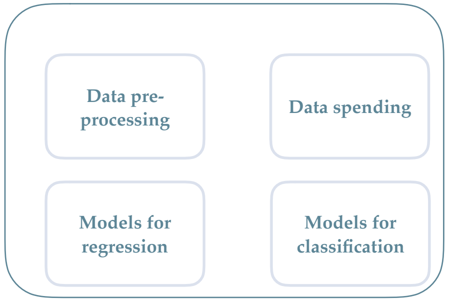
Data spending

Michela Mulas

A quick recap

During the last lectures, we did...

- Focus on **predictive modeling**.



Today's goal

Today, we going to ...

- Introduce the idea of data spending and methods for spending data in order to appropriately tune a model and assess its performance.

Today's goal

Today, we going to discuss...

over-fitting and model tuning.

Over-fitting is a concern for any predictive model regardless of field of research and it has been discussed in different fields:

- Forecasting.
- Medical research.
- Chemometrics.

Reading list

- Clark T (2004). *Can Out-of-Sample Forecast Comparisons Help Prevent Overfitting?* J Forecast, 23(2), 115-139
- Simon R, Radmacher M, Dobbin K, McShane L (2003). *Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification.* J Natl Cancer Inst, 95(1), 14-18.
- Steyerberg E (2010). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating.*

Today's goal

Today, we going to discuss ... over-fitting and model tuning.

Over-fitting is a concern for any predictive model regardless of field of research and it has been discussed in different fields:

- Chemometrics.
- Meteorology.

Reading list

Hawkins D (2004). *The Problem of Overfitting.* J Chem Inf Comput Sci, 44(1), 1-12.

Hsieh W, Tang B (1998). *Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography.* B Am Meteorol Soc, 79(9), 1855-1870.

Today's goal

Today, we going to discuss ... over-fitting and model tuning.

Over-fitting is a concern for any predictive model regardless of field of research and it has been discussed in different fields:

- Finance.
- Marital research.
- ...

Reading list

Dwyer D (2005). *Examples of Overfitting Encountered When Building Private Firm Default Prediction Models.* Technical report, Moody's KMV.

Heyman R, Slep A (2001). *The Hazards of Predicting Divorce Without Cross-validation.* J Marriage Fam, 63(2), 473.

Today's goal

Today, we going to discuss ... over-fitting and model tuning.

We describe strategies that enable us to have confidence on the model we build. Without this confidence, the model's predictions are useless.

- Define the problem of over-fitting.
- Define model tuning.
- Discuss data-splitting and resampling techniques.

Reference

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)

Today's lecture is mainly based on **Chapter 4** of the book.

The problem of over-fitting

All model building efforts are constrained by the existing data. **We must use the data at hand to find the best predictive model**.

Almost all predictive modeling techniques have tuning parameters that enable the model to flex to find the structure in the data.

We must use the existing data to identify settings for the model's parameters that yield the best and most realistic predictive performance (known as model tuning).

Traditionally, this is achieved by splitting the existing data into training and test sets.

- The **training set** is used to build and tune the model.
- The **test set** is used to estimate the model's predictive performance.

Modern approaches **split the data into multiple training and testing sets**.

- These set have been shown to often find more optimal tuning parameters and give a more accurate representation of the model's predictive performance.

The problem of over-fitting

Many techniques **learn the structure of a set of data** so well that when the model is applied to the **data on which the model was built**, it correctly predicts every sample.

In addition to learning the general patterns in the data, **the model has also learned the characteristics of each sample's unique noise**.

- This type of **model is said to be over-fit**
- Usually it have poor accuracy when predicting a new sample.

The problem of over-fitting

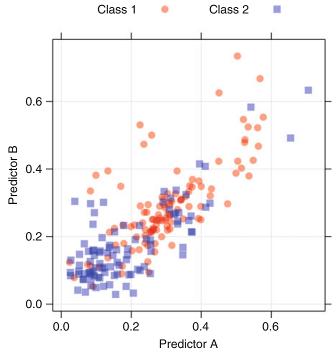
Consider the **simple classification** example with two predictor variables.

