Cross Validation

- Cross-validation is primarily a way of measuring the predictive performance of a statistical model. Every statistician knows that the model fit statistics are not a good guide to how well a model will predict: high R^2 does not necessarily mean a good model.

- It is easy to over-fit the data by including too many degrees of freedom and so inflate R^2 and other fit statistics. For example, in a simple polynomial regression I can just keep adding higher order terms and so get better and better fits to the data. But the predictions from the model on new data will usually get worse as higher order terms are added.

- One way to measure the predictive ability of a model is to test it on a set of data not used in estimation. Data miners call this a test set and the data used for estimation is the training set. For example, the predictive accuracy of a model can be measured by the mean squared error on the test set. This will generally be larger than the MSE on the training set because the test data were not used for estimation.

- However, there is often not enough data to allow some of it to be kept back for testing. A more sophisticated version of training/test sets is ***leave-one-out cross-validation (LOOCV)*** in which the accuracy measures are obtained as follows. Suppose there are n independent observations, $y_1, \ldots, y_n$.

- Let observation i form the test set, and fit the model using the remaining data. Then compute the error ($e_i^* = y_i - \hat{y}_i$) for the omitted observation. This is sometimes called a predicted residual to distinguish it from an ordinary residual. Repeat step 1 for $i = 1, \ldots, n$. Compute the MSE from $e_1^*, \ldots, e_n^*$. We shall call this the CV.

- This is a much more efficient use of the available data, as you only omit one observation at each step. However, it can be very time consuming to implement (except for linear models see below).

- Other statistics (e.g., the MAE) can be computed similarly. A related measure is the PRESS statistic (predicted residual sum of squares) equal to $n \times MSE$.

0.1 variations

Minimizing a CV statistic is a useful way to do model selection such as choosing variables in a regression or choosing the degrees of freedom of a nonparametric smoother. It is certainly far better than procedures based on statistical tests and provides a nearly unbiased measure of the true MSE on new observations.

However, as with any variable selection procedure, it can be misused. Beware of looking at statistical tests after selecting variables using cross-validation the tests do not take account of the variable selection that has taken place and so the p-values can mislead.

It is also important to realise that it doesnt always work. For example, if there are exact duplicate observations (i.e., two or more observations with equal values for all covariates and for the y variable) then leaving one observation out will not be effective.

Another problem is that a small change in the data can cause a large change in the model selected. Many authors have found that k-fold cross-validation works better in this respect.

In a famous paper, Shao (1993) showed that leave-one-out cross validation does not lead to a consistent estimate of the model. That is, if there is a true model, then LOOCV will not always find it, even with very large sample sizes. In contrast, certain kinds of leave-k-out cross-validation, where k increases with n, will be consistent. Frankly, I dont consider this is a very important result as there is never a true model. In reality, every model is wrong, so consistency is not really an interesting property.

0.2 Cross Validation: Training and Testing Data

Cross-validation, sometimes called rotation estimation, 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).
- 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.

0.3 Cross Validation

The confusion table is a table in which the rows are the observed categories of the dependent and the columns are the predicted categories. When prediction is perfect all cases will lie on the diagonal. The percentage of cases on the diagonal is the percentage of correct classifications. The cross validated set of data is a more honest presentation of the power of the discriminant function than that provided by the original classifications and often produces a poorer outcome. The cross validation is often termed a "jack-knife" classification, in that it successively classifies **all cases but one** to develop a discriminant function and then categorizes the case that was left out. This process is repeated with each case left out in turn.This is known as leave-1-out cross validation.

This cross validation produces a more reliable function. The argument behind it is that one should not use the case you are trying to predict as part of the categorization process.

0.4 Error Rates

We can evaluate error rates by means of a training sample (to construct the discrimination rule) and a test sample.

An optimistic error rate is obtained by reclassifying the training data. (In the ***training data*** sets, how many cases were misclassified). This is known as the **apparent error rate**.

The apparent error rate is obtained by using in the training set to estimate the error rates. It can be severely optimistically biased, particularly for complex classifiers, and in the presence of over-fitted models.

If an independent test sample is used for classifying, we arrive at the **true error rate**.The true error rate (or conditional error rate) of a classifier is the expected probability of misclassifying a randomly selected pattern. It is the error rate of an infinitely large test set drawn from the same distribution as the training data.

