random forest

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

A random forest is an ensemble of decision trees. Each tree is trained on a bootstrap sample of the training set, and each split may choose only among a small random subset of the features. The predictions of the trees are aggregated by a majority vote in classification or by averaging in regression. The two sources of randomness decorrelate the trees, so that averaging reduces the variance of the aggregated prediction. Adding trees does not lead to overfitting: the generalization error converges to a limit that is bounded via the strength of the trees and the correlation between them. A random forest also yields a widely used form of feature importance, read off from the features its trees split on.

Definition

A bank must decide, for each loan application, whether the applicant is creditworthy. A random forest answers by asking many different decision trees and following their majority: it is an ensemble of decision trees, each trained on a bootstrap sample of the training set (as in bootstrap aggregating (bagging)) and further randomized by allowing each split to choose only among a small random subset of the features (Breiman, 2001). The predictions of the trees are aggregated by a majority vote in classification or by averaging in regression. Fig. 1 shows a regression example: each data point is one day at the weather station Krems, its feature the minimum and its label the maximum air temperature of the day. Three decision trees, each trained on its own bootstrap sample of the days, yield three different piecewise constant hypotheses; the random forest averages them.

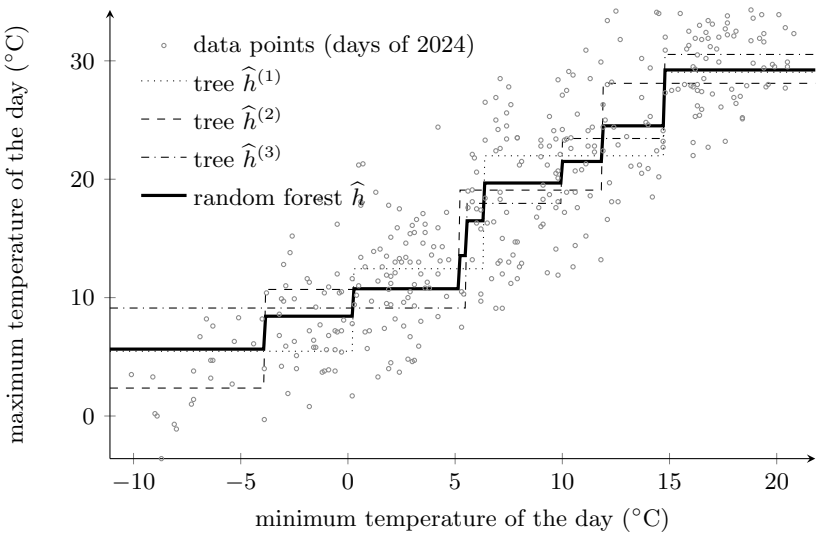

Figure 1: Scatterplot of the days of 2024 at the weather station Krems, with the minimum temperature of a day as feature and its maximum temperature as label. Three decision trees $\widehat{h}^{(1)}, \widehat{h}^{(2)}, \widehat{h}^{(3)}$, each trained on its own bootstrap sample of the days, are piecewise constant maps; the random forest $\widehat{h} = \frac{1}{3}\big(\widehat{h}^{(1)} + \widehat{h}^{(2)} + \widehat{h}^{(3)}\big)$ averages them. Data generated by `pythondemos/randomforest.py`

The ensemble idea behind this construction is that averaging many individually unreliable predictions yields a more reliable one, provided the errors of the individual predictions do not align. A deep decision tree is an ideal base learner for this purpose: it can fit the training set closely (small bias), but its prediction varies strongly with the training set (large variance). Averaging attacks precisely this variance: for B identically distributed predictions, each with variance σ^2 and with pairwise correlation ρ, the averaged prediction has variance

$$\rho\sigma^2 + \frac{1-\rho}{B}\,\sigma^2,$$

which decreases with the number B of trees toward the floor $\rho\sigma^2$ set by the correlation (Hastie et al., 2009, Sect. 15.2). Averaging leaves the bias unchanged, since each tree has the same expected prediction. In Fig. 1, the averaged curve is visibly less ragged than each single tree, and its squared-error risk on the depicted days is smaller than the average of the trees' risks.

The two sources of randomness serve one purpose: they make the trees less alike (Fig. 2). Averaging reduces the variance of a prediction most when the averaged predictions are weakly correlated, and restricting each split to a random subset of features prevents all trees from reusing the same dominant features (Hastie et al., 2009, Sect. 15.2). A random forest also reports which features its trees split on most productively — a widely used form of feature

importance.

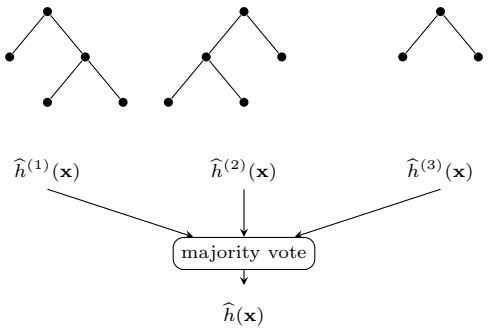

Figure 2: A random forest with three decision trees. Each tree is trained on its own bootstrap sample of the training set, with each split restricted to a random subset of the features. For a feature vector $\mathbf{x}$, the prediction $\widehat{h}(\mathbf{x})$ is the majority vote of the tree predictions $\widehat{h}^{(1)}(\mathbf{x}), \ldots, \widehat{h}^{(3)}(\mathbf{x})$

A random forest also admits precise guarantees. As trees are added, the generalization error of the forest converges almost surely to a limiting value: a random forest does not suffer from overfitting as the number of trees grows (Breiman, 2001, Thm. 1.2). The limiting error is at most $\bar{\rho}\,(1-s^2)/s^2$, where the strength s is the expected vote margin (the expected gap between the fraction of trees that vote for the correct label of a data point and the largest fraction that votes for any wrong one) and $\bar{\rho}$ is the mean correlation between the raw margin functions of two independently drawn trees (Breiman, 2001, Thm. 2.3). The bound is loose, but it names the two levers that the randomization operates on: strong trees and weak correlation.

The averaging also yields stability: the learned hypothesis depends only weakly on any single data point of the training set. Consider the subbagged version of any machine learning (ML) method with predictions in $[0, 1]$: each base learner is trained on a share p of the m data points, drawn without replacement, and the predictions are averaged over all such subsamples. Then, removing a single data point from the training set changes the prediction at a test point by more than ε for at most a δ-fraction of the removed data points, for every pair (ε, δ) with $\delta\varepsilon^2 \geq \frac{1}{4(m-1)} \cdot \frac{p}{1-p}$ (Soloff et al., 2024, Thm. 8). This guarantee holds for every training set and requires nothing of the base learner beyond the bounded predictions — the stability is created by the averaging itself.

In the credit-scoring application, the majority vote over hundreds of trees is far less sensitive to a few unusual customer records than any single decision tree, and the feature importance readout indicates which applicant features drive the decisions.

Cross References

ensemble, bagging, decision tree, bootstrap, feature importance, variance, stability

References

1. Breiman (2001). Random Forests. Machine Learning, 45(1), pp. 5–32.
2. Soloff et al. (2024). Bagging Provides Assumption-free Stability. J. Mach. Learn. Res., 25(131), pp. 1–35.
3. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.