208 samples designated either as "Class 1" or "Class 2".

There are 111 samples in the first class and 97 in the second.

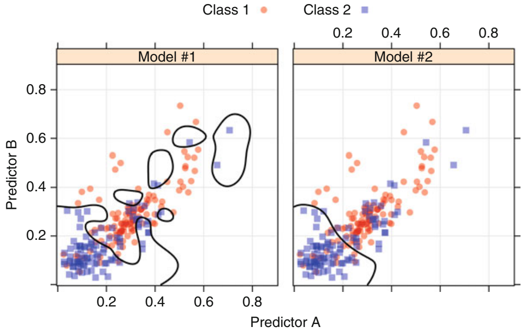
There is a significant overlap between the classes which is often the case for most applied modeling problems.

One objective for a data set such as this would be to **develop a model to classify new samples**.



The problem of over-fitting

In this 2D example, the classification models can be represented by boundary lines.

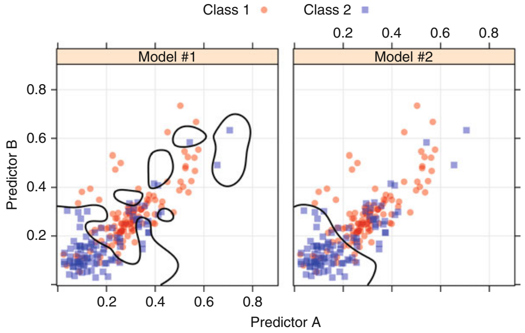


Model #1 shows a boundary that is complex and attempts to encircle every possible data point. The pattern in this panel is **not likely to generalize to new data**.

Model #2 it is an alternative model fit where the boundary is fairly smooth and does not overextend itself to correctly classify every data point in the training set.

The problem of over-fitting

In this 2D example, the classification models can be represented by boundary lines.

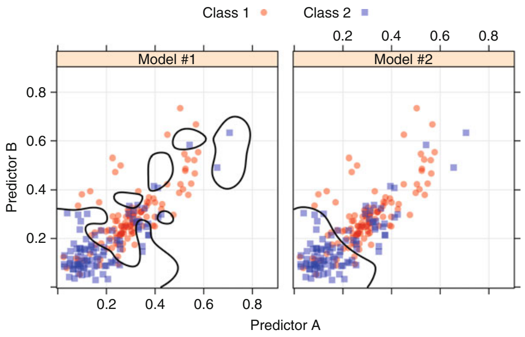


To define how well the model is classifying samples, one might use the training set.

- The estimated error rate for the Model #1would be overly optimistic.
- Estimating the utility of a model by re-predicting the training set is referred to **apparent performance** of the model (e.g., the apparent error rate).

The problem of over-fitting

In this 2D example, the classification models can be represented by boundary lines.



In two dimensions, it is not difficult to visualize that one model is over-fitting, but most modeling problems are in much higher dimensions.

In these situations, it is very important to have a tool for characterizing how much a model is over-fitting the training data.

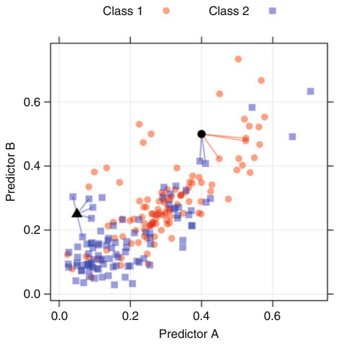
Model tuning

To understand the model tuning we start considering that many models have parameters that cannot be directly estimated from the data.

For example, in the **K-nearest neighbor classification model**, a new sample is predicted based on the K-closest data points in the training set.

The figure shows a 5-nearest neighbor model.

Here, two new samples (● and ▲) are being predicted.

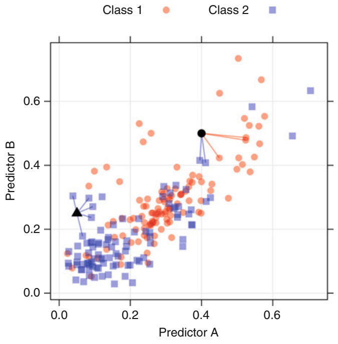


Model tuning

To understand the model tuning we start considering that many models have parameters that cannot be directly estimated from the data.

