Section 6: Cross-validation

STA 141A – Fundamentals of Statistical Data Science (UC Davis, Spring 2026)

Instructor: Akira Horiguchi

Overview

Based on Chapter 5 of ISL book James et al. (2021).

- For more R code examples, see R Markdown files in ISLR's website

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

Idea

Recall distinction between *test error rate* and *training error rate* of an estimator $\hat{f}$.

- Want to avoid overfitting and systematic bias (bias-variance tradeoff)
- $\hat{f}$'s predictive ability can be quantified by the *population* test error

$$E\left[\text{error}\left(Y, \hat{f}(X)\right)\right] \tag{1}$$

  which we typically cannot directly calculate in practice because the entire population is typically unknown or inaccessible.

- The population test error (1) can be estimated by the *empirical* test error

$$\frac{1}{m}\sum_{i=1}^{m}\text{error}\left(y_{n+i},\ \hat{f}(x_{n+i})\right) \tag{2}$$

- Choose estimator that produces smallest empirical test error (2).
- Evaluating an estimator's performance is known as *model assessment*.

Idea

However, a designated test set is typically not available.

- How to estimate test error (1) in such cases?
- Can instead train the estimator on a *subset* of the available data, then assess performance on the unused data.
- Also helps to select proper level of flexibility for a model; process known as *model selection*.

For now we consider only regression (classification is similar).

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

Validation set approach

Randomly partitions the available data in two sets of the same size: a *training set* and a *validation set* (or *hold-out set*). Procedure:

1. Randomly partition the available data in two sets of the same size.
2. Fit the model on the training set.
3. Use the validation set to assess the performance of the fit (e.g., MSE).

Example: We want to do linear regression given the data set.

We split the whole data set into two groups with three elements each.

$$(x_1, y_1) = (1, 12), \quad (x_2, y_2) = (2, 14), \quad (x_3, y_3) = (4, 12),$$
$$(x_4, y_4) = (6, 15), \quad (x_5, y_5) = (8, 17), \quad (x_6, y_6) = (9, 22).$$

```
set.seed(37)  # allows these "random" numbers to be reproduced later
N <- 6
train_inds <- sample(N, N/2)  # 6 2 3
valid_inds <- (1:N)[-train_inds]  # 1 4 5
```

1. Fit a linear model to the training set $\{(x_6, y_6), (x_2, y_2), (x_3, y_3)\}$.
2. Compute MSE of fitted linear function $\hat{f}$ on validation set: $\frac{1}{3}\sum_{i\in\{1,4,5\}} \left(y_i - \hat{f}(x_i)\right)^2$.

Conceptually simple and easy to implement, but two major drawbacks

1. The validation estimate of the test error rate highly depends on the values in the validation set.

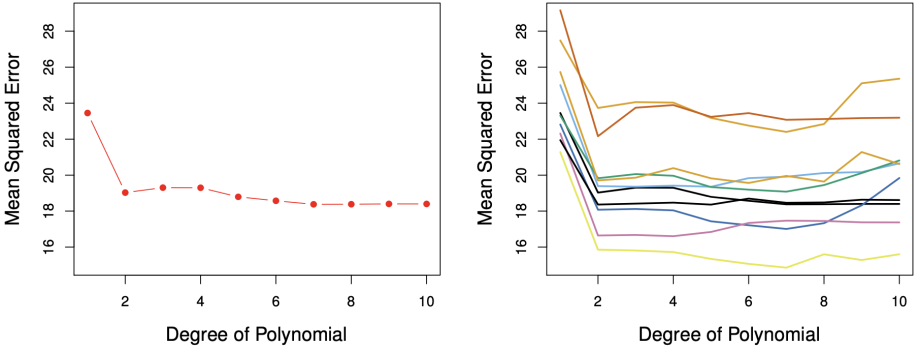

Figure 1: Image by James et al. (2021) using `Auto` data set of validation errors from predicting `mpg` using polynomial functions of `horsepower`. Left: one random split. Right: 10 random splits, illustrating variability in the estimated test MSE.

2. Statistical methods tend to perform worse if trained on half of the whole data set compared to using the whole data set.

