Cross-validation (statistics)

Cross-validation, sometimes called **rotation estimation**,[1][2][3] or **out-of-sample testing** is any of various similar model validation techniques 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. 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 (called the validation dataset or *testing set*).[4],[5] The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias[6] and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

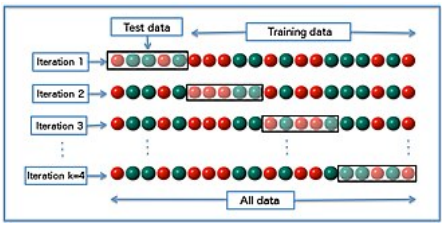

Diagram of k-fold cross-validation with k=4.

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, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance.

In summary, cross-validation combines (averages) measures of fitness in prediction to derive a more accurate estimate of model prediction performance.[7]

Contents

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. The size of this difference is likely to be large especially 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 estimate the size of this effect.

In linear regression we have real *response values* $y_1, \ldots, y_n$, and n p-dimensional vector *covariates* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. The components of the vector $\boldsymbol{x}_i$ are denoted $x_{i1}, \ldots, x_{ip}$. If we use least squares to fit a function in the form of a hyperplane $y = a + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}$ to the data $(\boldsymbol{x}_i, y_i)_{\,1 \le i \le n}$, we could then assess the fit using the mean squared error (MSE). The MSE for given estimated parameter values a and $\boldsymbol{\beta}$ on the training set $(\boldsymbol{x}_i, y_i)_{\,1 \le i \le n}$ is

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - a - \boldsymbol{\beta}^T\mathbf{x}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - a - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2$$

If the model is correctly specified, 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[8] (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 under the assumption that the model specification is valid, cross-validation can be used for checking whether the model has been overfitted, in which case the MSE in the validation set will substantially exceed its anticipated value. (Cross-validation in the context of linear regression is also useful 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 compute the expected out-of-sample fit. Cross-validation is, thus, a generally applicable way to predict the performance of a model on unavailable data using numerical computation in place of theoretical 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

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 training and validating the model C_p^n times, where n is the number of observations in the original sample, and where C_p^n is the binomial coefficient. For $p > 1$ and for even moderately large n, LpO CV can become computationally infeasible. For example, with $n = 100$ and $p = 30 = 30$ percent of 100 (as suggested above), $C_{30}^{100} \approx 3 \times 10^{25}$.

Leave-one-out cross-validation

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

The process looks similar to jackknife; however, with cross-validation one computes a statistic on the left-out sample(s), while with jackknifing one computes a statistic from the kept samples only.

LOO cross-validation requires less computation time than LpO cross-validation because there are only $C_1^n = n$ passes rather than C_k^n. However, n passes may still require quite a large computation time, in which case other approaches such as k-fold cross validation may be more appropriate.

Pseudo-Code-Algorithm:

Input:

x, {vector of length N with x-values of data points}

y, {vector of length N with y-values of data points}

Output:

err, {estimate for the prediction error}

Steps:

err ← 0

for i ← 1, . . . , N do

// define the cross-validation subsets

x_in ← (x[1], . . . , x[i − 1], x[i + 1], . . . , x[N])

y_in ← (y[1], . . . , y[i − 1], y[i + 1], . . . , y[N]

x_out ← x[i]

y_out ← interpolate(x_in, y_in, x_out, y_out)

err ← err + (y[i] − y_out)^2

end for

err ← err/N

Non-exhaustive cross-validation

Non-exhaustive cross validation methods do not compute all ways of splitting the original sample. Those methods are approximations of leave-*p*-out cross-validation.

k-fold cross-validation

In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal sized 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, with each of the *k* subsamples used exactly once as the validation data. The *k* results can then be averaged 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,[9] but in general *k* remains an unfixed parameter.

For example, setting $k = 2$ results in 2-fold cross-validation. In 2-fold cross-validation, we randomly shuffle the dataset into 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 validate on d_1, followed by training on d_1 and validating on d_0.

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 binary classification, this means that each fold contains roughly the same proportions of the two types of class labels.

Holdout method

In the holdout method, we randomly assign data points to two sets d_0 and d_1, usually called the training set and the test set, respectively. The size of each of the sets is arbitrary although typically the test set is smaller than the training set. We then train (build a model) on d_0 and test (evaluate its performance) on d_1.

