Hyperparameter tuning and Cross Validation

Dr Rebecca Barter

Hyperparameters

RF

- Number of trees in the forest
- Min observation in terminal “leaf” nodes
- Number of variables to consider at each split

XGBoost

- Max “depth” (# splits) of each tree
- Min “loss” reduction
- Learning rate (how aggressively the algorithm updates for each iteration)
- Number of variables to consider at each split

Optimizing “*tuning*” these hyperparameters can lead to improved performance

Hyperparameter default values

RF

Number of trees in the forest **[500]**

Number of variables to consider at each split **[$\sqrt{\text{\# cols}}$]**

Min observation in terminal "leaf" nodes **[5 cont; 10 binary]**

XGBoost

Max "depth" (# splits) of each tree **[6]**

Learning rate (how aggressively the algorithm updates for each iteration **[0.3]**

Min "loss" reduction **[0]**

Number of variables to consider at each split **[all]**

Optimizing "*tuning*" these hyperparameters can lead to improved performance

Using cross validation to tune hyperparameters

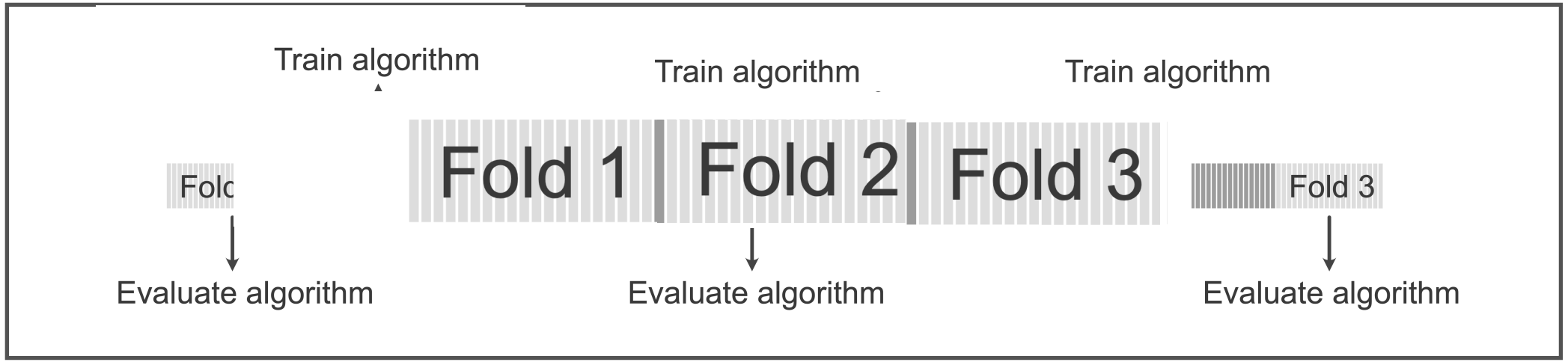
Idea: Train each algorithm many times using different hyperparameter value combinations and compare performance (on validation set)

Issue: Validation set used to train our algorithm—no longer independent!

Remedy: Use Cross Validation on the training set to determine optimal hyperparameter values

V-fold Cross Validation (CV)

1. Split the *training* data into V equal-sized non-overlapping subsets/“folds”
2. Withhold the first fold (to be “pseudo-validation” set)
3. Train algorithm on remaining V-1 folds
4. Use withheld fold to evaluate algorithm
5. Replace withheld fold, and repeat using the other V-1 folds as the withheld pseudo validation set.



Tuning hyperparameters

Create a **grid** of plausible hyperparameter combinations

mtry (# vars to consider at each split): 4, 8, 12

depth (max # of splits): 5, 10

n_trees (# of trees in forest): 250, 500

Comb.	mtry	depth	n_trees	AUC
1	4	5	250	0.65
2	4	5	500	0.67
3	4	10	250	0.62
4	4	10	500	0.63
5	8	5	250	0.71
6	8	5	500	0.73
7	8	10	250	0.68
8	8	10	500	0.67
9	12	5	250	0.70
10	12	5	500	0.71
11	12	10	250	0.69
12	12	10	500	0.67

For each hyperparameter combination, compute the average 5-fold CV or 10-fold CV error/accuracy (e.g., AUC for binary problems, rMSE for continuous problems).

Choose the combination of hyperparameters that have the highest best CV predictive performance (e.g., AUC/rMSE)