The sample ● is near a mixture of the two classes: 3 of the 5 neighbors indicate that the sample should be predicted as the first class.

The sample ▲ has 5 points indicating the second class should be predicted.



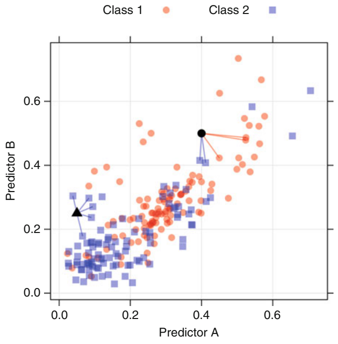
Model tuning

To understand the model tuning we start considering that many models have parameters that cannot be directly estimated from the data.

How many neighbors should be used?

- Too few neighbors may over-fit the individual points of the training set.
- Too many neighbors may not be sensitive enough to yield reasonable performance.

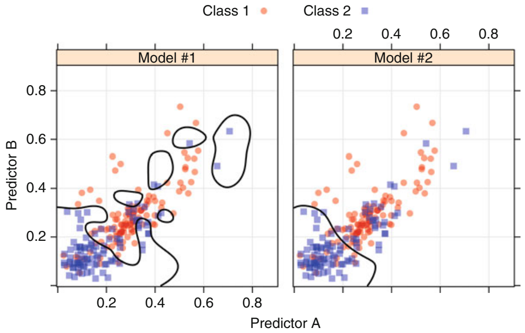
This type of model parameter is referred to as a **tuning parameter** because there is no analytical formula available to calculate an appropriate value.



Model tuning

Since many of these parameters control the complexity of the model, poor choices for the values can result in over-fitting.

Consider again the classification problem. A **support vector machine** was used to generate the class boundaries in each panel.

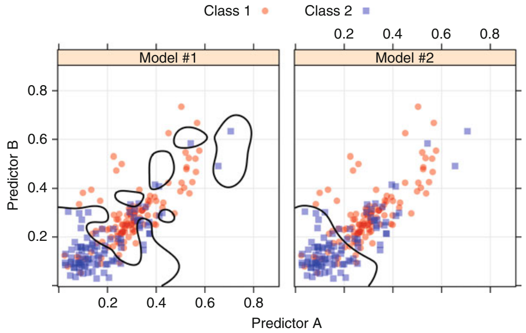


One of the tuning parameters for this model sets the price for misclassified samples in the training set and is generally referred to as the “cost” parameter.

Model tuning

Since many of these parameters control the complexity of the model, poor choices for the values can result in over-fitting.

Consider again the classification problem. A **support vector machine** was used to generate the class boundaries in each panel.

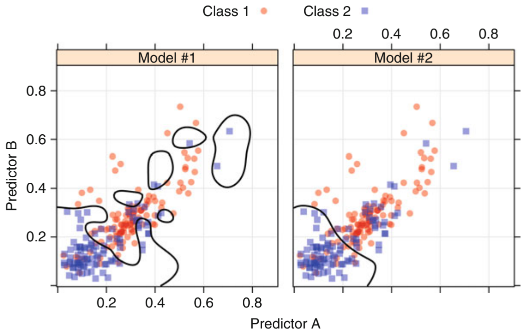


When the cost is large, the model will go to great lengths to correctly label every point (Model #1) while smaller values produce models that are not as aggressive.

Model tuning

Since many of these parameters control the complexity of the model, poor choices for the values can result in over-fitting.

Consider again the classification problem. A **support vector machine** was used to generate the class boundaries in each panel.

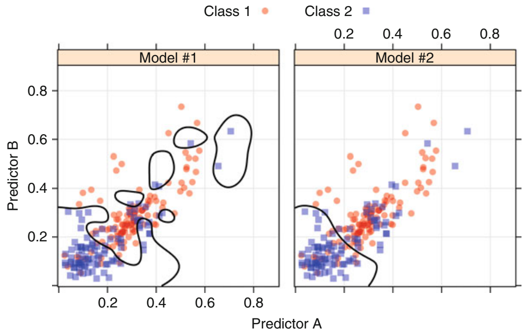


The class boundary for Model #1 was created by manually setting the cost parameter to a very high number.

