Supervised Learning II

Christoph Kern

Mannheim Machine Learning Modules

c.kern@uni-mannheim.de

October 20, 2017

UNIVERSITY OF MANNHEIM

Outline

Random Forests

Some limitations of (single) trees

- Difficulties in modeling additive structures
- Lack of smoothness of prediction surface
- High variance / **instability** due to hierarchical splitting process

→ Ensemble methods

- Address instability via combining multiple prediction models
- Can be applied to different base learners (e.g. CART)

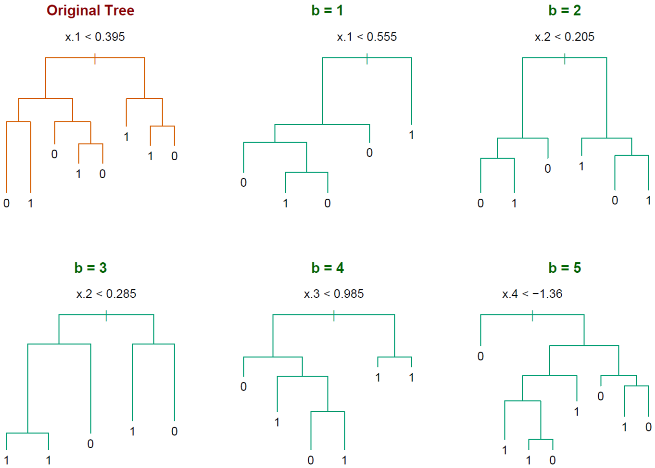
Bootstrap and Bagging

Bootstrap: Sampling B samples of size n with replacement from original data set

Applications

- Estimate the variability of model parameters
  - e.g. standard errors of regression coefficients
- Estimate test error with training data
  - Fit model on bootstrap samples and predict original training set
- Construct an ensemble of learners for prediction
  - **Bagging**: Bootstrap Aggregating
  - Train prediction models on bootstrap samples

Figure: Bagging Trees



Hastie et al. 2009

Growing a Forest

From Bagging to Random Forests

Variance of an average of B i.i.d. random variables

$$\frac{1}{B}\sigma^2$$

$\rightarrow$ Bagging: Averaging over B trees decreases variance

Variance of an average of B i.d. random variables with $\rho > 0$

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

$\rightarrow$ **Random Forests**: Averaging over B trees with m out of p predictors per split decreases variance and decorrelates trees

Algorithm 1: Grow a Random Forest

```
Set number of trees B;
Set predictor subset size m;
Define stopping criteria;
for b = 1 to B do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        draw a random sample m from the p predictors;
        find the optimal split point among m;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

A Random Forest

$$\{T_b\}_1^B$$

consists of a set of $b = 1, 2, \ldots, B$ trees which can be used for prediction by...

- Regression
  - ...averaging predictions over all trees
  - $\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$
- Classification
  - ...using most commonly occurring class among all trees
  - $\hat{C}_{rf}^B(x) = \text{majority vote}\{\hat{C}_b(x)\}_1^B$

OOB error and tuning

Observations in each bootstrap sample

$$
\begin{aligned}
P(\text{obs}\, i \in \text{sample}\, b) &= 1 - \left(1 - \frac{1}{n}\right)^{n} \\
&\approx 1 - e^{-1} \\
&= 0.632
\end{aligned}
$$

Out-of-bag (OOB) error

- Sampling with replacement leads to models based on subsets of the data
- Unused (OOB) observations can be used for test error estimation
  1. Generate predictions for case i using models where i was OOB
  2. Average predictions for i and estimate test error
  3. Compute OOB error over all cases

Tuning Random Forests

- Predictor subset size m out of p
  - Most important tuning parameter in RF
  - Starting value; $m = \sqrt{p}$ (classification), $m = p/3$ (regression)
  - Can be chosen using OOB errors based on different m
- Optional: Number of trees
  - sufficiently high (e.g. 500)
- Optional: Node size (number of observations in terminal nodes)
  - sufficiently low (e.g. 5)

Outlook on Boosting

Boosting

- Class of ensemble methods which combine **sequential** prediction models
- Adaptive approach with focus on "difficult observations"
- Different flavors exist
  - AdaBoost
  - Gradient Boosting Machines (GBM)
  - ...
- Can be applied to different (weak) base learners
  - Boosting trees
  - ...

Algorithm 2: Gradient Boosting for regression

Set number of trees B;
Set interaction depth D;
Set shrinkage parameter λ;
Use $\bar{y}$ as initial prediction;
for *b=1 to B* **do**
    compute residuals based on current predictions;
    assign data to root node, using the residuals as the outcome;
    while *current tree depth* $< D$ **do**
        tree growing process;
    end
    compute the predicted values of the current tree;
    add the shrinked new predictions to the previous predicted values;
end

Tuning Gradient Boosting Machines

- Number of trees B
  - Number of "iterations"
  - Overfitting can occur for large B
- Interaction depth D
  - Number of splits for each tree
  - Boosting stumps: $D = 1$
- Shrinkage parameter λ
  - Learning rate, slows down learning process
  - e.g. $\lambda = 0.01$, $\lambda = 0.001$
- ...

Resources

- R: ML Task View, caret & `mlr`
  - `https://cran.r-project.org/web/views/MachineLearning.html`
  - `https://topepo.github.io/caret/`
  - `https://mlr-org.github.io/mlr-tutorial/devel/html/`
- Competitions and community
  - `https://www.kaggle.com/`
  - `https://www.openml.org/`
- Books
  - `http://www.springer.com/de/book/9781461468486`
  - `http://www.springer.com/de/book/9783319440477`
- Data
  - `https://archive.ics.uci.edu/ml/datasets.html`

References

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.