

Cross-Validation

Payam Refaeilzadeh¹, Lei Tang², and Huan Liu³

¹Google Inc., Los Angeles, CA, USA

²Chief Data Scientist, Clari Inc., Sunnyvale, CA, USA

³Arizona State University, Tempe, AZ, USA

Synonyms

[Rotation estimation](#)

Definition

Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k -fold cross-validation. Other forms of cross-validation are special cases of k -fold cross-validation or involve repeated rounds of k -fold cross-validation.

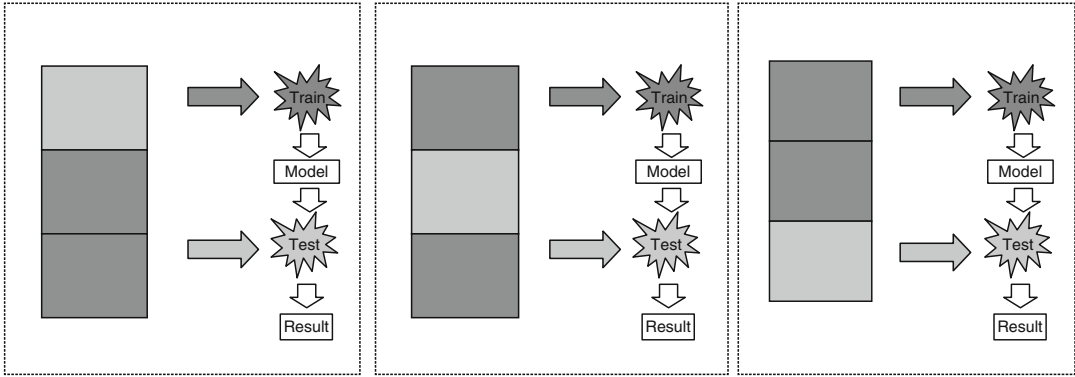
In k -fold cross-validation, the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that

within each iteration a different fold of the data is held out for validation, while the remaining $k - 1$ folds are used for learning. Figure 1 demonstrates an example with $k = 3$. The darker section of the data is used for training, while the lighter sections are used for validation. In data mining and machine learning, tenfold cross-validation ($k = 10$) is the most common.

Cross-validation is used to evaluate or compare learning algorithms as follows: in each iteration, one or more learning algorithms use $k - 1$ folds of data to learn one or more models, and subsequently the learned models are asked to make predictions about the data in the validation fold. The performance of each learning algorithm on each fold can be tracked using some predetermined performance metric-like accuracy. Upon completion, k samples of the performance metric will be available for each algorithm. Different methodologies such as averaging can be used to obtain an aggregate measure from these samples, or these samples can be used in a statistical hypothesis test to show that one algorithm is superior to another.

Historical Background

In statistics or data mining, a typical task is to learn a model from available data. Such a model may be a regression model or a classifier. The problem with evaluating such a model is that it may demonstrate adequate prediction capability on the training data but might fail to predict future



Cross-Validation, Fig. 1 Procedure of threefold cross-validation

unseen data. Cross-validation is a procedure for estimating the generalization performance in this context. The idea for cross-validation originated in the 1930s [6]. In the paper, one sample is used for regression and a second for prediction. Mosteller and Tukey [9] and various other people further developed the idea. A clear statement of cross-validation, which is similar to current version of k -fold cross-validation, first appeared in [8]. In the 1970s, both Stone [12] and Geisser [4] employed cross-validation as a means for choosing proper model parameters, as opposed to using cross-validation purely for estimating model performance. Currently, cross-validation is widely accepted in data mining and machine learning community and serves as a standard procedure for performance estimation and model selection.

Foundations

There are two possible goals in cross-validation:

- To estimate performance of the learned model from available data using one algorithm. In other words, to gauge the generalizability of an algorithm
- To compare the performance of two or more different algorithms and find out the best algorithm for the available data or alternatively to compare the performance of two or more variants of a parameterized model

The above two goals are highly related, since the second goal is automatically achieved if one knows the accurate estimates of performance. Given a sample of N data instances and a learning algorithm A , the average cross-validated accuracy of A on these N instances may be taken as an estimate for the accuracy of A on unseen data when A is trained on all N instances. Alternatively if the end goal is to compare two learning algorithms, the performance samples obtained through cross-validation can be used to perform two-sample statistical hypothesis tests, comparing a pair of learning algorithms.

Concerning these two goals, various procedures are proposed:

Resubstitution Validation

In resubstitution validation, the model is learned from all the available data and then tested on the same set of data. This validation process uses all the available data but suffers seriously from over-fitting. That is, the algorithm might perform well on the available data yet poorly on future unseen test data.

