Bagging, random forests, & boosting

Readings for today

- Chapter 8: Tree-based methods. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R (Vol. 6). New York: Springer

Topics

1. Bagging

2. Random forests

3. Boosting

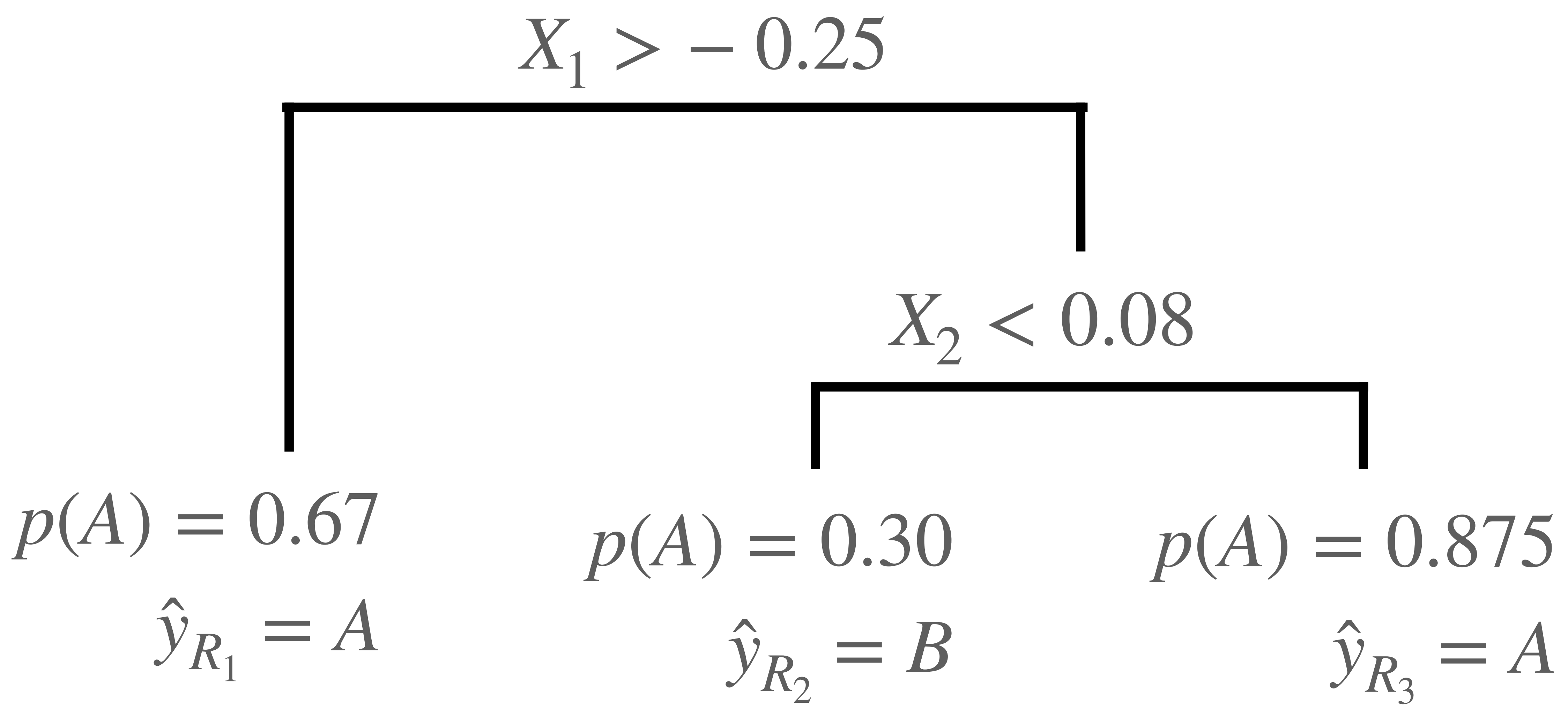
Bagging

Problems with decision trees

Disadvantages:

- Lower predictive accuracy.
- Non-robust

small changes in data have huge impacts on model fits.



High flexibility of decision trees leads to sensitivity to variation in data set.

(James et al. 2013)

Bootstrap aggregation (Bagging)

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_1^*} \overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_1^*} \rightarrow \hat{f}^1(X_1^*)$$

$$\vdots$$

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_B^*} \overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_B^*} \rightarrow \hat{f}^B(X_B^*)$$

$$\hat{y}_i = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x_i)$$

- Use bootstrapping to generate B predictive models.
- For any prediction of y_i, run x_i through all B models.
- Use the average of those predictions for $\hat{y}_i$.

