Introduction and Overview

Chapters 1 and 2

September 13, 2016

1. Introduction and examples
2. Two illustrative simple predictive methods
3. Prediction accuracy vs. model interpretability
4. Sparisty
5. Assessing accuracy of the estimation/prediction
6. Statistical decision theory
7. Classes of restricted estimators
8. Model selection and bias-variance tradeoff
9. Supervised learning and unsupervised learning

About the Course

- Basic and standard contents: linear regression models and classification
- (Slightly) more advanced topics: nonlinear models, tree methods, svm...
- Emphasize model selection (such as regularization, validation) that are directly related with learning/prediction.

Textbooks and Reference Books

- Textbook: An introduction to Statistical Learning (ISLR)
- Reference: Elements of Statistical Learning (ESL)
- Will stick with ISL and may cite ESL accassionally.
- data from the library of ISLR: type `library(ISLR)`
- Acknowledge the use of the graphics in the textbook/refernce for only the purpose of presentation.

Advice on statistical learning/prediction

- “It’s tough to make predictions, especially about the future.” (quoting Yogi Berra).
- “In God we trust, everything else bring data” (quoting Edwards Deming).
- “All models are wrong, some are useful” (quoting George Box).
- “Occam’s Razor” or “Simplicity is beauty” or “Everything else the same, bring the simplest model”. Vote for parsimonious models or bet on sparsity.

- Trade-off between bias and variance. This is one rule of thumb in model selection. Always keeping in mind of the risk of overfit
- Develop intimate feeling about ... data.

Statistical learning/prediction

- A set of statistical tools used in data analysis with the purpose of understanding data and/or make predictions.
- Usually a model, f, is established, containing parameters.
- The model is estimated through data analysis.
- The estimated model is used for understanding and making predictions.

Examples

- Predict a patient's chance of heart attack, based on, e.g., demographic diet or clinical variables.
- Predict price movement of stocks in the coming six months, based on, for example, the historical price processes or other fundamental variables, such as PE, PB.
- Identify the numbers of handwritten ZIP codes for digitized image or matching finger prints or face recognition.
- Finding disease related genes out of the thousands or millions of genes, with only hundreds of samples.

Example 1. Email spam

- Totally 4601 emails.
- Each email is either a “email” or “spam”.
- Two classes: $y_i = 1$ if “email”; $y_i = -1$, if “spam”; for the i-th emails.
- Other info: $x_i = (x_{i,1}, ..., x_{i,57})$: the frequency of 57 (most commonly occuring) words and punctuations, such as “you”, “george”, ...
- how to determine each email is a “email” or “spam”.

Terminology and notation

- y_i: response/output variable/dependen variable
- x_i: covariates/features/independent variables/input variables.

Example 1. Email Spam

TABLE 1.1. (from ESL) Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between spam and email.

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

Email Spam

- how to determine each email is a “email” or “spam”?
- Simple rule 1: if percentage of george < 0.5 and percent.you > 2, the classify as spam; else as email.
- Simple rule 2: if percentage. of free > 0.3 then classify as spam, else as email.
- Two types of errors. Cannot keep both down at the same time.

Example 2. ZIP code data in R

- Each image is a handwritten one-digit number, reprensented as a 16×16 matrix.
- Each pixel ranges in intensity from 0 to 255.
- $y_i = 0, 1, ..., 9$ (the written single digit number)
- $x_i = (x_{i,1}, ..., x_{i,256})$.
- How to correctly/quickly recognize each image.

