Random Forests

Phil Cutler

Statistical Data Mining I

Fall 2015

Rachael Hageman Blair

Machine Learning, 45, 5–32, 2001

Random Forests

LEO BREIMAN
Statistics Department, University of California, Berkeley, CA 94720

Editor: Robert E. Schapire

Random Forests
Leo Breiman and Adele Cutler

Statistical Methods for Prediction and Understanding.

Phil Cutler

Random Forests
Leo Breiman and Adele Cutler

Random Forests(tm) is a trademark of Leo Breiman and Adele Cutler and is licensed exclusively to Salford Systems for the commercial release of the software.
Our trademarks also include RF(tm), RandomForests(tm), RandomForest(tm) and Random Forest(tm).

Contents

Outline

- Recap Trees & AdaBoost
- RF Motivation, relation to bagging.
- Terminology & Definitions
- Properties of RF: convergence, error bounds.
- Out of bag estimates
- RF via random input selection
- RF via random linear combinations
- Examples
- Conclusions

Properties of Trees

- ✓ Can handle huge datasets
- ✓ Can handle mixed predictors – quantitative and qualitative
- ✓ Easily ignore redundant variables
- ✓ Handle missing data elegantly
- ✓ Small trees are easy to interpret
- ✓ If grown sufficiently deep, have low bias.

- ⊘ large trees are hard to interpret
- ⊘ often prediction performance is poor

Model Stabilization

- **Bagging and Boosting** – exploit the presence of instability in order to create a more accurate learning method (e.g., predictor or classifier).
- Operate by perturbing the learning set, and generating an ***ensemble*** of different ***base predictors*** or ***base classifiers.***
- The ensemble is combined appropriately to yield a final summary or single classifier that is more stable.
- Success of these methods depends on the degree of instability of the base predictors or classifiers.
- Approach: referred to as ***ensemble learning*** or ***committee machines***.

Model Stabilization

Classification trees can be simple, but often produce noisy (bushy) or weak (stunted) classifiers.

Committee Methods:

- Bagging – Fit many large trees to bootstrap-resampled versions of the training data, and classify by majority vote.
- Boosting – Fit many large or small trees to reweighted versions of the training data. Classify by majority vote.
- Random Forests – fancier version of bagging. – tree growing through random selection of input variables.

In general …

Boosting ****, Rforests***, Bagging**, Single Tree*

AdaBoost.M1

- At each boosting step, weights are applied $w_1, w_2, \ldots, w_N$, to each of the training observations (x_i, y_i), $i = 1, 2, \ldots, N$.

- At the initial step, $w_i = 1/N$, $\forall i$.

- At each successive iteration, $m = 2, 3, \ldots, M$, the observation weights are individually modified and the classification algorithm is reapplied to the weighted observations.

- At step m, those observations that were misclassified by the classifier $G_{m-1}(x)$ induced at the previous step have their weights increased, whereas the weights are decreased for those that were classified correctly.

AdaBoost.M1

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Random Forests: Motivation

Committee Methods give improvements to weak classifiers:

- With respect to trees, significant improvements have been found by growing ensembles of trees, and letting them vote on the "most popular class".
- There often exists a random component in tree stabilizing methods e.g. bagging.

Common Element:

- for the *k*th tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1, \Theta_2, \ldots, \Theta_{k-1}$ but with the same distribution.

- a tree is grown using the training set Θ_k resulting in a classifier $h(x, \Theta_k)$.

Random Forests: Motivation

Random Forest - A combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

Several Asymptotic Properties:

- Test error converges as the number of tree becomes large.
- The test error of a forest of tree classifiers depends on the **strength of the individual trees in the forest AND the correlation between them**.

Random Forests: Motivation

Does well compared to other methods:

- Less required in terms of complexity parameters.
- More robust than AdaBoost when dealing with noisy data.

Internal estimates provide good insight:

- Estimates of error, strength and correlation are used to show the response of increasing the number of features used in the splitting.

