ID5059 Practical 1 (2019)

Tom Kelsey & Carl Donovan

01 Feb 2019, due 08 March 2019 (end of week 6)

The purpose of this practical is to allow you to familiarise yourself with the tools we use and to get a feel for doing data mining and for comparison of candidate predictive models.

Data set

Get the Auto MPG data set from the UCI Machine Learning repository[1]. Familiarise yourself with it. Create some summary statistics and graphs.

Analysis

Analyse the pairwise relationships of the MPG attribute against the displacement, horsepower, weight and acceleration attributes i.e. seek to predict MPG as a function of the others. Do the following for each combination.

Construct

- linear models of varying degrees,
- bin smooth models with varying knot positions,
- a b-spline basis.

So there are 3 competing types of model for each of the 4 predictors of MPG. Evaluate the fit of these models to the data by calculating the residual sum of squares for each of them. Also calculate a measure of *generalisation* error - this can be your choice (penalised fit measures qualify).

1. use RSS to do model evaluation
2. calc some penalised measure of fitting(AIC/BIC or so)

[1] http://archive.ics.uci.edu/ml/

Resources

The file **P01-code-CS.R** contains code for loading data, producing bin-smooths, calculating linear models, and calculating nonlinear basis-function models. It also has a basic RSS function for model comparison. The code only compares MPG to displacement; you have to modify it for the other attributes. The choice of knots, basis functions and plot parameters are all illustrative: the code does not return models that are good enough to submit to a validation process. You should modify model attributes using this code as a basic working framework.

The files **P01-code-stats-functions.R** and **P01-code-stats-driver.R** contain more complicated code that does the same thing, but adds data analysis, scaling and fit estimation tools. This code probably does return models that are suitable for validation, but it's harder to verify this.

Deliverables

Upload by Friday 8th of March, on Moodle, a PDF report containing

- the code you used,
- graphs showing the linear models,
- graphs showing the bin smooths,
- graphs showing the b-spline bases,
- the residual sum of squares for each of the models.
- your measure of generalisation error.

1. Code
2. LR, bin_smooth, B–spline models and their RSS
3. Measure of generalisation error – AIC etc.
4. Check assumptions
5. Answer questions in report

SID and programme, not name

Mention any assumptions you made (e.g. with regards to missing data). Remember that submissions should only contain your matriculation number, *not* your name, but should indicate which degree you are studying for.

Additionally, answer the following questions.

✓ 1. Which attribute/predictor has the best predictive ability and why?

✓ 2. What size of bins did you choose for the bin smoothing and why? What effect would decreasing or increasing the number of bins have on the residual sum of squares?

✓ 3. What knots and degree of polynomial did you pick for the b-splines and why? 5–fold CV over 10 degrees of freedom

You should approach these questions using the background from the course you are studying for. For example:

- Maths students could disregard much of the structure of the supplied code, and focus on error rates, other definitions of error, and mathematical reasons for choosing knot positions

- CS students could disregard much of the underlying maths, and focus on working out how my bin-smooth knot positions were chosen then seeing if these positions could be used for other basis functions
- Students not studying either Maths nor CS could disregard much of how the code is structured and the maths behind error values, and focus on replication in spreadsheet software where possible, explanation & presentation of results to a lay audience.

Remember that the aim of constructing a model is to get one with a good *general* predictive ability. You can always get a very good fit to the training data if the number of bins/the degree of polynomials is large enough, but this would not generalise well to new, similar data. **don’t overfit**