ZIP code data in R

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 1

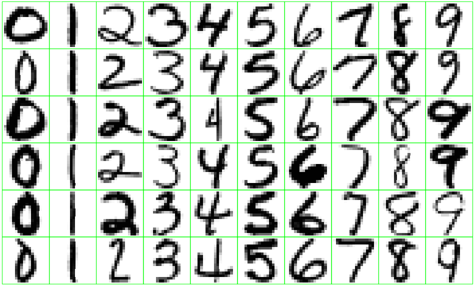
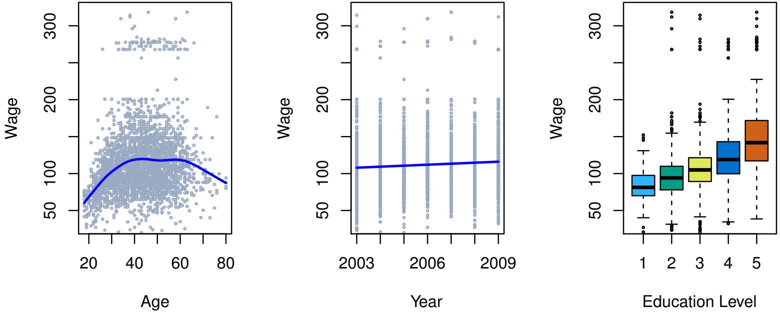


FIGURE 1.2. *Examples of handwritten digits from U.S. postal envelopes.*

Example 3. The Wage data

- Data from a group of males from the Atlantic region of the United States.
- Response variable: Wage.
- input variables: education, age and (calendar) year.
- Wish understand the influence of the input variables on the response.

Wage data


Wage
300
200
100
50
20
40
60
80
Age
2003
2006
2009
Year
1
2
3
4
5
Education Level

Wage data

Left: wage as a function of age. On average, wage increases with age until about 60 years of age, at which point it begins to decline. Center: wage as a function of year. There is a slow but steady increase of approximately $10,000 in the average wage between 2003 and 2009. Right: Boxplots displaying wage as a function of education, with 1 indicating the lowest level (no high school diploma) and 5 the highest level (an advanced graduate degree). On average, wage increases with the level of education.)

Example 4. The stock market data (Smarket)

- S&P 500 stock index from 2001-2005 (`Smarket`)
- y_i : up or down of day i.
- $x_i = (r_{i-1}, ..., r_{i-5})$, where r_j is the return of day j.
- How to predict the up/down of the index of any day given the prvious 5 day's index returns.

Smarket data

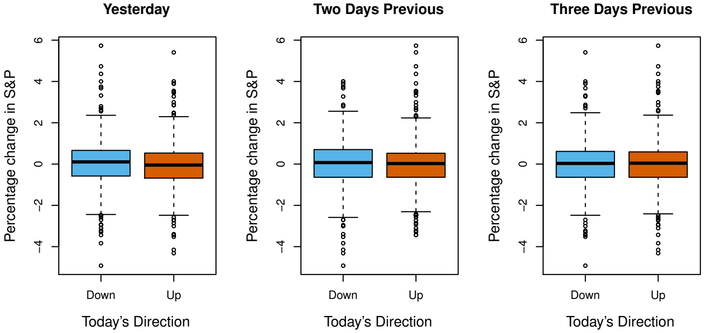
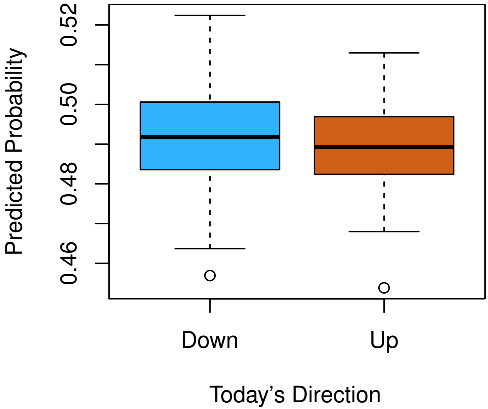


Figure: 1.1. *Left: Boxplots of the previous days percentage change in the S&P index for the days for which the market increased or decreased, obtained from the Smarket data. Center and Right: Same as left panel, but the percentage changes for 2 and 3 days previous are shown.*

Smarket data



Smarket data

ISLR fit a quadratic discriminant analysis model to the subset of the Smarket data corresponding to the 20012004 time period, and predicted the probability of a stock market decrease using the 2005 data. On average, the predicted probability of decrease is higher for the days in which the market does decrease. Based on these results, they are able to correctly predict the direction of movement in the market 60% of the time.

Example 5. The gene expression data

- No response variable (unsurpervised learning)
- 6830 gene expression measurements on each of 64 cancer cell lines (samples).
- Explore the characteristics such as groups, clusters, etc.

Gene expression

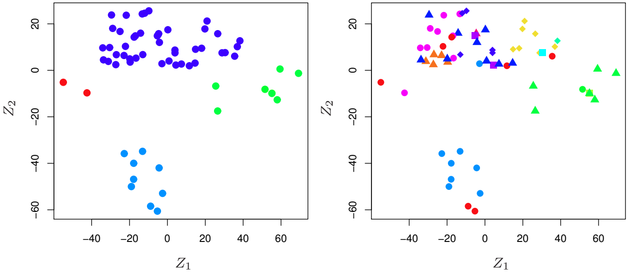


Figure: Representation of the NCI60 gene expression data set in a two-dimensional space, Z1 and Z2 (principal components). Each point corresponds to one of the 64 cell lines. There appear to be four groups of cell lines, which we have represented using different colors. Right: Same as left panel except that we have represented each of the 14 different types of cancer using a different colored symbol. Cell lines corresponding to the same cancer type tend to be nearby in the two-dimensional space.

Data presentation: multiple variables

$$\begin{aligned}\mathbf{X} &= \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ldots & \vdots \\ x_{31} & x_{32} & \cdots & x_{np} \end{pmatrix} \\ &= \left(\mathbf{x}_1 \vdots \mathbf{x}_2 \vdots \cdots \vdots \mathbf{x}_p \right)\end{aligned}$$

x_{ij} is the i-th observation of the j-th variable.
$\mathbf{x}_j$ is n-vector representing all observations of the j-th variable. We shall also use $x_i = (x_{i1}, ..., x_{ip})$ to denote the i-th observations.

Notations

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_n \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}, \quad \mathbf{x}_j = \begin{pmatrix} 1 \\ \vdots \\ x_{nj} \end{pmatrix},$$

sample size: n
number of covariates/inputs/features: p

Least squares vs KNN

- Data: (y_i, x_i), $i =, ..., n$, where $y_i = 0$ or 1 representing two categories.
- Least squares method: huge assumption, stable estimate, but may be inaccurate.
- Nearest neighbor method: minimal assumtoin, unstable estiamte, possibly accurate.

Least squares vs KNN

- Least squares method: Try to minimize $\mathrm{RSS}(\beta) = \sum_{i=1}^{n}(y_i - x_i^T\beta)^2$. On given x, predict $\hat{y} = \hat{\beta}^T x$, where

$$\hat{b}eta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

- Nearest neighbor method:

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

where $N_k(x) = \{x_i : \|x_i - x\|$ is among the smallest K data points $\}$, refers to the so called k-nearest neighbor.

- A simple rule: predict class 0 if $\hat{y} < 0.5$, and predict class 1 if $\hat{y} \geq 0.5$.

- The least squares is effective under some “huge” assumption about the distribution of x.
  It is less flexible, larger bias and smaller variance.
- The KNN (k-nearest neighbor), particularly for small K, implies more flexibility, smaller bias and larger variance.
  Understanding: each prediction relyies on effectively fewer number of observations for small K. Extreme case: $K = 1$ relies on just one observation.
- A proper way to caliberate the “effective” number of observations used is by the degree of freedom.

Bayesian Classifier

Suppose we know exactly the conditional probability of $P(y=1|x)$ and $P(y=0|x)$ for any given x, and classify into

$$\begin{cases} \text{class } 1 & \text{if } P(y=1|x) > 0.5 \\ \text{class } 0 & \text{if } P(y=1|x) \leq 0.5 \end{cases}$$

This is called “Bayesian classifier”.

Bayesian Classifier

- The curve
  $$g(x) = \{x : P(y = 1|x) = 0.5\}$$
  is called “Bayesian decision boundary”.
- The Bayes error rate is
  $$\begin{aligned} & P(\text{ mistaken classification}) \\ = & P(y = 1,\ p(x) \leq 0.5)) + P(y = 0,\ p(x) > 0.5) \\ = & 1 - E(max_{j=0,1} P(y = j|x)) \end{aligned}$$
  where $p(x) = P(y = 1|x)$. This is the probability of making errorneous classfication when apply the Bayes classfier.
- This can be extended to J classes, for any $J \geq 2$.