0.5 Misclassification Cost

As in all statistical procedures it is helpful to use diagnostic procedures to asses the efficacy of the discriminant analysis. We use **cross-validation** to assess the classification probability.

Typically you are going to have some prior rule as to what is an **acceptable misclassification rate**.

Those rules might involve things like, "what is the cost of misclassification?" Consider a medical study where you might be able to diagnose cancer.

There are really two alternative costs. The cost of misclassifying someone as having cancer when they don't. This could cause a certain amount of emotional grief. Additionally there would be the substantial cost of unnecessary treatment.

There is also the alternative cost of misclassifying someone as not having cancer when in fact they do have it.

A good classification procedure should

- result in few misclassifications
- take ***prior probabilities of occurrence*** into account
- consider the cost of misclassification

For example, suppose there tend to be more financially sound firms than bankrupt firm. If we really believe that the prior probability of a financially distressed and ultimately bankrupted firm is very small, then one should classify a randomly selected firm as non-bankrupt unless the data overwhelmingly favor bankruptcy.

There are two costs associated with discriminant analysis classification: The true misclassification cost per class, and the expected misclassification cost (ECM) per observation.

Suppose there we have a binary classification system, with two classes: class 1 and class 2. Suppose that classifying a class 1 object as belonging to class 2 represents a more serious error than classifying a class 2 object as belonging to class 1. There would an assignable cost to each error. $c(i|j)$ is the cost of classifying an observation into class j if its true class is i. The costs of misclassification can be defined by a cost matrix.

	Predicted Class 1	Predicted Class 2
Class 1	0	$c(2\|1)$
Class 2	$c(1\|2)$	0

¡¡¡¡¡¡¡ HEAD

1 Training and validation

Using Validation and Test Data

In some cases you might want to use only training and test data. For example, you might decide to use an information criterion to decide what effects to include and when to terminate the selection process. In this case no validation data are required, but test data can still be useful in assessing the predictive performance of the selected model. In other cases you might decide to use validation data during the selection process but forgo assessing the selected model on test data.

1.1 Cross-Validation and Testing

- In order to build the best possible mode, we will split our training data into two parts: a training set and a test set.

- The general idea is as follows. The model parameters (the regression coefficients) are learned using the training set as above.
- The error is evaluated on the test set, and the meta-parameters are adjusted so that this cross-validation error is minimized.

1.2 Cross Validation

- The cross validation is often termed a jack-knife classification, in that it successively classifies **all cases but one** to develop a predictive model and then categorizes the case that was left out. This process is repeated with each case left out in turn.This is known as leave-1-out cross validation.
- This cross validation produces a more reliable function. The argument behind it is that one should not use the case you are trying to predict as part of the categorization process.

1.3 Cross Validation

- 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/ testing dataset*) against which testing the model is performed.
- Cross-validation is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice, with unseen data.
- The goal of cross validation is to define a dataset to "test" the model in the training phase, 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.
- Cross-validation is important in guarding against testing hypotheses suggested by the data (called "***Type III errors***"), especially where further samples are hazardous, costly or impossible to collect

K-fold Cross Validation

- In k-fold cross-validation, the original data set is randomly partitioned into k equally sized subsamples (e.g. 10 samples).
- Of the k subsamples, a single subsample is retained as the testing 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 test 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 is that all observations are used for both training and testing, and each observation is used for testing exactly once.

Leave-One-Out Cross-Validation

- As the name suggests, **leave-one-out cross-validation (LOOCV)** involves using a single observation from the original sample as the validation data, and the remaining observations as the training data.
- This is repeated such that each observation in the sample is used once as the validation data.
- This is the same as a K-fold cross-validation with K being equal to the number of observations in the original sampling, i.e. **K=n**.

Standard Data Partition
Standard Data Partition

2 Data Partitioning

Most data mining projects use large volumes of data. Before building a model, typically you partition the data using a partition utility. Partitioning yields mutually exclusive datasets: a training dataset, a validation dataset and a test dataset.

Training Set

The training dataset is used to train or build a model. For example, in a linear regression, the training dataset is used to fit the linear regression model, i.e. to compute the regression coefficients. In a neural network model, the training dataset is used to obtain the network weights.

