ID5059 L02 - basis functions

C. Donovan

Big picture

Many apparently different methods are tied by a simple idea:

$$\mathbf{y} = f(\mathbf{X}, \theta) + \mathbf{e}$$

- We want to predict y using a set of x, but don't know the function f that connects them
- **Given** f we usually define some agreement between the model and observed y, then we can optimise parameters (θ) of f (e.g. OLS)
- But if permit lots of candidate f, which one is best?

ML methods tend to be agnostic about f (avoiding the "human intervention" often referred to)

Choosing amongst candidate models

We (statisticians/analysts) do this *a lot* already

- Add remove terms based on significance or AIC
- Stepwise or dredged regressions
- Typically the scope is not large though e.g. assume all linear terms and up to 2nd order interactions

Interpretation might be ill-advised after this, but probably gives better predictive performance.

The form of f is still quite tightly controlled.

Building complex models from simple blocks

Much in ML is made understandable by considering their building blocks - complex f from combining many simpler functions. This is familiar

- Polynomial regression
- Generalised Additive Models

Altering these building blocks can form:

- Tree models
- Ensemble models - bagged/boosted
- Thin-plate splines
- Multivariate Adaptive Regression Splines
- Support Vector Machines
- Neural networks
- ...

Polynomial regression as bases

Let's devise a simple method for varying our model complexity

- f is from the class of polynomials - represented by simple basis functions
- Increase/decrease the complexity by adding/removing bases
- Fit each to the data using some definition of good (e.g. OLS) - a **loss function**

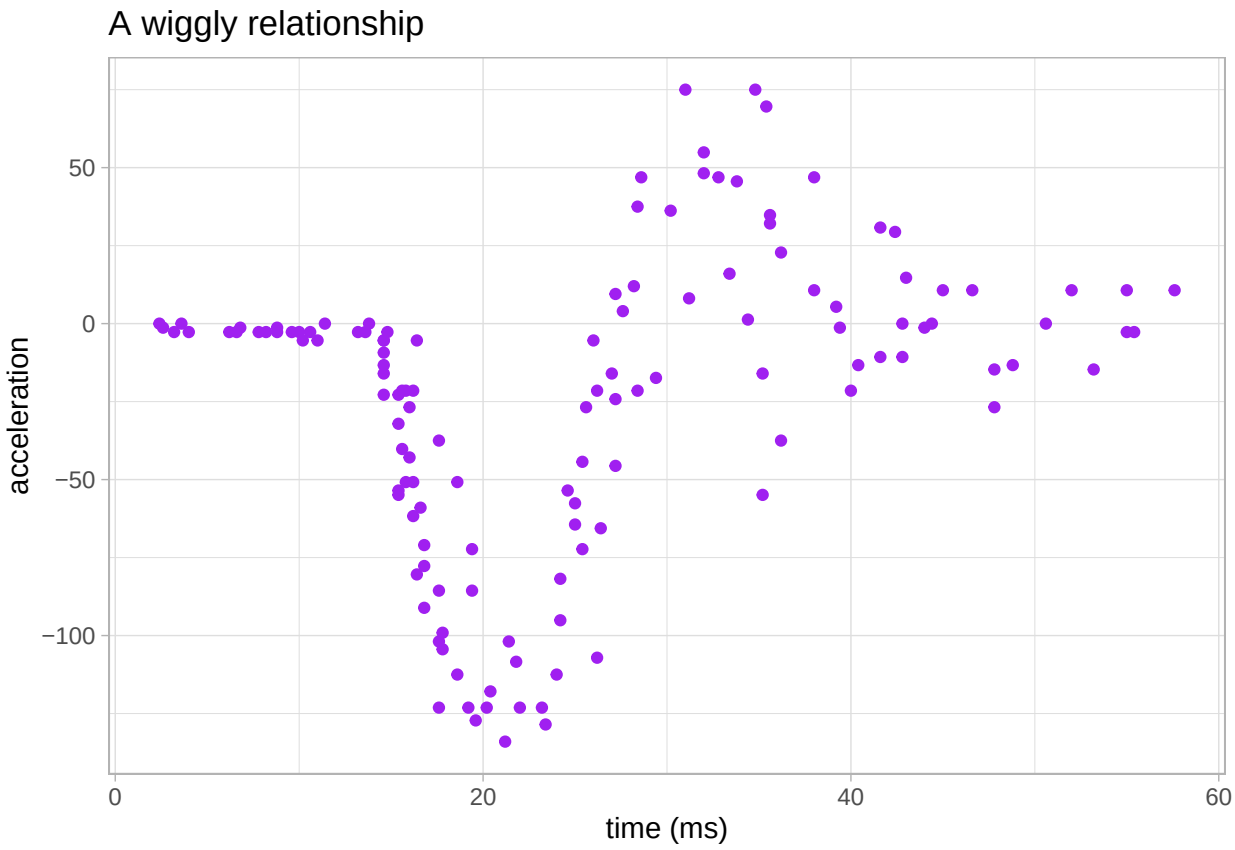
Determine which f is best? - we can't rely on our fitting criteria/loss function alone, it always likes complexity

