Statistical Testing 3 - Model Selection

Daniel Lawson University of Bristol

Lecture 02.2.3 (v1.0.2)

Model Selection

- Imagine that we have run two different inference procedures (models) on our data.
- We want to decide which of these gives the **best** description of the data.
    - (For the moment we will pretend we want to know which one is **right**…)
- Model selection formalises how to make this assessment.

Overview

- From **Residuals**…
- Towards **Leave one out Cross Validation**…
- Via **Information Criteria**…
- To **k-Fold Cross Validation**

General considerations

- To make Cross-Validation work, we need to be able to define our inference goal cleanly. Some scenarios:
  - **Same source, single datapoint**: Within a single datastream, how well can we predict the **next** point?
  - **Same source, segment of data**: Within a single datastream, how well could we predict everything that happens within an hour?
  - **New but understood source**: We have multiple datastreams, each of which might be different but all are generated by a similar process. How well can we predict a new such datasource?
  - **Unexpected source**: We have many classes of datastream. How well can we predict what would happen on a new class of datastream?

Motivation: Residuals

- The **residual sum of squares** for n predictions of a univariate y:

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

- The expected value of the prediction error $\mathbb{E}(e^2) = R^2/n$.
- What happens if **compare two models** M_1 and M_2, where M_1 is a subset of M_2?

Linear Models - Model selection

- For illustration, consider

$$Y = \mathbf{x}_1 A_1 + \epsilon_1$$

- and

$$Y = \mathbf{x}_1 A_1 + \mathbf{x}_2 A_2 + \epsilon_2.$$

- Unless $\mathbf{x}_2 = 0$ or $\mathbf{x}_2 \equiv \mathbf{x}_1$, then ϵ_2^2 will be smaller than ϵ_1^2.
  - This is an example of a more general rule: **larger models always have better predictions**.
- So prediction error is OK to use to fit models with the same dimension, but is incomplete for **model selection**.

Cross-Validation Motivation

- Usually we are not interested in properties of **our sample**.
- We instead wish to know how our inference will generalise to **new samples**.
- The most straight forward way to predict how a model generalises is to test in **held-out data**.
- **Cross Validation** is a procedure to leave-out some data for testing.
- How much data?
  - **Leave-one-out Cross-Validation** (LOOCV) leaves out one datapoint at a time for testing.
  - **k-Fold Cross Validation** (k-fold CV) keeps a fraction $(k-1)/k$ of the data for learning parameters and $1/k$ for testing.

Prediction accuracy in linear regression

- In linear regression, the errors are

$$\mathbf{e} = \mathbf{y} - \mathrm{X}\beta = \mathbf{y} - \mathrm{H}\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$$

- Recall the H matrix describes the influence of y_i on $\hat{y}_j$, i.e. that y_i and $\hat{y}_j$ covary.
- We show in Worksheet 2.2A that the expected MSE for the i-th datapoint is:

$$\mathbb{E}(e_i^2) = \mathbb{E}\left[(y_i - \hat{y}_i)^T (y_i - \hat{y}_i)\right] = \mathbb{E}\left[(y_i - \hat{y}_i)^2\right] \tag{1}$$

$$= \mathrm{Var}[y_i] + \mathrm{Var}[\hat{y}_i] - 2\mathrm{Cov}[y_i, \hat{y}_i] + [\mathbb{E}(y_i) - \mathbb{E}(\hat{y}_i)]^2 \tag{2}$$

- This is shown by rearranging the formula for $\mathrm{Var}[y_i - \hat{y}_i]$

Out-of-sample prediction accuracy in linear regression

- We can write the same thing when predicting an **out-of-sample** y'_i:

$$\mathbb{E}(e'^2_i) = \mathbb{E}\left[(y'_i - \hat{y}_i)^T (y'_i - \hat{y}_i)\right] \tag{3}$$

$$= \quad \mathrm{Var}[y'_i] + \mathrm{Var}[\hat{y}_i] - 2\mathrm{Cov}[y'_i, \hat{y}_i] + [\mathbb{E}(y'_i) - \mathbb{E}(\hat{y}_i)]^2 \tag{4}$$

- But out-of-sample, $\mathrm{Cov}[y'_i, \hat{y}_i] = 0$ whereas within-sample, $\mathrm{Cov}[y_i, \hat{y}_i] \neq 0$.
- Therefore:

$$\mathbb{E}(e'^2_i) = \mathbb{E}(e_i^2) + 2\mathrm{Cov}[y_i, \hat{y}_i]$$

Quantifying Out-of-sample prediction accuracy

- Fortunately we already did the work required to describe this:

$$\mathrm{Cov}[y_i, \hat{y}_i] = \sigma^2 \mathrm{H}_{ii}$$

- The mean out-of-sample prediction error is

$$\mathbb{E}(e'^2) = n^{-1} \sum_{i=1}^{n} e_i'^2 = n^{-1} \sum_{i=1}^{n} e_i^2 + 2n^{-1}\mathrm{tr}(\mathrm{H})$$

- We show in Worksheet 2.2A that $\mathrm{tr}(\mathrm{H}) = \sigma^2 p$ where p=number of predictors.
- The **optimism** is defined as $2n^{-1}\sigma^2 p$.
- The optimism grows with σ^2 and p but shrinks with n. It is used to define the **model selection criteria** ΔC_p which is minimised:

$$\Delta C_p = MSE_1 - MSE_2 + \frac{2}{n}\hat{\sigma}^2(p_1 - p_2)$$

Linear model optimism and AIC

- Minimising **Akaike's Information Criterion**:

$$AIC = -2\mathbb{L}(\hat{\theta}) + 2\mathrm{Dim}(\theta)$$

- reduces to the ΔC_p method when the Likelihood $\mathbb{L}$ is a Normal distribution.