Hold-Out Validation

To avoid over-fitting, an independent test set is preferred. A natural approach is to split the available data into two non-overlapped parts: one for training and the other for testing. The test data is held out and not looked at during training. Hold-out validation avoids the overlap between training data and test data, yielding a more accu-

rate estimate for the generalization performance of the algorithm. The downside is that this procedure does not use all the available data and the results are highly dependent on the choice for the training/test split. The instances chosen for inclusion in the test set may be too easy or too difficult to classify and this can skew the results. Furthermore, the data in the test set may be valuable for training, and if it is held-out prediction, performance may suffer, again leading to skewed results. These problems can be partially addressed by repeating hold-out validation multiple times and averaging the results, but unless this repetition is performed in a systematic manner, some data may be included in the test set multiple times while others are not included at all, or conversely some data may always fall in the test set and never get a chance to contribute to the learning phase. To deal with these challenges and utilize the available data to the max, k -fold cross-validation is used.

K-Fold Cross-Validation

In k -fold cross-validation, the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held out for validation, while the remaining $k - 1$ folds are used for learning. Data is commonly stratified prior to being split into k folds. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the whole. For example in a binary classification problem where each class comprises 50 % of the data, it is best to arrange the data such that in every fold, each class comprises around half the instances.

Leave-One-Out Cross-Validation

Leave-one-out cross-validation (LOOCV) is a special case of k -fold cross-validation where k equals the number of instances in the data. In other words, in each iteration nearly all the data except for a single observation are used for training, and the model is tested on that single observation. An accuracy estimate obtained using LOOCV is known to be almost unbiased, but it

has high variance, leading to unreliable estimates [3]. It is still widely used when the available data are very rare, especially in bioinformatics where only dozens of data samples are available.

Repeated K-Fold Cross-Validation

To obtain reliable performance estimation or comparison, a large number of estimates are always preferred. In k -fold cross-validation, only k estimates are obtained. A commonly used method to increase the number of estimates is to run k -fold cross-validation multiple times. The data is reshuffled and re-stratified before each round.

Pros and Cons

Kohavi [5] compared several approaches to estimate accuracy, cross-validation (including regular cross-validation, leave-one-out cross-validation, stratified cross-validation) and bootstrap (sample with replacement), and recommended *stratified tenfold cross-validation* as the best model selection method, as it tends to provide less biased estimation of the accuracy.

Salzberg [11] studies the issue of comparing two or more learning algorithms based on a performance metric and proposes using k -fold cross-validation followed by appropriate hypothesis test rather than directly comparing the average accuracy. Paired t -test is one test which takes into consideration the variance of training and test data and is widely used in machine learning. Dietterich [2] studied the properties of tenfold cross-validation followed by a paired t -test in detail and found that such a test suffers from higher than expected type I error. In this study, this high type I error was attributed to high variance. To correct for this, Dietterich proposed a new test: fivefold \times twofold cross-validation. In this test twofold cross-validation is run five times resulting in ten accuracy values. The data is reshuffled and re-stratified after each round. All ten values are used for average accuracy estimation in the t -test, but only values from one of the five twofold cross-validation rounds are used to estimate variance. In this study fivefold \times twofold cross-validation is shown to have acceptable type I error but not to be as powerful as tenfold cross-



validation and has not been widely accepted in data mining community.

Bouckaert [1] also studies the problem of inflated type I error with tenfold cross-validation and argues that since the samples are dependent (because the training sets overlap), the actual degrees of freedom are much lower than theoretically expected. This study compared a large number of hypothesis schemes and recommend tenfold \times tenfold cross-validation to obtain 100 samples, followed with t -test with degree of freedom equal to 10 (instead of 99). However, this method has not been widely adopted in data mining field either, and tenfold cross-validation remains the most widely used validation procedure.

A brief summary of the above results is presented in Table 1.

Why Tenfold Cross-Validation: From Ideal to Reality

Whether estimating the performance of a learning algorithm or comparing two or more algorithms in terms of their ability to learn, an ideal or statistically sound experimental design must provide a *sufficiently large* number of *independent* measurements of the algorithm(s) performance.

To make independent measurements of an algorithm's performance, one must ensure that the factors affecting the measurement are independent from one run to the next. These factors are (i) the training data the algorithm learns from and (ii) the test data one uses to measure the algorithm's performance. If some data is used for testing in more than one round, the obtained results, for example, the accuracy measurements from these two rounds, will be dependent, and a statistical comparison may not be valid. In fact, it has been shown that a paired t -test based on taking several random train/test splits tends to have an extremely high probability of type I error and should never be used [2].

