Comprehensive Review: Decision Trees, Boosting, and Random Forests

Master's Level Data Science

Contents

1 Decision Trees

Decision trees partition the feature space via axis-aligned splits to classify data. Starting from the root node containing all samples, the greedy algorithm selects at each leaf the split that maximally reduces an uncertainty measure.

1.1 Uncertainty Measures

For a node with class probabilities $p_1, \dots, p_K$:

$$\text{Misclassification rate: } u_{\text{mis}} = 1 - \max_i p_i,$$

$$\text{Gini index: } u_{\text{gini}} = 1 - \sum_{i=1}^{K} p_i^2,$$

$$\text{Entropy: } u_{\text{ent}} = -\sum_{i=1}^{K} p_i \ln p_i.$$

The benefit of a split of S into S_L, S_R with fractions p_L, p_R is

$$\Delta u = u(S) - \big(p_L\, u(S_L) + p_R\, u(S_R)\big).$$

1.2 Building and Pruning

1. **Grow full tree:** split until leaves are pure or meet stopping criteria.
2. **Prune:** collapse branches using validation error (cost-complexity pruning).

2 Boosting Weak Learners

Boosting combines weak learners into a strong classifier by reweighting examples. AdaBoost is the canonical algorithm.

2.1 AdaBoost Blueprint

Given (x_i, y_i), $y_i \in \{-1, +1\}$, initialize $D_1(i) = 1/n$. For $t = 1, \dots, T$:

1. Train weak learner h_t on distribution D_t.
2. Compute error $\varepsilon_t = \sum_i D_t(i)\, 1[h_t(x_i) \neq y_i]$.
3. Set weight $\alpha_t = \frac{1}{2} \ln\big((1 - \varepsilon_t)/\varepsilon_t\big)$.
4. Update and normalize:

$$D_{t+1}(i) \propto D_t(i)\, \exp\big(-\alpha_t\, y_i\, h_t(x_i)\big).$$

Final classifier:

$$H(x) = \operatorname{sign}\Big(\sum_{t=1}^{T} \alpha_t\, h_t(x)\Big).$$

2.2 Freund–Schapire Example

Round t	r_t	α_t	Comment
1	0.40	0.42	weak stump
2	0.58	0.65	refocus on errors
3	0.72	0.92	strong stump

Combined classifier:

$$H(x) = \operatorname{sign}(0.42\, h_1 + 0.65\, h_2 + 0.92\, h_3).$$

3 Random Forests

Random forests build T trees in parallel on bootstrap samples and random feature subsets, then aggregate by majority vote.

3.1 Algorithm

Given data S of size n and feature dimension d:

1. For $t = 1, \dots, T$ in parallel:
   (a) Sample n' points with replacement from S.
   (b) Grow tree h_t: at each node, consider a random subset of k features (e.g. $k = \lfloor \sqrt{d} \rfloor$) for the best split.
2. Predict $H(x) = \text{majority\_vote}\{h_t(x)\}$.

3.2 Covertype Dataset Results

- Single tree (depth 20): train error 1%, test error 12.6%.
- Boosted trees (10 trees, depth 20): test error 8.7%.
- Random forest (10 trees, 50% features, depth 40): test error 8.8%.

4 Interpretation and Guidelines

- Decision trees are high-variance; pruning or ensembling mitigates overfitting.
- Boosting reduces bias and variance via sequential reweighting.
- Random forests reduce variance by aggregating diverse trees.
- Hyperparameters (tree depth, T, k, sample size) should be tuned via cross-validation.