Model tuning

Since many of these parameters control the complexity of the model, poor choices for the values can result in over-fitting.

Consider again the classification problem. A **support vector machine** was used to generate the class boundaries in each panel.



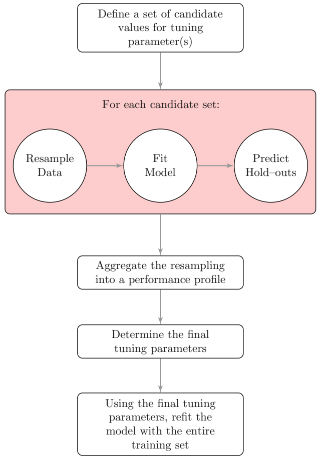
In the right panel, the cost value was determined using **cross-validation**.

Model tuning

There are different approaches for searching for the best parameters.

A general approach that can be applied to almost any model is:

- Define a set of candidate values.
- Generate reliable estimates of model utility across the candidates values
- Choose the optimal settings.



Define a set of candidate values for tuning parameter(s)

For each candidate set:

Resample Data

Fit Model

Predict Hold–outs

Aggregate the resampling into a performance profile

Determine the final tuning parameters

Using the final tuning parameters, refit the model with the entire training set

Model tuning

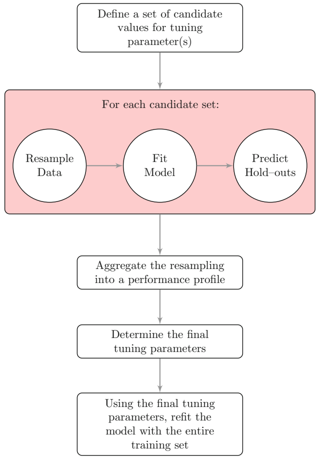
There are different approaches for searching for the best parameters.

Once a candidate set of parameter values has been selected, then we must obtain trustworthy estimates of model performance.

The performance on the hold-out samples is then aggregated into a performance profile which is then used to determine the final tuning parameters.

We then build a final model with all of the training data using the selected tuning parameters.

This procedure uses a set of candidate models that are defined by the tuning parameters.



Define a set of candidate values for tuning parameter(s)

For each candidate set:

Resample Data

Fit Model

Predict Hold–outs

Aggregate the resampling into a performance profile

Determine the final tuning parameters

Using the final tuning parameters, refit the model with the entire training set

Model tuning

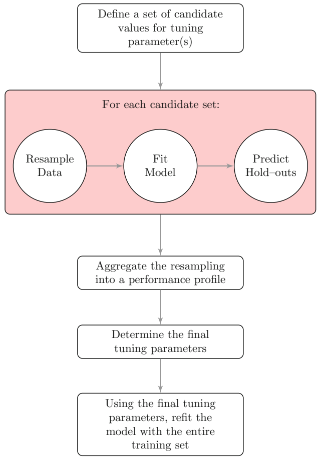
There are different approaches for searching for the best parameters.

The candidate set in the K-nearest neighbor example might include all odd values of K between 1 and 9

- Odd values are used in the two-class situation to avoid ties.

The training data would then be resampled and evaluated many times for each tuning parameter value.

These results would then be aggregated to find the optimal value of K.



Define a set of candidate values for tuning parameter(s)

For each candidate set:

Resample Data

Fit Model

Predict Hold–outs

Aggregate the resampling into a performance profile

Determine the final tuning parameters

Using the final tuning parameters, refit the model with the entire training set

Model tuning

Other approaches such as **genetic algorithms** or **simplex search methods** can also find optimal tuning parameters.

- These procedures algorithmically determine appropriate values for tuning parameters and iterate until they arrive at parameter settings with optimal performance.
- These techniques tend to evaluate a large number of candidate models and can be superior to a defined set of tuning parameters when model performance can be efficiently calculated.

Model tuning

A more difficult problem is obtaining trustworthy estimates of model performance for these candidate models.