(James et al. 2013)

Example: regression

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_1^*} = \overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_1^*} \overbrace{\begin{pmatrix} \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{pmatrix}}^{\hat{\beta}_1^*} \rightarrow \hat{f}^1(X_1^*) = \hat{\beta}^*_{1,1} X^*_{1,1} + \ldots + \hat{\beta}^*_{1,p} X^*_{1,p}$$

$$\vdots$$

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_B^*} = \overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_B^*} \overbrace{\begin{pmatrix} \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{pmatrix}}^{\hat{\beta}_B^*} \rightarrow \hat{f}^B(X_B^*) = \hat{\beta}^*_{B,1} X^*_{1,1} + \ldots + \hat{\beta}^*_{B,p} X^*_{B,p}$$

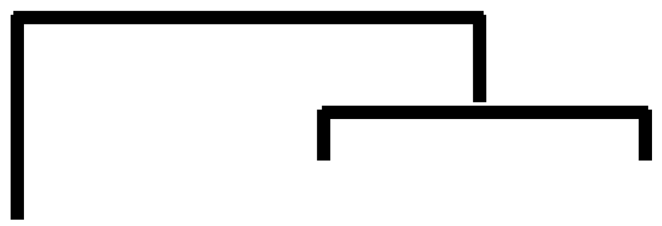
$$\hat{y}_i = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x_i)$$

$$= \frac{1}{B} \sum_{b=1}^{B} \sum_{j=1}^{p} \hat{\beta}^*_{b,j} x_{i,j}$$

(James et al. 2013)

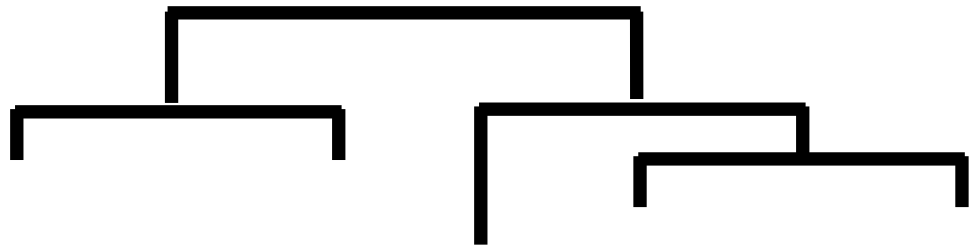
Example: trees

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_1^*} = \hat{T}_1(\overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_1^*}) \rightarrow \hat{f}^1(X_1^*) = \hat{T}_1(X_1^*) \rightarrow$$



$$\vdots$$

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_B^*} = \hat{T}_B(\overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_B^*}) \rightarrow \hat{f}^B(X_B^*) = \hat{T}_B(X_B^*) \rightarrow$$



$$\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x_i)$$

$$= \frac{1}{B}\sum_{b=1}^{B} \hat{T}_b(x_i)$$

(James et al. 2013)

Out-of-bag (OOB) error

Each pull of the bootstrap leaves out a few observations (on average 1/3 of observations are left out). Use these as a test set on bagged model.

