Cross-validation (statistics)

From Wikipedia, the free encyclopedia

Cross-validation, sometimes called **rotation estimation**,[1][2][3] is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. It is worth highlighting that in a prediction problem, a model is usually given a dataset of *known data* on which training is run (*training dataset*), and a dataset of *unknown data* (or *first seen* data) against which the model is tested (*testing dataset*).[4] The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the *validation dataset*), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent data set (i.e., an unknown dataset, for instance from a real problem), etc.

One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the *training set*), and validating the analysis on the other subset (called the *validation set* or *testing set*). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

Cross-validation is important in guarding against testing hypotheses suggested by the data (called "Type III errors"[5]), especially where further samples are hazardous, costly or impossible to collect.

Contents

- 1 Purpose of cross validation
- 2 Common types of cross-validation
  - 2.1 Exhaustive cross-validation
    - 2.1.1 Leave-p-out cross-validation
    - 2.1.2 Leave-one-out cross-validation
  - 2.2 Non-exhaustive cross-validation
    - 2.2.1 *k*-fold cross-validation
    - 2.2.2 2-fold cross-validation
    - 2.2.3 Repeated random sub-sampling validation
- 3 Measures of fit
- 4 Applications
- 5 Statistical properties
- 6 Computational issues
- 7 Relationship to other forms of validation
- 8 Limitations and misuse
- 9 See also
- 10 Notes and references

Purpose of cross validation

Suppose we have a model with one or more unknown parameters, and a data set to which the model can be fit (the training data set). The fitting process optimizes the model parameters to make the model fit the training data as well as possible. If we then take an independent sample of validation data from the same population as the training data, it will generally turn out that the model does not fit the validation data as well as it fits the training data. This is called overfitting, and is particularly likely to happen when the size of the training data set is small, or when the number of parameters in the model is large. Cross-validation is a way to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available.

Linear regression provides a simple illustration of overfitting. In linear regression we have real *response values* $Y_1, ..., Y_n$, and vector *covariates* $X_1, ..., X_p$. We can use least squares to fit a hyperplane $a + b_1X_1 + ... + b_pX_p$ between the Y and X data, and then assess the fit using the mean squared error (MSE):

$$\frac{\sum_i (Y_i - a - b_1X_{1i} - \cdots - b_pX_{pi})^2}{n}$$

where X_{ji} is the value of variable X_j corresponding to the i^{th} response value Y_i.

It can be shown under mild assumptions that the expected value of the MSE for the training set is $(n - p - 1)/(n + p + 1) < 1$ times the expected value of the MSE for the validation set (the expected value is taken over the distribution of training sets). Thus if we fit the model and compute the MSE on the training set, we will get an optimistically biased assessment of how well the model will fit an independent data set. This biased estimate is called the *in-sample* estimate of the fit, whereas the cross-validation estimate is an *out-of-sample* estimate.

Since in linear regression it is possible to directly compute the factor $(n - p - 1)/(n + p + 1)$ by which the training MSE underestimates the validation MSE, cross-validation is not practically useful in that setting (however, cross-validation remains useful in the context of linear regression in that it can be used to select an optimally regularized cost function). In most other regression procedures (e.g. logistic regression), there is no simple formula to make such an adjustment. Cross-validation is, thus, a generally applicable way to predict the performance of a model on a validation set using computation in place of mathematical analysis.

Common types of cross-validation

Two types of cross-validation can be distinguished, exhaustive and non-exhaustive cross-validation.

Exhaustive cross-validation

Exhaustive cross-validation methods are cross-validation methods which learn and test on all possible ways to divide the original sample into a training and a validation set.

Leave-p-out cross-validation

As the name suggests, leave-*p*-out cross-validation (**LpO CV**) involves using *p* observations as the validation set and the remaining observations as the training set. This is repeated on all ways to cut the original sample on a validation set of *p' observations and a training set.*

LpO cross-validation requires to learn and validate C_n^p times (where *n* is the number of observation in the original sample). So as soon as *n* is quite big it becomes impossible to calculate.

Leave-one-out cross-validation

Leave-*one*-out cross-validation (**LOOCV**) is a particular case of leave-*p*-out cross-validation with $p = 1$.

LOO cross-validation doesn't have the calculation problem of general LpO cross-validation because $C_n^1 = n$.

Non-exhaustive cross-validation

Non-exhaustive cross validation methods doesn't make the calculus on every splitting way of the original sample. Those methods are approximations of the leave-*p*-out cross-validation.

k-fold cross-validation

In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal size subsamples. Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $k - 1$ subsamples are used as training data. The cross-validation process is then repeated *k* times (the *folds*), with each of the *k* subsamples used exactly once as the validation data. The *k* results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used,[6] but in general *k* remains an unfixed parameter [1] (http://andrewgelman.com/2006/03/crossvalidation_2/).

