Assignment 3 Answers

ISLR2 Ch5. Ex 3.

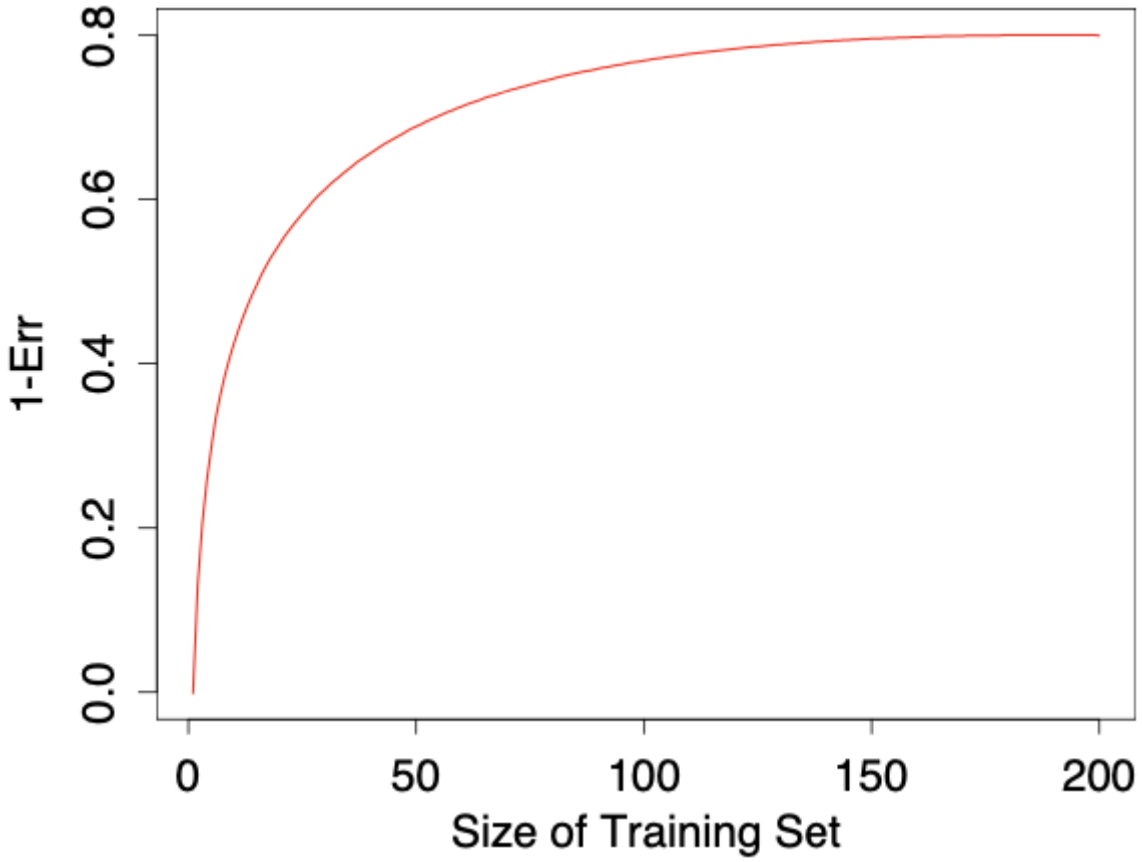
We now review k-fold cross-validation.

(a) Explain how k-fold cross-validation is implemented.

See class lecture.

(b) What are the advantages and disadvantages of k-fold cross-validation relative to:

i. The validation set approach? (We are considering relative small data set)



The validation set approach has two disadvantages:

1. Reduce the amount of data you have available to train your model on.
2. The estimate of the test error has high variance.

k-fold CV addresses both problems partially:

1. You are able to train your model on more data each time
2. Averaging over k times results lowers variance.

The disadvantage of k-fold CV is in its higher computational cost.

ii. LOOCV?

LOOCV has higher computational cost in general cases. But in special cases (see ESL2 Ex7.3 below), it has lower computational cost (in fact, you only need to do the training once in these cases).

