LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Introduction to Machine Learning

Chapter 16: Random Forest cont.

Bernd Bischl, Christoph Molnar

Department of Statistics – LMU Munich

Winter term 2017/18

VARIABLE IMPORTANCE

- Single trees are highly interpretable
- Random Forests as an ensemble of many trees lose this feature
- Hence, contributions of a single covariate to the fit are difficult to evaluate
- Way out: variable importance measures

VARIABLE IMPORTANCE

Measure based on permutations of OOB observations

1: After growing tree $\hat{b}^{[m]}(x)$, pass down OOB observations and record predictive accuracy.
2: Permute OOB observations of jth 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 1

	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	

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

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

VARIABLE IMPORTANCE

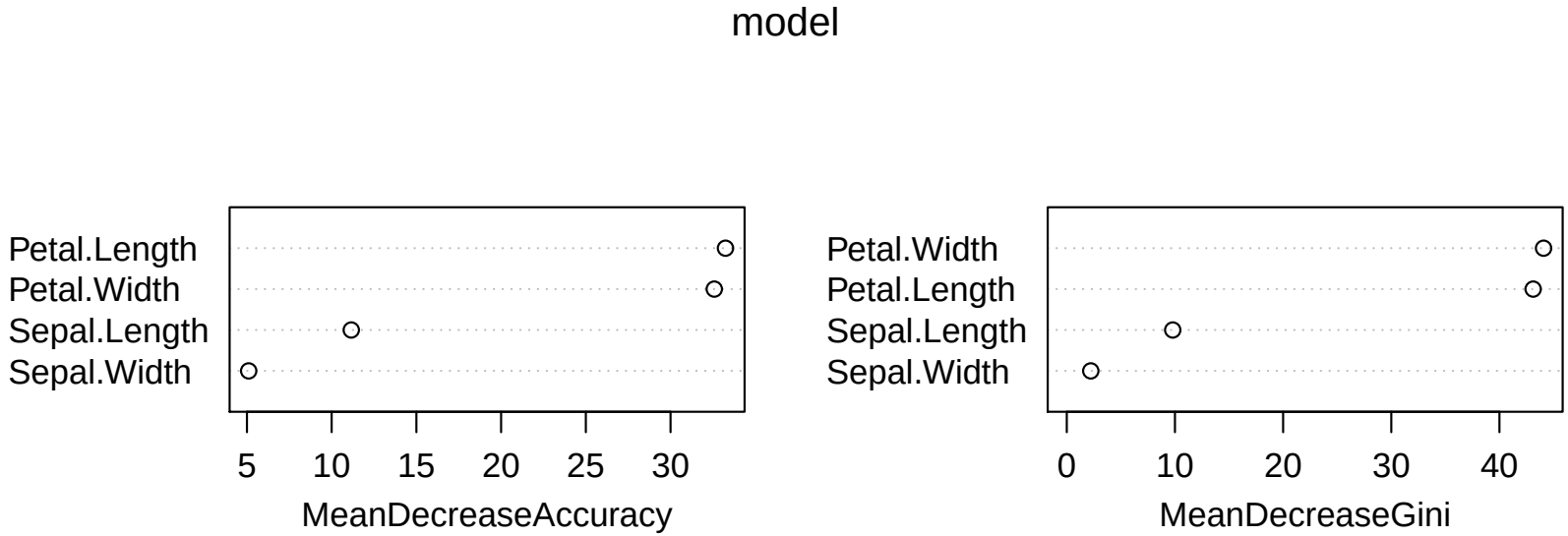


Figure: Two importance measures on iris.