Doing this

```
#= for pretty plots

  library(ggplot2)

#= use the mcycle data in the MASS library

  library(MASS)
  data(mcycle)

  head(mcycle)
```

```
##   times accel
## 1   2.4   0.0
## 2   2.6  -1.3
## 3   3.2  -2.7
## 4   3.6   0.0
## 5   4.0  -2.7
## 6   6.2  -2.7
```

Which looks like



A general process

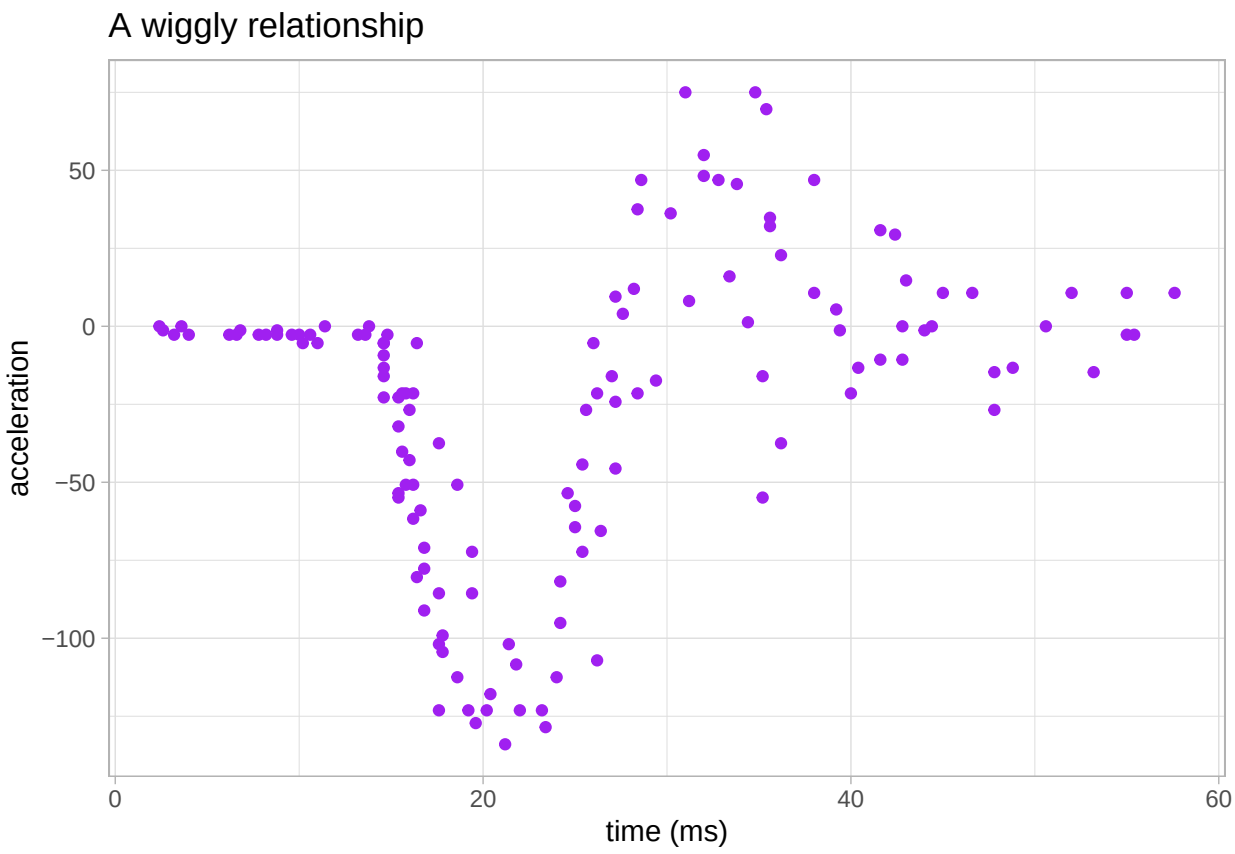
- Look at the nature of the xs and y
- Devise a process for generating candidate f
- Given f, fit to data - what is good? Basically get predicted y near observed y
- Pick the best f based on a *good* measure of predictive performance

