What is the concept behind Leave-One-Out Cross-Validation (LOOCV)?

Authored by

stats writer

April 21, 2024

Leave-One-Out Cross-Validation (LOOCV) is a concept used in statistical analysis to evaluate the performance of a predictive model. It involves splitting the dataset into n subsets, where n is the number of observations in the dataset. Each subset is then used as the test set while the remaining n-1 subsets are used as the training set. This process is repeated n times, with each iteration leaving out a different subset as the test set. The results from all n iterations are then averaged to obtain an overall performance measure of the model. LOOCV is useful in determining the generalizability of a model, as it uses all the available data for training and testing, while preventing overfitting.

A Quick Intro to Leave-One-Out Cross-Validation (LOOCV)

To evaluate the performance of a model on a dataset, we need to measure how well the predictions made by the model match the observed data.

The most common way to measure this is by using the mean squared error (MSE), which is calculated as:

MSE = (1/n)*Σ(yi - f(xi))2

where:

n: Total number of observationsyi: The response value of the ith observationf(xi): The predicted response value of the ith observation

The closer the model predictions are to the observations, the smaller the MSE will be.

In practice, we use the following process to calculate the MSE of a given model:

1. Split a dataset into a training set and a testing set.

	x_1	x_2	x_3	y	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					**Training Set**
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					**Testing Set**
26					
27					
28					
29					
30					

2. Build the model using only data from the training set.

	x_1	x_2	x_3	y
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Training Set

Use this training data to build some model:

$$y = 3.5(x_1) - 9.3(x_2) + 2.1(x_3)$$

Testing Set

3. Use the model to make predictions on the testing set and measure the MSE - this is know as the test MSE.

	x_1	x_2	x_3	y
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Training Set

Use this training data to build some model:
$y = 3.5(x_1) - 9.3(x_2) + 2.1(x_3)$

Testing Set

Use model to make predictions about *y* on this test data, then calculate the test MSE to measure how close the predictions were to the actual data

The test MSE gives us an idea of how well a model will perform on data it hasn't previously seen, i.e. data that wasn't used to "train" the model.

However, the drawback of using only one testing set is that the test MSE can vary greatly depending on which observations were used in the training and testing sets.

It's possible that if we use a different set of observations for the training set and the testing set that

our test MSE could turn out to be much larger or smaller.

One way to avoid this problem is to fit a model several times using a different training and testing set each time, then calculating the test MSE to be the average of all of the test MSE's.

This general method is known as cross-validation and a specific form of it is known as leave-one-out cross-validation.

Leave-One-Out Cross Validation

Leave-one-out cross-validation uses the following approach to evaluate a model:

1. Split a dataset into a training set and a testing set, using all but one observation as part of the training set:

	x_1	x_2	x_3	y	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					**Training Set**
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					**Testing Set**

Note that we only leave one observation "out" from the training set. This is where the method gets the name "leave-one-out" cross-validation.

2. Build the model using only data from the training set.

	x_1	x_2	x_3	y
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Training Set

Use this training data to build some model:

$y = 3.5(x_1) - 9.3(x_2) + 2.1(x_3)$

Testing Set

3. Use the model to predict the response value of the one observation left out of the model and calculate the MSE.

	x_1	x_2	x_3	y
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Training Set

Use this training data to build some model:
$y = 3.5(x_1) - 9.3(x_2) + 2.1(x_3)$

Use model to predict the response value of the one observation left out of the model and calculate the MSE.

4. Repeat the process *n* times.

Lastly, we repeat this process *n* times (where *n* is the total number of observations in the dataset), leaving out a different observation from the training set each time.

We then calculate the test MSE to be the average of all of the test MSE's:

Test MSE = (1/n)*ΣMSEi

where:

n: The total number of observations in the datasetMSEi: The test MSE during the ith time of fitting the model.

Pros & Cons of LOOCV

Leave-one-out cross-validation offers the following pros:

It provides a much less biased measure of test MSE compared to using a single test set because we repeatedly fit a model to a dataset that contains *n-1* observations.It tends not to overestimate the test MSE compared to using a single test set.

However, leave-one-out cross-validation comes with the following cons:

It can be a time-consuming process to use when *n* is large.It can also be time-consuming if a model is particularly complex and takes a long time to fit to a dataset.It can be computationally expensive.

Fortunately, modern computing has become so efficient in most fields that LOOCV is a much more reasonable

method to use compared to many years ago.

Note that LOOCV can be used in both regression and classification settings as well. For regression problems, it calculates the test MSE to be the mean squared difference between predictions and observations while in classification problems it calculates the test MSE to be the percentage of observations correctly classified during the *n* repeated model fittings.

How to Perform LOOCV in R & Python

The following tutorials provide step-by-step examples of how to perform LOOCV for a given model in R and Python:

Leave-One-Out Cross-Validation in R
Leave-One-Out Cross-Validation in Python