ESL2

Ex. 7.3 Let $\hat{\mathbf{f}} = \mathbf{S}\mathbf{y}$ be a linear smoothing of $\mathbf{y}$.

(a) If S_{ii} is the ith diagonal element of $\mathbf{S}$, show that for $\mathbf{S}$ arising from least squares projections and cubic smoothing splines, the cross-validated residual can be written as

$$y_i - \hat{f}^{-i}(x_i) = \frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}. \tag{7.64}$$

(b) Use this result to show that $|y_i - \hat{f}^{-i}(x_i)| \geq |y_i - \hat{f}(x_i)|$.

(c) Find general conditions on any smoother $\mathbf{S}$ to make result (7.64) hold.

Proof:
(a) We have $\mathbf{S} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ for least square and $\mathbf{S} = \mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\Omega_N)^{-1}\mathbf{N}^T$ for cubic smoothing splines. We show the proof for least square for simplicity.

We have

$$S_{ii} = x_i^T(\mathbf{X}^T\mathbf{X})^{-1}x_i$$

$$\hat{f}(x_i) = x_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$$\begin{aligned}
\hat{f}^{-i}(x_i) &= x_i^T(\mathbf{X}_{-i}^T\mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^T\mathbf{y}_{-i} \\
&= x_i^T(\mathbf{X}^T\mathbf{X} - x_ix_i^T)^{-1}(\mathbf{X}^T\mathbf{y} - x_iy_i) \\
&\quad \text{(using Woodbury matrix identity)} \\
&= x_i^T\left((\mathbf{X}^T\mathbf{X})^{-1} + \frac{(\mathbf{X}^T\mathbf{X})^{-1}x_ix_i^T(\mathbf{X}^T\mathbf{X})^{-1}}{1 - x_i^T(\mathbf{X}^T\mathbf{X})^{-1}x_i}\right)(\mathbf{X}^T\mathbf{y} - x_iy_i) \\
&= \left(x_i^T(\mathbf{X}^T\mathbf{X})^{-1} + \frac{S_{ii}x_i^T(\mathbf{X}^T\mathbf{X})^{-1}}{1 - S_{ii}}\right)(\mathbf{X}^T\mathbf{y} - x_iy_i) \\
&= x_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} + \frac{S_{ii}x_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}{1 - S_{ii}} - x_i^T(\mathbf{X}^T\mathbf{X})^{-1}x_iy_i - \frac{S_{ii}x_i^T(\mathbf{X}^T\mathbf{X})^{-1}x_iy_i}{1 - S_{ii}} \\
&= \hat{f}(x_i) + \frac{S_{ii}\hat{f}(x_i)}{1 - S_{ii}} - S_{ii}y_i - \frac{S_{ii}^2y_i}{1 - S_{ii}} \\
&= \frac{\hat{f}(x_i) - y_iS_{ii}}{1 - S_{ii}}
\end{aligned}$$

Hence,

$$y_i - \hat{f}^{-i}(x_i) = \frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}$$

(b) Since $0 \leq S_{ii} \leq 1$, we can easily deduct from (a) that $|y_i - \hat{f}^{-i}(x_i)| \geq |y_i - \hat{f}(x_i)|$

(c) In general, if $\mathbf{S}$ does not depend on $\mathbf{y}$ and replacing y_i by $\hat{f}^{-i}(x_i)$ in $\mathbf{y}$ (let's call this $\mathbf{y}'$) gives $\hat{f}^{-i}(x_i)$ back as the ith element, then (7.64) hold.
This condition says

$$\begin{aligned}\hat{f}^{-i}(x_i) &= (\mathbf{S}\mathbf{y}')_i \\ &= \Sigma_{j\neq i} S_{ij} y_j + S_{ii}\hat{f}^{-i}(x_i) \\ &= \Sigma_j S_{ij} y_j - S_{ii} y_i + S_{ii}\hat{f}^{-i}(x_i) \\ &= \hat{f}(x_i) - S_{ii} y_i + S_{ii}\hat{f}^{-i}(x_i)\end{aligned}$$

Rearranging the equation, we get (7.64).