- The apparent error rate can produce extremely optimistic performance estimates.
- A better approach is to test the model on samples that were not used for training.
- Evaluating the model on a test set is the obvious choice, but, to get reasonable precision of the performance values, the size of the test set may need to be large.

An alternate approach to evaluating a model on a single test set is to resample the training set.

This process uses several modified versions of the training set to build multiple models and then uses statistical methods to provide honest estimates of model performance (i.e., not overly optimistic).

Data splitting

A few of the **common steps in model building** are:

- Pre-processing the predictor data.
- Estimating model parameters.
- Selecting predictors for the model.
- Evaluating model performance.
- Fine tuning class prediction rules.

The modeler must decide **how to "spend" the fixed amount of data** to accommodate these activities.

One of the first decisions to make when modeling is to decide **which samples will be used to evaluate performance**.

Ideally, the model should be evaluated on samples that were not used to build or fine-tune the model, so that they provide an unbiased sense of model effectiveness.

Data splitting

A few of the **common steps in model building** are:

- Pre-processing the predictor data.
- Estimating model parameters.
- Selecting predictors for the model.
- Evaluating model performance.
- Fine tuning class prediction rules.

The modeler must decide **how to "spend" the fixed amount of data** to accommodate these activities.

When a **large amount of data** is at hand, a set of samples can be set aside to evaluate the final model.

The "training" data set is the general term for the samples used to create the model, while the "test" or "validation" data set is used to qualify performance.

Data splitting

A few of the **common steps in model building** are:

- Pre-processing the predictor data.
- Estimating model parameters.
- Selecting predictors for the model.
- Evaluating model performance.
- Fine tuning class prediction rules.

The modeler must decide **how to "spend" the fixed amount of data** to accommodate these activities.

When the a **number of samples is not large**, a strong case can be made that a test set should be avoided because every sample may be needed for model building.

Additionally, the size of the test set may not have sufficient power or precision to make reasonable judgements.

Data splitting

A few of the **common steps in model building** are:

- Pre-processing the predictor data.
- Estimating model parameters.
- Selecting predictors for the model.
- Evaluating model performance.
- Fine tuning class prediction rules.

The modeler must decide **how to “spend” the fixed amount of data** to accommodate these activities.

Resampling methods, such as cross-validation, can be used to produce appropriate estimates of model performance using the training set.

Although resampling techniques can be misapplied, they often produce performance estimates superior to a single test set because they evaluate many alternate versions of the data.

Data splitting

There are several methods for splitting the samples.

Nonrandom approaches to splitting the data are sometimes appropriate.

For example,

- If a model was being used to predict patient outcomes, the model may be created using certain patient sets (e.g., from the same clinical site or disease stage), and then tested on a different sample population to understand how well the model generalizes.
- In chemical modeling for drug discovery, new “chemical space” is constantly being explored. We are most interested in accurate predictions in the chemical space that is currently being investigated rather than the space that was evaluated years prior.
- The same could be said for spam filtering; it is more important for the model to catch the new spamming techniques rather than prior spamming schemes.

Data splitting

There are several methods for splitting the samples.

In most cases, there is the desire to make the training and test sets as homogeneous as possible.

Random sampling methods can be used to create similar data sets.

- The simplest way to split the data into a training and test set is to take a simple random sample.
- This does not control for any of the data attributes, such as the percentage of data in the classes.
- When one class has a disproportionately small frequency compared to the others, there is a chance that the distribution of the outcomes may be substantially different between the training and test sets.

Resampling techniques

Generally, resampling techniques for estimating model performance operate similarly:

- A subset of samples are used to fit a model and the remaining samples are used to estimate the efficacy of the model.
- This process is repeated multiple times and the results are aggregated and summarized.
- The differences in techniques usually center around the method in which subsamples are chosen.

