Chapter 2 Solution

Haoran Jiang

July 31, 2017

Ex. 2.1

Suppose each of K-classes has an associated target t_k, which is a vector of all zeros, except a one in the kth position. Show that classifying to the largest element of $\hat{y}$ amounts to choosing the closest target, $\min_k ||t_k - \hat{y}||$, if the elements of $\hat{y}$ sum to one.

Sol (Weatherwax):

$$
\begin{aligned}
\arg\min_k ||t_k - \hat{y}|| &= \arg\min_k \sum_{i=1}^{K} (t_{k(i)} - \hat{y}_i)^2 \\
&= \arg\min_k \sum_{i=1}^{K} t_{k(i)}^2 - 2t_{k(i)}\hat{y}_i + \hat{y}_i^2 \\
&= \arg\min_k \sum_{i=1}^{K} -2t_{k(i)}\hat{y}_i \\
&= \arg\min_k -2\hat{y}_k \\
&= \arg\max_k \hat{y}_k
\end{aligned}
$$

So, for any K-dimensional vector $\hat{y}$, the k for which $\hat{y}_k$ is largest coincides with the k for which t_k is nearest to $\hat{y}$.

Ex. 2.2

Show how to compute the Bayes decision boundary for the simulation example in Figure 2.5.

Sol:
The Bayes Boundary is defined as the equation of equality between the two probabilities:

$$\Pr(g = \text{Blue}|X) = \Pr(g = \text{Orange}|X)$$

Ex. 2.3

Derive equation (2.24).

Sol:
Let r_i denote $||x_i||$. Since the volume of the p dimensional ball of radius r is proportional to r^p, the Probability Density Function (PDF) of r_i is

$$f_{r_i}(r) = \begin{cases} \frac{1}{p} r^{p-1} & 0 \le r \le 1 \\ 0 & \text{o.w} \end{cases}$$

Let $d$denote the $\min(r_1, r_2, \cdots, r_N)$. By order statistic formula, we can get the PDF of d,

$$f_d(x) = \begin{cases} \frac{N}{p} x^{p-1}(1 - x^p)^{N-1} & 0 \le x \le 1 \\ 0 & \text{o.w} \end{cases}$$

The median distance from the origin to the closest data point solve the equation

$$\int_0^d \frac{N}{p} x^{p-1}(1 - x^p)^{N-1} = \frac{1}{2}$$

The left side of the equation is

$$1 - (1 - d^p)^N$$

So we get the final solution:

$$d(p, N) = (1 - \frac{1}{2}^{1/N})^{1/p}$$

Ex. 2.4

The edge effect problem discussed on page 23 is not peculiar to uniform sampling from bounded domains. Consider inputs drawn from spherical multinormal distribution $X \sim N(0, \mathbf{I}_p)$. The squared distance from any sample point to the origin has a χ_p^2 distribution with mean p. Consider a prediction point x_0 drawn from this distribution, and let $a = \frac{x_0}{||x_0||}$ be associated unit vector. Let $z_i = a^T x_i$ be the projection of each of the training points on this direction.

Show that the z_i are distributed $N(0,1)$ with expected squared distance from the origin 1, while the target point has expected squared distance p from the origin.
Hence for $p = 10$, a randomly drawn test point is about 3.1 standard deviations from the origin, while all the training points are on average one standard deviation along direction a. So most prediction points see themselves as lying on the edge of the training set.

Sol:
Since $x_i \sim N(0, \mathbf{I}_p)$, $z_i = a^T x_i$ follows the Normal distribution.

$$\begin{aligned} E(z_i) &= E(a^T x_i) = a^T E(x_i) = a^T 0 = 0 \\ Var(z_i) &= Var(a^T x_i) = a^T Var(x_i) a = a^T a = 1 \end{aligned}$$

Ex. 2.5

(a) Derive equation (2.27). The last line makes use of (3.8) through a conditioning argument.

Sol:

$$\begin{aligned} \mathrm{EPE}(x_0) &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(y_0 - \hat{y}_0)^2 \\ &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(x_0^T\beta + \epsilon - x_0^T\hat{\beta})^2 \\ &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(x_0^T\beta + \epsilon - x_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\boldsymbol{X}\beta + \overrightarrow{\epsilon}))^2 \\ &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(\epsilon - x_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\overrightarrow{\epsilon})^2 \\ &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(\epsilon^2 - 2\epsilon x_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\overrightarrow{\epsilon} + x_0^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\overrightarrow{\epsilon}\overrightarrow{\epsilon}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}x_0) \\ &= \mathrm{E}_{y_0|x_0}(\epsilon^2 + \sigma^2 x_0^T\mathrm{E}_{\mathcal{T}}(\boldsymbol{X}^T\boldsymbol{X})^{-1}x_0) \\ &= \sigma^2 + \sigma^2 x_0^T\mathrm{E}_{\mathcal{T}}(\boldsymbol{X}^T\boldsymbol{X})^{-1}x_0 \end{aligned}$$

(b) Derive equation (2.28), making use of the cyclic property of the trace operator [trace(AB) = trace(BA)], and its linearity (which allows us to interchange the order of trace and expectation).

