBMI, Northwestern University
  - Research Area
    - Big data analytics, Machine Learning, AI
    - Biomedicine, Finance, Climate, IoT, Materials, and etc.