

1 Bagging and Boosting

These are different approaches to improve the performance of your model (so-called meta-algorithms):

Bagging (stands for Bootstrap Aggregation) is the way decrease the variance of your prediction by generating additional data for training from your original dataset using combinations with repetitions to produce multisets of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome.

Boosting is a an approach to calculate the output using several different models and then average the result using a weighted average approach. By combining the advantages and pitfalls of these approaches by varying your weighting formula you can come up with a good predictive force for a wider range of input data, using different narrowly tuned models.

Stacking is a similar to boosting: you also apply several models to you original data. The difference here is, however, that you don't have just an empirical formula for your weight function, rather you introduce a meta-level and use another model/approach to estimate the input together with outputs of every model to estimate the weights or, in other words, to determine what models perform well and what badly given these input data. As you see, these all are different approaches to combine several models into a better one, and there is no single winner here: everything depends upon your domain and what you're going to do. You can still treat stacking as a sort of more advances boosting, however, the difficulty of finding a good approach for your meta-level makes it difficult to apply this approach in practice.

Short examples of each:

Bagging: Ozone data. Boosting: is used to improve optical character recognition (OCR) accuracy. Stacking: is used in K-fold cross validation algorithms.

Bagging and boosting are meta-algorithms that pool decisions from multiple classifiers

1.1 Overview on Bagging

- Invented by Leo Breiman: Bootstrap aggregating.

- L. Breiman, Bagging predictors, Machine Learning, 24(2):123-140, 1996.
- Majority vote from classifiers trained on bootstrap samples of the training data.
- Generate B bootstrap samples of the training data: random sampling with replacement.
- Train a classifier or a regression function using each bootstrap sample.
- For classification: majority vote on the classification results.
- For regression: average on the predicted values.
- Reduces variation.
- Improves performance for unstable classifiers which vary significantly with small changes in the data set, e.g., CART.
- Found to improve CART a lot, but not the nearest neighbor classifier.

1.2 Overview on Boosting

- Iteratively learning weak classifiers
- Final result is the weighted sum of the results of weak classifiers.
- Many different kinds of boosting algorithms: Adaboost (Adaptive boosting) by Y. Freund and R. Schapire is the first.
- Examples of other boosting algorithms:
- LPBoost: Linear Programming Boosting is a margin-maximizing classification algorithm with boosting.
- BrownBoost: increase robustness against noisy datasets. Discard points that are repeatedly misclassified.
- LogitBoost: J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics, 28(2), 337-407, 2000.

1.3 The bag function

The `bag` function offers a general platform for bagging classification and regression models. Like `rfe` and `sbfi`, it is open and models are specified by declaring functions for the model fitting and prediction code (and several built-in sets of functions exist in the package). The function `bagControl` has options to specify the functions (more details below).

The function also has a few non-standard features:

- The argument `var` can enable random sampling of the predictors at each bagging iteration. This is to de-correlate the bagged models in the same spirit of random forests (although here the sampling is done once for the whole model). The default is to use all the predictors for each model.
- The `bagControl` function has a logical argument called `downSample` that is useful for classification models with severe class imbalance. The bootstrapped data set is reduced so that the sample sizes for the classes with larger frequencies are the same as the sample size for the minority class.
- If a parallel backend for the `foreach` package has been loaded and registered, the bagged models can be trained in parallel.