In typical cross-validation, results of multiple runs of model-testing are averaged together; in contrast, the holdout method, in isolation, involves a single run. It should be used with caution because without such averaging of multiple runs, one may achieve highly misleading results. One's indicator of predictive accuracy (F*), as noted below, will tend to be unstable since it will not be smoothed out by multiple iterations. Similarly, indicators of the specific role played by various predictor variables (e.g., values of regression coefficients) will tend to be unstable.

While the holdout method can be framed as "the simplest kind of cross-validation",[10] many sources instead classify holdout as a type of simple validation, rather than a simple or degenerate form of cross-validation.[2][11]

Repeated random sub-sampling validation

This method, also known as Monte Carlo cross-validation,[12] 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.

As the number of random splits approaches infinity, the result of repeated random sub-sampling validation tends towards that of 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.

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 [13]. 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.[14][15] 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,[14] 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 can be 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).

Limitations and misuse

Cross-validation only yields meaningful results if the validation set and training set are drawn from the same population and only if human biases are controlled.

In many applications of predictive modeling, the structure of the system being studied evolves over time (i.e. it is "non-stationary"). Both of these 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.

In many applications, models also may be incorrectly specified and vary as a function of modeler biases and/or arbitrary choices. When this occurs, there may be an illusion that the system changes in external samples, whereas the reason is that the model has missed a critical predictor and/or included a confounded predictor. New evidence is that cross-validation by itself is not very predictive of external validity, whereas a form of experimental validation known as swap sampling that does control for human bias can be much more predictive of external validity.[16] As defined by this large MAQC-II study across 30,000 models, swap sampling incorporates cross-validation in the sense that predictions are tested across independent training and validation samples. Yet, models are also developed across these independent samples and by modelers who are blinded to one another. When there is a mismatch in these models developed across these swapped training and validation samples as happens quite frequently, MAQC-II shows that this will be much more predictive of poor external predictive validity than traditional cross-validation.

The reason for the success of the swapped sampling is a built-in control for human biases in model building. In addition to placing too much faith in predictions that may vary across modelers and lead to poor external validity due to these confounding modeler effects, these are some other 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. Otherwise, predictions will certainly be upwardly biased.[17] 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.[18]
- 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. Note that to some extent twinning always takes place even in perfectly independent training and validation samples. This is because some of the training sample observations will have nearly identical values of predictors as validation sample observations. And some of these will correlate with a target at better than chance levels in the same direction in both training and validation when they are actually driven by confounded predictors with poor external validity. If such a cross-validated model is selected from a *k*-fold set, human confirmation bias will be at work and determine that such a model has been validated. This is why traditional cross-validation needs to be supplemented with controls for human bias and confounded model specification like swap sampling and prospective studies.

Cross validation for time-series models

Since the order of the data is important, cross-validation might be problematic for time-series models. A more appropriate approach might be to use forward chaining.

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*.[19] 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.

A recent development in medical statistics is its use in meta-analysis. It forms the basis of the validation statistic, Vn which is used to test the statistical validity of meta-analysis summary estimates.[20] It has also been used in a more conventional sense in meta-analysis to estimate the likely prediction error of meta-analysis results.[21]

See also

- Boosting (machine learning)
- Bootstrap aggregating (bagging)
- Bootstrapping (statistics)
- Model selection
- Resampling (statistics)
- Stability (learning theory)
- Validity (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 (https://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. "What is the difference between test set and validation set?" (https://stats.stackexchange.com/questions/19048/what-is-the-difference-between-test-set-and-validation-set). Retrieved 10 October 2018.
5. "Newbie question: Confused about train, validation and test data!" (https://web.archive.org/web/20150314221014/http://www.heatonresearch.com/node/1823). Archived from the original on 2015-03-14. Retrieved 2013-11-14.
6. Cawley, Gavin C.; Talbot, Nicola L. C. (2010). "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation" (http://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf) (PDF). **11**. Journal of Machine Learning Research: 2079–2107.
7. Grossman,, Robert; Seni, Giovanni; Elder, John; Agarwal, Nitin; Liu, Huan (2010). *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan & Claypool. doi:10.2200/S00240ED1V01Y200912DMK002 (https://doi.org/10.2200%2FS00240ED1V01Y200912DMK002).
8. Trippa, Lorenzo; Waldron, Levi; Huttenhower, Curtis; Parmigiani, Giovanni (March 2015). "Bayesian nonparametric cross-study validation of prediction methods" (http://projecteuclid.org/euclid.aoas/1430226098). *The Annals of Applied Statistics*. **9** (1): 402–428. arXiv:1506.00474 (https://arxiv.org/abs/1506.00474). doi:10.1214/14-AOAS798 (https://doi.org/10.1214%2F14-AOAS798). ISSN 1932-6157 (https://www.worldcat.org/issn/1932-6157).
9. McLachlan, Geoffrey J.; Do, Kim-Anh; Ambroise, Christophe (2004). *Analyzing microarray gene expression data*. Wiley.
10. "Cross Validation" (http://www.cs.cmu.edu/~schneide/tut5/node42.html). Retrieved 11 November 2012.
11. Arlot, Sylvain; Celisse, Alain (2010). "A survey of cross-validation procedures for model selection". *Statistics Surveys*. **4**: 40–79. "In brief, CV consists in averaging several hold-out estimators of the risk corresponding to different data splits."

