

Lesson 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 connections to our training MSE
- Split dataset leaving out unseen data from the model
 - Then use unseen data to estimate test MSE

} AVOIDS OVERFITTING

FIRST ATTEMPT

- Divide the set into 50% - 50%

SET → [1 2 3 ... n]

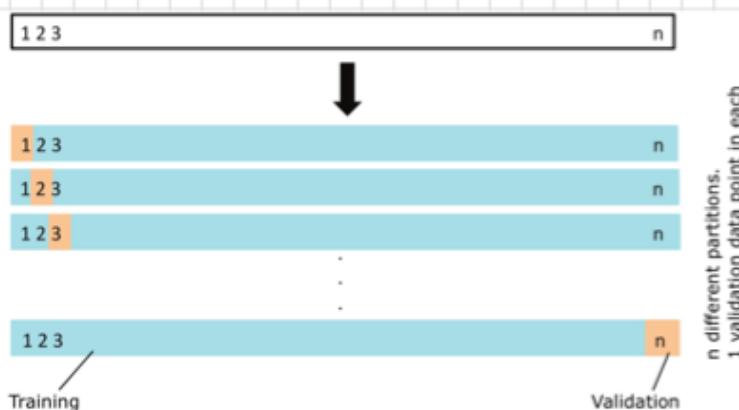
Division → [1 2 3 ...] [2 3 4 ...]

- 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 $\text{MSE}_{\text{train}}$ error.

• This leads to:

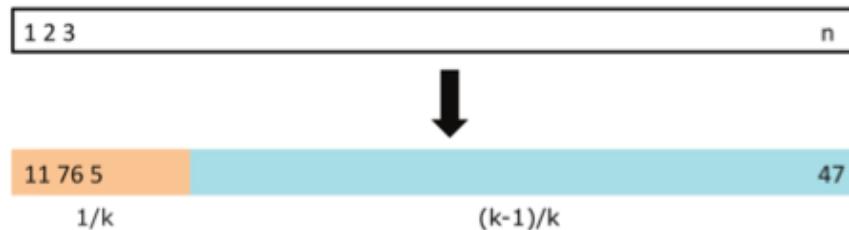
$$\text{For partition } i, \text{ the test MSE} = (y_i - \hat{y}_i)^2$$

- We can then say that the LOOCV estimate for test MSE is the $\text{AVG}(\text{test error})$

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

K-Fold Cross-validation

- In K-fold cross-validation, we leave out $1/k^{\text{th}}$ of the data at the time.



- In the i^{th} held-out fold, we compute MSE_i . At the end, we average k obtained MSE 's:

$$\frac{1}{k} \sum_{i=1}^k \text{MSE}_i$$

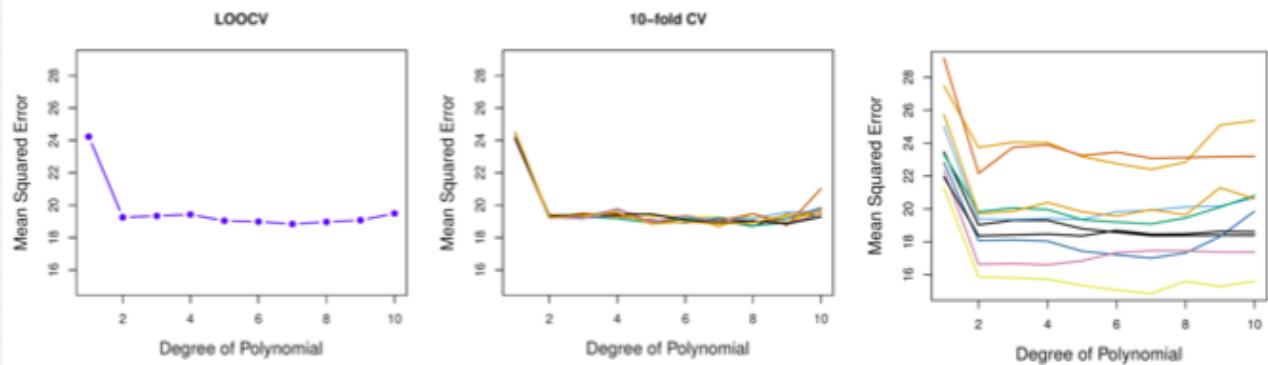
- Make sure to **randomly** allocate observations to folds, **shuffle every time**.

- 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 if also same problem arises:
 - Fewer folds leads to fewer observations
 - Too many leads 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.

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

Just a math
hand 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 P PARAMS THAN FEATURE $p > n$ THE MLF 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
- DIMINISHING REDUCTION OF FEATURE SPACE: LATER IN COURSE--

STRATEGIES FOR FEATURE Selection!

- Best Subset: EXPENSIVE, 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

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

OF PARAMS

MARSHALD LIKELIHOOD FUNCTION

We seek the model with the SMALLEST AIC

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

SIMILAR IS:

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

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 improvement**

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**:

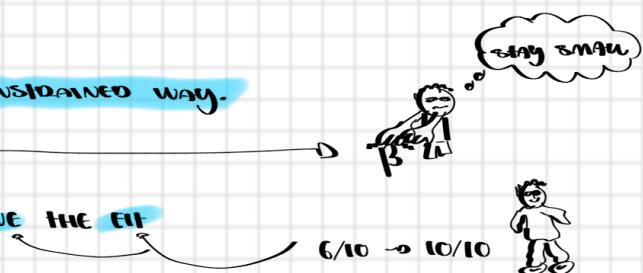
- **Keep certain variables** in the model even if they are **not significant**.
i.e. \rightarrow vars interesting for research question!
- **Remove variables**, even if **significant**.
i.e. \rightarrow complex interactions that complicate

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 **All/BIG often cannot**.
- One-standard-error rule chooses the simplest model among models with **staggered**

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**:

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

$\underbrace{\qquad\qquad\qquad}_{\|\beta\|_2^2}$

TUNING PARAM

- Gives us back a **PENALTY**
- This size is controlled by λ
- Intercept β_0 is **NOT REGULARIZED**!
- $\lambda = 0$ means **NO PENALTY**
- $\lambda \rightarrow \infty$ shrinks all β 's to 0

Lasso

- β 's estimated by **minimising**:

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

$\underbrace{\qquad\qquad\qquad}_{\|\beta\|_1}$

- Gives us back a **PENALTY**

This size is controlled by λ

- As **PENALTY λ** often more β 's are **set to 0**
- This means **eliminating features**!

Lasso VS Ridge

One can show that Ridge Regression coefficient estimates solve

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

And Lasso coefficient estimates solve

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

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

- This simply shows the minimization
- Both Process tend to minimize β 's
- Simply ≠ math formulas that lead to different minimization processes.
- Aim → Few 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 cross-validation
- Standardize feature before regularizing
- Ridge scales everything down
- Lasso shifts to zero & truncates