Lecture 4 - 05/09/24

MSE → train vs test

- In ML we want our models to generalize well to unseen data
- This implies the fact that we want the test MSE to be low
- Unfortunately, test MSE is unknown to us because we use train MSE

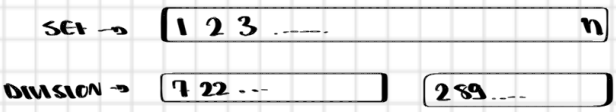
So what? 🤔

↳ We can estimate test MSE by:

- making clever mathematical corrections to our training MSE
- split dataset leaving out unseen data from the model } avoids overfitting
  - ↳ then use unseen data to estimate test MSE

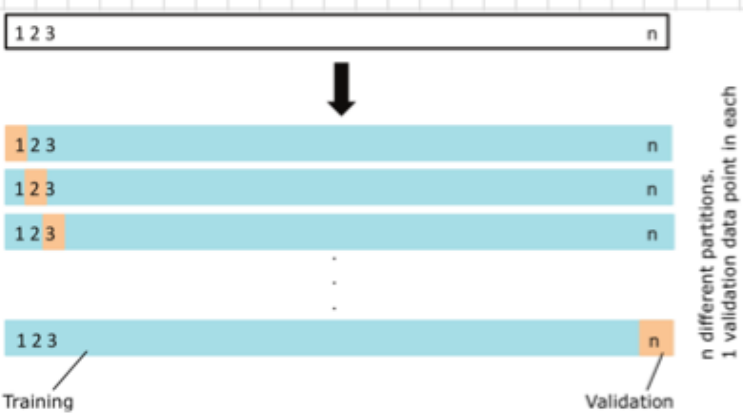
First attempt

- Divide the set into 50% - 50%



- This leads to big problems
  - Performance depends on what data points are in the two sets
  - Performance is underestimated because many data points are not used for training

Leave-one-out cross validation (LOOCV)



- What happens here is basically we divide the dataset into n random partitions & from each of those we leave one out
- The leave one out is the MSE(i) every time.
- This leads to:

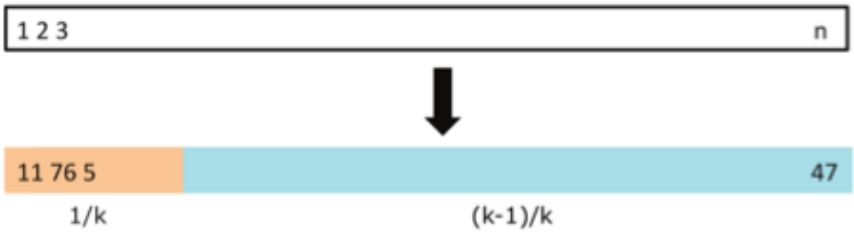
  ∀ partition, i, the test MSE = $(y_i - \hat{y}_i)^2$.

- We can then say that the LOOCV estimate for test MSE is the AVG(test error):

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

K-FOLD CROSSVALIDATION

- IN K-FOLD CROSSVALIDATION, WE LEAVE OUT $1/K^{th}$ OF THE DATA AT THE TIME.