7.4,

Ex. 7.4 Consider the in-sample prediction error (7.18) and the training error $\overline{\text{err}}$ in the case of squared-error loss:

$$\begin{aligned}\text{Err}_{\text{in}} &= \frac{1}{N}\sum_{i=1}^{N} \text{E}_{Y^0}(Y_i^0 - \hat{f}(x_i))^2 \\ \overline{\text{err}} &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(x_i))^2.\end{aligned}$$

Add and subtract $f(x_i)$ and $\text{E}\hat{f}(x_i)$ in each expression and expand. Hence establish that the average optimism in the training error is

$$\frac{2}{N}\sum_{i=1}^{N}\text{Cov}(\hat{y}_i, y_i),$$

as given in (7.21).

Proof:
Let's denote $\hat{y}_i = \hat{f}(x_i)$

$$\begin{aligned}\text{Err}_{\text{in}} &= \frac{1}{N}\sum_{i=1}^{N} E_{Y^0}\left(Y_i^0 - f(x_i) + f(x_i) - E\hat{y}_i + E\hat{y}_i - \hat{y}_i\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} A_i + B_i + C_i + D_i + E_i + F_i,\end{aligned}$$

where

$$
\begin{aligned}
A_i &= E_{Y^0}(Y_i^0 - f(x_i))^2 = \sigma^2 \\
B_i &= E_{Y^0}(f(x_i) - E\hat{y}_i)^2 = (f(x_i) - E\hat{y}_i)^2 \\
C_i &= E_{Y^0}(E\hat{y}_i - \hat{y}_i)^2 = (E\hat{y}_i - \hat{y}_i)^2 \\
D_i &= 2E_{Y^0}(Y_i^0 - f(x_i))(f(x_i) - E\hat{y}_i) \\
E_i &= 2E_{Y^0}(Y_i^0 - f(x_i))(E\hat{y}_i - \hat{y}_i) = 0 \\
F_i &= 2E_{Y^0}(f(x_i) - E\hat{y}_i)(E\hat{y}_i - \hat{y}_i) = 2(f(x_i) - E\hat{y}_i)(E\hat{y}_i - \hat{y}_i)
\end{aligned}
$$

Similarly for $\overline{\text{err}}$ we have

$$
\begin{aligned}
\overline{\text{err}} &= \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i) + f(x_i) - E\hat{y}_i + E\hat{y}_i - \hat{y}_i)^2 \\
&= \frac{1}{N}\sum_{i=1}^{N} G_i + B_i + C_i + H_i + J_i + F_i,
\end{aligned}
$$

where

$$
\begin{aligned}
G_i &= (y_i - f(x_i))^2 \\
H_i &= 2(y_i - f(x_i))(f(x_i) - E\hat{y}_i) \\
J_i &= 2(y_i - f(x_i))(E\hat{y}_i - \hat{y}_i).
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
E_y(\text{op}) &= E_y(\text{Err}_{\text{in}} - \overline{\text{err}}) \\
&= \frac{1}{N}\sum_{i=1}^{N} E_y[(A_i - G_i) + D_i - H_i - J_i].
\end{aligned}
$$

$E_y[G_i] = \sigma^2$, thus $E_y(A_i - G_i) = 0$. Similarly we have
$E_y D_i = E_y H_i = 0$, and thus

$$
\begin{aligned}
E_y(\text{op}) &= -\frac{2}{N}\sum_{i=1}^{N} J_i \\
&= \frac{2}{N}\sum_{i=1}^{N} E_y(y_i - f(x_i))(\hat{y}_i - E\hat{y}_i) \\
&= 2\text{Cov}(y_i, \hat{y}_i).
\end{aligned}
$$

Ex. 7.5 For a linear smoother $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$, show that

$$
\sum_{i=1}^{N} \text{Cov}(\hat{y}_i, y_i) = \text{trace}(\mathbf{S})\sigma_\varepsilon^2, \tag{7.65}
$$

which justifies its use as the effective number of parameters.

