Report on Cross Validation, Validation sets and leave one out Cross Validation in R

Course: 837 Machine Learning

Submitted to

Kishan Kumar Ganguly

Lecturer

Institute of Information Technology

University of Dhaka

Submitted by

Maloy Kanti Sarker

BSSE0834

Institute of Information Technology

University of Dhaka

Date: 03-10-2019

1. Introduction

Cross-validation, sometimes called rotational or off-sample testing, is some of several comparable model validation methods to evaluate how the statistical analysis findings are generalized into an autonomous set of information. The main purpose is to assess how correctly a predictive model works in practice. In this context, the main objective is prediction. In an issue with the forecast, a model generally provides a dataset of known information on which training is performed (training data set), and a data set of unknown information (or first-viewed information) for which the model is tested. The objective of cross-validation is to test the ability of the model to predict new information that wasn't used in its estimation, in order to indicate issues such as override or choice distortions and to show how the model generalizes into an autonomous data set (for example an unknown dataset from a true issue). One round is for the information sample to be divided into additional subsets, analyzed into one subgroup (called training set), and the evaluation validated on the other subgroup (called validation set or test set). Multiple rounds of cross-validation using separate parts are carried out in most techniques in order to decrease variability, and validation outcomes over the rounds are combined (e.g. averaged) to provide a predictive performance estimate.

2. Objective

The objective of this report is to understand the significance of doing cross validation for assessing the predictive performance of the models. Cross validation is a way to address the trade of between bias and variance. When we obtain a training set, our goal is to minimize the variance. The purpose of using cross-validation methods is to fit a model to a training dataset. We have information without cross validation only on how our model performs with our in-sample data. Ideally we want to see how the model works when we have fresh information on the precision of its forecasts. In science, its predictive efficiency assesses theories.

3. Methodology

Cross validation is a method involving the reservation of a certain sample of a dataset on which the model is not trained. Later, before finalizing, we will test our model for this sample.

Here are the steps involved in cross validation:

1. Reserve a sample data set
2. Train the model using the remaining part of the dataset
3. Use the reserve sample of the test (validation) set

4. Result and Discussion

4.1 Validation Set Approach

I overserve 20% of the validation dataset and the remaining 80% of the model for training.

```
library(tidyverse)
library(caret)
data("swiss")
sample_n(swiss, 3)
set.seed(123)
training.samples <- swiss$Fertility%>%
createDataPartition(p = 0.8, list = FALSE)
train.data  <- swiss[training.samples, ]
test.data <- swiss[-training.samples, ]
# Build the model
model <- lm(Fertility ~., data = train.data)
predictions <- model %>% predict(test.data)
data.frame( R2 = R2(predictions, test.data$Fertility),
            RMSE = RMSE(predictions, test.data$Fertility),
            MAE = MAE(predictions, test.data$Fertility))
```

Figure 1: Validating set splitting approach

Code Explanation:

1. **training.samples <- swiss$Fertility%>% createDataPartition(p = 0.8, list = FALSE):** Splitting the data into traing and test set. Here I used 80% data for training and 20% for validation set.
2. **model <- lm(Fertility ~., data = train.data):** build the model.
3. **predictions <- model %>% predict(test.data):** Make prediction on test data.
4. **data.frame(R2 = R2(predictions, test.data$Fertility),**
   RMSE = RMSE(predictions, test.data$Fertility),
   MAE = MAE(predictions, test.data$Fertility)) : Compute R^2 , Root Mean Squared Error(RMSE), Mean Absolute Error(MAE)

Output:

```
         R2      RMSE      MAE
1 0.5946201 6.410914 5.651552
```

When comparing two models, the one that produces the lowest test sample RMSE is the preferred model. The RMSE and the MAE are measured in the same scale as the outcome variable. Dividing the RMSE by the average value of the outcome variable will give you the prediction error rate, which should be as small as possible:

```
RMSE(predictions, test.data$Fertility)/mean(test.data$Fertility)
```

Output:

```
[1] 0.08800157
```

Note that, if we have a big, partitioned dataset, the validation set technique is helpful only. One disadvantage is that we only create a pattern on a fraction of the data set, which might lead to greater variations in some interesting information on the data. The test error rate can therefore be extremely variable based on what observations in the training set are included and what observations in the validation set are included.

