Random Forests

Charlotte Wickham

March 23, 2007

Random Forests

Definition

A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

- Bagging is one example. Here the Θ_i is vector representing a bootstrap sample from the training set.
- Another example is random split selection (to be discussed later). Here Θ_i is a vector of random integers indexing the input to be split on at each node.

The nature and dimension of Θ_k depends on its use in the tree construction process.

Why do random forests work?

- Weak learners have low bias but high variance.
- Averaging weak learners, over many training sets, gives estimates with low bias and low variance.

Why do random forests work?

Let the margin of the random forest be,

$$mr(X, Y) = P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(h(X, \Theta) = j).$$

And define the strength of the set of classifiers $h(X, \Theta)$ to be,

$$s = E_{X,Y} mr(X, Y).$$

Also define,

$$\bar{\rho} = E_{\Theta,\Theta'}(\rho(\Theta, \Theta')sd(\Theta)sd(\Theta'))/E_{\Theta,\Theta'}(sd(\Theta)sd(\Theta')),$$

to be the mean value of the correlation between trees.

Why do random forests work?

Can show that the generalization error of the forest, PE^* is bounded above by,

$$PE^* \leq \bar{\rho}(1 - s^2)/s^2.$$

So, for low error we want high strength and low correlation.

- High Strength
  - Grow tree to maximum depth
- Low correlation
  - Grow each tree from a bootstrap sample
  - Add some other form of randomness

Random input selection

Fix parameter K

- Draw a bootstrap sample from the training set
- Grow full size tree as usual except at each node choose K variables randomly on which to search for the best split.
- Repeat N times to generate N trees.

Breiman suggests $K = \sqrt{\text{number of variables}}$.
Implemented in R in `randomForest` .

Random linear combination of variables

Fix parameters L and F

- Draw a bootstrap sample from the training set
- Choose L variables randomly and combine them using coefficients drawn uniformly on $[-1, 1]$. Create F of these linear combinations and search for which gives the best split.
- Repeat N times to generate N trees.

Should standardize variables first. Some problems with categorical variables

- Transform to a dummy variable by picking a random selection of levels to be 1.
- Let categorical variables be $I - 1$ times as likely to be picked as continuous variables.

Not implemented in R.

Out of bag estimates.

For each tree in the forest we can use the observations that were not in the bootstrap sample to estimate measures of interest.

Misclassification rate

- For each tree find the classification for the out of bag observations
- Classification of a point is the plurality in classification over all trees in which an observation was out of bag
- Use classification to calculate error rate

Exploring the Random Forest mechanism.

Results in a black box. How can we investigate the structure of the data.

- Variable Importance
  - Use out of bag observations and randomly permute one variable.
  - Run the observations down the tree and record classification.
  - Repeat for each tree.
  - Compare the misclassification rate with the noised up variable to the out of bag estimate without permutation.
  - Variable Importance = percent increase in misclassification.
- Proximity
  - Proximity between observations i and j is the proportion of trees in which the observations occur together in a terminal node.

Supernova Data

```
Call:
 randomForest(formula = class ~ ., data = nova.train,
  importance = TRUE, proximity = TRUE, keep.forest = FALSE,
  ntree = 1000)

               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 4

        OOB estimate of  error rate: 4.2%
Confusion matrix:
      Other   SN class.error
Other  4386  114  0.02533333
SN      264 4236  0.05866667
```

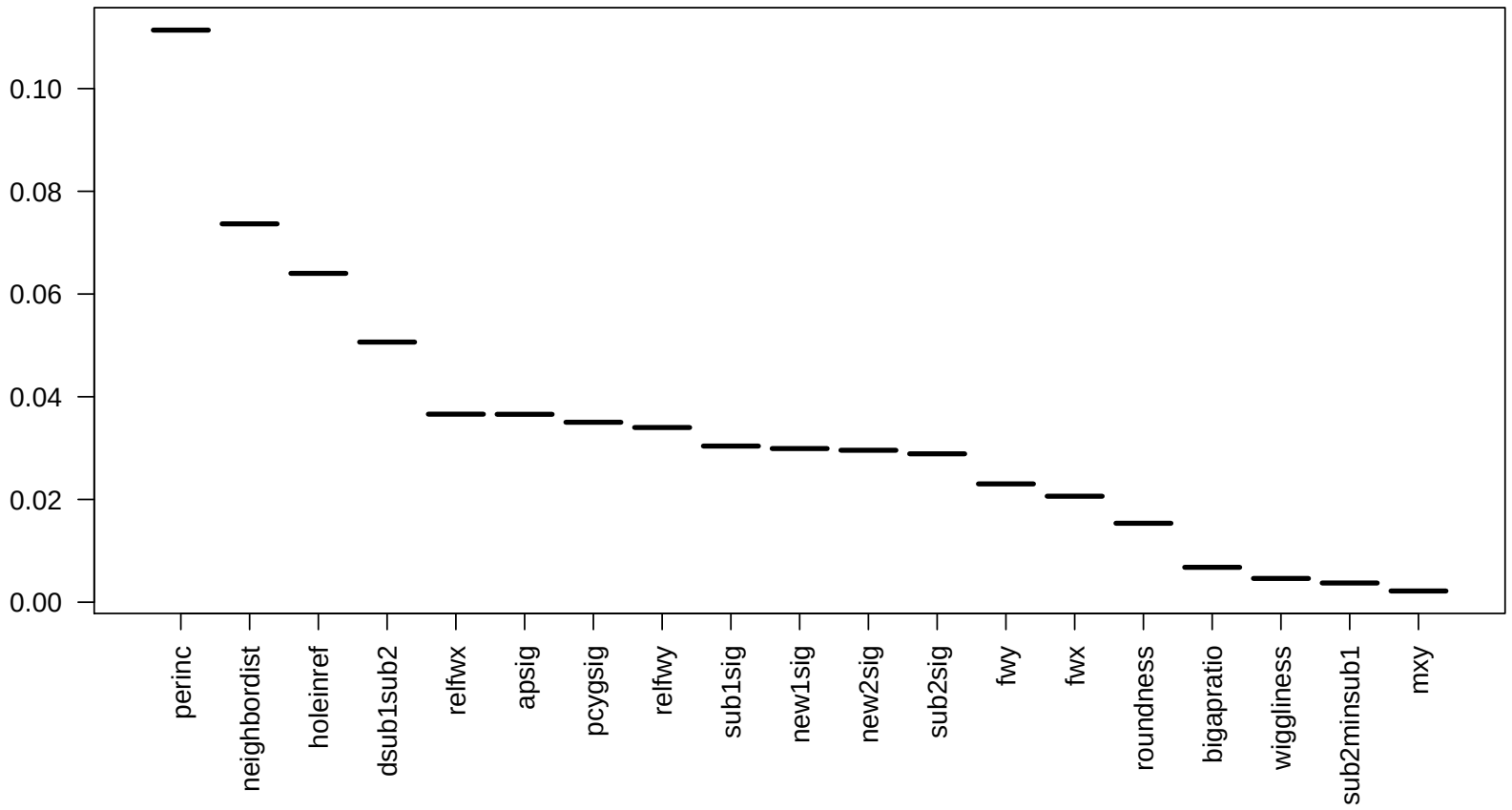
Preformance on test set

		Prediction	
		Other	Supernova
Actual	Other	484	16
	Supernova	39	461

Error: 5.5%

(were getting about 8% from CART and 6% from bagging.)

Supernova - Variable Importance



Supernova Data - Try different K

- K = 2

		Prediction	
		Other	Supernova
Actual	Other	484	16
	Supernova	38	462

Error: 5.4%

- K = 8

		Prediction	
		Other	Supernova
Actual	Other	480	20
	Supernova	37	463

Error: 5.7%