Proof:

$$\begin{aligned}\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i) &= \mathrm{trace}(\mathrm{Cov}(\hat{\mathbf{y}}, y)) \\ &= \mathrm{trace}(\mathrm{Cov}(\mathbf{S}y, y)) \\ &= \mathrm{trace}(\mathbf{S}\mathrm{Cov}(y, y)) \\ &= \mathrm{trace}(\mathbf{S}\sigma_\epsilon^2 I) \\ &= \mathrm{trace}(\mathbf{S})\sigma_\epsilon^2.\end{aligned}$$

Ex. 7.6 Show that for an additive-error model, the effective degrees-of-freedom for the k-nearest-neighbors regression fit is N/k.

Proof:
For k-nearest-neighbors, we have $\hat{f} = \mathbf{S}y$, where

$$S_{ij} = \begin{cases} \frac{1}{k} & \text{if } x_j \text{ is one of the k nearest neighbors of } x_i \\ 0 & \text{otherwise} \end{cases}$$

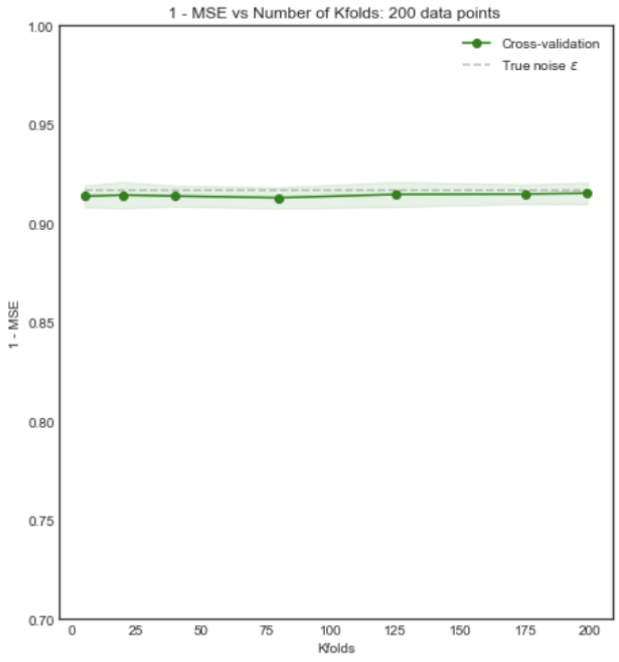
So the degree-of-freedom for the k-nearest-neighbors regression fit is $\mathrm{trace}(\mathbf{S}) = \frac{N}{k}$.

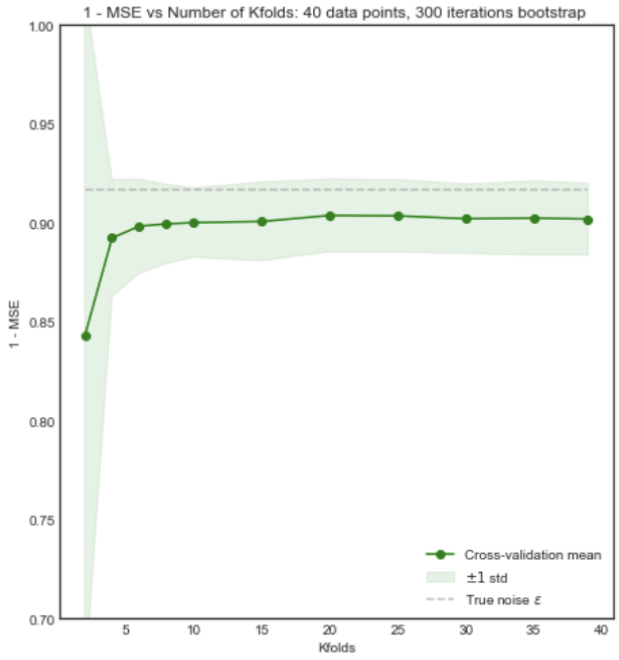
Experiment about Variance/Bias in K-fold Cross-Validation

1. Generate 10,000 points from the distribution $\sin(x) + \epsilon$ where the true variance of ϵ is known
2. Iterate i times (e.g. 100 or 200 times). At each iteration, change the dataset by resampling N points (try different Ns, e.g. 40, 200 to see the difference) from the original distribution
3. For each data set ii:
   - Perform K-fold cross validation for one value of K
   - Store the average Mean Square Error (MSE) across the K-folds
4. Once the loop over i is complete, calculate the mean and standard deviation of the MSE across the i datasets for the same value of K
5. Repeat the above steps for all K in range $\{5, \ldots, N\}$ all the way to Leave One Out CV (LOOCV)

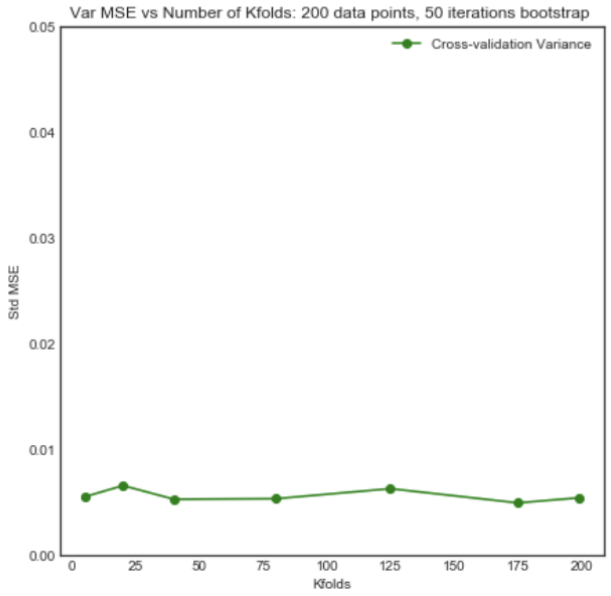
You should get plots like these

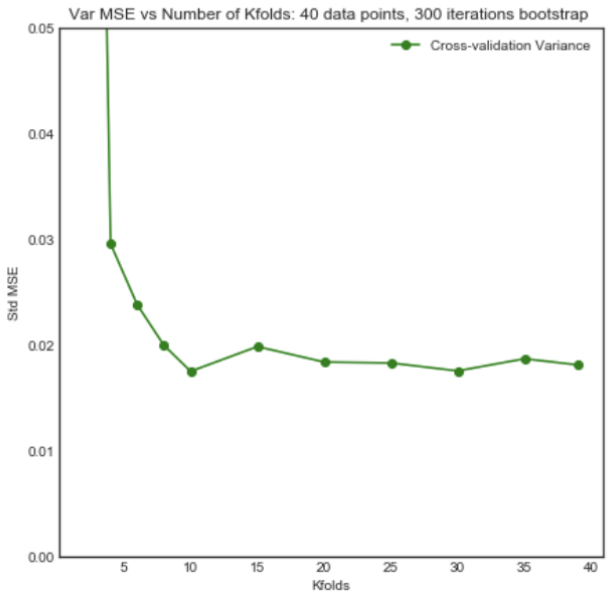
Left Hand Side: Kfolds for 200 data points, **Right Hand Side**: Kfolds for 40 data points





Standard Deviation of MSE (across data sets i) vs Kfolds





What does this experiment tell you?

Experiment about Wrong and Right Ways to do Cross-Validation

1. Generate 50 samples for 5000 standard Gaussian random variables as predictors and binary label (0 or 1 with 50/50 odds). All predictors and label are independent.
2. Find the 100 predictors having the largest correlation with the class labels.

3. Find 5-fold CV error (% of incorrectly classified cases) for 1 nearest neighbor classifier based on these 100 selected predictors.
4. Repeat step 1-3 50 times to get a distribution for CV error.
   This corresponds to the wrong way in class.
   Repeat this with the right way of doing CV.
   Since the label is independent of the predictors, any classifier should have roughly 50% error rate. What do you find in the wrong way and right way of doing CV?

ChatGPT can do it the wrong way very easily. But it takes me several tries before it understood that it did it the wrong way even if I instructed it to do it in the right way. Here is the moment that it realized its mistake: