Elements of Statistical Learning for Regression Analysis

Predictive Modeling & Statistical Learning

Gaston Sanchez

So Far ...

What's coming next

We've talked about Linear Regression—simple and multiple via OLS—from a traditional/classic perspective.

In order to continue introducing more contemporary (modernish) approaches, we need to discuss ideas like:

- modeling purposes
- model evaluation (*aka* model assessment)
- measures of model performance
- over-fitting
- learning and test sets
- resampling methods (cross-validation and bootstrapping)

Modeling ...
What for?

Modeling for what?

Goals

Understanding -vs- Prediction

Introduction

A statistical model typically aims to

Provide a certain comprehension of the data and the mechanism that generated them through a parsimonious representation of a random phenomenon.

Sometimes also, a statistical model seeks to

Predict new observations with "good" accuracy without necessary providing an explanation about the data generation/variation mechanism.

Understanding?

Understand could mean a model of a distribution for a random vector but it could also mean a regression model.

From a classic point of view, a model should be simple, and its parameters should be interpretable in terms of its domain of application (e.g. elasticity, odds-ratio, etc).

Paradox 1

Paradox 1

A “good” statistical model does not necessarily gives accurate predictions (at an individual level). E.g. risk factors in epidemiology.

Paradox 1

Paradox 1

A "good" statistical model does not necessarily gives accurate predictions (at an individual level). E.g. risk factors in epidemiology.

Paradox 2

We can predict without understanding

- no need for a theory of consumer to predict marketing target
- a model may be just simply an algorithm

Understanding vs Prediction

Models for Understanding versus Models for Prediction

Gilbert Saporta

Chaire de statistique appliquée & CEDRIC, CNAM
292 rue Saint Martin, Paris, France *saporta@cnam.fr*

Abstract. According to a standard point of view, statistical modelling consists in establishing a parsimonious representation of a random phenomenon, generally based upon the knowledge of an expert of the application field: the aim of a model is to provide a better understanding of data and of the underlying mechanism which have produced it. On the other hand, Data Mining and KDD deal with predictive modelling: models are merely algorithms and the quality of a model is assessed by its performance for predicting new observations. In this communication, we develop some general considerations about both aspects of modelling.

Model Performance

Model Performance

How do we define what a “good” model is?

- A model that fits the data well?
  (e.g. minimize resubstitution error)
- A model with optimal parameters?
  (e.g. most likely coefficients)
- A model that adequately predicts new (unseen) observations?
  (e.g. minimize generalization error)

In the Predictive Modeling arena ...

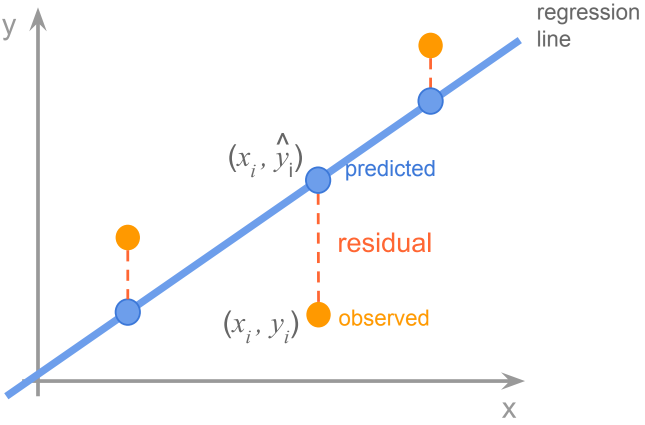
- A “good” model is one which gives accurate predictions.
- By *predictions* we mean predictions of new data.
- Therefore we focus on the generalization ability of the model to predict unobserved data
- This involves finding a measure of accuracy for predictions.

Starting Point: Residuals

- The main idea consists of comparing observed inputs y_i against predicted inputs $\hat{y}_i$.
- We need to measure the discrepancy between observed inputs and predicted inputs.
- This means our starting point involves considering residuals $e_i = y_i - \hat{y}_i$
- The summary considered up to this point is the RSS:

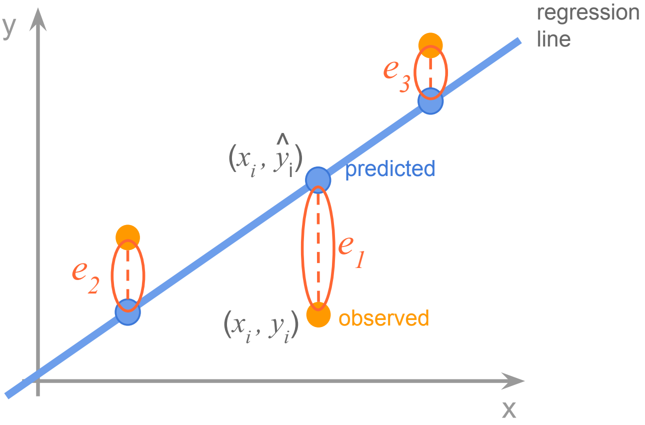
$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Looking at the amount of residuals



Measuring the overall size of the differences

Looking at the amount of residuals



How big are these differences?

Is RSS a good measure of model performance?

kind of ...

Residual Sum of Squares

One issue with the RSS is that it is a *total* value that depends on the number of residuals:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Preferably we would like to have a summary that is a "representative" measure of the **typical** error

Mean Squared Error (MSE)

A more "representative" measure of the typical error can be achieved by averaging RSS, getting what is called the Mean Squared Error (MSE)

$$MSE = \frac{1}{n} RSS = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Think of MSE as the "typical" squared error

Mean Squared Error (MSE)

- The MSE says how far typical points are above or below the regression line.
- Think of MSE as the typical size of prediction errors.
- In a general sense, MSE is a measure of scatter for residuals.
- MSE is a measure of how accurate our predictions are.

Root Mean Squared Error (RMSE)

A side effect of using squared residuals is that of having to deal with square units.

Some authors and practitioners prefer to take the square root of MSE in order to recover the original units. This produces the Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Mean Absolute Error (MAE)

MSE (and RMSE) are not the only possible options.

We can also consider the **absolute value** of residuals $|y_i - \hat{y}_i|$.
and the corresponding Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Absolute value is more robust than quadratic value because squaring may exacerbate atypical values (e.g. very large residuals).

Course Assumption

For this course, unless stated otherwise, we are going to use **MSE** *as the standard measure of model performance.*

Over-Fitting

Summary

- A “good” model is one which gives accurate predictions.
- Predictions of observed data? Predictions of unobserved data?
- Prediction of unobserved data is different from fitting observed data.
- We refer to predictive performance over new observations (generalization).

Motivation

- Suppose you have to fit a regression model.
- You decide to use all your available data to get predicted values $\hat{\mathbf{y}}$
- You use MSE to measure the predicting performance of the model.
- How reliable is the MSE value that you would obtain? (How "honest" would be the MSE?)

Over-fitting

When you use all the available data to train (fit) a model, you run the risk to obtain a model that has learned not only the systematic part of the model, but also the unique noise of the data.

This situation is called **over-fitting**. And any measure of model performance will tend to be too optimistic.

Over-fitting

- A highly accurate model may suffer from overfitting
- A very robust model (rigid) won't be able to adequately fit the data

Motivation

Evaluating the model by using the data employed to fit the model produces a resubstitution or apparent measure of error.

We are not really evaluating the model with new/unseen data.

In other words, we are not really evaluating the generalization ability of the model.

Learning Dilemma

On one hand ...

- You want to use as much data as possible to train a model.
- You want to feed your model with as many examples as possible.

Learning Dilemma

On one hand ...

- You want to use as much data as possible to train a model.
- You want to feed your model with as many examples as possible.

On the other hand ...

- You also want to know how will your model behave when new input data is available?
- How will your model generalize (when predicting unseen data)?

Dilemma

We face a BIG dilemma

- On one hand, you want to use as much data as possible to train a model.
- On the other hand, you want to have new/unseen data to evaluate the performance of your model and see how well it generalizes.

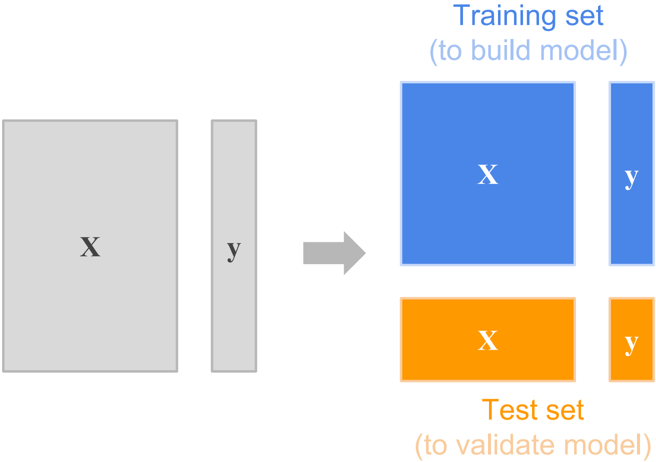
Training and Test Sets

The solution to this conundrum is to dispose of two data sets:

- **Training Dataset**: used to train the model
- **Test Dataset**: used to test the model and evaluate predicting performance

We assume that both the training and the test data sets are representative of the studied phenomenon. And that the test dataset is NOT used in the training stage.

Training and Test Sets

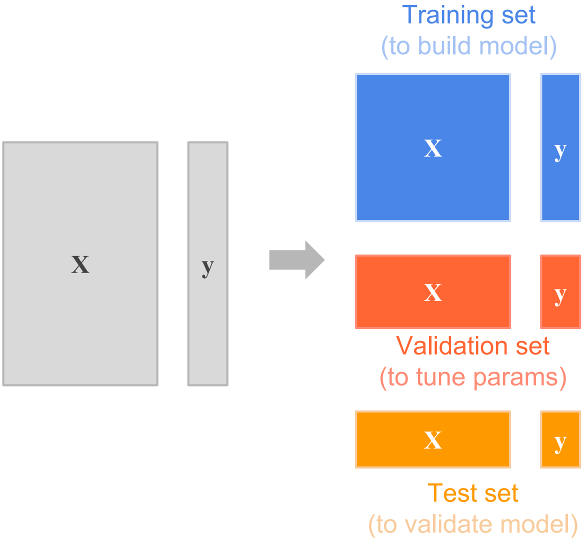


Training and Test Sets

Some authors go further and propose a three dataset scheme:

- Training set to fit the model
- Validation set to tune parameters
- Test set to assess predicting performance and select the best model

Training-Validation-Test Sets



Training and Test Sets

How do you decide what samples go into the training set, and what samples go into the test set?

Training and Test Sets

Selecting an adequate strategy to form the training and test sets obviously mainly depends on the amount of data.

With a vast amount of data, one could split the data set in halfs, one half as the training set, the other half as the test set.

Training and Test Sets

The problem is: we don't always have vast amounts of data.

Either because the size of the data is small ("few observations"), or because the quality of the data is not good, and the best samples form a small subset.

Test-MSE

In order to have an “honest” measure of performance, we calculate **MSE on test data** (the so-called *test-MSE*).

Evaluating the predicting power on unseen data will give us a better idea of the model performance than when we just use the train-MSE.

Idea: Various Test Sets

Depending on how you form your train and test sests, you may end up with sets that are not truly representative of the studied phenomenon.

So instead of using just one test set, some authors propose to use several training-test sets.

Of course, with this suggestion we go back again to the issue of the size of the data. The solution to this limitation comes from using re-sampling methods.