<u>Example:</u> $X = (13, 45, 32, 16, 25, 33)$

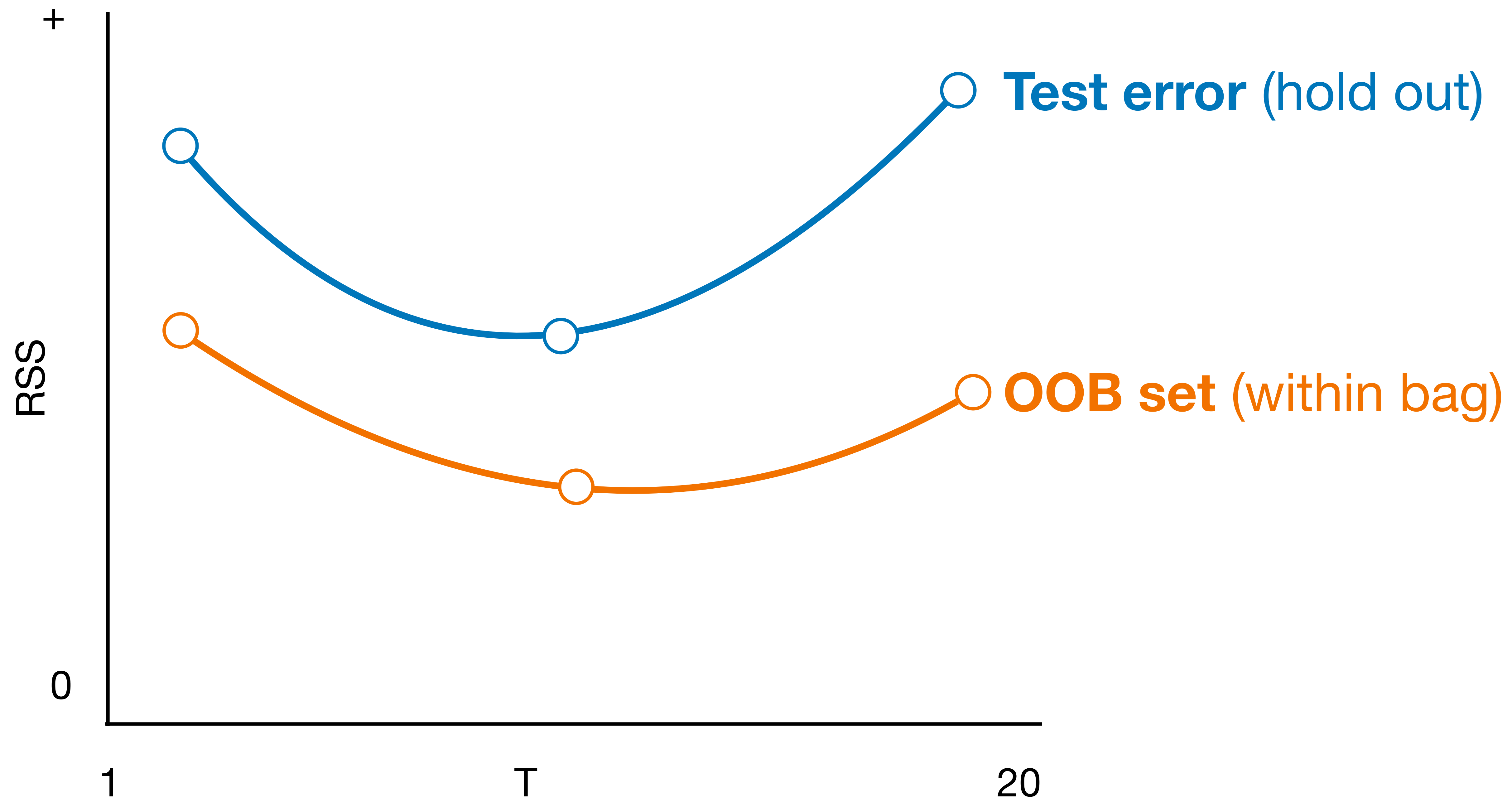
original data

$$X_1^* = \underbrace{(33, 33, 25\ 16, 16, 25)}_{\text{Training set}}\ \underbrace{(45, 13)}_{\text{Test set}}$$

$$\hat{f}^1(X_1^*) \qquad OOB = \sum_{i=1}^{m} (y_i - \hat{f}^1(x_i))^2$$

(James et al. 2013)

Out-of-bag (OOB) error



OOB error is usually lower than traditional hold out set error, while maintaining independence between the test and training data.

(James et al. 2013)