Validation Set

Once a model is built on training data, you need to find out the accuracy of the model on unseen data. For this, the model should be used on a dataset that was not used in the training process – a dataset where you know the actual value of the target variable. The discrepancy between the actual value and the predicted value of the target variable is the error in prediction. Some form of average error (MSE of average If you were to use the training data itself to compute the accuracy of the model fit, you would get an overly optimistic estimate of the accuracy of the model. This is because the training or model fitting process ensures that the accuracy of the model for the training data is as high as possible – the model is specifically suited to the training data. To get a more realistic estimate of how the model would perform with unseen data, you need to set aside a part of the original data and not use it in the training process. This dataset is known as the validation dataset. After fitting the model on the training dataset, you should test its performance on the validation dataset.

Test Set

The validation dataset is often used to fine-tune models. For example, you might try out neural network models with various architectures and test the accuracy of each on the validation dataset to choose among the competing architectures. In such a case, when a model is finally

chosen, its accuracy with the validation dataset is still an optimistic estimate of how it would perform with unseen data. This is because the final model has come out as the winner among the competing models based on the fact that its accuracy with the validation dataset is highest. Thus, you need to set aside yet another portion of data which is used neither in training nor in validation. This set is known as the test dataset. The accuracy of the model on the test data gives a realistic estimate of the performance of the model on completely unseen data.

3 Training and validation

Using Validation and Test Data

In some cases you might want to use only training and test data. For example, you might decide to use an information criterion to decide what effects to include and when to terminate the selection process. In this case no validation data are required, but test data can still be useful in assessing the predictive performance of the selected model. In other cases you might decide to use validation data during the selection process but forgo assessing the selected model on test data.

3.1 Cross-Validation and Testing

- In order to build the best possible mode, we will split our training data into two parts: a training set and a test set.
- The general idea is as follows. The model parameters (the regression coefficients) are learned using the training set as above.
- The error is evaluated on the test set, and the meta-parameters are adjusted so that this cross-validation error is minimized.

3.2 Cross Validation

- The cross validation is often termed a ***jack-knife*** classification, in that it successively classifies **all cases but one** to develop a predictive model and then categorizes the case that was left out. This process is repeated with each case left out in turn.This is known as leave-1-out cross validation.
- This cross validation produces a more reliable function. The argument behind it is that one should not use the case you are trying to predict as part of the categorization process.

3.3 Cross Validation

- 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/ testing dataset*) against which testing the model is performed.
- Cross-validation is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice, with unseen data.
- The goal of cross validation is to define a dataset to "test" the model in the training phase, 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.
- Cross-validation is important in guarding against testing hypotheses suggested by the data (called "***Type III errors***"), especially where further samples are hazardous, costly or impossible to collect

K-fold Cross Validation

- In k-fold cross-validation, the original data set is randomly partitioned into k equally sized subsamples (e.g. 10 samples).
- Of the k subsamples, a single subsample is retained as the testing 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 test 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 is that all observations are used for both training and testing, and each observation is used for testing exactly once.

Leave-One-Out Cross-Validation

- As the name suggests, **leave-one-out cross-validation (LOOCV)** involves using a single observation from the original sample as the validation data, and the remaining observations as the training data.
- This is repeated such that each observation in the sample is used once as the validation data.
- This is the same as a K-fold cross-validation with K being equal to the number of observations in the original sampling, i.e. **K=n**.

4 Standard Data Partition

5 Data Partitioning

Most data mining projects use large volumes of data. Before building a model, typically you partition the data using a partition utility. Partitioning yields mutually exclusive datasets: a training dataset, a validation dataset and a test dataset.

Training Set

The training dataset is used to train or build a model. For example, in a linear regression, the training dataset is used to fit the linear regression model, i.e. to compute the regression coefficients. In a neural network model, the training dataset is used to obtain the network weights.

Validation Set

Once a model is built on training data, you need to find out the accuracy of the model on unseen data. For this, the model should be used on a dataset that was not used in the training process – a dataset where you know the actual value of the target variable. The discrepancy between the actual value and the predicted value of the target variable is the error in prediction. Some form of average error (MSE of average If you were to use the training data itself to compute the accuracy of the model fit, you would get an overly optimistic estimate of the accuracy of the model. This is because the training or model fitting process ensures that the accuracy of the model for the training data is as high as possible – the model is specifically suited to the training data. To get a more realistic estimate of how the model would perform with unseen data, you need to set aside a part of the original data and not use it in the training process. This dataset is known as the validation dataset. After fitting the model on the training dataset, you should test its performance on the validation dataset.

