Introduction to Machine Learning

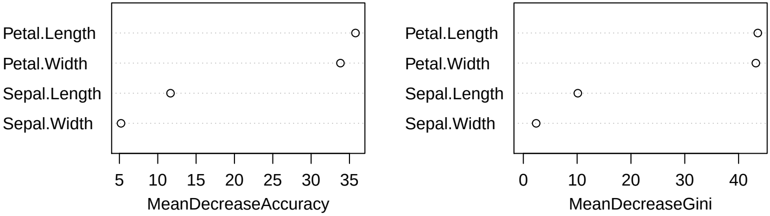
Bagging and Random Forest 2

Bernd Bischl, Christoph Molnar, Daniel Schalk, Fabian Scheipl

Department of Statistics – LMU Munich

VARIABLE IMPORTANCE

- Single trees are highly interpretable
- Random Forests as combinations of trees loose this feature
- Contributions of features to model are difficult to evaluate
- Way out: variable importance measures



VARIABLE IMPORTANCE

Measure based on permutations of OOB observations

1: While growing tree, pass down OOB observations and record predictive accuracy.
2: Permute OOB observations of j-th variable.
3: Pass down the permuted OOB observations and evaluate predictive accuracy again.
4: The loss of goodness induced by permutation is averaged over all trees and is used as a measure for the importance of the j-th variable.

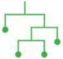
Measure based on improvement in split criterion

1: At each split in tree $\hat{b}^{[m]}(x)$ the improvement in the split criterion is attributed as variable importance measure for the splitting variable.
2: For each variable, this improvement is accumulated over all trees for the importance measure.

VARIABLE IMPORTANCE BASED ON PERMUTATIONS OF OOB OBSERVATIONS

	x_1	...	x_p	y	$\hat{y}$
1	1.4			1	1
2	2			0	0
3	1.55			1	1
4	1.72			0	0
5	1.89			1	0
⋮					
n	2.01			1	1

Tree 1



Tree M

Inbag + oob of tree 1

	x_1	...	x_p	y	$\hat{y}$
1	1.4			1	
2	2			0	
3	1.55			1	1
4	1.72			0	0
5	1.89			1	
⋮					
n	2.01			1	1

Permuted oob obs. of x_1

	x_1	...	x_p	y	$\hat{y}$
1	1.4			1	
2	2			0	
3	2.01			1	0
4	1.55			0	0
5	1.89			1	
⋮					
n	1.72			1	0

.....

Inbag + oob of tree M

	x_1	...	x_p	y	$\hat{y}$
1	1.4			1	1
2	2			0	
3	1.55			1	0
4	1.72			0	
5	1.89			1	1
⋮					
n	2.01			1	

Permuted oob obs. of x_1

	x_1	...	x_p	y	$\hat{y}$
1	1.89			1	0
2	2			0	
3	1.4			1	0
4	1.72			0	
5	1.55			1	1
⋮					
n	2.01			1	

$$acc_{1,\text{ without permutation}} - acc_{1,\text{with permutation}} = diff_1 \qquad acc_{M,\text{ without permutation}} - acc_{M,\text{with permutation}} = diff_M$$

$$\frac{1}{M}\sum_{i=1}^{M} diff_i = \text{variable importance for } x_1$$

RANDOM FOREST: ADVANTAGES

- Bagging is easy to implement
- Can be applied to basically any model
- All advantages of trees propagate to the RF, especially w.r.t. preprocessing
- Easy to parallelize
- Often works well (enough)
- Integrated variable importance
- Integrated estimation of OOB error
- Can work on high-dimensional data
- Often not much tuning necessary

RANDOM FOREST: DISADVANTAGES

- Often suboptimal for regression
- Same extrapolation problem as for trees
- Harder to interpret than trees, but many extra tools are nowadays available for interpreting RFs
- Does not really optimize loss aggressively in comparison to boosting
- Implementations sometimes memory-hungry
- Prediction can be (slightly) slower, as it is an ensemble