When *k=n* (the number of observations), the *k*-fold cross-validation is exactly the leave-one-out cross-validation.

In *stratified k*-fold cross-validation, the folds are selected so that the mean response value is approximately equal in all the folds. In the case of a dichotomous classification, this means that each fold contains roughly the same proportions of the two types of class labels.

2-fold cross-validation

This is the simplest variation of *k*-fold cross-validation. Also, called holdout method.[7] For each fold, we randomly assign data points to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d_0 and test on d_1, followed by training on d_1 and testing on d_0.

This has the advantage that our training and test sets are both large, and each data point is used for both training and validation on each fold.

Repeated random sub-sampling validation

This method randomly splits the dataset into training and validation data. For each such split, the model is fit to the training data, and predictive accuracy is assessed using the validation data. The results are then averaged over the splits. The advantage of this method (over *k*-fold cross validation) is that the proportion of the training/validation split is not dependent on the number of iterations (folds). The disadvantage of this method is that some observations may never be selected in the validation subsample, whereas others may be selected more than once. In other words, validation subsets may overlap. This method also exhibits Monte Carlo variation, meaning that the results will vary if the analysis is repeated with different random splits.

When the number of random splits goes to infinity, the Repeated random sub-sampling validation become arbitrary close to the leave-p-out cross-validation.

In a stratified variant of this approach, the random samples are generated in such a way that the mean response value (i.e. the dependent variable in the regression) is equal in the training and testing sets. This is particularly useful if the responses are dichotomous with an unbalanced representation of the two response values in the data.

Measures of fit

The goal of cross-validation is to estimate the expected level of fit of a model to a data set that is independent of the data that were used to train the model. It can be used to estimate any quantitative measure of fit that is appropriate for the data and model. For example, for binary classification problems, each case in the validation set is either predicted correctly or incorrectly. In this situation the misclassification error rate can be used to summarize the fit, although other measures like positive predictive value could also be used. When the value being predicted is continuously distributed, the mean squared error, root mean squared error or median absolute deviation could be used to summarize the errors.

Applications

Cross-validation can be used to compare the performances of different predictive modeling procedures. For example, suppose we are interested in optical character recognition, and we are considering using either support vector machines (SVM) or k nearest neighbors (KNN) to predict the true character from an image of a handwritten character. Using cross-validation, we could objectively compare these two methods in terms of their respective fractions of misclassified characters. If we simply compared the methods based on their in-sample error rates, the KNN method would likely appear to perform better, since it is more flexible and hence more prone to overfitting compared to the SVM method.

Cross-validation can also be used in *variable selection*.[8] Suppose we are using the expression levels of 20 proteins to predict whether a cancer patient will respond to a drug. A practical goal would be to determine which subset of the 20 features should be used to produce the best predictive model. For most modeling procedures, if we compare feature subsets using the in-sample error rates, the best performance will occur when all 20 features are used. However under cross-validation, the model with the best fit will generally include only a subset of the features that are deemed truly informative.

Statistical properties

Suppose we choose a measure of fit F, and use cross-validation to produce an estimate F^* of the expected fit EF of a model to an independent data set drawn from the same population as the training data. If we imagine sampling multiple independent training sets following the same distribution, the resulting values for F^* will vary. The statistical properties of F^* result from this variation.

The cross-validation estimator F^* is very nearly unbiased for EF. The reason that it is slightly biased is that the training set in cross-validation is slightly smaller than the actual data set (e.g. for LOOCV the training set size is $n - 1$ when there are n observed cases). In nearly all situations, the effect of this bias will be conservative in that the estimated fit will be slightly biased in the direction suggesting a poorer fit. In practice, this bias is rarely a concern.

The variance of F^* can be large.[9][10] For this reason, if two statistical procedures are compared based on the results of cross-validation, it is important to note that the procedure with the better estimated performance may not actually be the better of the two procedures (i.e. it may not have the better value of EF). Some progress has been made on constructing confidence intervals around cross-validation estimates,[9] but this is considered a difficult problem.

Computational issues

Most forms of cross-validation are straightforward to implement as long as an implementation of the prediction method being studied is available. In particular, the prediction method need only be available as a "black box" – there is no need to have access to the internals of its implementation. If the prediction method is expensive to train, cross-validation can be very slow since the training must be carried out repeatedly. In some cases such as least squares and kernel regression, cross-validation can be sped up significantly by pre-computing certain values that are needed repeatedly in the training, or by using fast "updating rules" such as the Sherman–Morrison formula. However one must be careful to preserve the "total blinding" of the validation set from the training procedure, otherwise bias may result. An extreme example of accelerating cross-validation occurs in linear regression, where the results of cross-validation have a closed-form expression known as the *prediction residual error sum of squares* (PRESS).

Relationship to other forms of validation

In "true validation," or "holdout validation," a subset of observations is chosen randomly from the initial sample to form a validation or testing set, and the remaining observations are retained as the training data. Normally, less than a third of the initial sample is used for validation data.[11] This would generally not be considered to be cross-validation since only a single partition of the data into training and testing sets is used.

Limitations and misuse

Cross-validation only yields meaningful results if the validation set and training set are drawn from the same population. In many applications of predictive modeling, the structure of the system being studied evolves over time. This can introduce systematic differences between the training and validation sets. For example, if a model for predicting stock values is trained on data for a certain five-year period, it is unrealistic to treat the subsequent five-year period as a draw from the same population. As another example, suppose a model is developed to predict an individual's risk for being diagnosed with a particular disease within the next year. If the model is trained using data from a study involving only a specific population group (e.g. young people or males), but is then applied to the general population, the cross-validation results from the training set could differ greatly from the actual predictive performance.

If carried out properly, and if the validation set and training set are from the same population, cross-validation is nearly unbiased. However there are many ways that cross-validation can be misused. If it is misused and a true validation study is subsequently performed, the prediction errors in the true validation are likely to be much worse than would be expected based on the results of cross-validation.

These are some ways that cross-validation can be misused:

- By performing an initial analysis to identify the most informative features using the entire data set – if feature selection or model tuning is required by the modeling procedure, this must be repeated on every training set. If cross-validation is used to decide which features to use, an *inner cross-validation* to carry out the feature selection on every training set must be performed.

- By allowing some of the training data to also be included in the test set – this can happen due to "twinning" in the data set, whereby some exactly identical or nearly identical samples are present in the data set.

It should be noted that some statisticians have questioned the usefulness of validation samples.[12]

See also

- Boosting (machine learning)
- Bootstrap aggregating (bagging)
- Bootstrapping (statistics)
- Resampling (statistics)

Notes and references

1. ^ Geisser, Seymour (1993). *Predictive Inference*. New York, NY: Chapman and Hall. ISBN 0-412-03471-9.
2. ^ Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (San Mateo, CA: Morgan Kaufmann) **2** (12): 1137–1143. CiteSeerX: 10.1.1.48.529 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529).
3. ^ Devijver, Pierre A.; Kittler, Josef (1982). *Pattern Recognition: A Statistical Approach*. London, GB: Prentice-Hall.
4. ^ "Newbie question: Confused about train, validation and test data!" (http://www.heatonresearch.com/node/1823). Retrieved 2013-11-14.
5. ^ Mosteller, Frederick (1948). "A k-sample slippage test for an extreme population". *Annals of Mathematical Statistics* **19** (1): 58–65. doi:10.1214/aoms/1177730290 (http://dx.doi.org/10.1214%2Faoms%2F1177730290). JSTOR 2236056 (https://www.jstor.org/stable/2236056). MR 0024116 (https://www.ams.org/mathscinet-getitem?mr=0024116).
6. ^ McLachlan, Geoffrey J.; Do, Kim-Anh; Ambroise, Christophe (2004). *Analyzing microarray gene expression data*. Wiley.
7. ^ "Cross Validation" (http://www.cs.cmu.edu/~schneide/tut5/node42.html). Retrieved 11 November 2012.
8. ^ Picard, Richard; Cook, Dennis (1984). "Cross-Validation of Regression Models". *Journal of the American Statistical Association* **79** (387): 575–583. doi:10.2307/2288403 (http://dx.doi.org/10.2307%2F2288403). JSTOR 2288403 (https://www.jstor.org/stable/2288403).
9. ^ ***a b*** Efron, Bradley; Tibshirani, Robert (1997). "Improvements on cross-validation: The .632 + Bootstrap Method". *Journal of the American Statistical Association* **92** (438): 548–560. doi:10.2307/2965703 (http://dx.doi.org/10.2307%2F2965703). JSTOR 2965703 (https://www.jstor.org/stable/2965703). MR 1467848 (https://www.ams.org/mathscinet-getitem?mr=1467848).
10. ^ Stone, Mervyn (1977). "Asymptotics for and against cross-validation". *Biometrika* **64** (1): 29–35.

doi:10.1093/biomet/64.1.29 (http://dx.doi.org/10.1093%2Fbiomet%2F64.1.29). JSTOR 2335766 (https://www.jstor.org/stable/2335766). MR 0474601 (https://www.ams.org/mathscinet-getitem?mr=0474601).

11. ^ "Tutorial 12" (http://web.archive.org/web/20060623055814/http://decisiontrees.net/node/36). *Decision Trees Interactive Tutorial and Resources*. Archived from the original (http://decisiontrees.net/node/36) on 2006-06-23. Retrieved 2006-06-21.
12. ^ Hirsch, Robert (1991). "Validation Samples". *Biometrics* **47** (3): 1193–1194.

Retrieved from "http://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=613302202"

Categories: Model selection | Regression variable selection | Machine learning

- This page was last modified on 17 June 2014 at 16:28.
-