4.2 Leave one out cross validation-LOOCV

In this approach, we reserve only one data point from the available dataset, and train the model on the rest of the data. This process iterates for each data point. This also has its own advantages and disadvantages. The method works as follows:

1. Leave out one data point and build the model on the rest of the data set.
2. Test the model against the data point that is left out at step 1 and record the test error associated with the prediction
3. Repeat the process for all data points
4. Compute the overall prediction error by taking the average of all these test error estimates recorded at step 2.

```
17 train.control <- trainControl(method = "LOOCV")
18 model <- train(Fertility ~., data = swiss, method = "lm",
19                 trControl = train.control)
20 print(model)
```

Figure 2: Leave one cross validation

Code Explanation:

1. **train.control <- trainControl(method = "LOOCV") :** Define training control
2. **model <- train(Fertility ~., data = swiss, method = "lm", trControl = train.control):** Train the model
3. **print(model) :** Summarize model

Output:

```
47 samples
 5 predictor

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 46, 46, 46, 46, 46, 46, ...
Resampling results:

  RMSE      Rsquared   MAE
  7.738618  0.6128307  6.116021
```

Figure 3: Output of Leave one out cross Validation

4.3 K-fold Cross Validation

The k-fold cross-validation technique evaluates the model efficiency on separate training data subsets and then calculates the average error rate for forecast. The algorithm as follows:

1. Randomly split the data set into k-subsets (or k-fold) (for example 5 subsets)
2. Reserve one subset and train the model on all other subsets
3. Test the model on the reserved subset and record the prediction error
4. Repeat this process until each of the k subsets has served as the test set.
5. Compute the average of the k recorded errors. This is called the cross-validation error serving as the performance metric for the model.

The most obvious advantage of k-fold CV compared to LOOCV is computational. A less obvious but potentially more important advantage of k-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV. Lower value of K is more biased and hence undesirable. On the other hand, higher value of K is less biased, but can suffer from large variability. It is not hard to see that a smaller value of k (say k = 2) always takes us towards validation set approach, whereas a higher value of k (say k = number of data points) leads us to LOOCV approach.

```
21 train.control <- trainControl(method = "cv", number = 10)
22 model <- train(Fertility ~., data = swiss, method = "lm",
23                trControl = train.control)
24 print(model)
```

Figure 4: K-fold Cross validation

Code Explanation:

1. **train.control <- trainControl(method = "cv", number = 10) :** train the model with 10-fold cross validation
2. **model <- train(Fertility ~., data = swiss, method = "lm"**
   trControl = train.control) : Build the model
3. **print(model) :** Summarize the result

Output:

```
47 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 42, 44, 42, 43, 41, 42, ...
Resampling results:

  RMSE      Rsquared   MAE
  7.126707  0.6863589  6.046966
```

Figure 5: K-fold cross validation

4.4 Repeated K-fold Cross Validation

The process of splitting the data into k-folds can be repeated a number of times, this is called repeated k-fold cross validation. The final model error is taken as the mean error from the number of repeats. The following example uses 10-fold cross validation with 3 repeats:

```
25 train.control <- trainControl(method = "repeatedcv",
26                                number = 10, repeats = 3)
27 model <- train(Fertility ~., data = swiss, method = "lm",
28                trControl = train.control)
29 print(model)
```

Figure 6: Repeated k-fold cross validation

Code Explanation:

1. **train.control <- trainControl(method = "repeatedcv",**
   number = 10, repeats = 3) : Train 10-fold cross validation with 3 repeations.
2. **model <- train(Fertility ~., data = swiss, method = "lm"**
   trControl = train.control) : Build the model
3. **print(model) :** Summarize the result

Output:

```
47 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 43, 42, 42, 43, 41, 42, ...
Resampling results:

  RMSE      Rsquared   MAE
  7.304991  0.7211256  6.030067
```

Discussion

In this chapter, I described 4 different methods for assessing the performance of a model on unseen test data. These methods include: validation set approach, leave-one-out cross-validation, k-fold cross-validation and repeated k-fold cross validation. I have used several statistical metrics for quantifying the overall quality of regression models. These include:

1. **R-squared (R2):** Representing the squared correlation between the observed outcome values and the predicted values by the model. The higher the adjusted R2, the better the model.
2. **Root Mean Squared Error (RMSE):** It measures the average prediction error made by the model in predicting the outcome for an observation. That is, the average difference between the observed known outcome values and the values predicted by the model. The lower the RMSE, the better the model.
3. **Mean Absolute Error (MAE):** An alternative to the RMSE that is less sensitive to outliers. It corresponds to the average absolute difference between observed and predicted outcomes. The lower the MAE, the better the model

The R^2 value for these 3 methods are 0.5946201, 0.6128307, 0.6863589 and 0.7211256 correspondingly. The RMSE values are 6.410194, 7.738618, 7.126707 and 7.304991. The MAE values are 5.651552, 6.116021, 6.046966 and 6.030067.

5. Conclusion

Cross-validation is a great way of testing a predictive model. Although a system can minimize the mean squared error of the training information, its predictive error can be optimistic. Cross-validation partitions assist simulate an autonomous information set and improve the evaluation of the predictive results of a model.