Sol:
If N is large and $\mathcal{T}$ were selected at random, and assuming $\mathrm{E}(X) = 0$, then $\boldsymbol{X}^T\boldsymbol{X} \to N\mathrm{Cov}(X)$ and

$$
\begin{aligned}
\mathrm{E}_{x_0}\mathrm{EPE}(x_0) &\sim \sigma^2 + \sigma^2 \mathrm{E}_{x_0} x_0^T (N\mathrm{Cov}(X))^{-1} x_0 \\
&= \sigma^2 + \frac{\sigma^2}{N} \mathrm{Tr}(\mathrm{E}_{x_0} x_0^T \mathrm{Cov}(X)^{-1} x_0) \\
&= \sigma^2 + \frac{\sigma^2}{N} \mathrm{E}_{x_0}(\mathrm{Tr}(x_0^T \mathrm{Cov}(X)^{-1} x_0)) \\
&= \sigma^2 + \frac{\sigma^2}{N} \mathrm{E}_{x_0}(\mathrm{Tr}(x_0 x_0^T \mathrm{Cov}(X)^{-1})) \\
&= \sigma^2 + \frac{\sigma^2}{N} \mathrm{Tr}(\mathrm{Cov}(x_0)\mathrm{Cov}(X)^{-1}) \\
&= \sigma^2 + \frac{p}{N}\sigma^2
\end{aligned}
$$

Ex. 2.6

Consider a regression problem with inputs x_i and outputs y_i, and a parameterized model $f_\theta(x)$ to be fit by least squares. Show that if there are observations with tied or identical values of x, then the fit can be obtained from a reduced weighted least squares problem.

Sol:

Ex. 2.7

Suppose we have a sample of N pairs x_i , y_i drawn i.i.d. from the distribution characterized as follows:

$$
\begin{aligned}
&x_i \sim h(x), \text{ the design density} \\
&y_i = f(x_i) + \epsilon_i, \ f \text{ is the regression function} \\
&\epsilon_i \sim (0,\ \sigma^2) \text{ (mean zero, variance } \sigma^2)
\end{aligned}
$$

We construct an estimator for f linear in the y_i,

$$
\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0; \chi) y_i,
$$

where the weights $l_i(x_0; \chi)$ do not depend on the y_i, but do depend on the entire training sequence of x_i, denoted here by χ.

(a) Show that linear regression and k-nearest-neighbor regression are members of this class of estimators. Describe explicitly the weights $l_i(x_0; \chi)$ in each of these cases.

Sol:
For the linear regression:

$$\begin{aligned}
\hat{f}(x_0) &= x_0^T \hat{\beta} \\
&= x_0^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \\
&= \sum_{i=1}^{N} x_i^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} x_0 y_i \\
&= \sum_{i=1}^{N} l_i(x_0; \chi) y_i,
\end{aligned}$$

where $l_i(x_0; \chi) = x_i^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} x_0$.
For the k - nearest-neighbor regression:

$$\begin{aligned}
\hat{f}(x_0) &= \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i \\
&= \sum_{i=1}^{N} l_i(x_0; \chi) y_i,
\end{aligned}$$

where $l_i(x_0; \chi) = \frac{1}{k} I(x_i \in N_k(x_0))$.

(b) Decompose the conditional mean-squared error

$$\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2$$

into a squared bias and a variance component.

Sol:

$$\begin{aligned}
\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2 &= \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0) + \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0) - \hat{f}(x_0))^2 \\
&= \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(f(x_0) - \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2 + \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0) - \hat{f}(x_0))^2 \\
&\quad + 2\mathrm{E}_{\mathcal{Y}|\mathcal{X}}((f(x_0) - \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))(\mathrm{E}_{\mathcal{Y}|\mathcal{X}}\hat{f}(x_0) - \hat{f}(x_0))) \\
&= (f(x_0) - \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2 + \mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0)) - \hat{f}(x_0))^2 \\
&= \mathrm{Bias}^2_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0)) + \mathrm{Var}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))
\end{aligned}$$

It is not hard to prove the cross product part (the red part) is zero.

(c) Decompose the (unconditional) mean-squared error

$$\mathrm{E}_{\mathcal{Y},\mathcal{X}}(f(x_0) - \hat{f}(x_0))^2$$

into a squared bias and a variance component.