VARIABLE IMPORTANCE

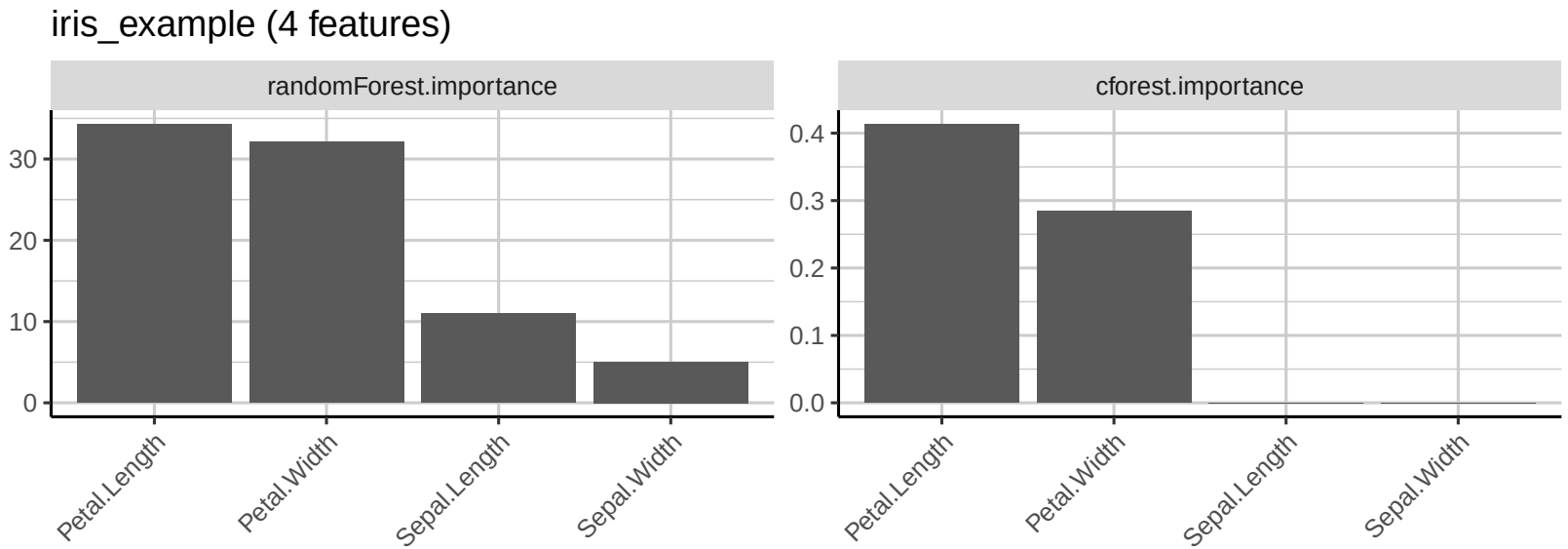


Figure: RF importance as filters in mlr.

RANDOM FOREST: PROXIMITIES

- the "closeness" or "nearness" between pairs of cases.
- Algorithm
  - After a tree is grown, put all of the data down the tree.
  - If cases x_1 and x_2 are in the same terminal node through one tree increase their proximity by one.
  - At the end of the run of all trees, normalize the proximities by dividing by the number of trees.
- The proximities originally form a NxN matrix.
- Proximities are used in replacing missing data, locating outliers, and producing illuminating low-dimensional views of the data.