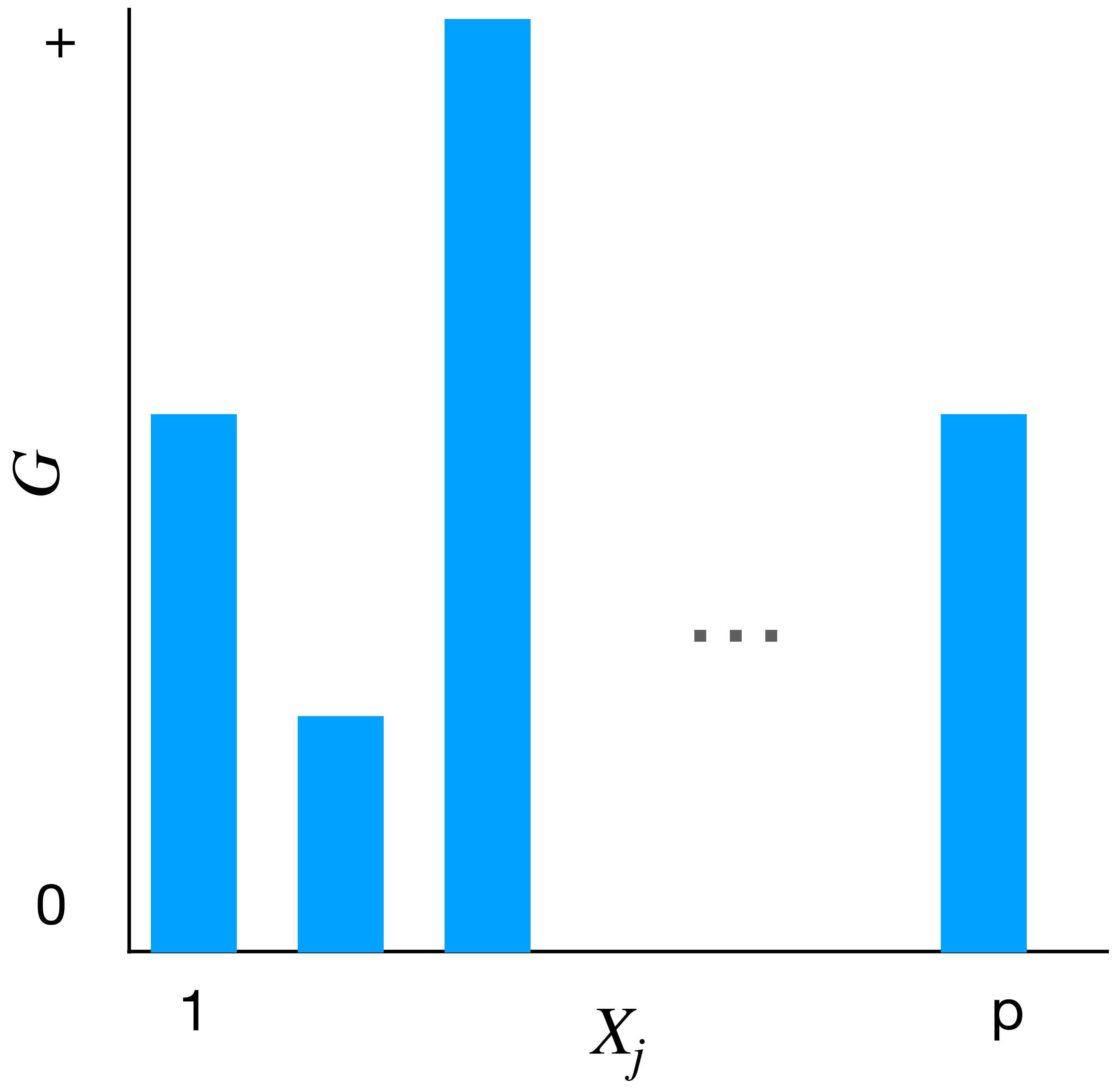
Variable importance

<u>Goal:</u> On each pull from the bag, b, take each predictor variable, j, and calculate goodness-of-fit using <u>just</u> that variable. Average across all pulls, B.