LOOCV

- We write a statistic $\hat{s}$ based on all data $\{y\}$ except i as $\hat{s}^{(-i)}$ and the data is $\{y\}^{(-i)}$.
- For a general **loss function** we can write:

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}\left(y_i; \hat{\theta}|y^{(-i)}\right)$$

- i.e. we evaluate the loss function for each datapoint using the estimate from the remaining data.
- NB A loss function is something that we choose the parameters θ to minimise. It can be:
  - the MSE,
  - the (negative log) likelihood,
  - a penalised version of these,
  - or any other convenient quantity.

LOOCV for linear models

- For the MSE of a linear model we can write:

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(-i)}\right)^2$$

- It is not particularly straightforward[1] to show that:

$$LOOCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - H_{ii}}\right)^2$$

- This is a very important quantity, often called the Studentized residual
- i.e. the LOOCV can be directly computed from a regression containing all data, by "downweighting" low-leverage data and upweighting high-leverage (hard to predict) data.

[1] Our references avoid proving this, but do discuss the motivation. Proofs are available but beyond scope.

Leave-one-out Cross-Validation

- Leaving out a single datapoint is going to be insufficient unless the **data are independent**.
- The real world is rarely completely independent.
- However, there is often a computationally convenient way to compute LOOCV, and it is still better than leaving nothing out. It converges to C_p for large n.
- Analogous tricks work for:
  - **Linear models** including **Best Linear Unbiased Predictors** (BLUPs)
  - **Kernel methods**
  - **Nearest neighbour** methods
  - And others

Asymptotics

- Here are some facts about the asymptotic behaviour of LOOCV:
  - As $n \to \infty$, the expected out-of-sample MSE of the model picked by LOO cross-validation is **close to that of the best model** considered.
  - As $n \to \infty$, if the true model is among those being compared, LOOCV tends to pick a **strictly larger model** than the truth.
- LOOCV is not the right tool for choosing the **right model**.
- It is however an excellent tool for choosing the model with the best out-of-sample **predictive power**.
- …when the data to be predicted come from the **same distribution as the data**!

Problems with LOOCV

- We might worry that leaving out one datapoint at a time isn't enough:
  - **Cost**. It is straightforward to apply LOOCV to an arbitrary loss function, including a Likelihood. However, it can be costly.
  - **Quality**. LOOCV estimates of out-of-sample loss has high variance because each test datapoint using $n - 2$ of the **same training datapoints**…
    - Empirically, we often choose a different model on different data generated under the same distribution!
  - **Correlation**. Any correlation breaks LOOCV.

K-fold CV

- Naive **k-fold CV** addresses the first issue by creating a **bias-variance tradeoff**: we introduce a bias (towards simpler models) but also significantly reduce the variance of the MSE estimation.
- More complicated sampling in k-fold settings can also address correlation.
- **Split** the data into k "folds" $f(i)$, that is, **random non-overlapping samples** of the data of size n/k. Then:
- **For each fold** i:
  - Call $X^{-(f(i))}$ the "training" dataset and $X^{(f(i))}$ the "test" dataset
  - Learn parameters $\hat{\theta}_i$ with data $X^{-(f(i))}$
  - Evaluate $l_i = \text{Loss}(X^{(f(i))}|\hat{\theta}_i)$
- And report $\frac{1}{n}\sum_{i=1}^{k} l_i$

How many folds?

- k-fold CV loses a fraction of the data, whereas LOOCV only loses a constant.
- This means that (under the assumption that the **true model is not in the model space**) k-fold CV will choose a **simpler model** with less predictive power than was possible.
- However, smaller k can make the inference more consistent across different data.
- For **small data**, LOOCV is recommended. For **larger data**, $k = 10$ is often chosen:
  - **cost**. k defines the minimum number of times you need to run the models. If you can afford to run a model once, you can probably afford 10 times.
  - **practicality**. If you had only 10% more data you might expect to get the same performance as LOOCV. We frequently lose this amount of data to quality control, etc.

Handling correlation

- **Correlation** structures can be handled in k-fold CV by **careful sampling**:
  - a-priori there is a correlation in time or space expected. we can therefore **remove windows**.
  - the data have some associated covariate, which can be removed en-masse.
  - empirical correlation structures can be used to select a point i and all points correlated with it above some **correlation threshold**.
- Some of these can be used in other contexts. Examples include:
  - **block bootstrap**
  - Using a different definition of a "datapoint" in a leave-one-out context, for example: datapoints are countries instead of countries at timepoints

Reflection

- What is model selection, and how is it different to statistical testing and parameter estimation?
- Be able to perform basic calculations with Leave-one-out cross validation (CV) and to make judgement calls about the appropriate use of k-fold CV.

Signposting

- Cross Validation is extremely poplular **because it works**. It is probably the most important component of machine learning.
- Further reading in:
- Chapters 2.3 and 7.10 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).
- Cosma Shalizi's Modern Regression Lectures (Lectures 20, 26)
- Next up: Workshop on Statistical Model Selection
- That is the end of this block.