Test Set

The validation dataset is often used to fine-tune models. For example, you might try out neural network models with various architectures and test the accuracy of each on the validation dataset to choose among the competing architectures. In such a case, when a model is finally chosen, its accuracy with the validation dataset is still an optimistic estimate of how it would perform with unseen data. This is because the final model has come out as the winner among the competing models based on the fact that its accuracy with the validation dataset is highest. Thus, you need to set aside yet another portion of data which is used neither in training nor in validation. This set is known as the test dataset. The accuracy of the model on the test data gives a realistic estimate of the performance of the model on completely unseen data.

5.1 Expected cost of misclassification (ECM)

Let p_1 and p_2 be the prior probability of class 1 and class 2 respectively. Necessarily $p_1 + p_2 = 1$.

The conditional probability of classifying an object as class 1 when it is in fact from class 2 is denoted $p(1|2)$. Similarly the conditional probability of classifying an object as class 2 when it is in fact from class 1 is denoted $p(2|1)$.

$$ECM = c(2|1)p(2|1)p_1 + c(1|2)p(1|2)p_2$$

(In other words: the sum of the cost of misclassification times the (joint) probability of that misclassification.

A reasonable classification rule should have ECM as small as possible.

6 Training data sets

A training set is a set of data used in various areas of information science to discover potentially predictive relationships. Training sets are used in artificial intelligence, machine learning, genetic programming, intelligent systems, and statistics. In all these fields, a training set has much the same role and is often used in conjunction with a test set.

7 Cross Validation

Cross validation techniques for linear regression employ the use 'leave one out' re-calculations. In such procedures the regression coefficients are estimated for $n-1$ covariates, with the Q^{th} observation omitted.

Let $\hat{\beta}$ denote the least square estimate of β based upon the full set of observations, and let $\hat{\beta}^{-Q}$ denoted the estimate with the Q^{th} case excluded.

In leave-one-out cross validation, each observation is omitted in turn, and a regression model is fitted on the rest of the data. Cross validation is used to estimate the generalization error of a given model. alternatively it can be used for model selection by determining the candidate model that has the smallest generalization error.

Evidently leave-one-out cross validation has similarities with 'jackknifing', a well known statistical technique. However cross validation is used to estimate generalization error, whereas the jackknife technique is used to estimate bias.

7.1 Cross Validation: Updating standard deviation

The variance of a data set can be calculated using the following formula.

$$S^2 = \frac{\sum_{i=1}^{n}(x_i^2) - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}{n-1} \tag{1}$$

While using bivariate data, the notation Sxx and Syy shall apply to the variance of x and of y respectively. The covariance term Sxy is given by

$$Sxy = \frac{\sum_{i=1}^{n}(x_i y_i) - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n}}{n-1} \tag{2}$$

Let the observation j be omitted from the data set. The estimates for the variance identities can be updating using minor adjustments to the full sample estimates. Where (j) denotes that the jth has been omitted, these identities are

$$Sxx^{(j)} = \frac{\sum_{i=1}^{n}(x_i^2) - (x_j)^2 - \frac{((\sum_{i=1}^{n} x_i) - x_j)^2}{n-1}}{n-2} \tag{3}$$

$$Syy^{(j)} = \frac{\sum_{i=1}^{n}(y_i^2) - (y_j)^2 - \frac{((\sum_{i=1}^{n} y_i) - y_j)^2}{n-1}}{n-2} \tag{4}$$

$$Sxy^{(j)} = \frac{\sum_{i=1}^{n}(x_i y_i) - (y_j x_j) - \frac{((\sum_{i=1}^{n} x_i) - x_j)(\sum_{i=1}^{n} y_i) - y_k)}{n-1}}{n-2} \tag{5}$$

The updated estimate for the slope is therefore

$$\hat{\beta}_1^{(j)} = \frac{Sxy^{(j)}}{Sxx^{(j)}} \tag{6}$$

It is necessary to determine the mean for x and y of the remaining $n-1$ terms

$$\bar{x}^{(j)} = \frac{(\sum_{i=1}^{n} x_i) - (x_j)}{n-1}, \tag{7}$$

$$\bar{y}^{(j)} = \frac{(\sum_{i=1}^{n} y_i) - (y_j)}{n-1}. \tag{8}$$

The updated intercept estimate is therefore

$$\hat{\beta}_0^{(j)} = \bar{y}^{(j)} - \hat{\beta}_1^{(j)} \bar{x}^{(j)}. \tag{9}$$

8 Updating Estimates

8.1 Updating of Regression Estimates

Updating techniques are used in regression analysis to add or delete rows from a model, allowing the analyst the effect of the observation associated with that row. In time series problems, there will be scientific interest in the changing relationship between variables. In cases where there a single row is to be added or deleted, the procedure used is equivalent to a geometric rotation of a plane.