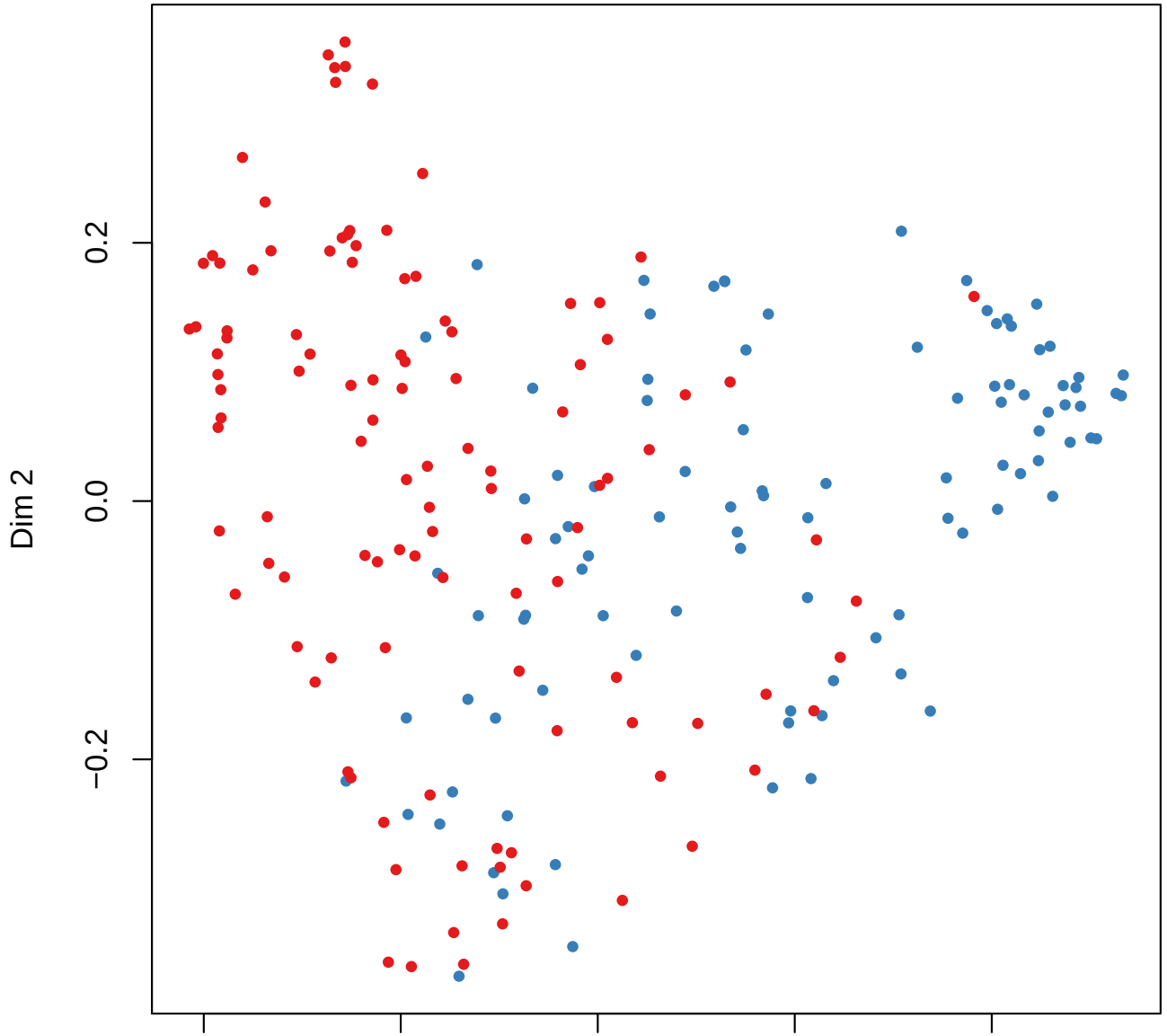
We can visualize our Proximities $P(x_i, x_j)$ for each i $\in \{1, ..., n\}$ example by MDS.
Our data contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions and 97 patterns obtained from rocks under similar conditions.

```
##       V53    V54    V55    V56    V57    V58    V59    V60 Class
## 1  0.0065 0.0159 0.0072 0.0167 0.0180 0.0084 0.0090 0.0032     R
## 2  0.0089 0.0048 0.0094 0.0191 0.0140 0.0049 0.0052 0.0044     R
## 3  0.0166 0.0095 0.0180 0.0244 0.0316 0.0164 0.0095 0.0078     R
## 4  0.0036 0.0150 0.0085 0.0073 0.0050 0.0044 0.0040 0.0117     R
## 5  0.0054 0.0105 0.0110 0.0015 0.0072 0.0048 0.0107 0.0094     R
## 6  0.0014 0.0038 0.0013 0.0089 0.0057 0.0027 0.0051 0.0062     R
## 7  0.0248 0.0131 0.0070 0.0138 0.0092 0.0143 0.0036 0.0103     R
## 8  0.0120 0.0045 0.0121 0.0097 0.0085 0.0047 0.0048 0.0053     R
## 9  0.0128 0.0145 0.0058 0.0049 0.0065 0.0093 0.0059 0.0022     R
## 10 0.0223 0.0179 0.0084 0.0068 0.0032 0.0035 0.0056 0.0040     R
```

We try to predict the type, based on signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock
We calculate the proximities $P(x_i, x_j)$ based on out-of-bag observations.

```
##
## Call:
##  randomForest(x = X, y = Y, ntree = 500, proximity = TRUE, oob.prox = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 16.8%
## Confusion matrix:
##    M  R class.error
## M 98 13       0.117
## R 22 75       0.227
```

Now we can visualize proximities $P(x_i, x_j)$ by MDS, using as distance matrix D= 1 - P


Dim 2
0.2
0.0
−0.2

ADAPTIVE NEAREST NEIGHBORS

Let $P(x, x_i) \in [0, 1]$ be the Proximity between the observation x and our original point x_i.

- For classification, the prediction will be the weighted greatest number of hits, which is proportional to the proximities $P(x, x_i)$.
- For a regression we can calculate the prediction of Random Forests at x as:
  - if every leaf node contains the same number of observations.

$$\hat{Y}_{RF}(x) = \frac{\sum_{n=0}^{n} P(x, x_i) Y_i}{\sum_{n=0}^{n} P(x, x_i)}$$

  - if some leaf node contains the different number of observations, $P(x, x_i)$ is the percentage of trees where x and x_i fall into the same leaf node, and weights are inversely proportional for each tree to the number of samples in the leaf node where x_i falls into.

RANDOM FOREST: ADVANTAGES

- Easy to implement
- Can be applied to basically any model
- Easy to parallelize
- Often works well (enough)
- Enables variance analysis
- Integrated estimation of OOB error
- Can work on high-dimensional data
- Often not much tuning necessary

RANDOM FOREST: DISADVANTAGES

- Often suboptimal for regression
- Hard to interpret, especially interactions
- Does not really optimize loss aggressively
- No real way to adapt to problem
  (see e.g. loss in GBM, kernel in SVM)
- Implementations sometimes memory-hungry
- Prediction can be slow