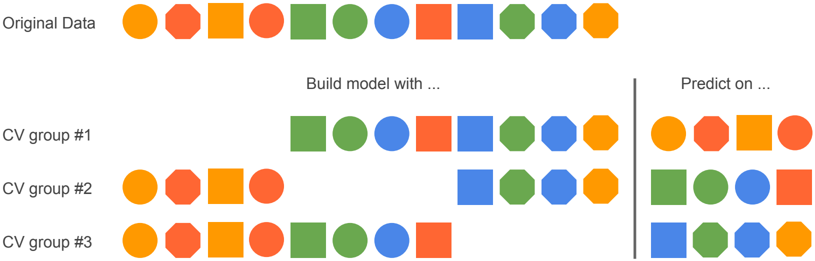
Cross-Validation

Cross-Validation (CV)

In Cross-Validation the main idea is to hold-out part of the data.
CV uses a systematic way of sampling the data.

- k-fold CV
- Leave-One-Out CV (loocv)
- 10-fold CV

Cross-validation three-fold



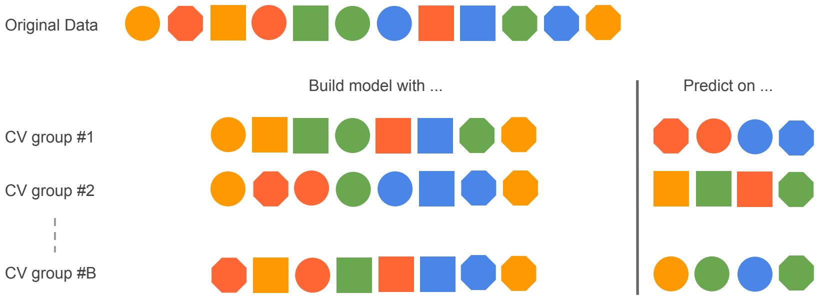
k-fold Cross-Validation

- The samples are randomly partitioned into k sets of roughly equal size.
- A model is fit using all the samples except the first subset.
- The hold-out samples are predicted by this model and used to estimate performance measures.
- The first subset is returned to the training set, and the procedure repeates with the second subset held out.
- The k resampled estimates of performance are summarized (usually with the mean and standard error).
- The choice of k is usually 5 or 10, but there is no formal rule.

Leave-One-Out Cross-Validation (loocv)

- A special version is the leave-one-out cross-validation.
- This is a special case for $k = 1$.
- One one sample is held out at a time.
- The overall performance is calculated from the k individual held out predictions.

B Repeated cross-validation



B Repeated cross-validation

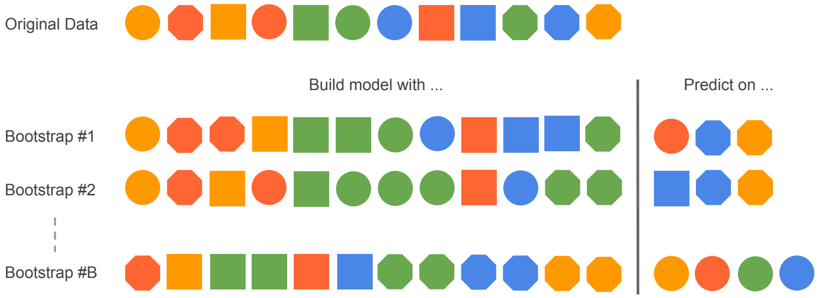
- Repeated training/set splits is also known as "leave-group-out" cross-validation.
- You simply create multiple splits of the data intro training and test sets.
- A good rule of thumb is about 75-80% training, 25-20% test.
- Increasing the number of subsets has the effect of decreasing the uncertainty of the performance estimates.
- It is suggested to choose a larger number of repetitions (say 50-200).

The Bootstrap

Bootstrap Resampling

- A bootstrap sample is a random sample of the data taken *with replacement*.
- 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 other will not be selected at all.
- The samples not selected are usually referred to as the out-of-bag samples.

Bootstrap Resampling



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

Bootstrap Resampling

- On average, 63.2% of the data points in the bootstrap sample are represented at least once.
- Bootstrap resampling has bias similar to k-fold cross-validation when $k = 2$.
- If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger.