Random Forests

Random forests is an ensemble learning algorithm. The basic premise of the algorithm is that building a small decision-tree with few features is a computationally cheap process. If we can build many small, weak decision trees in parallel, we can then combine the trees to form a single, strong learner by averaging or taking the majority vote. In practice, random forests are often found to be the most accurate learning algorithms to date. The pseudocode is illustrated in Algorithm 1.

The algorithm works as follows: for each tree in the forest, we select a bootstrap sample from S where $S^{(i)}$ denotes the ith bootstrap. We then learn a decision-tree using a modified decision-tree learning algorithm. The algorithm is modified as follows: at each node of the tree, instead of examining all possible feature-splits, we randomly select some subset of the features $f \subseteq F$. where F is the set of features. The node then splits on the best feature in f rather than F. In practice f is much, much smaller than F. Deciding on which feature to split is oftentimes the most computationally expensive aspect of decision tree learning. By narrowing the set of features, we drastically speed up the learning of the tree.

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForest(S, F)
    H ← ∅
    for i ∈ 1, ..., B do
        S^(i) ← A bootstrap sample from S
        h_i ← RandomizedTreeLearn(S^(i), F)
        H ← H ∪ {h_i}
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        f ← very small subset of F (without replacement)
        Split on best feature in f
    return The learned tree
end function
```

Why do random forests work?

The random forest algorithm uses the bagging technique for building an ensemble of decision trees. Bagging is known to reduce the variance of the algorithm.

However, the natural question to ask is why does the ensemble work better when we choose features from random subsets rather than learn the tree using the traditional algorithm? Recall, that ensembles are more effective when the individual models that comprise them are uncorrelated. In traditional bagging with decision-trees, the constituent decision trees may end up to be very correlated because the same features will tend to be used repeatedly to split the bootstrap samples. By restricting each split-test to a small, random sample of features, we can decrease the correlation between trees in the ensemble.

Furthermore, by restricting the features that we consider at each node, we can learn each tree much faster, and therefore, can learn more decision trees in a given amount of time. Thus, not only can we build many more trees using the randomized tree learning algorithm, but these trees will also be less correlated. For these reasons, random forests tend to have excellent performance.