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

LOOCV: idea

Leave-one-out cross validation (LOOCV): one data point for the validation set, and the remaining $n - 1$ data points for the training set.

- Start by leaving (x_1, y_1) out, train our model on $(x_2, y_2), \ldots, (x_n, y_n)$, and predict y_1 by $\hat{y}_1$ based on the trained model, and calculate MSE_1.
- MSE_1 is based on a single observation (x_1, y_1), making it highly variable and hence a poor estimate for the test error.
- Thus we repeat the LOOCV by leaving out (x_2, y_2), then (x_3, y_3), etc.

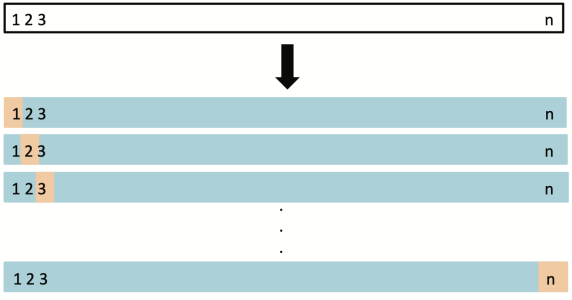

Figure 2: Image by James et al. (2021). Cartoon of folds in leave-one-out cross validation.

LOOCV: procedure given the data $(x_1, y_1), \ldots, (x_n, y_n)$

- 1st step:
  - Leave (x_1, y_1) out, and use it as validation set.
  - Derive an estimator $\hat{f}_1$ based on the training set $(x_2, y_2), \ldots, (x_n, y_n)$.
  - Calculate $MSE_1 := (y_1 - \hat{y}_1)^2$, where $\hat{y}_1 = \hat{f}_1(x_1)$.
- $\vdots$
- nth step:
  - Leave (x_n, y_n) out, and use it as validation set.
  - Derive an estimator $\hat{f}_n$ based on the training set $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$.
  - Calculate $MSE_n := (y_n - \hat{y}_n)^2$ where $\hat{y}_n = \hat{f}_n(x_n)$.
- $(n+1)$st step: Calculate the LOOCV estimate for the test MSE, namely

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} MSE_i.$$

LOOCV: example

Example: estimate test MSE for linear regression using LOOCV.

Data set $(x_1, y_1) = (5, 50), (x_2, y_2) = (6, 60), (x_3, y_3) = (4, 20)$, so $n = 3$.

1. Leave out $(x_1, y_1) = (5, 50)$.
   Train $\hat{f}_1$ on $(x_2, y_2) = (6, 60), (x_3, y_3) = (4, 20) \implies \hat{f}_1(x) = 20x - 60$.
   As $\hat{f}_1(5) = \hat{y}_1 = 40$, get $MSE_1 = (y_1 - \hat{y}_1)^2 = (50 - 40)^2 = 100$.
2. Leave out $(x_2, y_2) = (6, 60)$.
   Train $\hat{f}_2$ on $(x_1, y_1) = (5, 50), (x_3, y_3) = (4, 20) \implies \hat{f}_2(x) = 30x - 100$.
   As $\hat{f}_2(6) = \hat{y}_2 = 80$, get $MSE_2 = (y_2 - \hat{y}_2)^2 = (60 - 80)^2 = 400$.
3. Leave out $(x_3, y_3) = (4, 20)$.
   Train $\hat{f}_3$ on $(x_1, y_1) = (5, 50), (x_2, y_2) = (6, 60) \implies \hat{f}_3(x) = 10x$.
   As $\hat{f}_3(4) = \hat{y}_3 = 40$, get $MSE_3 = (y_3 - \hat{y}_3)^2 = (20 - 40)^2 = 400$.

Thus the test-MSE estimate for linear regression is $\boxed{CV_{(3)} = (100 + 400 + 400)/3 = 300}$.

We could also compute $CV_{(3)}$ for a quadratic fit, and then choose the model — linear fit vs quadratic fit — that produces the smaller $CV_{(3)}$ value.

LOOCV: pros and cons

Pros:

- $n-1$ training data points; thus LOOCV tends not to overestimate the test error rate (compared to validation set approach).
- In LOOCV each data point is left out exactly once, so data splits are not random.
- LOOCV is a general method that can be used for many statistical learning methods.