12. Dubitzky,, Werner; Granzow, Martin; Berrar, Daniel (2007). *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media. p. 178.
13. "Thoughts on prediction and cross-validation." Ronald Christensen, Department of Mathematics and Statistics University of New Mexico, May 21, 2015. Retrieved from http://www.math.unm.edu/~fletcher/Prediction.pdf on May 31, 2017.
14. 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 (https://doi.org/10.2307%2F2965703). JSTOR 2965703 (https://www.jstor.org/stable/2965703). MR 1467848 (https://www.ams.org/mathscinet-getitem?mr=1467848).
15. Stone, Mervyn (1977). "Asymptotics for and against cross-validation". *Biometrika*. **64** (1): 29–35. doi:10.1093/biomet/64.1.29 (https://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).
16. Consortium, MAQC (2010). "The Microarray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3315840). *Nature Biotechnology*. London: Nature Publishing Group. **28**: 827–838. doi:10.1038/nbt.1665 (https://doi.org/10.1038%2Fnbt.1665). PMC 3315840 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3315840). PMID 20676074 (https://www.ncbi.nlm.nih.gov/pubmed/20676074).
17. Bermingham, Mairead L.; Pong-Wong, Ricardo; Spiliopoulou, Athina; Hayward, Caroline; Rudan, Igor; Campbell, Harry; Wright, Alan F.; Wilson, James F.; Agakov, Felix; Navarro, Pau; Haley, Chris S. (2015). "Application of high-dimensional feature selection: evaluation for genomic prediction in man" (http://www.nature.com/srep/2015/150519/srep10312/full/srep10312.html). *Sci. Rep.* **5**: 10312. Bibcode:2015NatSR...510312B (http://adsabs.harvard.edu/abs/2015NatSR...510312B). doi:10.1038/srep10312 (https://doi.org/10.1038%2Fsrep10312).
18. Varma, Sudhir; Simon, Richard (2006). "Bias in error estimation when using cross-validation for model selection" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1397873). *BMC Bioinformatics*. **7**: 91. doi:10.1186/1471-2105-7-91 (https://doi.org/10.1186%2F1471-2105-7-91). PMC 1397873 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1397873). PMID 16504092 (https://www.ncbi.nlm.nih.gov/pubmed/16504092).
19. Picard, Richard; Cook, Dennis (1984). "Cross-Validation of Regression Models". *Journal of the American Statistical Association*. **79** (387): 575–583. doi:10.2307/2288403 (https://doi.org/10.2307%2F2288403). JSTOR 2288403 (https://www.jstor.org/stable/2288403).
20. Willis BH, Riley RD (2017). "Measuring the statistical validity of summary meta-analysis and meta-regression results for use in clinical practice" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5575530/pdf/SIM-36-3283.pdf) (PDF). *Statistics in Medicine*. **36** (21): 3283–3301. doi:10.1002/sim.7372 (https://doi.org/10.1002%2Fsim.7372). PMC 5575530 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5575530). PMID 28620945 (https://www.ncbi.nlm.nih.gov/pubmed/28620945).
21. Riley RD, Ahmed I, Debray TP, Willis BH, Noordzij P, Higgins JP, Deeks JJ (2015). "Summarising and validating test accuracy results across multiple studies for use in clinical practice" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4973708/pdf/SIM-34-2081.pdf) (PDF). *Statistics in Medicine*. **34** (13): 2081–2103. doi:10.1002/sim.6471 (https://doi.org/10.1002%2Fsim.6471). PMC 4973708 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4973708). PMID 25800943 (https://www.ncbi.nlm.nih.gov/pubmed/25800943).