Sol:

$$
\begin{aligned}
\mathrm{E}_{\mathcal{Y},\mathcal{X}}(f(x_0)-\hat{f}(x_0))^2 &= \mathrm{E}_{\mathcal{Y},\mathcal{X}}(f(x_0)-\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))+\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))-\hat{f}(x_0))^2 \\
&= \mathrm{E}_{\mathcal{Y},\mathcal{X}}(f(x_0)-\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))^2+\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\mathrm{E}_{\mathcal{Y}|,X}(\hat{f}(x_0)-\hat{f}(x_0))^2 \\
&\quad + 2\mathrm{E}_{\mathcal{Y},\mathcal{X}}((f(x_0)-\mathrm{E}_{\mathcal{Y}|,X}(\hat{f}(x_0))(\mathrm{E}_{\mathcal{Y},X}\hat{f}(x_0)-\hat{f}(x_0))) \\
&= (f(x_0)-\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))^2+\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))-\hat{f}(x_0))^2 \\
&= \mathrm{Bias}^2_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))+\mathrm{Var}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))
\end{aligned}
$$

Like part b, it is not hard to prove the cross product part (the red part) is zero.

(d) Establish a relationship between the squared biases and variance in the above two cases.

Sol:
By Law of Total Expectation, we have

$$\mathrm{E}_{\mathcal{Y},\mathcal{X}}(f(x_0)-\hat{f}(x_0))^2 = \mathrm{E}_{\mathcal{X}}(\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(f(x_0)-\hat{f}(x_0))^2)$$

Further, we can get

$$\mathrm{Bias}^2_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))+\mathrm{Var}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0)) = \mathrm{E}_{\mathcal{X}}(\mathrm{Bias}^2_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))+\mathrm{Var}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0)))$$

Now, let's look at this equation in details.

$$
\begin{aligned}
\mathrm{Bias}^2_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0) &= (f(x_0)-\mathrm{E}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0))^2 \\
&= (f(x_0)-\mathrm{E}_{\mathcal{X}}\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2 \\
&= (f(x_0)-\mathrm{E}_{\mathcal{X}}\sum_{i=1}^{N} l_i(x_0;\mathcal{X})f(x_i))^2 \\
&= (\mathrm{E}_{\mathcal{X}}(f(x_0)-\sum_{i=1}^{N} l_i(x_0;\mathcal{X})f(x_i)))^2 \\
&\leq \mathrm{E}_{\mathcal{X}}((f(x_0)-\sum_{i=1}^{N} l_i(x_0;\mathcal{X})f(x_i))^2 \\
&= \mathrm{E}_{\mathcal{X}}(f(x_0)-\mathrm{E}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2 \\
&= \mathrm{E}_{\mathcal{X}}\mathrm{Bias}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2
\end{aligned}
$$

We can achieve the relationship between the squared biases and variances:

$$\mathrm{Bias}^2_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0) \leq \mathrm{E}_{\mathcal{X}}\mathrm{Bias}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))^2$$
$$\mathrm{Var}_{\mathcal{Y},\mathcal{X}}(\hat{f}(x_0)) \geq \mathrm{E}_{\mathcal{X}}\mathrm{Var}_{\mathcal{Y}|\mathcal{X}}(\hat{f}(x_0))$$

Ex. 2.8

Compare the classification performance of linear regression and k - nearest neighbor classification on the zip code data. In particular, consider only the 2's and 3's and $k = 1,\ 3,\ 5,\ 7,$ and 15. Show both the training and test error for each choice. The zip code data are available from the book website `http://web.stanford.edu/~hastie/ElemStatLearn/`

```
# Data Loading
setwd("~/Desktop/Statistical Learning/ESL/zip_code/")
zip_train = read.csv(file = "zip_train.csv",sep = "",header = F)
zip_test = read.csv(file = "zip_test.csv", sep = "", header = F)
colnames(zip_train) = c("y", paste0("x", 1:256))
colnames(zip_test) = c("y",paste0("x",1:256))
zip_train_filter = subset(zip_train, zip_train$y == 2|zip_train$y == 3)
zip_test_filter = subset(zip_test, zip_test$y == 2|zip_test$y == 3)
```

```
# Linear Regression Classification
LR = lm(y~., zip_train_filter)
LR_predict_train = data.frame(y = ifelse(predict(LR,zip_train_filter)>=2.5,3,2))
LR_predict_test = data.frame(y = ifelse(predict(LR, zip_test_filter)>=2.5,3,2))

# Accurate Rate for Training Data
sum(LR_predict_train == zip_train_filter$y)/(dim(LR_predict_train)[1])

## [1] 0.9942405

# Accurate Rate for Test Data
sum(LR_predict_test == zip_test_filter$y)/(dim(LR_predict_test)[1])

## [1] 0.9587912
```

```
# KNN Classification
library('class')

# Function for Accurate Rate Calculation
knn_acc = function(k, train = zip_train_filter, test = zip_test_filter){
                     knn_model = knn(train = train[,-1], test = test[,-1],
                                          cl = train$y, k)
                     len = dim(test)[1]
                     return(acc = sum(knn_model == test$y)/len)
}

# K = 1 Train Accurate Rate
knn_acc(1, test = zip_train_filter)
```

```
## [1] 1

# K = 1 Test Accurate Rate
knn_acc(1)

## [1] 0.9752747

# K = 3 Train Accurate Rate
knn_acc(3, test = zip_train_filter)

## [1] 0.9949604

# K = 3 Test Accurate Rate
knn_acc(3)

## [1] 0.9697802

# K = 5 Train Accurate Rate
knn_acc(5, test = zip_train_filter)

## [1] 0.9942405

# K = 5 Test Accurate Rate
knn_acc(5)

## [1] 0.9697802

# K = 7 Train Accurate Rate
knn_acc(7, test = zip_train_filter)

## [1] 0.9935205

# K = 7 Test Accurate Rate
knn_acc(7)

## [1] 0.967033

# K = 15 Train Accurate Rate
knn_acc(15, test = zip_train_filter)

## [1] 0.9906407

# K = 15 Test Accurate Rate
knn_acc(15)

## [1] 0.9615385
```