BENCHMARK: RANDOM FOREST VS. (BAGGED) CART VS. (BAGGED) K-NN

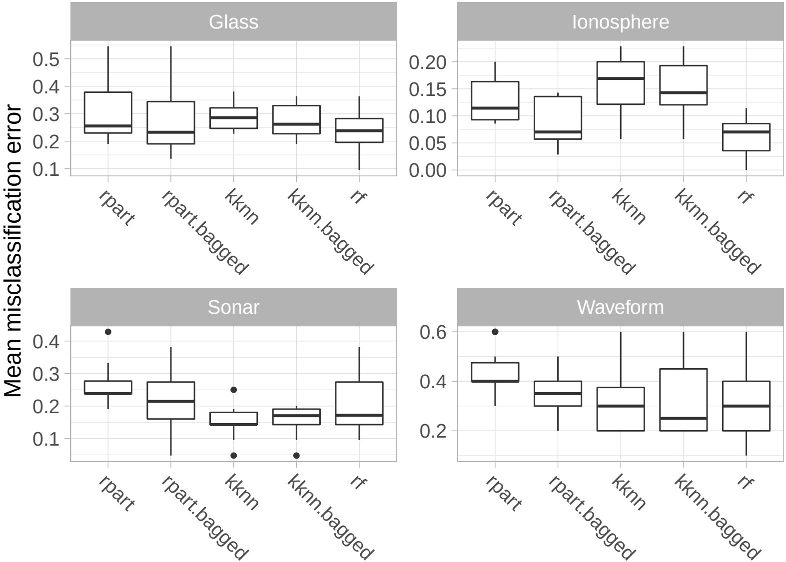
- Goal: Compare performance of random forest against (bagged) stable and (bagged) unstable methods
- Algorithms:
  - classification tree (CART, implemented in `rpart`, `max.depth`: 30, `min.split`: 20, `cp`: 0.01)
  - bagged classification tree using 50 bagging iterations (`bagged.rpart`)
  - k-nearest neighbors (k-NN, implemented in `kknn`, $k = 7$)
  - bagged k-nearest neighbors using 50 bagging iterations (`bagged.knn`)
  - random forest with 50 trees (implemented in `randomForest`)
- Method to evaluate performance: 10-fold cross-validation
- Performance measure: mean missclassification error on test sets

BENCHMARK: RANDOM FOREST VS. (BAGGED) CART VS. (BAGGED) K-NN

- Datasets from **mlbench**:

Name	Kind of data	n	p	Task
Glass	Glass identification data	214	10	Predict the type of glass (6 levels) on the basis of the chemical analysis of the glasses represented by the 10 features
Ionosphere	Radar data	351	35	Predict whether the radar returns show evidence of some type of structure in the ionosphere (“good”) or not (“bad”)
Sonar	Sonar data	208	61	Discriminate between sonar signals bounced off a metal cylinder (“M”) and those bounced off a cylindrical rock (“R”)
Waveform	Artificial data	100	21	Simulated 3-class problem which is considered to be a difficult pattern recognition problem. Each class is generated by the waveform generator.

BENCHMARK: RANDOM FOREST VS. (BAGGED) CART VS. (BAGGED) K-NN



BENCHMARK: RANDOM FOREST VS. (BAGGED) CART VS. (BAGGED) K-NN

Bagging k-NN does not improve performance because:

- k-NN is stable w.r.t. perturbations
- In a 2-class problem, nearest neighbor based classification only changes under bagging if both
  - the nearest neighbor in the learning set is **not** in at least half of the bootstrap samples, but the probability that any given observation is in the bootstrap sample is 63% which is greater than 50%,
  - and, simultaneously, the *new* nearest neighbor(s) all have a different label than the missing nearest neighbor in those bootstrap samples, which is unlikely for most regions of $\mathcal{X} \times \mathcal{Y}$.