Not only must the datasets be independently controlled across different runs, there must not be any overlap between the data used for learning and the data used for validation in the same run. Typically, a learning algorithm can make more accurate predictions on a data that it has seen dur-

ing the learning phase than those it has not. For this reason, an overlap between the training and validation set can lead to an overestimation of the performance metric and is forbidden. To satisfy the other requirement, namely, a sufficiently large sample, most statisticians call for 30+ samples.

For a truly sound experimental design, one would have to split the available data into $30 \times 2 = 60$ partitions to perform 30 truly independent train/test runs. However, this is not practical because the performance of learning algorithms and their ranking is generally not invariant with respect to the number of samples available for learning. In other words, an estimate of accuracy in such a case would correspond to the accuracy of the learning algorithm when it learns from just 1/60 of the available data (assuming training and validation sets are of the same size). However, the accuracy of the learning algorithm on unseen data when the algorithm is trained on all the currently available data is likely much higher since learning algorithms generally improve in accuracy as more data becomes available for learning. Similarly, when comparing two algorithms A and B, even if A is discovered to be the superior algorithm when using 1/60 of the available data, there is no guarantee that it will also be the superior algorithm when using all the available data for learning. Many high-performing learning algorithms use complex models with many parameters, and they simply will not perform well with a very small amount of data. But they may be exceptional when sufficient data is available to learn from.

Recall that two factors affect the performance measure: the training set and the test set. The training set affects the measurement indirectly through the learning algorithm, whereas the composition of the test set has a direct impact on the performance measure. A reasonable experimental compromise may be to *allow for overlapping training sets, while keeping the test sets independent*. K -fold cross-validation does just that.

Now the issue becomes selecting an appropriate value for k . A large k is seemingly desirable, since with a larger k (i) there are more performance estimates and (ii) the training set size is closer to the full data size, thus increasing the

Cross-Validation, Table 1 Pros and cons of different validation methods

Validation method	Pros	Cons
Resubstitution validation	Simple	Over-fitting
Hold-out validation	Independent training and test	Reduced data for training and testing, large variance
k -fold cross-validation	Accurate performance estimation	Small samples of performance estimation, overlapped training data, elevated type I error for comparison, underestimated performance variance or overestimated degree of freedom for comparison
Leave-one-out cross-validation	Unbiased performance estimation	Very large variance
Repeated k -fold cross-validation	Large number of performance estimates	Overlapped training and test data between each round, underestimated performance variance or overestimated degree of freedom for comparison

possibility that any conclusion made about the learning algorithm(s) under test will generalize to the case where all the data is used to train the learning model. As k increases, however, the overlap between training sets also increases. For example, with fivefold cross-validation, each training set shares only 3/4 of its instances with each of the other four training sets, whereas with tenfold cross-validation, each training set shares 8/9 of its instances with each of the other nine training sets. Furthermore, increasing k shrinks the size of the test set, leading to less precise, less fine-grained measurements of the performance metric. For example, with a test set size of 10 instances, one can only measure accuracy to the nearest 10 %, whereas with 20 instances the accuracy can be measured to the nearest 5 %. These competing factors have all been considered, and the general consensus in the data mining community seems to be that $k = 10$ is a good compromise. This value of k is particularly attractive because it makes predictions using 90 % of the data, making it more likely to be generalizable to the full data.

Key Applications

Cross-validation can be applied in three contexts: performance estimation, model selection, and tuning learning model parameters.

Performance Estimation

As previously mentioned, cross-validation can be used to estimate the performance of a learning algorithm. One may be interested in obtaining an estimate for any of the many performance indicators such as accuracy, precision, recall, or F-score. Cross-validation allows for all the data to be used in obtaining an estimate. Most commonly one wishes to estimate the accuracy of a classifier in a supervised-learning environment. In such a setting, a certain amount of labeled data is available, and one wishes to predict how well a certain classifier would perform if the available data is used to train the classifier and subsequently ask it to label unseen data. Using tenfold cross-validation one repeatedly uses 90 % of the data to build a model and test its accuracy on the remaining 10 %. The resulting average accuracy is likely somewhat of an underestimate for the true accuracy when the model is trained on all data and tested on unseen data, but in most cases this estimate is reliable, particularly if the amount of labeled data is sufficiently large and if the unseen data follows the same distribution as the labeled examples.