Resampling techniques

k-fold cross-validation

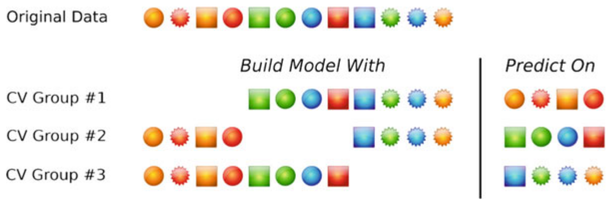
- The samples are randomly partitioned into *k* sets of roughly equal size.
- A model is fit using the all samples except the first subset (called the **first fold**).
- The held-out samples are predicted by this model and used to estimate performance measures.
- The first subset is returned to the training set and procedure repeats with the second subset held out, and so on.
- The *k* resampled estimates of performance are summarized (usually with the mean and standard error) and used to understand the relationship between the tuning parameter(s) and model utility.

Resampling techniques

k-fold cross-validation

A schematic of threefold cross-validation is shown below.

- Twelve training set samples are represented as symbols and are allocated to three groups.
- These groups are left out in turn as models are fit.

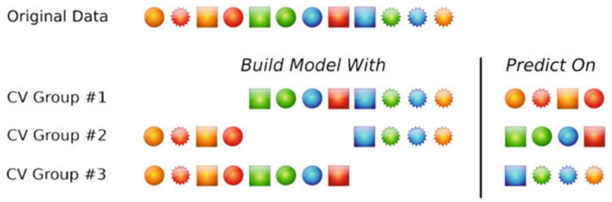


Resampling techniques

k-fold cross-validation

A schematic of threefold cross-validation is shown below.

- Performance estimates, such as the error rate or R^2 are calculated from each set of held-out samples.
- The average of the three performance estimates would be the cross-validation estimate of model performance.
- In practice, the number of samples in the held-out subsets can vary but are roughly equal size.



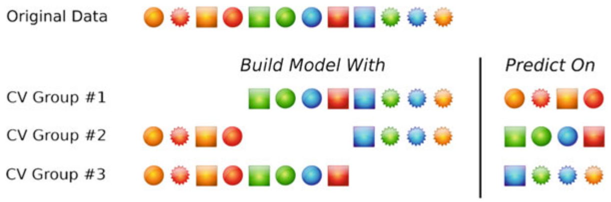
Resampling techniques

k-fold cross-validation

Leave-one-out cross-validation (LOOCV), is the special case where *k* is the number of samples.

In this case, since only one sample is held-out at a time, the final performance is calculated from the *k* individual held-out predictions.

For example, if 10-fold cross-validation was repeated five times, 50 different held-out sets would be used to estimate model efficacy.



Resampling techniques

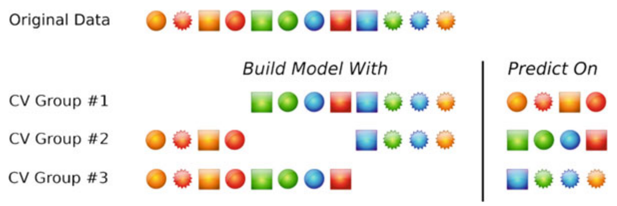
k-fold cross-validation

The choice of k is usually 5 or 10, but there is no formal rule.

As k gets larger, the difference in size between the training set and the resampling subsets gets smaller.

As this difference decreases, the bias of the technique becomes smaller (i.e., the bias is smaller for $k = 10$ than $k = 5$).

Here, the bias is the difference between the estimated and true values of performance.



Resampling techniques

k-fold cross-validation

Uncertainty (i.e., variance or noise) is an important aspect of resampling.

- An unbiased method may be estimating the correct value (e.g., the true theoretical performance) but may pay a high price in uncertainty.
- This means that repeating the resampling procedure may produce a very different value (but done enough times, it will estimate the true value).
- k-fold cross-validation generally has high variance compared to other methods and, for this reason, might not be attractive.
- It should be said that for large training sets, the potential issues with variance and bias become negligible.

Resampling techniques

k-fold cross-validation

From a practical viewpoint, **larger values of k are more computationally burdensome**.

In the extreme, LOOCV is most computationally taxing because it requires as many model fits as data points and each model fit uses a subset that is nearly the same size of the training set.

- It has been found[1] that leave-one-out and k =10-fold cross-validation yielded similar results, indicating that $k = 10$ is more attractive from the perspective of computational efficiency.