Ex. 2.9

Consider a linear regression model with p parameters, fit by least squares to a set of training data $(x_1,\ y_1), \cdots, (x_N,\ y_N)$ drawn at random from a population. Let $\hat{\beta}$ be the least squares estimate. Suppose we have some test data $(\tilde{x}_1, \tilde{y}_1), \cdots, (\tilde{x}_M, \tilde{y}_M)$ drawn at random from the same population as the training data. If $R_{tr}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i^T\beta)^2$ and $R_{te}(\beta) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \tilde{x}_i^T\beta)^2$, prove that

$$\mathrm{E}[R_{tr}(\hat{\beta})] = \mathrm{E}[R_{te}(\hat{\beta})],$$

where the expectations are over all that is random in each expression.

Sol:
To simplify analysis, I make a strong assumption here:

$$y = x^T\beta + \epsilon,$$

where $\mathrm{E}(\epsilon|x) = 0,\ \mathrm{Var}(\epsilon) = \sigma^2$.

$$\begin{aligned}
R_{tr}(\hat{\beta}) &= \frac{1}{N}(Y - X\hat{\beta})^T(Y - X\hat{\beta}) \\
&= \frac{1}{N}(Y - X(X^TX)^{-1}X^TY)^T(Y - X(X^TX)^{-1}X^TY) \\
&= \frac{1}{N}(\epsilon^T\epsilon - \epsilon^TX(X^TX)^{-1}X^T\epsilon)
\end{aligned}$$

Since $X(X^TX)^{-1}X^T$ is a symmetric matrix (semi-definite), $\epsilon^TX(X^TX)^{-1}X^T\epsilon \geq 0$.

$$\begin{aligned}
\mathrm{E}(R_{tr}(\hat{\beta})) &= \mathrm{E}(\frac{1}{N}(\epsilon^T\epsilon - \epsilon^TX(X^TX)^{-1}X^T\epsilon)) \\
&\leq \mathrm{E}(\frac{1}{N}\epsilon^T\epsilon) \\
&= \sigma^2
\end{aligned}$$

$$\begin{aligned}
R_{te}(\hat{\beta}) &= \frac{1}{M}(\tilde{Y} - \tilde{X}\hat{\beta})^T(\tilde{Y} - \tilde{X}\hat{\beta}) \\
&= \frac{1}{M}(\tilde{X}\beta + \tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T(X\beta + \epsilon))^T(\tilde{X}\beta + \tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T(X\beta + \epsilon)) \\
&= \frac{1}{M}(\tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T\epsilon)^T(\tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T\epsilon)
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}R_{te}(\hat{\beta}) &= \mathrm{E}(\frac{1}{M}(\tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T\epsilon)^T(\tilde{\epsilon} - \tilde{X}(X^TX)^{-1}X^T\epsilon)) \\
&= \frac{1}{M}\mathrm{E}(\tilde{\epsilon}^T\tilde{\epsilon}) + \frac{1}{M}\mathrm{E}(\epsilon^TX(X^TX)^{-1}\tilde{X}^T\tilde{X}(X^TX)^{-1}X^T\epsilon) \\
&\geq \frac{1}{M}\mathrm{E}(\tilde{\epsilon}^T\tilde{\epsilon}) \\
&= \sigma^2
\end{aligned}$$

Q.E.D