- Regression = RSS
- Classification = G

Variable Importance measures

Approximate estimate of importance.

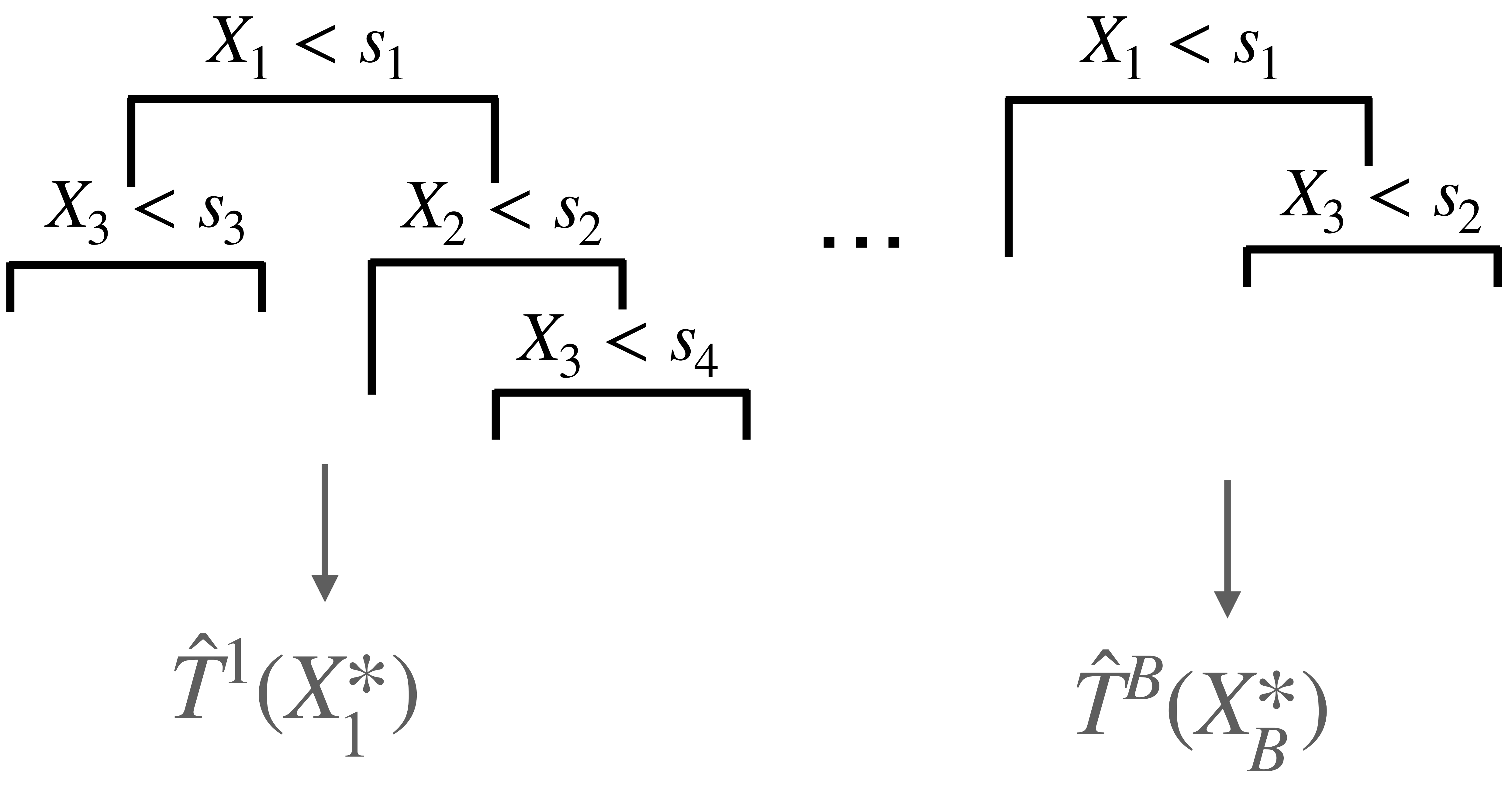


(James et al. 2013)