Cons: LOOCV can computationally be very expensive since n estimators are fit.

- Exception: with least squares linear or polynomial regression, the cost of LOOCV is (amazingly!) the same as that of a single model fit:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2,$$

  where the leverage h_i is defined in the textbook (no need to remember this for HW/exam).

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

k-fold CV: idea

k-fold CV randomly partitions the data with n elements in k groups (*folds*) of about equal size, by leaving the first fold out as a validation set, using the remaining $k-1$ folds as a training set, and repeating the procedure k times.

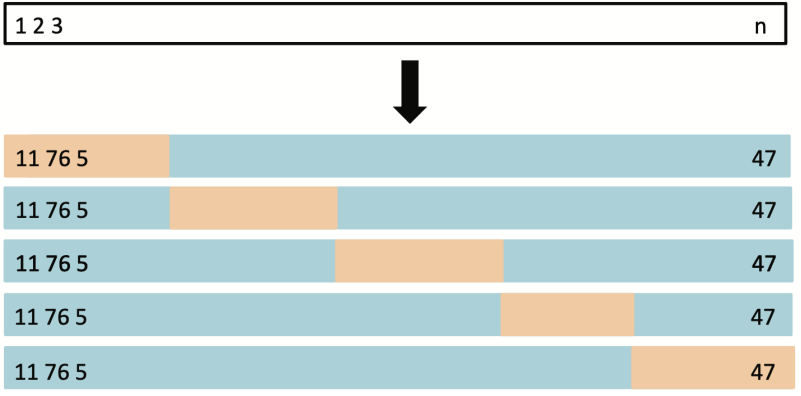

Figure 3: Image by James et al. (2021). Cartoon of folds in 5-fold cross validation. Here the random partition is achieved by first permuting indices $1, 2, \ldots, n$, and then partition into k folds.

k-fold CV: procedure, given the data $(x_1, y_1), \ldots, (x_n, y_n)$

- 0st step: Randomly split the given data in k folds (k is predefined).
- 1st step:
  - Leave the 1st fold out, and use it as validation set.
  - Derive an estimator $\hat{f}$ based on the remaining $k-1$ folds.
  - Calculate MSE_1 based on the 1st left out fold (if $n = 100$ and $k = 5$, so we have $k = 5$ folds with $n/k = 20$ elements each, then with I_1 denoting the set of the indices of all elements in the first fold (e.g. $I_1 = \{1, 3, 5, 10, 11, 86, \ldots, 100\}$), we have $MSE_1 = \frac{1}{n/k} \sum_{i \in I_1} (y_i - \hat{y}_i)^2$).
- $\vdots$
- kth step:
  - Leave the kth fold out, and use it as validation set.
  - Derive an estimator $\hat{f}$ based on the remaining $k-1$ folds.
  - Calculate MSE_k based on the kth left out fold.
- $(k+1)$th step: Calculate the k-fold CV estimate for the test MSE, namely

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i \,. \tag{3}$$

k-fold CV generalizes LOOCV ($k = n$), but often use $k = 5$ or $k = 10$

- If $k < n$, then k-fold CV requires fewer computations than does LOOCV.
- Another advantage of k-fold CV involves the *bias-variance trade-off*. Consider two sources of variability: (1) random data split and (2) data from unknown distribution.

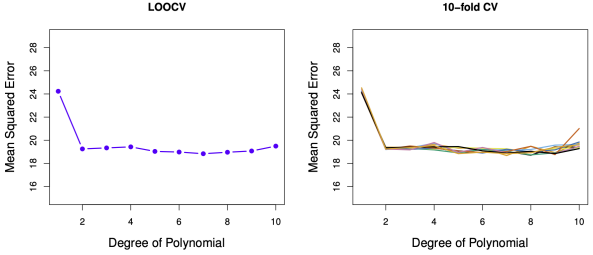

Figure 4: Image by James et al. (2021) using **single** `Auto` data set of validation errors from predicting `mpg` using polynomial functions of `horsepower`.

