# 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

1. Motivation ✓
2. Decision Tree
3. Ensemble Learning
4. Stacking algorithms
   - 4.1 Bagging and Random Forest
   - 4.2 Boosting and Gradient Boosting
   - 4.3 Bayes??
   - 4.4 Stacked Generalization
5. Potentials and Problems of Ensemble Learning
6. The German Credit Dataset
7. Sources

# 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 $n_i$

- 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?

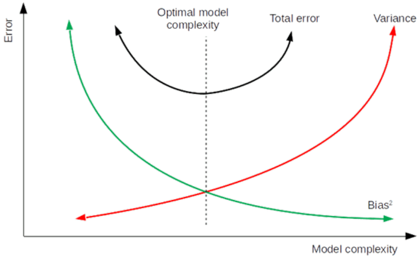


Figure 1: The bias-variance-trade-off.

- 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

---

for base learner $m$ in $\{1, 2, ..., M\}$
    uniformly draw sample $D_m$ with size $N$ from dataset $D$
    (with replacement)
    build model $T_m$ on dataset $D_m$
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

---

for base learner $m$ in $\{1, 2, ..., M\}$

  uniformly draw sample $k_m$ of size $L$ from features $\{1, 2, ..., K\}$
  (with replacement)

  uniformly draw sample $D_m$ with size $N$ from dataset $D$
  (with replacement)

  build model $T_m$ on dataset $D_m$ with feature set $k_m$

$\hat{C}_{rf}^{L,N}(x) = \text{majority vote}\{\hat{T}_m\}_1^M$

---

# Random Forest algorithm for classification

Random Forest vs. single Tree

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

# Boosting

# Gradient Boosting

# Bayes??

# Stacked Generalization

# Potentials of Ensemble Learning

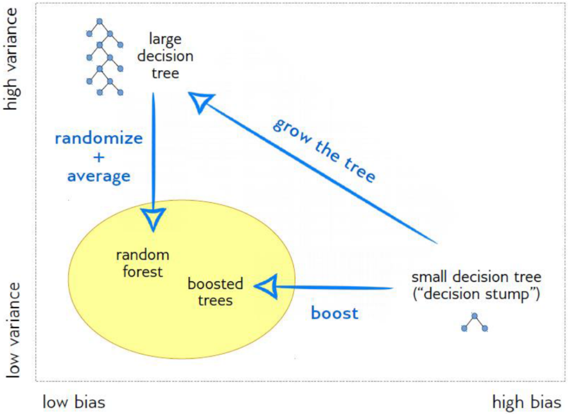


Figure 2: How Gradient Boosting and Random Forest improve performance.

# Problems of Ensemble Learning

# Sources

Breiman, L. (2001).
Random forests.
*Machine learning*, 45(1):5–32.