Updating techniques are used in regression analysis to add or delete rows from a model, allowing the analyst the effect of the observation associated with that row.

8.2 Updating Standard deviation

A simple, but useful, example of updating is the updating of the standard deviation when an observation is omitted, as practised in statistical process control analyzes. From first principles, the variance of a data set can be calculated using the following formula.

$$S^2 = \frac{\sum_{i=1}^{n}(x_i^2) - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}{n-1} \tag{10}$$

While using bivariate data, the notation Sxx and Syy shall apply hither to the variance of x and of y respectively. The covariance term Sxy is given by

$$Sxy = \frac{\sum_{i=1}^{n}(x_i y_i) - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n}}{n-1}. \tag{11}$$

8.3 Updating of Regression Estimates

Updating techniques are used in regression analysis to add or delete rows from a model, allowing the analyst the effect of the observation associated with that row. In time series problems, there will be scientific interest in the changing relationship between variables. In cases where there a single row is to be added or deleted, the procedure used is equivalent to a geometric rotation of a plane.

$$(X^T X \pm x_i x_i^T)^{-1} = \quad (X^T X)^{-1} \mp \quad \frac{(X^T X)^{-1}(x_i x_i^T (X^T X)^{-1}}{1 - x_i^T (X^T X)^{-1} x_i} \tag{12}$$

8.4 Updating Regression Estimates

Let the observation j be omitted from the data set. The estimates for the variance identities can be updating using minor adjustments to the full sample estimates. Where (j) denotes that the jth has been omitted, these identities are

$$Sxx^{(j)} = \frac{\sum_{i=1}^{n}(x_i^2) - (x_j)^2 - \frac{((\sum_{i=1}^{n} x_i) - x_j)^2}{n-1}}{n-2} \tag{13}$$

$$Syy^{(j)} = \frac{\sum_{i=1}^{n}(y_i^2) - (y_j)^2 - \frac{((\sum_{i=1}^{n} y_i) - y_j)^2}{n-1}}{n-2} \tag{14}$$

$$Sxy^{(j)} = \frac{\sum_{i=1}^{n}(x_i y_i) - (y_j x_j) - \frac{((\sum_{i=1}^{n} x_i) - x_j)(\sum_{i=1}^{n} y_i) - y_k)}{n-1}}{n-2} \tag{15}$$

The updated estimate for the slope is therefore

$$\hat{\beta}_1^{(j)} = \frac{Sxy^{(j)}}{Sxx^{(j)}} \tag{16}$$

It is necessary to determine the mean for x and y of the remaining $n-1$ terms

$$\bar{x}^{(j)} = \frac{(\sum_{i=1}^{n} x_i) - (x_j)}{n-1}, \tag{17}$$

$$\bar{y}^{(j)} = \frac{(\sum_{i=1}^{n} y_i) - (y_j)}{n-1}. \tag{18}$$

The updated intercept estimate is therefore

$$\hat{\beta}_0^{(j)} = \bar{y}^{(j)} - \hat{\beta}_1^{(j)} \bar{x}^{(j)}. \tag{19}$$

8.5 Inference on intercept and slope

$$\hat{\beta}_1 \pm t_{(\alpha, n-2)} \sqrt{\frac{S^2}{(n-1)S_x^2}} \tag{20}$$

$$\frac{\hat{\beta}_0 - \beta_0}{SE(\hat{\beta}_0)} \tag{21}$$

$$\frac{\hat{\beta}_1 - \beta_1}{SE(\hat{\beta}_0)} \tag{22}$$

8.5.1 Inference on correlation coefficient

This test of the slope is coincidentally the equivalent of a test of the correlation of the n observations of X and Y.

$$\begin{aligned} H_0 &: \rho_{XY} = 0 \\ H_A &: \rho_{XY} \neq 0 \end{aligned} \tag{23}$$