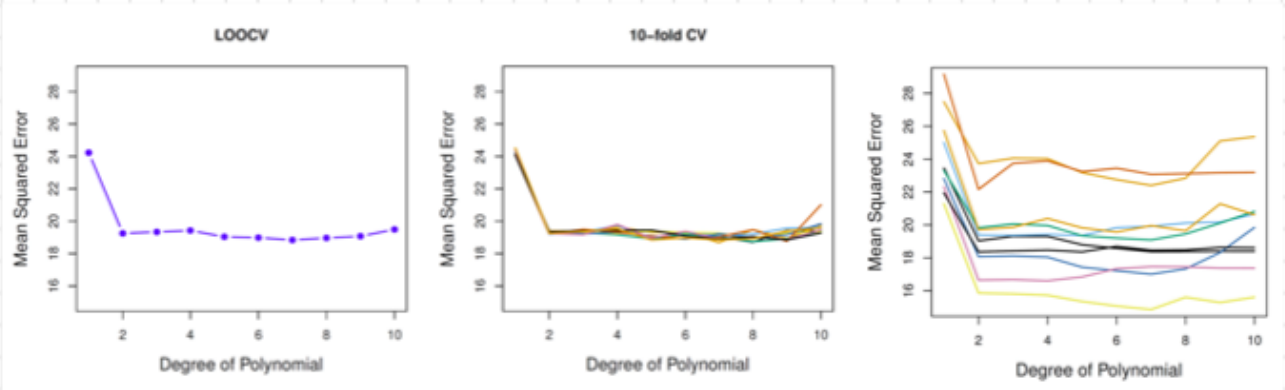


- IN THE i^{th} HELD-OUT FOLD, WE COMPUTE MSE_i. AT THE END AVG OVER K OBTAINED MSE_i'S:

$$\frac{1}{K}\sum_{i=1}^{K} MSE_K$$

- MAKE SURE TO RANDOMLY ALLOCATE OBSERVATIONS TO FOLDS, SHUFFLE EVERYTIME.

- IN BOTH THE FOLD METHODS:
  - NEED TO FIT THE MODEL AS MANY TIMES AS THERE ARE FOLDS
- FOR LOOCV THERE IS JUST 1 WAY OF ALLOCATING OBSERVATIONS TO FOLDS:
  - FOR K-FOLD CROSS-VALIDATION THERE IS RANDOMNESS IN HOW OBSERVATIONS ARE ALLOCATED TO FOLDS, SO THE CROSS-VALIDATION CURVE WILL BE DIFFERENT IF YOU DO IT AGAIN.



OF FOLDS

- COMMON CHOICES ARE K=5 OR K=10 FOLDS
- THERE IS NOT A PERFECT K OBVIOUSLY & ALSO SAME PROBLEM ARISES:
  - FEWER FOLDS LEADS TO FEWER OBSERVATION
  - TOO MANY LEAD TO OVERFITTING
- WHAT'S THE VARIANCE OF THE TEST ERROR WITH K FOLDS?
  - IT IS AN AVERAGE OF K ESTIMATED ERRORS. THEY ARE POSITIVELY CORRELATED.
  - THIS LEADS TO:

VARIANCE IS LARGER THAN IF THERE WERE NO CORRELATIONS.

$$\text{Var}\left(\frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i\right) = \frac{1}{k^2}\text{Var}\left(\sum_{i=1}^{k}\text{MSE}_i\right)$$
$$= \frac{1}{k^2}\sum_{i=1}^{k}\left(\text{Var}\,\text{MSE}_i + 2\sum_{j=1}^{i}\text{Cov}(\text{MSE}_i, \text{MSE}_j)\right)$$

JUST A MATH HARD PROOF!

REGRESSION MODELS: STRATEGIES WITH "MANY" FEATURES

- WITH HIGH NUMBER OF FEATURE $p \approx n$ ✗ NOT ENOUGH INFO IN DATA TO ESTIMATE A CURVE WELL
- WITH > PARAMS THAN FEATURE $p > n$ THE MLE DOES NOT EVEN EXIST.

WHAT SHOULD WE DO?

- MODEL SELECTION: INCLUDE ONLY A SUBSET OF THE FEATURES IN THE MODEL
- SHRINKAGE: ESTIMATE PARAMS BY MINIMISING LOG-LIKELIHOOD + PENALTY TERM
- DIMENSIONALITY REDUCTION OF FEATURE SPACE: LATER IN COURSE...

STRATEGIES FOR FEATURE SELECTION!