Random forests

Problem with bagging trees

Because of their flexibility, a few important factors will drive splits in all iterations of the bootstrap, leading to ↑ correlations across models.

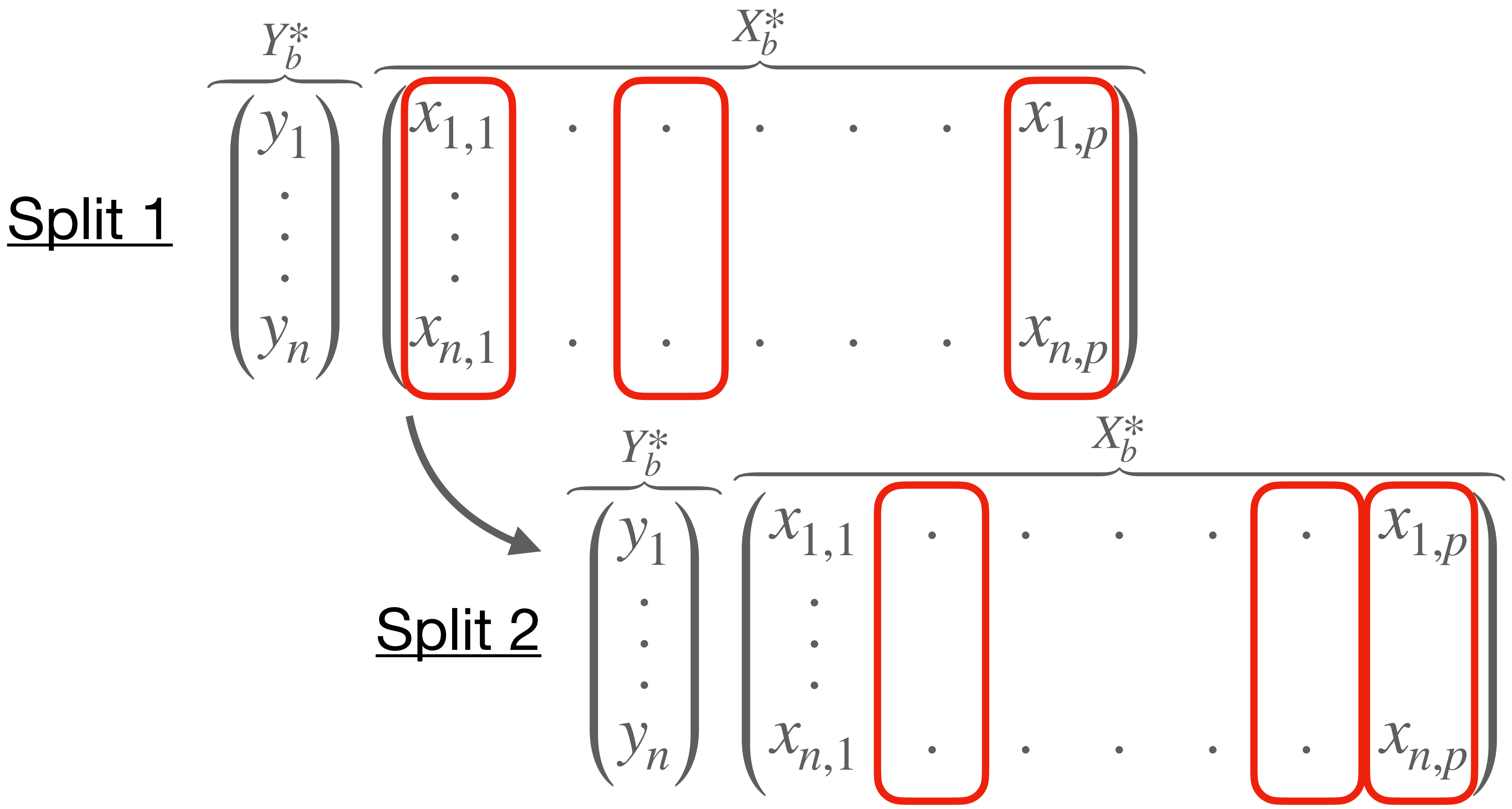


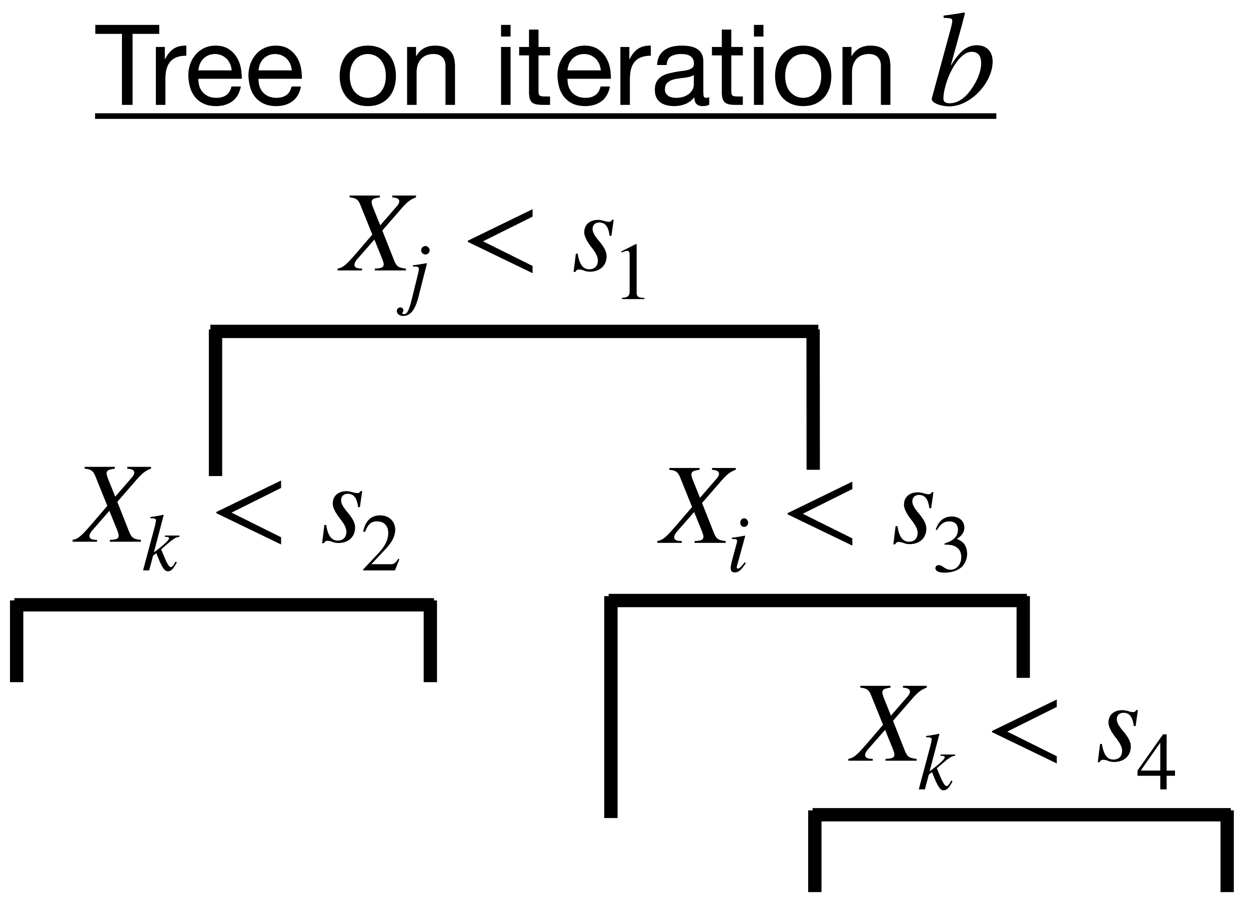
X_1 is <u>always</u> in first split

(James et al. 2013)

