STAT562 Lecture 7 Cross Validation

Beidi Qiang

SIUE

Introduction

Cross-Validation is a well-used resampling method, which involves repeatedly drawing samples from set and refitting a model in order to obtain additional information about the fitted model.

- Cross-Validation is used to estimate the test error associated with a given model.
- Very general method, and can be used with any kind of predictive modeling.
- It is used for model assessment, i.e. evaluating a model's performance.
- It can also be used for model selection, i.e. selecting the proper level of model flexibility, or tuning the model parameters

Training and Test Error

- The test error is the average error that results from using a trained model to predict the response on a new set of observation
- The training error is the average error by applying the trained model to the observations used in its training.
- The training error rate often can be quite different from the test error rate, especially when overfitting.
- The test error can be easily calculated if a designated test set is available, what if not?

The Training - Validation - Test split

Dividing the available set of observations into three parts, a training set, validation set (optional) and a test set.

- First, the model is fit on the training set.
- If you'd like to tweak model hyper-parameters or perform model selection, use the fitted model to predict the labels for the observations in the validation set. Tweak the model based on the errors in the validation set. This step can be optional depending on whether model tweaking is needed.
- Then the final model is used to predict the labels for the observations in the test set. And test error rate (or MSE) can be calculated.

Drawbacks

- The estimate of the test error rate can be highly variable, depending how the data is slit into training and test.
- Only a subset of the observations (those that are included in the training set) are used to fit the model. The model training tends to suffer from reduced number of training observations.

Leave-One-Out Cross-Validation

Leave-one-out cross-validation (LOOCV) involves splitting the set of training samples into two parts repeatedly. A test set should still be held out for final evaluation, but the validation set is no longer needed when doing. At each iteration i,

- The ith observation is used as the validation set, and the remaining observations make up the training set.
- The model is fit on the $n-1$ training observations, and a prediction is made for the excluded ith observation, which provides a test error estimate (MSE_i).

Repeat the steps for each observation i, and the LOOCV estimate for the test MSE is the average of these n test error estimates:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i \text{ or } CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i$$

The LOOCV schematic

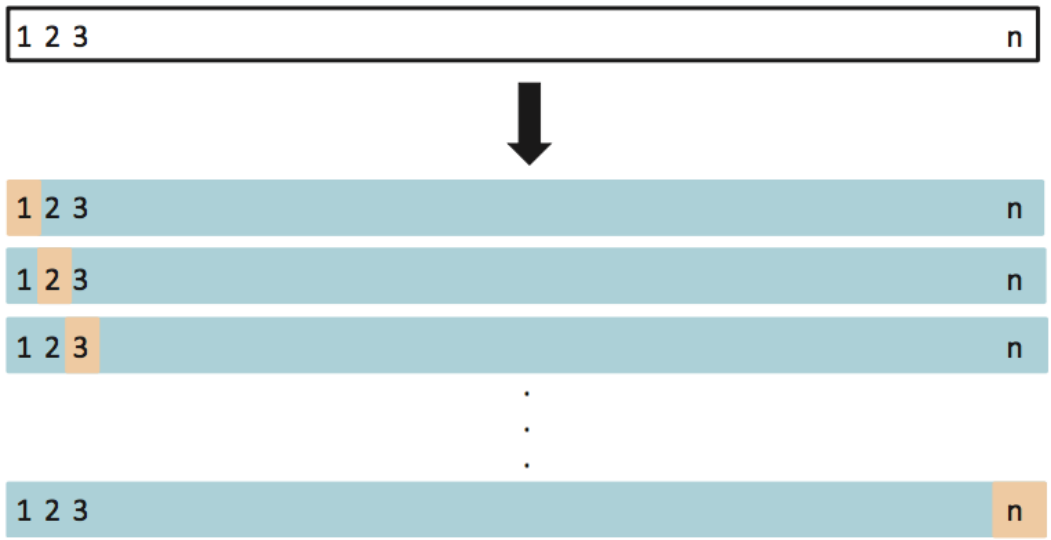

FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

Advantages and Drawbacks of LOOCV

- We repeatedly fit the model using the $n-1$ training data, almost as many as are in the entire data set.
- There is no randomness in the training/validation set splits, LOOCV give the same result each time.
- However, LOOCV can be very time consuming if n is large, since the model has to be fit n times and each fitting procedures can be computationally intensive.
- The MSE_i's from LOOCV are highly (positively) correlated with each other since the model is trained each time on an almost identical set of observations.

