Stacking Algorithm for Ensemble Modelling

Frederik Schreck

Course: Numerical Introductory Course
Lecturer: Prof. Dr. Brenda López Cabrera
Humboldt–Universität zu Berlin

Motivation - The wisdom of the crowd

- The aggregation of individual guesses in groups is often superior to individual guesses - even to experts
- BUT: Only fulfilled under certain criteria
  - Variation of guesses
  - Independence of guesses
  - Decentralization
  - Algorithm

Outline

Decision Tree

- Idea: use a set of splitting rules to recursively partition the dataset.
- Classification trees:
  - Minimize impurity within nodes
- Regression trees:
  - Minimize variance of the response variable within nodes

Decision Tree for classification

- Choice of splitting rule: maximizing information gain (IG) by decreasing node impurity (I)

$$IG_n = I_n - p_{n_1} * I(n_1) - p_{n_2} * I(n_2), \tag{1}$$

for node n with branching nodes n_1 and n_2, and p_{n_i} as the fraction of cases in branching node n_i

Decision Tree for classification

- How to measure impurity? Choices of splitting criteria:

$$\text{Entropy:} I(n) = -\sum_{j}^{J} p(c_j|n) * \log_2(p(c_j|n)) \tag{2}$$

$$\text{Gini impurity:} I(n) = 1 - \sum_{j}^{J} p(c_j|n)^2 \tag{3}$$

$$\text{Misclassification impurity:} I(n) = 1 - \max_j p(c_j), \tag{4}$$

for classes $c_j, j \in J = \{1, 2, ...\}$

Decision Tree for classification

- Choice of stopping rule:
  A fully grown tree has pure leaf nodes and may overfit the data
  However, a too small tree may not capture all relevant structure of the data
  - Pre-pruning
  - Post-pruning

Ensemble Learning - Terminology

Machine Learning

- Part of computer science that uses statistical techniques to train models on data
- Typically used for prediction purposes

Stacking and Ensemble Learning

- Idea is to combine hypotheses of multiple learning algorithms (base learners)
- Goal is to obtain a better predictive performance than with each of the single algorithms alone
- Mainly used in supervised learning
- Very flexible method

Ensemble Learning

Which models to combine?

- Effective ensembling builds on diverse and little correlated models
- Best to use strong base learners

Similar criteria as mentioned in the Motivation!

Ensemble Learning

Which models to combine?

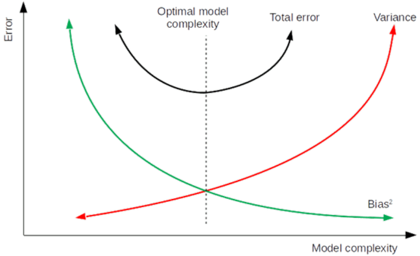

Figure 1: The bias-variance-trade-off.

- Combining complex classifiers may reduce variance.
- Combining simple classifiers may reduce bias.

Bagging (= Bootstrap Aggregating)

- Proposed by Leo Breiman
- Meta-algorithm, designed to
  - improve accuracy of base algorithms
  - reduce MSE by reducing variance
  - avoid overfitting problems
  - obtain smoother prediction boundaries
- Can be applied to all kinds of base learners
- However best to use unstable methods that tend to have high variance, like trees

Bagging algorithm

Suppose we have training data $\{(x_1, y_1), \dots, (x_N, y_N)\}$

---

for base learner m in $\{1, 2, \dots, M\}$
    uniformly draw sample D_m from dataset D (with repl.)
    build model T_m on dataset D_m to obtain hypothesis $h_m(x)$
combine hypotheses

- Combining by averaging in regression problems
- Combining by majority vote in classification problems

Random Forest

- Also proposed by Leo Breiman
- Random forests combine bagging with random subspace approach
- Random subspace randomly samples features from set of all features for each learner (with replacement)
  - Reduces the correlation between estimators
  - Thus decreases variance in the ensemble learner
- Random feature sampling happens at tree level or at split level
- Random Forest only possible with tree-based base learners

Random Forest algorithm for classification

Suppose we have training data $\{(x_1, y_1), ..., (x_N, y_N)\}$

---

for base learner m in $\{1, 2, ..., M\}$
    uniformly draw sample k_m of size L from features $\{1, 2, ..., K\}$ (with repl.)
    uniformly draw sample D_m from dataset D (with repl.)
    build model T_m on dataset D_m using feature set k_m
$\hat{C}_{rf}^{L,N}(x) = \text{majority vote}\{\hat{T}_m\}_1^M$

Random Forest

Random Forest vs. single Tree

Random Forest	Single Tree
– higher computational costs – blackbox + easy to tune parameters + smaller prediction variance + scalability – many parameter choices to make	+ computationally simple + insights into decision rules + easy to tune parameters – tends to overfit and have high variance

Boosting

- Method proposed by Freund & Shapire
- Original idea only applies to classification problems
- Idea: simple learners are easier to find. Combining many simple learners can produce a powerful learner.
- The ensemble first considers only one base learner. Then we iteratively enlarge it by another base learner that aims to correct the error of the current ensemble.

The Adaboost algorithm

Suppose we have training data $\{(x_1, y_1), ..., (x_N, y_N)\}$,
initialize weightings $d_i^{(1)} = \frac{1}{N}, \forall i \in \{1, ..., N\}$

---

for base learner m in $\{1, 2, ..., M\}$
 train base learner according to weighted data $d^{(m)}$
 and obtain hypothesis $h_m : \mathbf{x} \mapsto \{-1, +1\}$
 calculate weighted classifier error
 $\epsilon_m = \sum_{i=1}^{N} d_i^m I(y_i \neq h_t(x_i))$
 calculate hypothesis weighting $\beta_m = \frac{1}{2}\log(\frac{1-\epsilon_m}{\epsilon_m})$
 update data weighting, e.g. by
 $h_m(x_i) = y_i : d_i^{m+1} = d_i^m \exp(-\beta_m)$
 $h_m(x_i) \neq y_i : d_i^{m+1} = d_i^m \exp(\beta_m)$
$\hat{y}(x) = H_{final}(x) = \frac{1}{M}\sum_1^M \beta_m h_m(x)$

Gradient Boosting

- Developed by Friedman
- Extended boosting to regression problems
- Shortcomings of current ensemble is identified by gradients instead of weightings of data
- In each stage m, a new learner improves the current ensemble H_{m-1} and is fitted to $(x_i, y_i - H_{m-1}(x_i)), \forall i \in \{1, 2, ..., N\}$

Stochastic Gradient Boosting

- Advancement of Gradient Boosting, again by Friedman
- Takes ideas from Bagging:
  - Using trees as base learners
  - Fit trees to negative gradient of random sample of dataset
  - Less prune to overfitting

Gradient Boosting

Random Forest vs. single Tree vs. Gradient Boosting

Random Forest	Single Tree	Gradient Boosting
– higher computational costs – blackbox + easy to tune parameters + smaller prediction variance + scalability – many parameter choices to make	+ computationally simple + insights into decision rules + easy to tune parameters – tends to overfit and have high variance	+ relatively fast to train + insights by feature importance and partial dependence plots + one of the best of-the-shelf methods – tends to overfit – parallelization difficult – many tunable parameters

Bayes??

Stacked Generalization

Potentials of Ensemble Learning

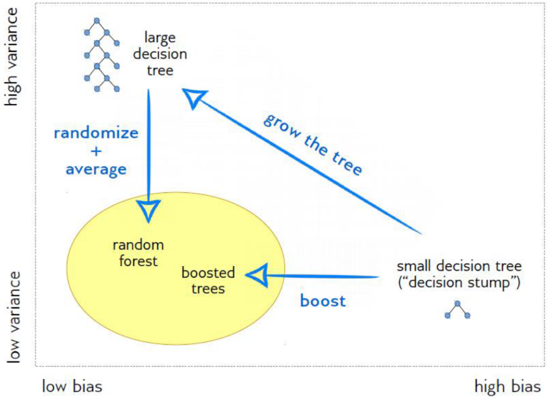

Figure 2: How Gradient Boosting and Random Forest improve performance.

Potentials and Problems of Ensemble Learning

Potentials	Problems
+ currently best predictive methods available	– needs high computational resources
+ ensembling decreases variance and bias	– blackbox problems
+ often scalable	– many parameters to tune
+	– lack of proven statistical properties
+	
+	

Current research

- Scalability
- Evolving statistical properties
- ?Out of the black box

Sources

Friedman, J. H. (2001).
Greedy function approximation: a gradient boosting machine.
Annals of statistics, pages 1189–1232.