STAT 24620=FINM 34700, STAT 32950
Multivariate Data Analysis
Lecture 13: Bagging and Random Forests

Jingshu Wang

The University of Chicago

Outline

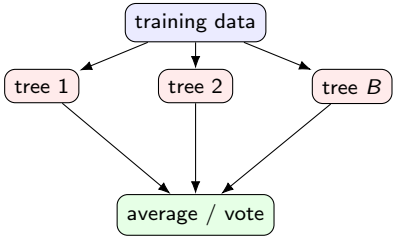
Why not stop with one tree?

- A tree is flexible and interpretable.
- But the first few splits are high-leverage decisions.
- A small change in the training data can change the tree structure.
- This is a high-variance estimator.

Random forests keep the flexibility of trees but reduce variance by averaging.

The ensemble idea

- Fit many prediction rules, not one.
- Make them different by perturbing the training data or the fitting procedure.
- Average their predictions.



For regression:

$$\hat{f}_{\text{ens}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x).$$

For classification: average class probabilities or use majority vote.

A variance calculation

Suppose each tree has variance σ^2, and any pair of trees has correlation ρ.
Then the variance of the average of B trees is

$$\mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B} T_b(x)\right) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

- Increasing B reduces the second term.
- The first term remains if the trees are strongly correlated.
- Therefore we want many accurate trees that are not too correlated.

Random forests are designed to reduce correlation among trees.

Bagging: how to create different trees

Question: How do we make the trees in an ensemble different?

Bagging (bootstrap aggregation):

- Keep the same model (decision tree)
- Change the **training data**
- For each tree:
  - Sample n observations **with replacement**
  - Fit a large tree on this resampled dataset (do not prune)

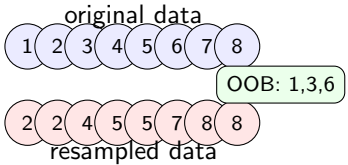
Same method + different data $\Rightarrow$ different trees

Averaging these trees reduces variance

What does resampling do?

Key idea: Each tree is trained on a slightly different dataset.

- Draw n observations **with replacement**
- Result:
  - Some points appear multiple times
  - Some points are not selected
- Out-of-bag (OOB) observations:
  - Points not used in this tree



Each tree sees a slightly different version of the data

Out-of-bag (OOB) error

Question: How do we evaluate bagging without a test set?
Key idea: Use OOB observations as evaluation data.

For a fixed observation i,

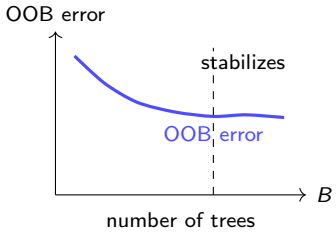
$$P(i \text{ is OOB for one tree}) = P(i \text{ is not selected in one bootstrap sample}) = \left(1 - \frac{1}{n}\right)^n \approx 0.37.$$

OOB validation:

- For observation i, use only trees where i is OOB
- Predict y_i using those trees
- Compare with observed y_i

Practical use:

- Track OOB error as the number of trees B increases
- Choose B large enough so the error stabilizes



From bagging to random forests

Bagging trees still tend to be correlated:

- if one predictor is very strong, many trees split on it near the root;
- then many trees look similar;
- averaging similar trees does not reduce variance as much.

Random forest modification: at each split, consider only a random subset of predictors.

$$m_{\text{try}} < p.$$

This decorrelates the trees and improves the variance reduction from averaging.

Random forest algorithm

1. For $b = 1, \ldots, B$, draw a bootstrap sample (resample observations).
2. Grow a large tree on this sample.
3. **At each split**, randomly select m_{try} predictors.
4. Choose the best split only among those predictors.
5. Aggregate predictions across all trees.

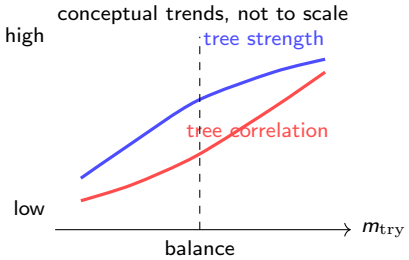
Each tree uses all variables overall, but only a random subset at each split.

Randomness is added both in data (bootstrap) and in splits (feature subsampling)

What does m_{try} do?

Tree strength: prediction accuracy of an individual tree

- Larger m_{try}: stronger trees, but more correlated.
- Smaller m_{try}: more diverse trees, but weaker.
- The best choice balances strength and diversity.



Common defaults:

$$m_{\text{try}} \approx \sqrt{p}, \text{classification}, \quad m_{\text{try}} \approx p/3, \text{regression}.$$

Why these defaults?

- Classification: avoid dominance of a few strong variables $\rightarrow$ more diverse trees
- Regression: Signal spread across variables $\rightarrow$ include more variables to keep trees strong

Practical aspects of random forests

Computation

- Build B trees $\Rightarrow$ cost grows roughly linearly in B
- Each tree is cheaper (only m_{try} variables per split)
- Trees are independent $\Rightarrow$ easy to parallelize

Choosing B

- Increase B until OOB error stabilizes
- No overfitting from large B
- Typical choices: $B = 100$–1000

Tuning m_{try}

- Try a few values and compare OOB error
- Defaults often work well

Variable importance (permutation)

Random forests are harder to visualize than a single tree, so we often summarize variable importance.

Question: Which variables matter for prediction?

Idea: break the variable and see what happens

- Randomly shuffle X_j across observations
- This destroys its relationship with the response
- Recompute prediction error (e.g., using OOB data)

$$\text{Importance of } X_j = \text{error increase after shuffling } X_j$$

$$\text{Large increase} \Rightarrow \text{important variable}$$

Importance is predictive, not automatically causal.

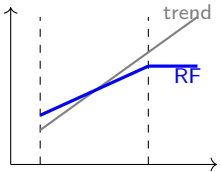
When do random forests work well?

- **Complex structure:** nonlinear effects and interactions
  - trees adapt automatically without specifying a model
- **Many predictors:**
  - feature randomness helps avoid over-reliance on a few variables
- **Prediction-focused tasks:**
  - averaging many trees gives strong predictive accuracy
- **Moderate to large sample size:**
  - flexible models need enough data to generalize well

Random forests are a strong default when the true model is complex or unknown.

Limitations of random forests

- **Interpretability:**
  - harder to understand than a single tree
- **Extrapolation:**
  - predictions are averages of observed responses
  - cannot extend trends beyond the training range
- **Variable importance:**
  - correlated predictors can share importance
- **Probabilities (classification):**
  - may be overconfident (too extreme)
- **Causal questions:**
  - captures associations, not causal effects



extrapolation fails

Strong predictor, but limited for interpretation and extrapolation

R demo

Notebook file: `Lecture13_demo.nb.html`, Bagging and random forest section.

Lecture 13 summary

- Bagging averages trees fit to bootstrap samples.
- Random forests add random feature selection at each split.
- Averaging reduces variance; random feature selection reduces tree correlation.
- OOB error gives a built-in estimate of predictive performance.
- Variable importance helps interpret the fitted forest.

Suggested reading

- James, Witten, Hastie, Tibshirani (2nd edition), Chapter 8.2.