Random forests

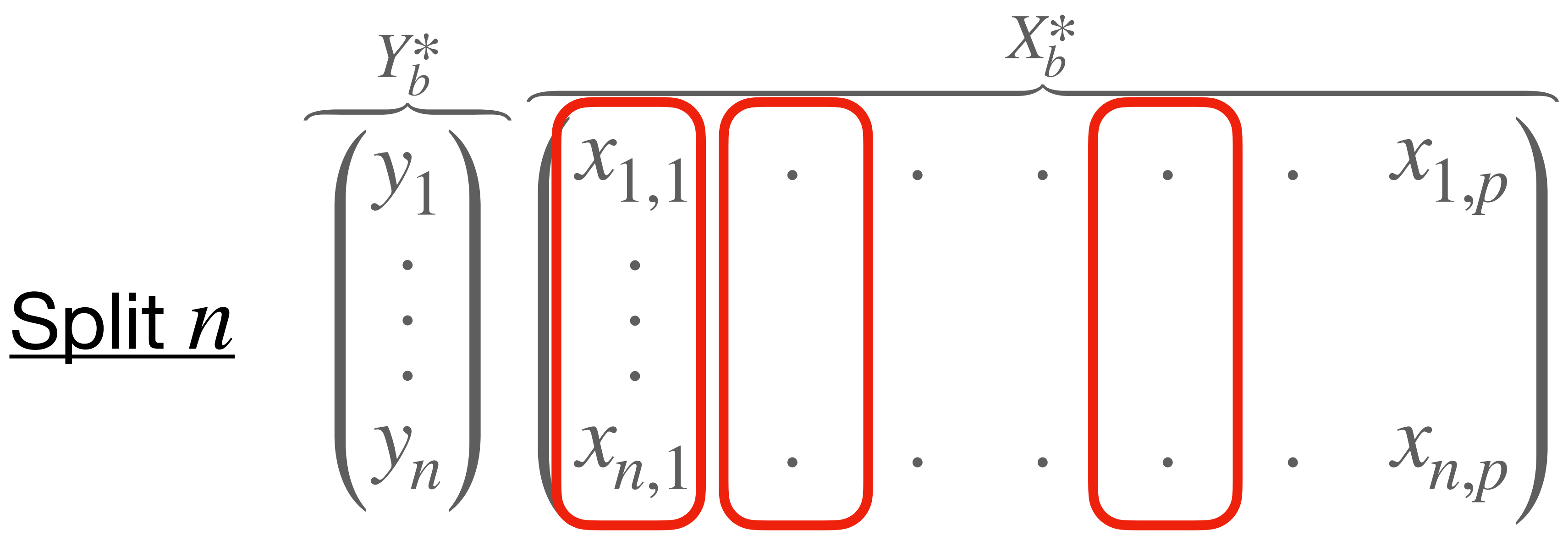
<u>Solution:</u> On each split allow only a subset of predictor variables, m, out of all variables, p, to be included in the model.





$$m = \sqrt{p}$$

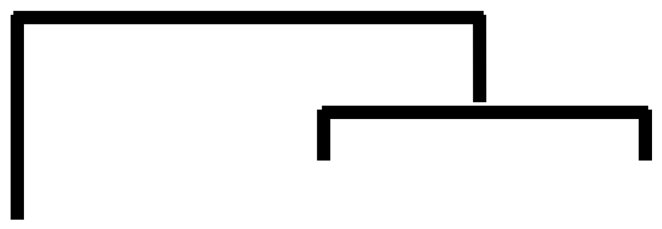
…



(James et al. 2013)

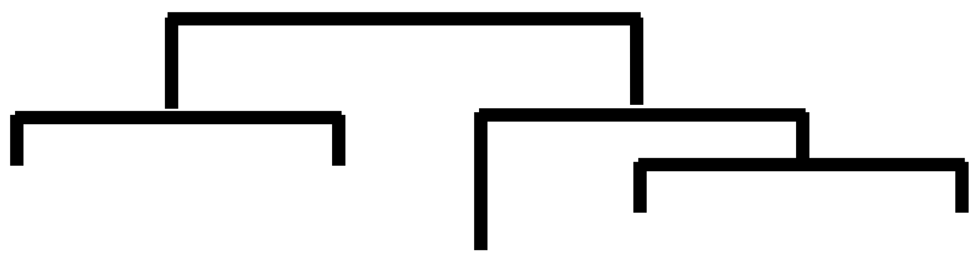
Forest of random trees

$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_1^*} = \hat{T}_1(\overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_1^*}) \rightarrow \hat{f}^1(X_1^*) = \hat{T}_1(X_1^*) \rightarrow$$