Random Forests: Definition

A **random forest** is a classifier consisting of a collection of tree-structured classifiers $\{h(x,\Theta_k), k=1,\ldots,M\}$ 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.

Terminology

- The margin ~ measures the extent to which the average number of votes at X, Y, for the right class exceeds the average vote for any other class:

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j).$$

**The larger the margin, the more confidence in the classification.

Random Forests: Terminology

- Generalization error:

$$PE^{*} = P_{X,Y}\left(mg\left(X,Y\right) < 0\right)$$

  where the subscripts indicate that the probability is over the X,Y, space.

- In a random forest, $h_k(X) = h(X,\Theta_k).$

- Random forests converge: as the number of trees increases, for almost all sequences $\Theta_1,\ldots,\Theta_\infty$ the test error converges to a minimum. Therefore, we are not in a situation of overfitting by adding more and more trees (Theorem 1.2).

Random Forests: Terminology

- The margin for a random forest is:

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

and the strength of the set of classifiers $\{h(X,\Theta)\}$ is defined as

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

Assuming $s \geq 0,$ it can be shown that:

$$PE^{*} \leq \mathrm{var}(mr) / s^{2}.$$

*In other words, performance is bounded by (1) error in terms of margin, and the (2) strength of the classifiers.

Random Forests: Properties

- Another way to write the upper bound, is in terms of correlation and strength of the classifier:

$$PE^* \le \bar{\rho}\left(1 - s^2\right) / s^2,$$

where $\bar{\rho}$ is the mean value of the correlation between trees.

Two ingredients control test error –
- Strength of individual trees
- Correlation between trees.

- The c/s2 ratio for a random forest is defined as:

$$c / s2 = \bar{\rho} / s^2.$$

*The smaller the better

Random Features

The use of Random Features

- **Different flavors of random:** random split selection, addition of random noise to the output. None come close to AdaBoost.
- Random vs. AdaBoost
- To improve accuracy, the randomness has to minimize the correlation $\bar{\rho}$ while maintaining strength s.
- Randomness in RF – randomly selected inputs or combinations of inputs to grow the tree at each node. Performance comparable to AdaBoost.

Random Features

RF Characteristics:

1. Accuracy often as good as AdaBoost sometimes better.
2. Relatively more robust to outliers.
3. Faster than bagging or boosting.
4. Useful internal estimates of error, strength, correlation and variable importance.
5. Simple and easily parallelized.

Out of Bag Estimates

- Bagging is used in tandem with random feature selection.
- Each new training set is drawn, with replacement, from the original training set.
- The tree is grown on the new training set using random feature selection, and the tree is left unpruned.

Bagging Advantages:

1. Enhances accuracy when random features are used.
2. Can be used to provide ongoing updates regarding the estimated generalization error of the ensemble of trees, and estimates for the strength and correlation.

Out of Bag Estimates

Consider prediction by bagging in the usual sense.

- Given a training set T form bootstrap training sets T_k, construct classifiers $h(x, T_k)$ and let those vote to form a bagged predictor.

Now, the "out of bag" sense –

- For each y, x in the training set, aggregate the votes only over classifiers for which T_k does not contain y, x. This is the out of bag classifier.
- The out of bag generalization error is the error rate of the out of bag classifiers on the training set.

