MODEL SELECTION / MODEL ASSESSMENT

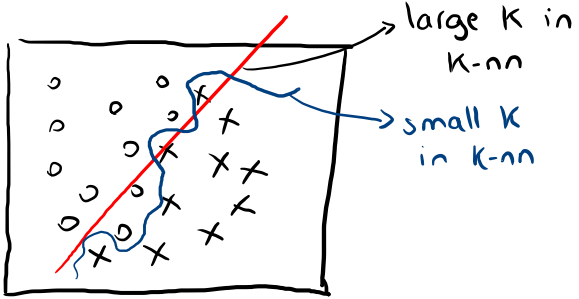
22/03/21

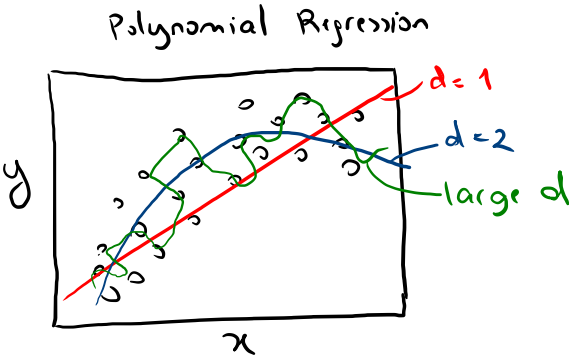
We have looked at many methods so far: K-nn, LR, LDA, QDA, NB, ...

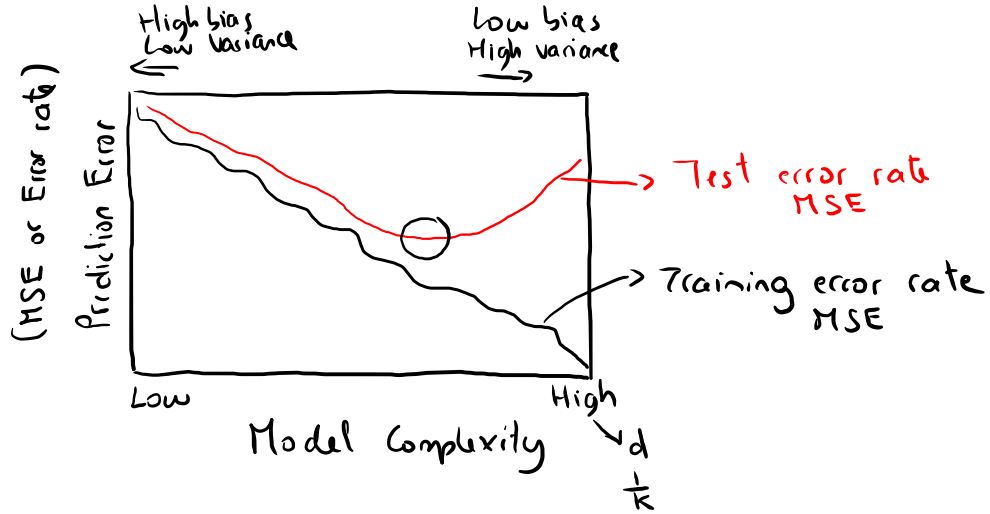
Two objectives:

1. Model assessment: evaluate a model performance
2. Model selection: selecting the appropriate level of flexibility

It is crucial to find a good trade-off between bias and variance.







Best solution : a large designated Test set, left aside to select the optimal model. often not available.

Two main approaches :

① Make some mathematical adjustments to the training error rate in order to estimate the test error rate.
→ C_p statistic, AIC, BIC, adjusted R^2, ...

② Resampling methods : estimating directly the test error rate by doing some resampling strategy → cross-validation

RESAMPLING METHODS

General idea: hold out a subset of the training observations from the fitting process, so that I can use these observations for testing

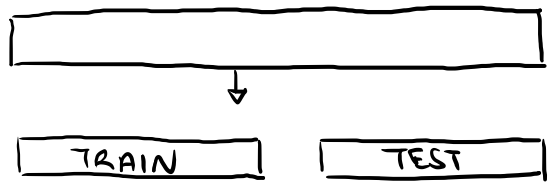
We need to reach a good compromise in terms of:

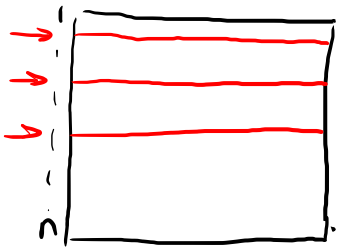
1. computational complexity
2. using enough data for fitting while testing the model on sufficiently different / representative datasets

Three main approaches:

① VALIDATION SET APPROACH

Randomly divide the dataset into two parts: a training set and a validation / hold-out / test set:





- 50-50 split
- 70-30 split

(x, y) from same individual must be kept together

EXAMPLE

Polynomial regression

- $d = 2$ (quadratic)
- estimate $\hat{\beta}$ on training data
- $\hat{y}$ on test data using $\hat{\beta}$ estimated from training data
- Calculate MSE on test

k-nn

- Fix k
- Estimate $\hat{p}(1|x)$ on test based on k nearest neighbours from training data
- Calculate error rate

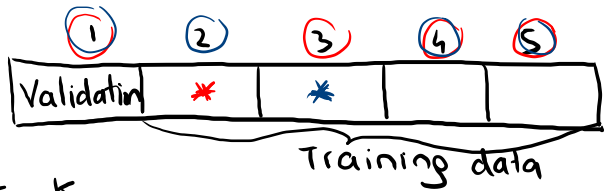
In general, there are two main issues:

① Variability in MSE / Test error for different data split ← Could be addressed by repeated train-test splits then average

② Not using all the data to fit the model
→ tend to over-estimate the test error

A more sophisticated approach is K-FOLD CROSS-VALIDATION

Idea: Randomly divide the data into K equally-sized parts $\sim \frac{n}{K}$



K = 5

1. Leave out part K
2. Fit the model on the other K-1 parts (combined)
3. Use this model to make predictions on the K-left out part
4. Repeat for all the parts and combine the results

In detail :

Fix a particular model, one particular degree ($d=2$) or a specific value of k in K-nn.

① Let $C_1, \dots, C_K$ be the indices of the observations in each fold

② For each fold i, calculate the error E_i by making predictions from the model fitted on the other folds:

$$E_i = \begin{cases} MSE_i = \dfrac{\sum_{j \in C_i} (y_j - \hat{y}_j^*)^2}{|C_i|} \\ ER_i = \dfrac{\sum_{j \in C_i} I(y_j \neq \hat{y}_j^*)}{|C_i|} \end{cases}$$

→ prediction of observation j using $\hat{\beta}$ estimated from the other folds

③ Summarize the results by taking the average error:

$$CV_{(K)} = \frac{\sum_{i=1}^{K} E_i}{K}$$

Some remarks

1. All data gets used at some point, as each observation will belong to one of the folds
2. Computationally more expensive: model has to be fitted K times
3. Much less variability between different runs of cross-validation (different random splits of the data)
4. Common choices of K are K=5 or K=10

   A special case is of K=n: n-fold cross-validation

   LEAVE-ONE-OUT CV (LOOCV)

LOOCV

Cross-validation with K=n. No randomness

Each observation is used as test set in turn, with the model fitted on the remaining n-1 observations. In the end:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} E_i$$

Disadvantage: ① computational time (n models)

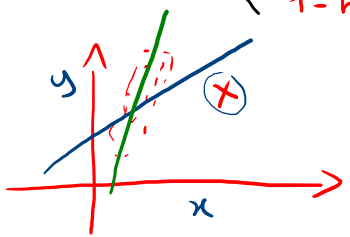
In linear regression, there is a magic formula that allows to calculate $CV_{(n)}$ with a single fit of the model on the full data

MULTIPLE REGRESSION ONLY!

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

$\hat{y}_i$ is prediction of observation i (with predictors $\underline{x}_i$) using $\hat{\beta}$ estimated on full data

h_i → leverage (between 0 and 1 – the closer to 1 the more influencial that point is in the estimate of $\hat{\beta}$)



② It doesn't "shake up" the data enough. Models are fitted on almost the same data.

If the model is too complex and overfits the data, it may continue to do so in each fold)

A better choice is to use $K=5$ or $K=10$.

ALTERNATIVE TO CV

Indirectly estimating the test error, by adjusting the Training error.

① Mallow's C_p : (regression)

$$C_p = \frac{1}{n}\left(RSS + 2d\hat{\sigma}^2\right)$$

- RSS: residual sum of squares
- $\hat{\sigma}^2$: estimate of σ (typically obtained from the most complex model)
- d: number of parameters

② AIC (AKaike Information Criterion) - defined on every model where likelihood is used for inference

$$AIC = -\frac{2}{n}\log L + 2\frac{d}{n}$$

- d: number of parameters
- $\log L$: maximized value of likelihood of chosen model

For regression models

$$\log L \propto -\frac{RSS}{2\sigma^2}$$

③ $BIC = -\frac{2}{n}\log L + \frac{\log(n)}{n} \cdot d$

Bayesian Information Criterion

AIC vs BIC : 2 replaced by $\log(n)$ in BIC

$\log(n) > 2$ for any $n > 7$

BIC tends to result in simpler models than AIC

④ Adjusted R^2 (regression)

$$R^2 = 1 - \frac{RSS}{TSS}$$

For model selection:

$$adj\text{-}R^2 = 1 - (1-R^2)\frac{n-1}{n-d-1} = 1 - \frac{RSS}{n-d-1}\,\frac{(n-1)}{TSS}$$

depends on d

does not depend on d

Use more complex model only if it reduces RSS "enough".