$$\vdots$$

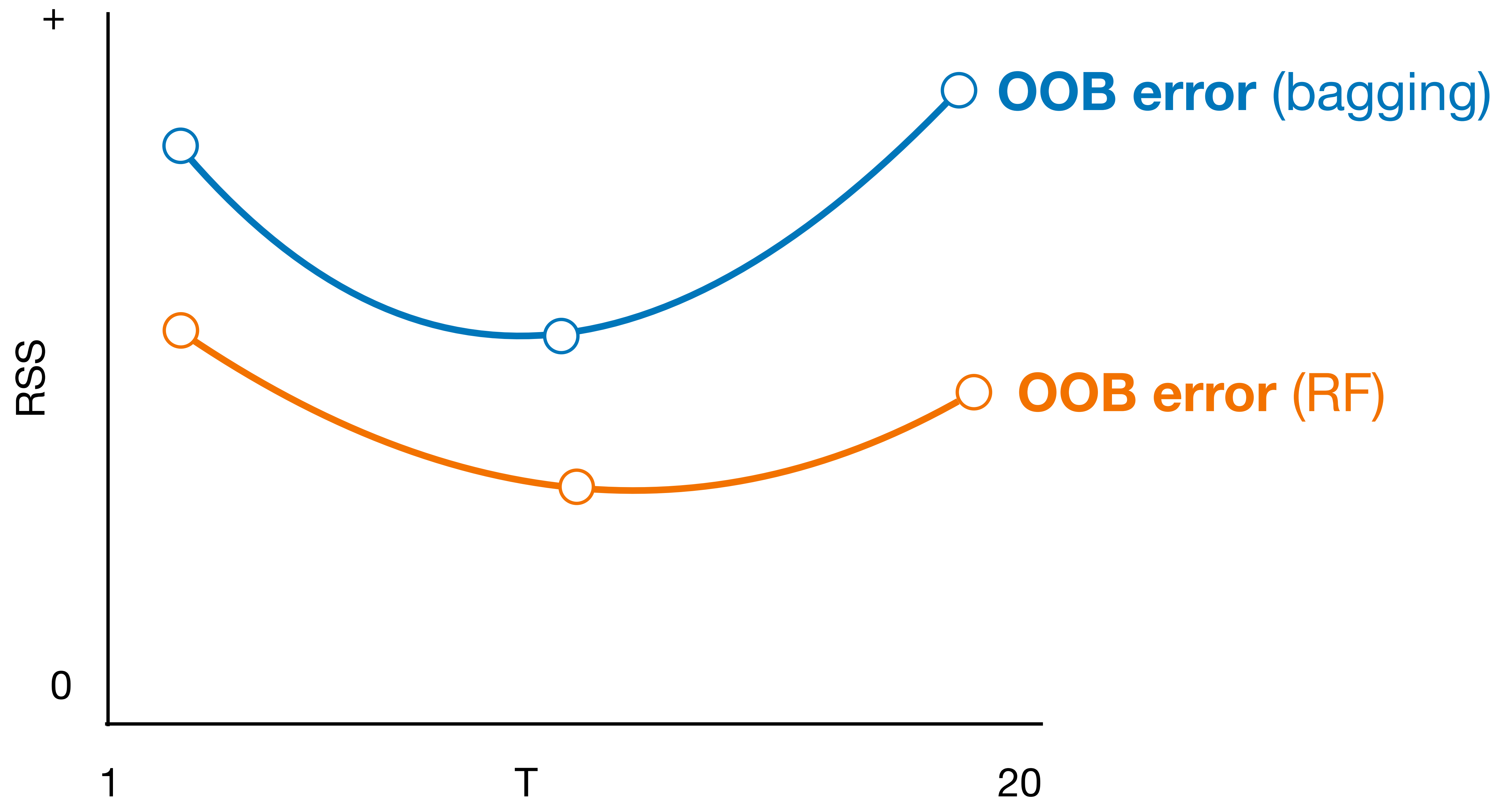
$$\overbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}^{Y_B^*} = \hat{T}_B(\overbrace{\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix}}^{X_B^*}) \rightarrow \hat{f}^B(X_B^*) = \hat{T}_B(X_B^*) \rightarrow$$



$$\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x_i) = \frac{1}{B}\sum_{b=1}^{B} \hat{T}_b(x_i)$$

(James et al. 2013)

Random vs. static forests



- Random forests (RF) forces weaker predictor variables to contribute to predictions.
- On average, a single strong predictor will be in the "approved list" only $\frac{p-m}{p}$ times.
- Improves test accuracy when your predictor variables are highly correlated.

(James et al. 2013)

Boosting

Boosting

Bagging & RF: $\underbrace{\hat{f}^1(X_1^*), \ldots, \hat{f}^B(X_B^*)}_{\text{independent}}$

Boosting: $\underbrace{\hat{f}^1(X_1^*) \to \hat{f}^2(X_2^*) \to \ldots \to \hat{f}^B(X_B^*)}_{\text{sequential}}$

Let each model inform the next so that you *boost* the overall variance explained by the collective set.

(James et al. 2013)

Boosting algorithm

<u>Goal:</u> "Eat up" all the residual variance that you can.

<u>Step 1</u>: Start with a null model, $\hat{f}(X) = 0$, such that $r_i = y_i - \hat{y}_i = y_i$.

<u>Step 2</u>: Run B iterations where, on each iteration b:

- Calculate a new model $\hat{f}^b(X)$ using the objective function.
  $$\min \sum_{i=1}^{n} (r_i - \hat{f}^b(x_i))^2.$$
- Update the previous model with the current model
  $$\hat{f}(X) \leftarrow \hat{f}(X) + \lambda \hat{f}^b(X).$$
  sparsity parameter
- Update the residuals $r_i \leftarrow r_i - \lambda \hat{f}^b(X)$.

<u>Step 3</u>: Output the boosted model $\hat{f}(X) = \sum_{b=1}^{B} \lambda \hat{f}^b(X)$

(James et al. 2013)

Parameters to tune

Boosting relies on 3 free parameters that have to be tuned.

1. B → number of models generated

2. λ → sparsity constraint

3. d → number of splits (if using trees)

Care must be taken with cross validation sets when selecting these parameters.

(James et al. 2013)

Take home message

- Bagging, random forests, and boosting are very powerful methods that improve overall prediction accuracy of high variance methods (e.g., decision trees), but at the expense of interpretability.