k-Fold Cross-Validation

K-fold CV involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. At each iteration i,

- The ith fold is treated as a validation set, and the model is fit on the remaining $k-1$ folds.
- The estimated error rate Err_i or MSE_i is then computed on the observations in the held-out fold.

Repeat the steps for each fold i, and the K-fold CV estimate for the test error rate is the average of these k test error estimates:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i \text{ or } CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} Err_i$$

The k-Fold Cross-Validation schematic

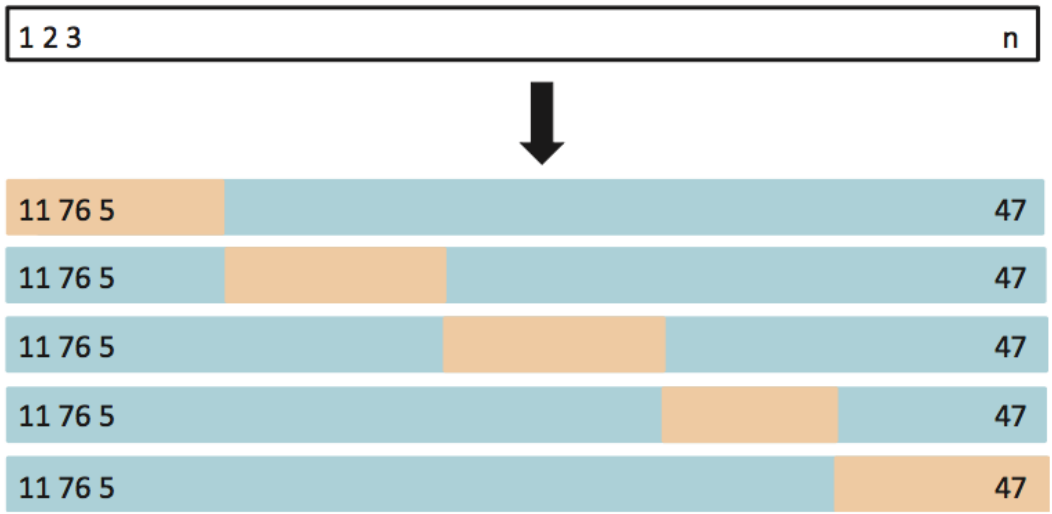

FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

Comments on k-Fold Cross-Validation

- LOOCV is a special case of k-fold CV in with $k = n$
- k-Fold CV requires fitting the learning procedure only k times, which provides a computational advantage over LOOCV.
- The MSE_i's from k-Fold CV are less correlated.
- There is some variability in the k-fold CV estimates as a result of the randomness in how the observations are divided into folds. This variability is lower with larger k.
- Typically choice of k are $k = 5$ or $k = 10$. these values have been shown empirically to yield good test error rate estimates.

Example: Iris data

```
> data(iris)
> library(caret)
> train.y=iris[,5]
> train.x=iris[,1:4]
> model = train(train.x,train.y,
+               method = "knn",
+               preProcess = c("center", "scale"),
+               tuneLength = 10,
+               trControl = trainControl(method = "cv",
+                                        number = 10))
> print(model)
k-Nearest Neighbors

150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

Pre-processing: centered (4), scaled (4)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.9466667  0.92
   7  0.9666667  0.95
   9  0.9533333  0.93
  11  0.9533333  0.93
  13  0.9666667  0.95
  15  0.9733333  0.96
  17  0.9600000  0.94
  19  0.9466667  0.92
  21  0.9600000  0.94
  23  0.9533333  0.93

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 15.
```