- LOOCV has the *smallest bias* compared to k-fold CV for any other k; gives approximately unbiased estimates of the test error since each training set has $(n - 1)$ observations.
- LOOCV also has the *largest variance* compared to k-fold CV for any other k; because the n fitted models are trained on almost identical data sets, their outputs are highly positively correlated, so the variance does not lessen much when averaging over the n fitted models.

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

Let's compare LOOCV and 10-fold CV

Without the true test MSE, it is difficult to assess the CV estimate.

- The true test MSE can be computed for *simulated data*.

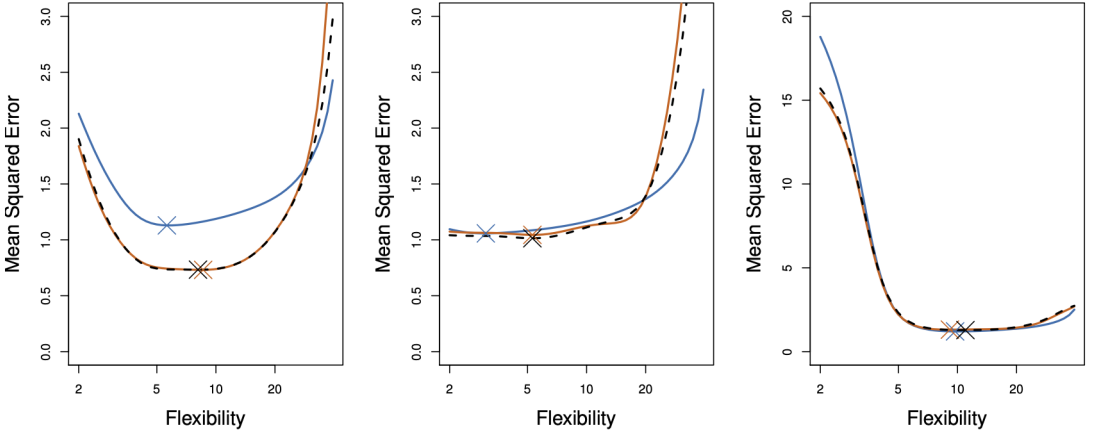

Figure 5: Image by James et al. (2021). Each panel corresponds to one of three simulated data sets; true test MSE (blue), LOOCV estimate (black dashed), and 10-fold CV estimate (orange). Cross indicates minimum of MSE curve.

What do we care about? Model assessment vs model selection

We can use the CV error as an estimate of the test error. Two possible analysis goals:

1. Determine how well a given statistical learning procedure can be expected to perform on independent data; in this case, the actual estimate of the test MSE is of interest. Evaluating a model's performance is known as *model assessment*.
2. Determine the level of flexibility that produces the smallest estimated test MSE; process is known as *model selection*.

Let's use these two lenses to examine the previous figure.

Comments on previous figure

Panel	Model assessment	Model selection
All	10-fold CV estimate is pretty close to the LOOCV estimate (and is computationally faster).	If there is a range of flexibility values that produce roughly the smallest estimated test MSE, we want to choose the smallest value in this "good range" (typically enables easier inference).
Left	Both CV estimates seem to underestimate the true test MSE.	For both CV approaches, any flexibility between 5 and 10 produces roughly the smallest estimated test MSE.
Center	Both CV estimates seem to roughly match the true test MSE for flexibility values smaller than 12.	For both CV approaches, any flexibility smaller than 8 produces roughly the smallest estimated test MSE.
Right	Both CV estimates seem to very closely match the true test MSE for flexibility values between 4 and 20.	For both CV approaches, any flexibility between 6 and 20 produces roughly the smallest estimated test MSE.

Validation set approach

Leave-one-out cross validation approach

k-fold cross validation approach

Comparison

Classification

Classification

Cross-validation can also be used for qualitative responses (in classification).

- The LOOCV error rate in the classification setting takes the form

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i\,, \tag{4}$$

where

$$Err_i \coloneqq I(y_i \neq \hat{y}_i) = \begin{cases} 1, & \text{if } y_i \neq \hat{y}_i \text{ (obs } i \text{ is misclassified)} \\ 0, & \text{otherwise (obs } i \text{ is assigned to the correct class)} \end{cases} \tag{5}$$

- Bias-variance tradeoff again in Figures 5.7 and 5.8 of James et al. (2021).