General models/methods

- linear models, generalized linear models.
- KNN
- Kernel methods
- local polynomial regression
- regression and smoothing splines
- Tree based methods
- SVM
- projection persuit and nearal networks.
- A proper way to caliberate the “effective” number of observations used is by the
  textcolor[rgb]1, .5,0 degree of freedom.

- Assume $E(y|x) = f(x)$.
- f is contrained $\Longrightarrow$ less flexibility, more restrictive and thus a small model.
- Usually, small model $\Longrightarrow$ more interpretability.
- For example: Linear model: $f(x) = x^T\beta$ versus additive model: $f(x) = \sum_{j=1}^{p} g_j(x_j)$
- For example: Linear model: $f(x) = x^T\beta$ versus deep neural nets: a complex parametric model.
- Small model: not necessarily less predictability.
- Small model: small variance but large bias.
- Model error: variance + bias (square)

Sparse modeling

- Sparse modeling refers to finding a “small” (sparse) model, with smaller number of parameters, out of an originally large model, without losing much of predictability.
- Example: For linear model $y_i = x_i^T \beta + \epsilon_i$, $i = 1, ..., n$ and x_i are of p dimension.
- Suppose p is large relative to n. For example, $p = \sqrt{n}$ or $p = \exp(n)$. (Recall the gene expresion example)

Sparse modeling

- Lasso is one of the regression methods, which minimize

$$\sum_{i=1}^{n}(y_i - \beta^T x_i)^2 + \lambda\|\beta\|_1,$$

  where λ is a tuning paramter and $\|\cdot\|_1$ is the l_1 norm.
- The Lasso estimator can be sparse in the sense that many of the components of β is 0.
- It implies that those corresponding variables are “not important” in predicting the response y.
- Bet on sparsity principle.

No free-lunch

- No single method dominate all others on all data sets.
- Strength of a method depends on whether its underlying assumptions are satisfied.
- Training error is one of the measurement of the quality of fit.
- Training MSE (mean squared error)

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

for the training data $(x_i, y_i), i = 1, ..., n$.

- Setting $\hat{f}(x_i) = y_i$ leading to the training MSE $= 0$.
- It shows training MSE is not the best measurement of the model accuracy.
- The ultimate measurement is the test MSE.

$$\text{Test MSE} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}^{n}(y_j - \hat{f}(x_j))^2$$

  where $\mathcal{J}$ refers to the test set of data, and $\hat{f}$ is obtained from the training data.
- Methods, such as cross validation, AIC, BIC, are often employed to compute or estimate the size of the test error

- With a loss function $L(\cdot,\cdot)$ to measure the "loss" of the error between the response y and the predictor $f(x)$.
- The squared loss is a typical loss function:

$$L(y, f(x)) = (y - f(x))^2.$$

- The expected prediction error

$$\text{EPE}(f) = E(L(y, f(x))^2)) = \int L(y, f(x))P(dx, dy)$$

- For squared loss,

$$E(y|x) = \text{argmin}_f \ \text{EPE}(f)$$

Remark

- The loss function can be more general.
- The L_1 loss: $L(y, f(x)) = |y - f(x)|$
- Then, the median of the distribution of y given x is the minimizer of the EPE.

1. Roughness penalty

- Consider the RSS (redisdual sum of square) criterion:

$$\text{RSS}(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2.$$

  Minimizing over all f leads to $f(x_i) = y_i$, and no generalization to $f(x)$ for $x \neq x_i$.

- Thus we add a "complexity" constraint to the model: restricting $f(\cdot)$ to be smooth of certain order.
- Penalizing RSS:

$$\text{PRSS}(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int \ddot{f}^2(x) dx$$

- This is called roughness penalty

2. Local regression

- At each x,

$$\mathrm{RSS}(f;x) = \sum_{i=1}^{n} k_\lambda(x, x_i)(y_i - f(x_i))^2$$

  and

$$\hat{f}(x) = \mathrm{argmin}_f \mathrm{RSS}(f, x)$$

- Here $k_\lambda(x, x^*)$ is a kernel function, such as

Typical kernel functions

- Gaussian radial kernel:

$$k_\lambda(x, x^*) = \frac{1}{\lambda} \exp\left(\|x - x^*\|^2/\lambda\right)$$

- Epanechnikov kernel:

$$k_\lambda(x, x^*) = 1 - \frac{3}{4}\|x - x^*\|^2/\lambda^2$$

- KNN: (data dependent kernel)

$$k_\lambda(x, x^*) = I(\|x - x^*\|^2 \leq d_k))$$

where d_k is the distance from the k-th nearest data points to x. Kernel functions are generally nonnegative and is decreasing when the distance between x and x^* increases.

3. Basis functions and dictionary methods.

- Assume

$$f(x) = \sum_{i=1}^{M} \theta_i h_i(x)$$

where h_i are known functions, called basis functions.

- Examples are: a). polynomial splines:

$$h_1 = 1; \quad h_2 = x, ..., h_{m+2} = (x - t_m)_+ ...$$

where $t_1, ..., t_{m-2}$ are the so called knots. Here $m \leq M - 2$.

- b). kernels:

$$h_j(x) = k_{\lambda_j}(\mu_j, x)$$

  for example, $k_\lambda(\mu, x) = \exp(-\|x - \mu\|^2/(2\lambda))$.

- c). single layer feed-forward neural net:

$$h_j(x) = \sigma(\alpha_j^T x + b_j)$$

  where $\sigma(x) = 1/(1 + e^{-x})$ is the activation function.

- In summary, the collection of functions $h_1, ..., h_m$ provide a "dictionary", thus called dictionary method.

- Generally, as the model becomes more complex, the estimation/prediction accuracy tends to have lower bias but larger variance.
- Model complexity characterized by smoothing or complexity parameters.
- For KNN: smaller K, larger model, less smoothness, smaller RSS (training), less bias, larger variance.
- For linear model: more inputs, larger model, smaller RSS (training), less bias, larger variance.

More calculation of test error for KNN

Consider $y_i = f(x_i) + \epsilon_i$, $i = 1, ..., n$ where ϵ_i are mean 0 with variance σ^2. With KNN denoted as $\hat{f}_k(x)$, then

$$\begin{aligned} \text{EPE}(x) &= E((Y - \hat{f}_k(x))^2 | X = x) \\ &= E((\epsilon + f(x) - \hat{f}_k(x))^2 | X = x) \\ &= E\{((\epsilon + f(x) - E(\hat{f}_k(x)) + E(\hat{f}_k(x)) - \hat{f}_k(x))^2 | X = x\} \\ &= \sigma^2 + (f(x) - E(\hat{f}_k(x)))^2 + \text{var}(\hat{f}_k(x)) \\ &= \sigma^2 + \text{Bias}^2 + \text{var} \end{aligned}$$

which is

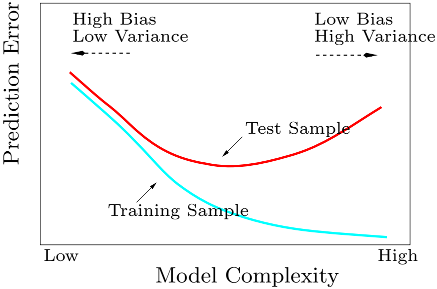
$$\sigma^2 + \left[\frac{1}{k}\sum_{l=1}^{k} f(x_{(l)}) - f(x)\right]^2 + \sigma^2/k$$

where $x_{(l)}$ refers to the l-th nearest data point to x.

If k increases, model is smaller, and smoother, variance σ^2/k decreases, and the bias term increases.

Bias-variance tradeoff

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 2



- Supervised learning: for simplicity

$$y_i = f(x_i) + \epsilon_i, \qquad i = 1, \ldots, n$$

- Try to learn f with the supervision of the "teacher" y
- f is adjusted so that error $y_i - f(x_i)$ becomes small.
- Eventually, we hope the learned f can be applied to all data, particularly data not in the training set.

- Unsupervised: We only have $x_i, i = 1, .., n$.
- No supervision from teachers.
- Seek to understand the relations among the variables and data points.
- Typical methods: PCA, ICA, cluster analysis.

Homework

- ISLR Chapter 2
- 1; 2; 7; 8.

End of Chapters 1-2