Outline: **Cross-Validation** (F. Chiaromonte)

Introduction to Statistical Learning Chapter 5 Section 1 Lab 3

COMPUTATIONAL ~~ASSESSMENT~~/TUNING OF STATISTICAL PROCEDURES

For a very long time, the properties of statistical procedures were assessed using mathematical manipulations, facts about probability distributions and asymptotic results.

Mathematical tractability defined a narrow scope for statistics:

- Consider only simple procedures
- Introduce strong assumptions on the stochastic mechanism generating the data, and/or
- Prove properties only for large samples

’70s onward: replacing math with computation has substantially expanded the scope.

IN PROCEDURES WHOSE PERFORMANCE DEPENDS CRUCIALLY ON TUNING PARAMETERS, HOW DO WE IDENTIFY THEIR OPTIMAL/SATISFACTORY VALUES… BASED ON THE DATA?

CROSS VALIDATION: A computational approach to evaluate the out-of-sample accuracy of a statistical procedure used for prediction (supervised problems). Computationally expensive but now viable.

- Given a model or procedure comprising a tuning parameter (e.g. a polynomial regression of degree r; a lowess with smoothing parameter s; a classifier with threshold t)… how do we choose the tuning parameter?
- More generally, given a set of alternative models… how do we chose among them?

E.G. **Mean Squared Error (MSE)** for a regression; prediction of a continuous response (the quantity that is minimized in-sample by Least Squares fitting in the case of parametric linear models).

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

$$\hat{y}_i = \begin{cases} \hat{\beta}_o + \hat{\beta}' x_i & \text{(parametric)} \\ \ell(x_i) & \text{(lowess fit)} \end{cases}$$

Or E.G. **Misclassification rate (Err)** for a classifier; prediction of a categorical response.

$$Err(in) = \frac{1}{n}\sum_{k=0}^{n} Ind(\hat{y}_i \neq y_i)$$

$$\hat{y}_i = classpred(x_i) \quad \text{(classifier)}$$

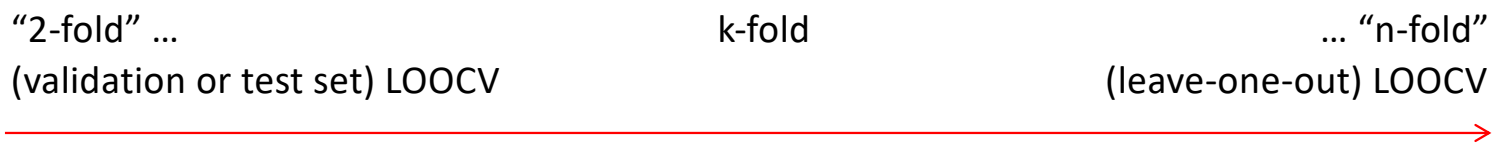
"in" = in-sample, same as "train" (training data for the procedure)

In-sample MSE (Err) can be a poor approximation of **out-of-sample MSE (Err)**; how closely we manage to "reproduce" the training data, especially with a large or complex model having lots of degrees of freedom, can say little about how accurately we would predict the response on independent test data.

In-sample MSE (Err) can substantially underestimate out-of-sample MSE (Err); if we overfit we are learning idiosynchratic components of the training data, not the underlying mechanism.

Traditionally, formulae were developed for specific problems (e.g. Mallow's C_p for regression – good estimator of the out-of-sample MSE under assumptions and for large samples).

Cross Validation (CV) allows us to produce a reliable estimate of the out-of-sample MSE for a collection of values of the tuning parameter (a collection of possible models), and then select the one that provides the lowest.



- Divide the data at random in k subsets of equal size S_1, S_2 … S_k

- For j=1,2 … k

  Form Training and Test Sets
  (sizes n(k-1)/k) and n/k , respectively)

  $$TrSet = \bigcup_{h \neq j} S_h \qquad TsSet = S_j$$

  Train model/procedure on Training Set $\hat{M}_j$

  Compute MSE on Test Set

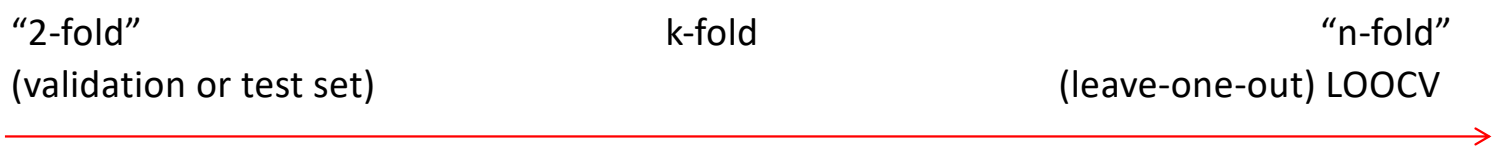
  $$MSE_j = \frac{1}{n/k} \sum_{i \in S_j} (y_i - \hat{M}_j(x_i))^2$$

- Average measurements over the folds

$$MSE(out)_k^{CV} = \frac{1}{k} \sum_{j=1}^{k} MSE_j$$

Estimates the out-of-sample MSE (Err) without the *underestimation* one has in-sample, but with an *overestimation* due to using a smaller sample size in training!

Select k based on computational burden, as well as a variance-bias trade off.



As k increases (from 2 to n)

- Computational burden increases.
- Overestimation bias for MSE(out) decreases: training sets used in each fold grow in size, up to n-1 ~ n.
- Variance for estimation of MSE(out) increases: training sets used in each fold have larger overlaps; *while we average more numbers (k) they are more dependent* – less actual replication.

Common reasoning: using k>10 often induces an increase in computational burden and variance that is not justified by a substantial decrease in the overestimation bias…

Note: in the old fashioned "validation (test) set approach", the calculation is not even repeated flipping the roles of the two folds as training and testing sets.