Creating candidate f

- A common idea is to use simple building blocks to create complex models
- Have small (basis) functions that you can simply add together to increase complexity
- Polynomial regression, splines are examples
- Trees, NNs can be thought of similarly

A naive attempt

Some R....



Training error, Generalization error & Overfitting

How complex should f be? Key ideas:

- **Training error**: how much our model fails to predict the data used to develop it.
- **Generalisation error**: how much our model fails to predict data not used in model development i.e. real error.
- **Overfitting**: fitting a model that is too complicated.
- **Underfitting**: fitting a model that is too simple.

Training error, Generalization error & Overfitting

Overfitting - which can variously mean (these are different angles of the same thing) the model is:

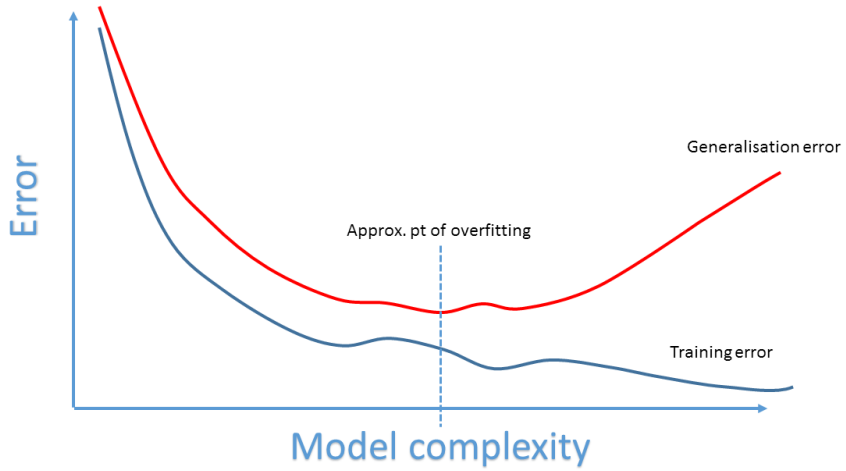
- more complicated than the underlying signal we are trying to capture.
- capturing some noise and deeming this to be signal.
- too closely approximating our sample, but not generally not so good for other samples.

from our perspective, let's consider it not being optimal in terms of generalisation error by being too complex.

- It follows there is **underfitting**: fitting a model that is 'too simple' - not being optimal in terms of generalisation error by being too simple.

Selection & Validation Summary

Commonly encounter:



Some standard measures of model fit

- R^2 & other things related to the sum-of-squared errors, or likelihood - these cannot be used directly if the model complexity is fluid.
- Things that include model complexity, penalised measures e.g. adjusted R^2, AIC (penalised likelihood).

ML tends to favour measures that simulate unseen data - very general and excellent

Model Validation

Model fit against unseen data – two approaches:

1. Hold back some of the data - data is genuinely unseen at the model derivation stage
2. Use *all* the data in a managed way - data is unseen at chosen iterations

Cross-Validation (CV)

Model fit against 'new' data

- Very important - this is a very intuitive measure which is used extensively within data-mining/ML/predictive modelling.
- A reasonable definition of a **good** model - a model that best predicts data that is as-yet unseen
- Put another way, a model that best predicts data that was *not* used in the construction of the model in question

k-fold cross validation

- Leaving each observation out in turn has been criticised for not perturbing the data enough
- A simple variant called k-fold CV was proposed - the excluded amount of data is greater than one observation at each iteration
- The data is 'folded' by the number specified e.g. a 5-fold CV would entail:

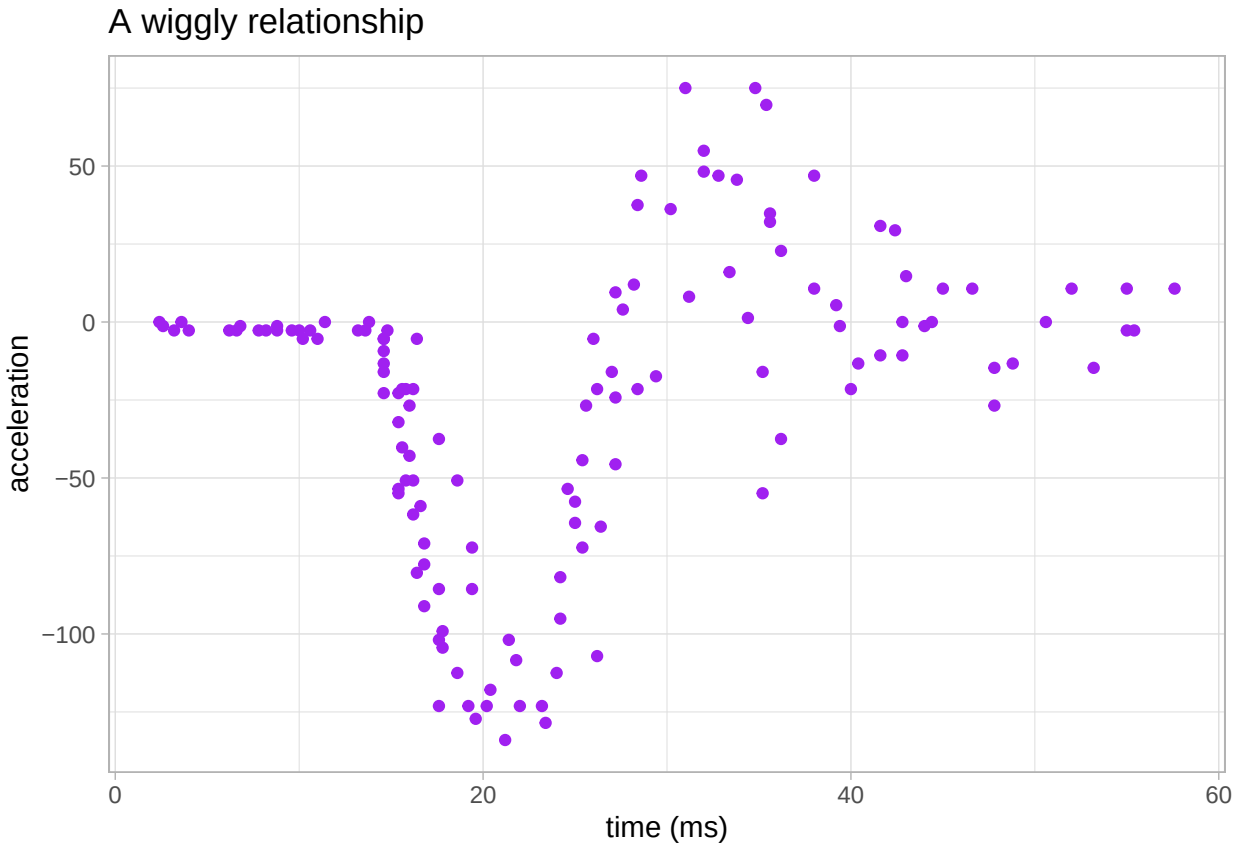
1. folding/dividing the data into 5 roughly equal portions,
2. fitting the model 5 times, omitting 20% of the data for each iteration of model fitting.
3. the subsequent results are summarised.

Out-Of-Bag (OOB)

- Bootstrapping revolves around resampling n elements from a pool of n elements **with replacement**
- Almost certain that a proportion of data not selected
- The non-sampled fraction of the data can be used as validation data
- This Out-Of-Bag (OOB) sample can be used to test our model performance

Adding to our naive attempt

Some R....



Some observations

- We usually estimate our parameters on the basis of training error (maximum likelihood, RSS, misclass error).
- **Complexity** (which may be a parameter(s)) needs to be assessed on the basis of generalisation error.
- Penalised fit measures attempt this based on the fit to the sample. Problem: how many parameters do we really have?
- Cross-validation/validation measures assess this by 'simulating' new data.
- Generalisation error is our focus, we do not know appropriate complexity *a priori*.

Keywords and Reading

- **Supervised/unsupervised learning, training error, generalisation error, bootstrapping and OOB, k-fold cross-validation, validation data, over/under-fitting, basis functions, loss function/fitting criteria/objective function**
- James *et al*: pages 289-292
- HT&F: Section 5.1
- Geron: pages 111-116, 128-134

Work through the associated L02 markdown document