Small values of k, say 2 or 3, have high bias but are very computationally efficient.

[1] Molinaro A (2005). “Prediction Error Estimation: A Comparison of Resampling Methods”. Bioinformatics, 21(15), 3301–3307.

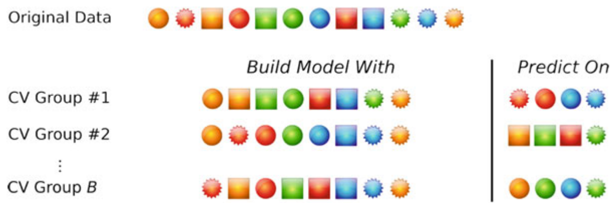
Resampling techniques

Repeated Training/Test Splits

Repeated training/test splits is also known as **leave-group-out cross-validation** or **Monte Carlo cross-validation**.

Simply we create multiple splits of the data into modeling and prediction sets.

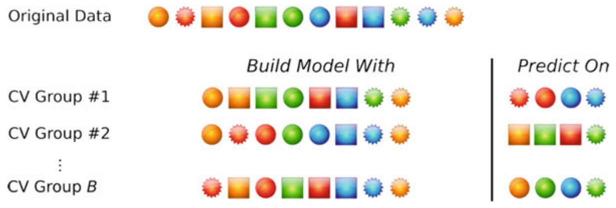
- The proportion of the data going into each subset is controlled by the practitioner as is the number of repetitions.



Resampling techniques

Repeated Training/Test Splits

- Twelve training set samples are represented as symbols and are allocated to B subsets that are 2/3 of the original training set.
- One difference between this procedure and *k*-fold cross-validation are that samples can be represented in multiple held-out subsets.
- Also, the number of repetitions is usually larger than in *k*-fold cross-validation.

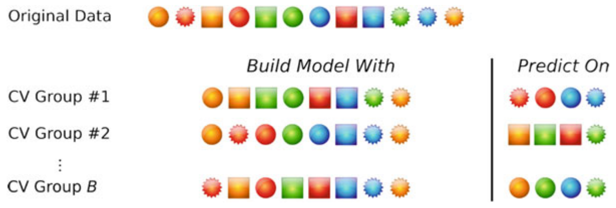


Resampling techniques

Repeated Training/Test Splits

Increasing the number of subsets has the effect of decreasing the uncertainty of the performance estimates.

- For example, to get a gross estimate of model performance, 25 repetitions will be adequate if the user is willing to accept some instability in the resulting values.
- However, to get stable estimates of performance, it is suggested to choose a larger number of repetitions (say 50-200).

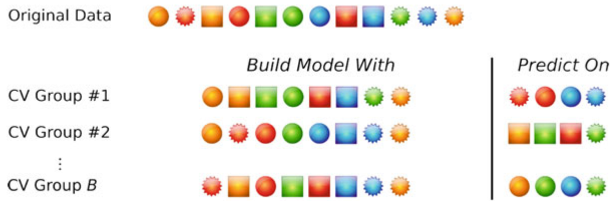


Resampling techniques

Repeated Training/Test Splits

Increasing the number of subsets has the effect of decreasing the uncertainty of the performance estimates.

- This is also a function of the proportion of samples being randomly allocated to the prediction set; the larger the percentage, the more repetitions are needed to reduce the uncertainty in the performance estimates.



Resampling techniques

The Bootstrap

A bootstrap sample is a random sample of the data taken with replacement.

- This means that, after a data point is selected for the subset, it is still available for further selection.

The bootstrap sample is the same size as the original data set.

As a result, some samples will be represented multiple times in the bootstrap sample while others will not be selected at all (“out-of-bag” samples).

For a given iteration of bootstrap resampling, a model is built on the selected samples and is used to predict the out-of-bag samples.

Resampling techniques

The Bootstrap

In general, bootstrap error rates tend to have less uncertainty than *k*-fold cross-validation.

- However, on average, 63.2% of the data points the bootstrap sample are represented at least once, so this technique has bias similar to *k*-fold cross-validation when $k \approx 2$.
- If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger.