Tentative solution to RecEx Module 5: Resampling

Cross-Validation

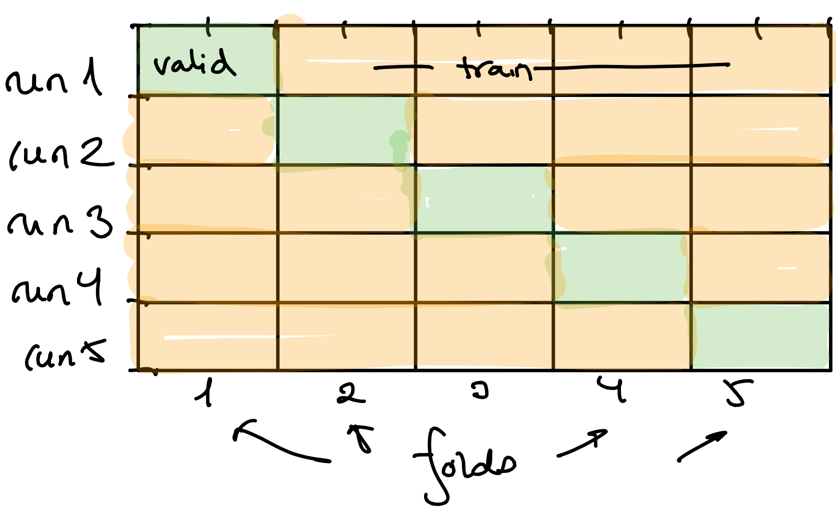
1) Explain how k-fold cross-validation is implemented.

Drawing: $k=5$ (for simplicity)

First - shuffle the data: indeces [11, 2, 17, ..., 28]

Then partition into k groups = folds.

↑ in general $\frac{n!}{k!\left[\left(\frac{n}{k}\right)!\right]^k}$ possible ways to do this (multinomial with a twist) see below



train =

validate =

In run 1 fold 1 is kept aside and folds 2-5 are used to train the method (maybe many times, once for every model complexity). Then error is calculated on the validation fold.

Repeat k times: $CV_n = \frac{1}{n}\sum_{j=1}^{k} MSE_j \cdot n_j$

$$MSE_j = \frac{1}{n_j} \sum_{i \in G_j} (y_i - \hat{y}_i)^2 \text{ for MSE}$$

obs in validation fold

prediction of x_i in validation fold using fitted model from folds "$-j$"

not j.

other loss functions may be 0/1 loss.

Regression: find the optimal number of neighbours in KNN-regression.

Classification: choose between QDA or LDA in classification

n obj. deles inn i k gruppen med m i hver, $n = km$

$(\text{ant. perm. av de } n) = n!$

$$(\text{ant. perm. av de } n) = (\underbrace{\text{ant. måter å dele inn i k gruppen}}_{B})$$

$$\cdot (\underbrace{\text{ant. perm. av gruppene}}_{= k!}) \cdot (\underbrace{\text{ant. måter å permutere innen gruppe}}_{m!})^k$$

$$= Bk!(m!)^k$$

$$B = \frac{n!}{k!(m!)^k} = \binom{n}{k} \frac{(n-k)!}{(m!)^k}$$

2) Advantages & disadvantages of k-fold CV relative to

a) the validation set

D: computational complexity

A: bias = generally larger sample size for (each k-1 in) k-fold then validation set, which means "more data → better fit" and therefore not overestimate the test set error

Bias?: compared to using the full data set for model fit.

A: different validation sets may give very different test error, so the results are variable - much more then for k-fold.

b) LOOCV ← no randomness in splits!

A: less computational efforts for k then n, unless nice formula as for multiple linear regression.

A: less bias - in the sense that LOOCV use a larger set to fit to data (n-1 obs) which gives a less biased version of the test set error.

D: higher variance: we are averaging the output from n fitted models that are trained on nearly the same data ⇒ correlated positively.

This happens to a less degree with k-fold, since the k models more different data and are thus less variable.

$$CV_n = \left((y_1-\hat{y}_1)^2 + (y_2-\hat{y}_2)^2 + \cdots + (y_n-\hat{y}_n)^2 \right) \frac{1}{n}$$

$Var(CV_n) =$ sum variances of terms + 2 · covariances of terms

this part tend to be larger for LOOCV than k-fold when correlation higher between models

c) Choice of k in k-fold. We just know that

$k = n =$ LOOCV (small bias – high variance) of estimator for test set error
+ generally high comp. demand

$k = 2$ (larger bias – lower variance)
+ less comp. challenging

and empirical research has found $k = 5$ or $k = 10$ to be good choices!

3) Case (as in R-code) : classification set-up with two classes

- $n = 50$ observations of $p = 5000$ predictors

a) Choose to use only $d = 25$ predictors, but choose the top d from absolute correlation coeff between the p preds. and the class label.

b) then use logistic regression with the d predictors.

$\Rightarrow$ How to do CV? On $\underbrace{a+b}_{\text{right}}$ or only on $\underbrace{b}_{\text{wrong}}$?

Wrong : if only b, then all data used to find the predictors $\rightarrow$ gives

rigth : both a+b $\Rightarrow$ all is good

See R-code in problem and run to see what the misclassification rates are..

Bootstrepping

1)

a) $P(\text{draw } x_i) = \frac{1}{n}$, $P(\text{not draw } x_i) = 1 - \frac{1}{n}$

b) $P(\text{not any } x_i\text{'s}) = (1-\frac{1}{n})^n$

$P(\text{at least one } x_i) = 1 - (1-\frac{1}{n})^n$

c) $P(x_i \text{ in boot sample}) = 1-(1-\frac{1}{n})^n \approx 1 - \exp(-1)$

$= 0.682$

d) R-code to check result and see how fast $1-(1-\frac{1}{n})^n \to 0.632$ (in n).

2) Bootstrep to estimate $SD(\hat{\beta})$:

$(X, Y)^*_b$: bootstrep sample b, $b = 1, \dots, B$.

for $(b \text{ in } 1{:}B)$ {

fit $Y = X\beta + \varepsilon$ and keep $\hat{\beta}_b$

}

Calculate $\hat{SD}(\hat{\beta}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\beta}_b - \frac{1}{B} \sum \hat{\beta}_b \right)^2}$

Why do we went to do this - when we really know that $\hat{SD}(\hat{\beta}) = \hat{\sigma} \cdot \text{diag}((X^TX)^{-1})$

And we might also do $\hat{Cov}(\hat{\beta})$ but then use

$$\hat{Cov}(\hat{\beta}) = \frac{1}{B-1} \sum_{b=1}^{B} (\hat{\beta}_b - \bar{\hat{\beta}}_b)(\beta_b - \bar{\hat{\beta}}_b)^T$$

$$\bar{\hat{\beta}}_b = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}_b$$

3) is covered on page 195 of the ISL book