Title: Purged cross-validation

URL: https://en.wikipedia.org/wiki/Purged_cross-validation

PageID: 79709539

Categories: Category:Investment management, Category:Machine learning

---

Purged cross-validation is a variant of k -fold cross-validation designed to prevent look-ahead bias in time series and other structured data, developed in 2017 by Marcos López de Prado at Guggenheim Partners and Cornell University . [1] It is primarily used in financial machine learning to ensure the independence of training and testing samples when labels depend on future events. It provides an alternative to conventional cross-validation and walk-forward backtesting methods, which often yield overly optimistic performance estimates due to information leakage and overfitting. [2] [3]

Motivation

Standard cross-validation assumes that observations are independently and identically distributed (IID), which often does not hold in time series or financial datasets. If the label of a test sample overlaps in time with the features or labels in the training set, the result may be data leakage and overfitting. Purged cross-validation addresses this issue by removing overlapping observations and, optionally, adding a temporal buffer ("embargo") around the test set to further reduce the risk of leakage. [4] [3] [5] [6]

The figure below illustrates standard 5 Fold Cross-Validation [7]

Purging

Purging removes from the training set any observation whose timestamp falls within the time range of formation of a label in the test set. This can be the case for train set observations before and after the test set. Their removal ensures that the algorithm cannot learn during train time information that will be used to assess the performance of the algorithm. See the figure below for an illustration of purging. [8]

Embargoing

Embargoing addresses a more subtle form of leakage: even if an observation does not directly overlap the test set, it may still be affected by test events due to market reaction lag or downstream dependencies. To guard against this, a percentage-based embargo is imposed after each test fold. For example, with a 5% embargo and 1000 observations, the 50 observations following each test fold are excluded from training.

Unlike purging, embargoing can only occur after the test set. The figure below illustrates the application of embargo: [8]

Applications

Purged and embargoed cross-validation has been useful in:

Backtesting of trading strategies [2] [9]

Validation of classifiers on labeled event-driven returns [6] [10]

Any machine learning task with overlapping label horizons [8] [4]

Example

To illustrate the effect of purging and embargoing, consider the figures below. Both diagrams show the structure of 5-fold cross-validation over a 20-day period. In each row, blue squares indicate training samples and red squares denote test samples. Each label is defined based on the value of the next two observations, hence creating an overlap. If this overlap is left untreated, test set

information leaks into the train set.

The second figure applies the Purged CV procedure. Notice how purging removes overlapping observations from the training set and the embargo widens the gap between test and training data. This approach ensures that the evaluation more closely resembles a true out-of-sample test and reduces the risk of backtest overfitting.

Combinatorial Purged Cross-Validation

Walk-forward backtesting analysis, another common cross-validation technique in finance, preserves temporal order but evaluates the model on a single sequence of test sets. This leads to high variance in performance estimation, as results are contingent on a specific historical path. [2]

Combinatorial Purged Cross-Validation (CPCV) addresses this limitation by systematically constructing multiple train-test splits, purging overlapping samples, and enforcing an embargo period to prevent information leakage. The result is a distribution of out-of-sample performance estimates, enabling robust statistical inference and more realistic assessment of a model's predictive power. [8]

Methodology

CPCV divides a time-series dataset into N sequential, non-overlapping groups. These groups preserve the temporal order of observations. Then, all combinations of k groups (where $k < N$) are selected as test sets, with the remaining $N - k$ groups used for training. For each combination, the model is trained and evaluated under strict controls to prevent leakage. [8]

To eliminate potential contamination between training and test sets, CPCV introduces two additional mechanisms:

Purging : Any training observations whose label horizon overlaps with the test period are excluded. This ensures that future information does not influence model training.

Embargoing : After the end of each test period, a fixed number of observations (typically a small percentage) are removed from the training set. This prevents leakage due to delayed market reactions or auto-correlated features.

Each data point appears in multiple test sets across different combinations. Because test groups are drawn combinatorially, this process produces multiple backtest "paths," each of which simulates a plausible market scenario. From these paths, practitioners can compute a distribution of performance statistics such as the Sharpe ratio , drawdown , or classification accuracy.

Formal definition

Let N be the number of sequential groups into which the dataset is divided, and let k be the number of groups selected as the test set for each split. Then:

The number of unique train-test combinations is given by the binomial coefficient:

Each observation is used in k {\displaystyle k} test sets and contributes to $\phi [N , k]$ {\displaystyle \varphi [N,k]} unique backtest paths:

This yields a distribution of performance metrics rather than a single point estimate, making it possible to apply Monte Carlo-based or probabilistic techniques to assess model robustness.

Illustrative example

Consider the case where $N = 6$ and $k = 2$. The number of possible test set combinations is $(6\ 2) = 15$ {\displaystyle {\binom {6}{2}}=15} . Each of the six groups appears in five test splits. Consequently, five distinct backtest paths can be constructed, each incorporating one appearance from every group.

Test group assignment matrix

This table shows the 15 test combinations. An "x" indicates that the corresponding group is included in the test set for that split.

Backtest path assignment

Each group contributes to five different backtest paths. The number in each cell indicates the path to which the group's result is assigned for that split.

Advantages

Combinatorial Purged Cross-Validation offers several key benefits over conventional methods:

It produces a distribution of performance metrics, enabling more rigorous statistical inference.

The method systematically eliminates lookahead bias through purging and embargoing.

By simulating multiple historical scenarios, it reduces the dependence on any single market regime or realization.

It supports high-confidence comparisons between competing models or strategies.

CPCV is commonly used in quantitative strategy research, especially for evaluating predictive models such as classifiers, regressors, and portfolio optimizers. [4] It has been applied to estimate realistic Sharpe ratios, assess the risk of overfitting, and support the use of statistical tools such as the Deflated Sharpe Ratio (DSR). [10] [6]

Limitations

The main limitation of CPCV stems from its high computational cost. However, this cost can be managed by sampling a finite number of splits from the space of all possible combinations.

See also

Information leakage

Machine learning in finance

References