- BEST SUBSET: EXTENSIVE, SOMETIMES EVEN IMPOSSIBLE!
- FORWARD SELECTION: START BY INCLUDING NO/FEW VARIABLES
- BACKWARD SELECTION: START BY INCLUDING ALL/MANY VARIABLES
- ALTERNATE: ALTERNATE BETWEEN FORWARD & BACKWARD

COMPARISON METHODS:

- AIC
- BIC
- TEST ERROR
- F-TEST CAN BE USED FOR TWO NESTED MODELS

AIC → AN INFO CRITERION

$$\text{AIC} = -2\log\hat{L} + 2p$$

p = # OF PARAMS

$\hat{L}$ = MAXIMISED LIKELIHOOD FUNCTION

WE SEEK THE MODEL WITH THE SMALLEST AIC

SMALL MEANS BETTER, THOUGH, GIVES NO INDICATION OF GOODNESS FIT.

SIMILAR IS:

$$\text{BIC} = -2\log\hat{L} + p\log n.$$

Stepwise Selection

Backward

1. Decide on the largest model
2. Iteratively remove the least relevant feature
3. Stop if removing a feature leads to no improve

Forward

1. Decide on smallest model (only intercept)
2. Iteratively add most relevant feature
3. Stop if adding = no improvement

Automatic Model Selection:

We might want to, from interpretational perspective:

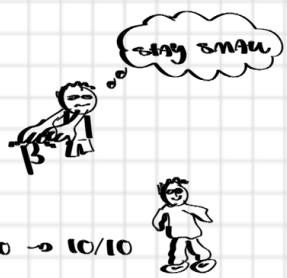
- Keep certain variables in the model even if they are not significant
  i.e. → vars interesting for research question!
- Remove variables, even if significant.
  i.e → complex interactions that complicate

Estimating Test Error directly

- With validation we estimate test error directly.
- Can use test error to choose/select between models.
- The estimated MSE used to select between models where AIC/BIC often cannot.
- One-standard-error rule chooses the simplest model among models with st.error

Shrinkage Methods

- Help us estimate coefficients β in a constrained way.
- Help us keep β small
- Reduces coefficient variance & can improve the fit



2 are the shrinkage methods:

Ridge Regression

- β's estimated by minimising:

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

λ = tuning param

$\lambda \|\beta\|_2^2$

- Gives us back a penalty
  - It's size is controlled by λ
- Intercept β_0 is not regularized!
- $\lambda = 0$ means no penalty
- $\lambda \to \infty$ shrinks all β's to 0

Lasso

- β's estimated by minimising:

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

$\lambda \|\beta\|_1$

- Gives us back a penalty
  - It's size is controlled by λ
- As penalty ↑↑↑ often more β's are set to 0
  - this means eliminating features!

Lasso Vs Ridge

One can show that Ridge Regression coefficient estimates solve

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

And Lasso coefficient estimates solve

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

s can be thought of as a "budget" for $\sum_{i=1}^{p} \beta_i$.

- This simply shows the minimizing
- Both Process tend to minimize β's
- Simply $\neq$ math formulas that lead to different minimization processes.
- Aim → find only most relevant variables
- Both method help to reduce overfitting & improve prediction accuracy
- With Ridge β's never reach 0, while with Lasso it is possible!

Lasso Regression Vs Ridge Regression	
Lasso Regression uses L1 regularization (absolute value of coefficients).	Ridge Regression uses L2 regularization (square of coefficients).
Lasso Regression can force them to be exactly zero.	Ridge Regression shrinks coefficients of less significant features towards zero.
Lasso Regression performs both regularization and feature selection, making it more suitable for high-dimensional datasets.	Ridge Regression does not perform feature selection and can only shrink the coefficient values. This makes it more suitable for datasets with highly correlated predictors since it avoids including all of them in the model.
Lasso Regression may be more effective in situations where only a subset of features contribute significantly to the output	Ridge Regression generally works better in scenarios where there are fewer significant features.
Lasso Regression can lead to a sparse model, which means it can create a model with fewer features.	Ridge Regression does not produce sparse models.

Last Considerations!

- Choose λ by crossvalidation
- Standardize feature before regularizing
- Ridge scales everything down
- Lasso shifts to zero & bounces