CROSS-VALIDATION

So far, we have used model selection criteria that are based on an estimate of the *expected* gap between pMSE and MSE e.g. C_p, AIC and BIC.

Now, cross-validation — a direct way of estimating the pMSE itself

Data set split into

TRAINING	VALIDATION	TESTING

Use TRAINING to fit models $M_1, \ldots, M_J$ to data.
Use VALIDATION to compute $pMSE(M_1), \ldots, pMSE(M_J)$ and select the model M^* with the smallest pMSE

Apply M^* to the TESTING Data to obtain a valid estimate of $pMSE(M^*)$

(Why not $\boxed{\text{TRAIN} \mid \text{VALIDATION}}$ only? Well, if VALIDATION is used to find the M^* with the smallest pMSE, then this pMSE is a biased estimate of true pMSE ~~[illegible]~~ (called selection bias)

It's important to remember that a valid pMSE estimate can only be obtained from an un-touched TESTING data.

The pMSE's we get from the VALIDATION data are used to <u>rank</u> the models, not to provide a final estimate of the true predictive performance.

Note, if you only fit one model M_1 to data and evaluate pMSE(M_1) on VALIDATION data, that's ok. It's the ranking of many models that generates the bias.

Now — how to split the data.

Focus on the | TRAIN | VALIDATION | part

| | | is called 2-fold CV

① | TRAIN | VAL | $\longrightarrow$ pMSE$(M_j)^1$, $j=1,\ldots,J$

models pMSEs

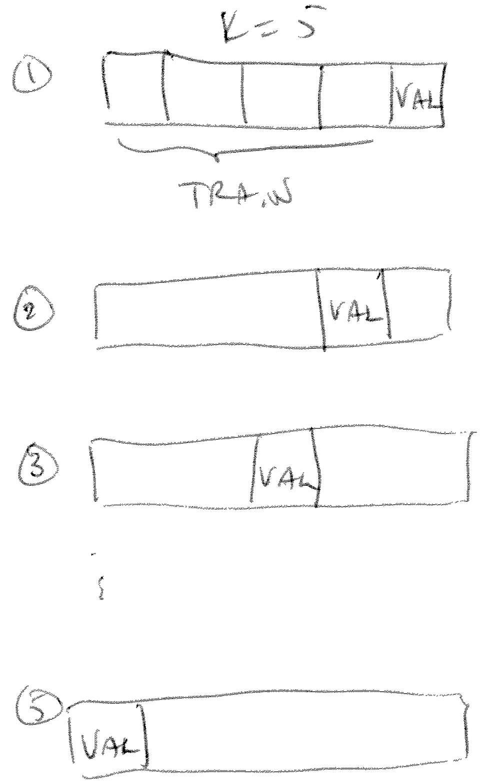
② | VAL | TRAIN | $\longrightarrow$ pMSE$(M_j)^2$, $j=1,\ldots,J$

That is, both parts of the data take a turn at training & validation.

We base our final ranking of models on the average

$$pMSE(M_j) = \frac{pMSE(M_j)^1 + pMSE(M_j)^2}{2}$$

Can also do <u>K-fold</u>



$\rightarrow pMSE(M_j)^1$

$\rightarrow pMSE(M_j)^2$

$\vdots$

$\rightarrow pMSE(M_j)^5$

$$pMSE(M_j) = \frac{1}{K} \sum_{k=1}^{K} pMSE(M_j)^k$$

average pMSE on each of the K VALIDATION sets.

How large should K be?

Small K	Large K
• TRAINING data much smaller than original data → more difficult to get good parameter estimates → bias against complex models but • computationally fast	• TRAIN almost same size as original data, so almost no bias but since each TRAIN share a lot of observations with other TRAIN ⇒ individual $pMSE(M_j)^k$ estimates are correlated ⇒ high variance estimate of pMSEs → ranking highly variable too

So small K → may bias against selecting big models

large K ⇒ pMSEs are highly variable so selection is highly variable too.

Common K's used — $K=3$

$K=10$

and $K=n$ (leave-one-out CV)

LOOCV (leave-one-out)

- Let each observation take turn to be the VALIDATION set.

- $$\text{PMSE} = \frac{1}{n} \sum_{i=1}^{n} \Big(y_i - \underbrace{\hat{f}^{-i}(x_i)}\Big)^2$$

fitted value @ x_i when (x_i, y_i) <u>not</u> used for estimation

Studentized residual $(y_i - \hat{f}^{-i}(x_i))$ can be obtained without refitting the model

$$\hat{f}(x_i) = \sum_{j=1}^{n} h_{ij} y_j \qquad \text{(linear model)}$$

$$\Rightarrow \hat{f}^{-i}(x_i) = \frac{\sum_{i \neq j} h_{ij} y_j}{1 - h_{ii}} \qquad \Rightarrow \hat{f}^{-i}(x_i) = \sum_{i \neq j} h_{ij} y_j + h_{ii} \hat{f}^{-i}(x_i)$$

multiply up

$$\Rightarrow (y_i - \hat{f}^{-i}(x_i))^2 = \Big(y_i - \sum_{i \neq j} h_{ij} y_j - h_{ii} \hat{f}^{-i}(x_i)\Big)^2$$

$$= \Big(y_i - \underbrace{\sum_{i} h_{ij} y_j}_{\hat{f}(x_i)} + h_{ii}(y_i - \hat{f}^{-i}(x_i)\Big)^2$$

$$\Rightarrow y_i - \hat{f}^{-i}(x_i) = y_i - \hat{f}(x_i) + h_{ii}(y_i - \hat{f}^{-i}(x_i))$$

$$\Rightarrow y_i - \hat{f}^{-i}(x_i) = \frac{y_i - \hat{f}(x_i)}{1 - h_{ii}}$$

$$\Rightarrow \text{LOOCV} = \frac{1}{n} \sum (y_i - \hat{f}^{-i}(x_i))^2$$
$$= \frac{1}{n} \sum \left(\frac{y_i - \hat{f}(x_i)}{1-h_{ii}} \right)^2$$

So no refitting needed to get the leave-one-out CV.

<u>BUT</u>, for some types of model, hard to compute all leverage values h_{ii}

$\Rightarrow$ approximation

$$GCV = \frac{1}{n} \sum \left(\frac{y_i - \hat{f}(x_i)}{1 - \underbrace{\frac{tr\{H\}}{n}}} \right)^2$$

replace h_{ii} by <u>average</u> $\frac{tr\{H\}}{n} = \frac{\sum_{j=1}^{n} h_{jj}}{n}$

For many models $Tr\{H\}$ is very easy to compute.

E.g. linear regression $\Rightarrow$ $\underline{tr\{H\} = p}$, # variables

<u>2nd approx</u>

Note $\frac{1}{(1-z)^2} \approx 1+2z$ if z is small. Here $z = \frac{tr\{H\}}{n}$ small if n is large

$$\Rightarrow GCV = \frac{\frac{1}{n}\sum(y_i - \hat{f}(x_i))^2}{\left(1 - \frac{tr\{H\}}{n}\right)^2} \approx \underbrace{\frac{1}{n}\sum(y_i - \hat{f}(x_i))^2}_{MSE} + 2 \cdot \frac{tr\{H\}}{n} \underbrace{\frac{1}{n}\sum(y_i - \hat{f}(x_i))^2}_{\approx \hat{\sigma}^2}$$

$$= MSE + 2 * p * \hat{\sigma}^2 = C_p \, !$$

So C_p an estimate of the LOOCV almost