Model Selection

Alternatively cross-validation may be used to compare a pair of learning algorithms. This may be done in the case of newly developed learning algorithms, in which case the designer may wish to compare the performance of the classifier

with some existing baseline classifier on some benchmark dataset, or it may be done in a generalized model selection setting. In generalized model selection, one has a large library of learning algorithms or classifiers to choose from and wish to select the model that will perform best for a particular dataset. In either case the basic unit of work is pair-wise comparison of learning algorithms. For generalized model selection, combining the results of many pair-wise comparisons to obtain a single *best* algorithm may be difficult, but this is beyond the scope of this entry. Researchers have shown that when comparing a pair of algorithms using cross-validation, it is best to employ proper two-sample hypothesis testing instead of directly comparing the average accuracies. Cross-validation yields k pairs of accuracy values for the two algorithms under test. It is possible to make a null hypothesis assumption that the two algorithms perform equally well and set out to gather evidence against this null hypothesis using a two-sample test. The most widely used test is the paired t -test. Alternatively the nonparametric sign test can be used.

A special case of model selection comes into play when dealing with non-classification model selection, for example, when trying to pick a feature selection [7] algorithm that will maximize a classifier's performance on a particular dataset. Refaeilzadeh et al. [10] explore this issue in detail and explain that there are in fact two variants of cross-validation in this case: performing feature selection before splitting data into folds (OUT) or performing feature selection k times inside the cross-validation loop (IN). The paper explains that there is potential for bias in both cases: With OUT, the feature selection algorithm has looked at the test set, so the accuracy estimate is likely inflated. On the other hand, with IN the feature selection algorithm is looking at less data than would be available in a real experimental setting, leading to underestimated accuracy. Experimental results confirm these hypotheses and further show that:

- In cases where the two feature selection algorithms are not statistically differentiable, IN tends to be more truthful.
- In cases where one algorithm is better than another, IN often favors one algorithm and OUT the other.

OUT can in fact be the better choice even if it demonstrates a larger bias than IN in estimating accuracy. In other words, *estimation bias is not necessarily an indication of poor pair-wise comparison*. These subtleties about the potential for bias and validity of conclusions obtained through cross-validation should always be kept in mind, particularly when the model selection task is a complicated one involving preprocessing as well as learning steps.

Tuning

Many classifiers are parameterized, and their parameters can be *tuned* to achieve the best result with a particular dataset. In most cases it is easy to learn the proper value for a parameter from the available data. Suppose a naïve Bayes classifier is being trained on a dataset with two classes: $\{+, -\}$. One of the parameters for this classifier is the prior probability $p(+)$. The best value for this parameter according to the available data can be obtained by simply counting the number of instances that are labeled positive and dividing this number by the total number of instances. However, in some cases parameters do not have such intrinsic meaning, and there is no good way to pick a best value other than trying out many values and picking the one that yields the highest performance. For example, support vector machines (SVMs) use soft margins to deal with noisy data. There is no easy way of learning the best value for the soft margin parameter for a particular dataset other than trying it out and seeing how it works. In such cases, cross-validation can be performed on the training data as to measure the performance with each value being tested. Alternatively a portion of the training set can be reserved for this purpose and not used in the rest of the learning process. But if the amount of labeled data is limited, this can significantly

degrade the performance of the learned model, and cross-validation may be the best option.

Cross-References

- [Classification](#)
- [Evaluation Metrics for Structured Text Retrieval](#)
- [Feature Selection for Clustering](#)

Recommended Reading

1. Bouckaert RR. Choosing between two learning algorithms based on calibrated tests. In: Proceedings of 20th International Conference on Machine Learning; 2003. p. 51–8.
2. Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* 1998;10(7):1895–923.
3. Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation. *J Am Stat Assoc.* 1983;78:316–31.
4. Geisser S. The predictive sample reuse method with applications. *J Am Stat Assoc.* 1975;70(350):320–8.
5. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of 14th International Joint Conference on AI; 1995. p. 1137–45.
6. Larson S. The shrinkage of the coefficient of multiple correlation. *J Educat Psychol.* 1931;22:45–55.
7. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans Knowl Data Eng.* 2005;17(4):491–502.
8. Mosteller F, Tukey JW. Data analysis, including statistics. In: Handbook of social psychology. Reading: Addison-Wesley; 1968.
9. Mosteller F, Wallace DL. Inference in an authorship problem. *J Am Stat Assoc.* 1963;58:275–309.
10. Refaeilzadeh P, Tang L, Liu H. On comparison of feature selection algorithms. In: Proceedings of AAAI-07 Workshop on Evaluation Methods in Machine Learning II; 2007. p. 34–9.
11. Salzberg S. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Min Knowl Disc.* 1997;1(3):317–28.
12. Stone M. Cross-validatory choice and assessment of statistical predictions. *J Royal Stat Soc.* 1974;36(2):111–47.
13. Tang L, Liu H. Community detection and mining in social media. Morgan & Claypool Publishers, San Rafael; 2010.
14. Zafarani R, Abbasi MA, Liu H. Social media mining: an introduction. Cambridge University Press, New York; 2014.