Bagging, Random Forests, Boosting

Eltecon Data Science Course by Emarsys

Sándor Sóvágó

November 10, 2021

About me

- Data Scientist @ Emarsys AI Labs
- (Previously: Assistant Professor in Economics at the University of Groningen)
- Contact: *sovago.sandor@gmail.com*

Recap: What we have learnt thus far?

- Classification problems are about predicting categorical variables (making a purchase in the next 30 days)
- The logistic regression is a regression technique for binary classifications
- Decision trees are non-linear classifiers
- Accuracy may not be informative for classification problems
- The ROC curve visualizes the trade-off between True Positive Rates (TPR) and False Positive Rates (FPR)

Learning outcomes

After this lecture students will be able to

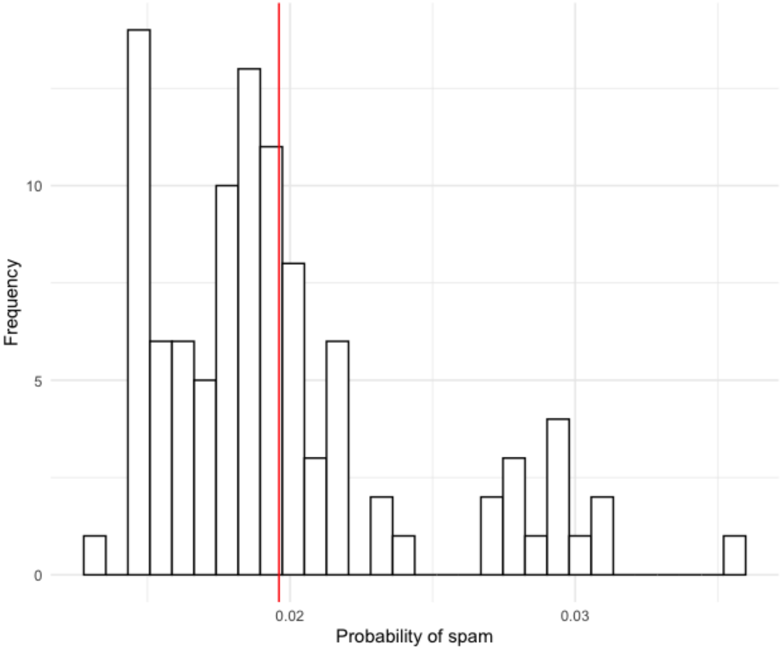
- understand the intuition behind more advanced tree-based methods, such as
  - bagging,
  - random forests,
  - boosting;
- implement these methods in R;
- interpret the output of tree-based methods
- understand pros and cons of these methods

Bagging – Background

- Tree-based models yield high variance predictions
- High variance predictions: different training sets give very different predictions

High-variance predictions: illustration

- Spam dataset: can we predict spams with the length of the e-mails using a decision trees?
- What is the likelihood that an e-mail is a spam with a length of 61 characters?
- Mean: 0.02
- Standard deviation: 0.005
- Homework: conduct the same exercise using a logit model insted of a tree


10
5
0
Frequency
0.02
0.03
Probability of spam

Bagging (Cont'd)

- Resample the data with replacement and fit a tree-model
- Create a prediction with the fitted model
- Repeat these steps B times, and compute the average prediction
- Intuition:
  - The variance of the average predictions is smaller than the variance of the individual prediction
- Bootstrapping: random sampling with replacement
  - Often used to estimate the sampling distribution of statistics
  - Example: the distribution of the t-statistics (t-distribution) relies on asymptotic approximations. In finite samples, the distribution of the t-statistic can be obtained via bootstrapping

Issues related to bagging

1. Out-of-Bagging Error: accuracy based on left out observations
2. Black box method (average of many trees) – hard to interpret the importance of covariates
    - Solution: variance importance graphs
3. Limitation: trees will be correlated, which may result in high variance predictions
    - Potential cause: presence of a strong predictor
    - Why?

Bagging: summary

- Bootstrap samples, and average predictions
- Pros:
  - Lower prediction variance relative to trees
  - No need to prune
  - No need to perform cross-validation when using out-of-bagging errors
  - The idea of bagging can be applied to other statistical learning methods
- Cons:
  - Hard to interpret the predictions
  - Trees can still be correlated, which increases the prediction variance

Random Forests

- Aims at reducing the correlation between trees (cf. bagging)
- For a given bootstrap draw:
  - For each split, use a random subset of predictors (m)
  - If there are p predictors, $m = \sqrt{p}$ is a typical choice
- The random forest prediction is the average of the bootstrapped predictions
- Intuition:
  - Important predictors are not always considered when splitting the tree
  - E.g., on average, an important predictor is not even considered in $(p - m)/p$ of the splits
- Why bagging is a special *random forest*?

Boosting

- Instead of bootstrapping trees (like bagging or random forests), we grow trees sequentially

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$
3. Output the boosted model,
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$

Boosting (Cont'd)

- Overcomes the overfitting problems of large decision trees by growing the tree slowly (see λ)
- By fitting small additional trees (d is small) we slowly improve the prediction
- Parameters:
  - B: number of trees
  - d: the depth of the trees
  - λ: learning rate (shrinkage parameter)
- Advantage over random forests: smaller trees (more like an additive model) make interpretation easier
- We need to cross-validation to avoid overfitting

Summary

- Bagging, random forests and boosting could improve decison-tree-based predictions
- Bagging and random forests are based on bootstrapping, i.e., random resampling with replacement
- Bagging is a special random forest, where $m = p$
- Model interpretability is problemativ for random forests
- Boosting grows trees sequentially