

“RandomForest” for Powerful Stable Prediction

Main ingredient ideas:

1. trees
2. bootstrap
3. aggregation (→ bagging, ensemble learning, see chapter 8)

The “random forests” or “RandomForest” algorithm (for both classification and regression) consists of the following steps:¹

1. Draw n_{tree} bootstrap samples from the original data.
2. For each of the bootstrap samples, grow an *unpruned* classification or regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m_{try} of the predictors and choose the best split from among those variables.
3. Predict new data by *aggregating* the predictions of the n_{tree} trees (i.e., majority votes for classification, average for regression).

Note that the usual “Bagging” (:= “Bootstrap aggregating”) can be thought of as the special case of random forests obtained when $m_{\text{try}} = p$, the number of predictors.

An estimate of the error rate can be obtained, based on the training data, by the following:

1. At each bootstrap iteration, predict the data not in the bootstrap sample (what Breiman calls “out-of-bag”, or **OOB**, data) using the tree grown with the bootstrap sample.
2. Aggregate the OOB predictions. (On the average, each data point would be out-of-bag around 36% of the times, so aggregate these predictions.) Calculate the error rate, and call it the OOB estimate of error rate.

Our experience has been that the OOB estimate of error rate is quite accurate, given that enough trees have been grown (otherwise the OOB estimate can bias upward; see Bylander(2002).

Extra information from Random Forests :

The `randomForest` R package optionally produces two additional pieces of information:

1. a measure of the importance of the predictor variables
2. a measure of the internal structure of the data (the proximity of different data points to one another).

¹adapted from *RNews (3) 2002, p.18*