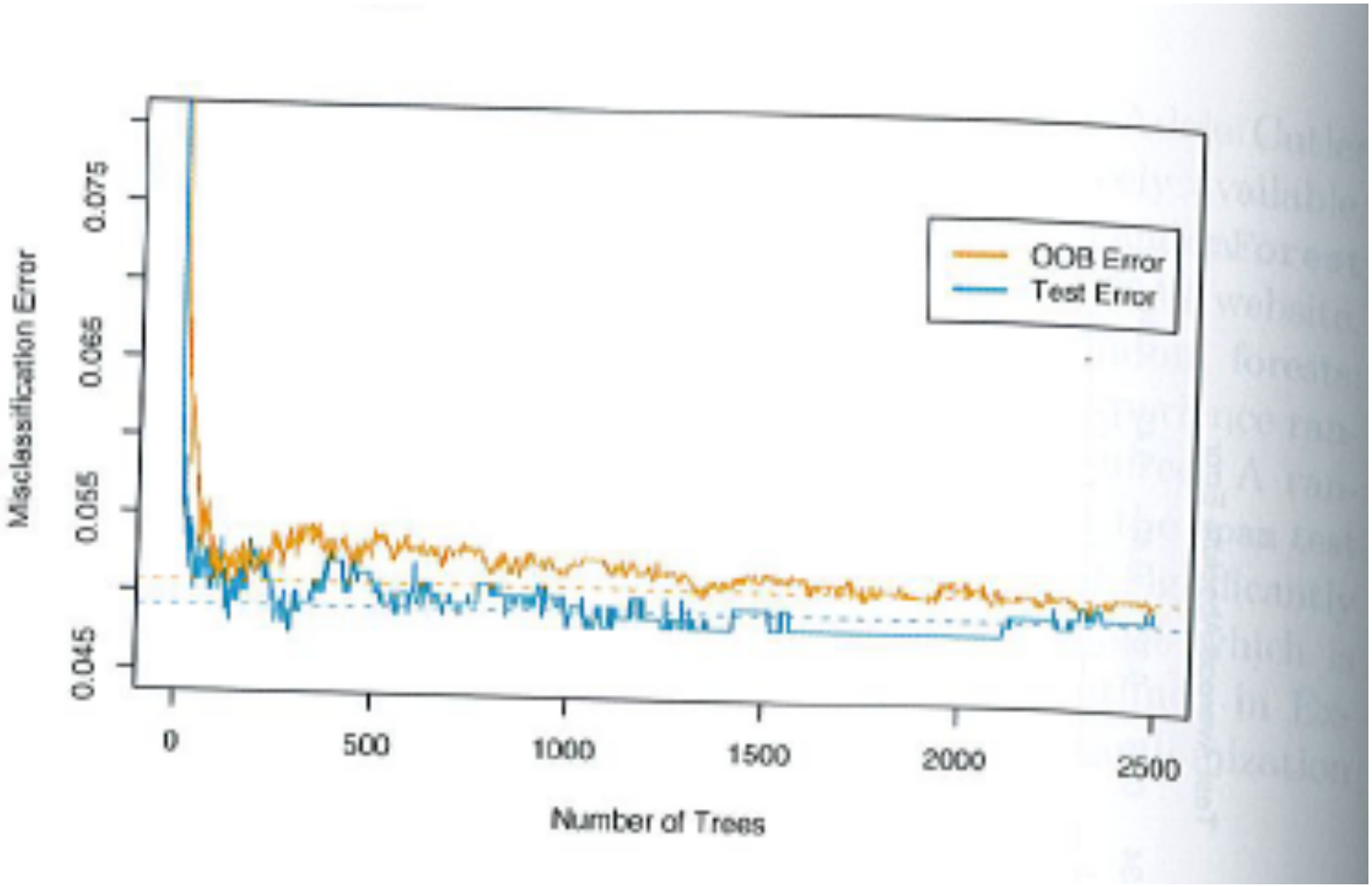
Out of Bag Estimates

Website Explanation:

- **Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree.**
- **Put each case left out in the construction of the kth tree down the kth tree to get a classification.** In this way, a test set classification is obtained for each case in about one-third of the trees. At the end of the run, take j to be the class that got most of the votes every time case n was oob**. The proportion of times that j is not equal to the true class of n averaged over all cases is the oob error estimate. This has proven to be unbiased in many tests.**

Out of Bag Estimates

- Out of bag error estimates have been shown empirically to be as accurate as a test data set of the same size as the training set. Therefore, removes the need to set aside test error.
- Selection properties cause out of bag error estimation to be an overestimate (asymptotics).
- Strength and correlation can also be estimated (monitored) using out of bag methods.


Misclassification Error
0.045
0.055
0.065
0.075
0
500
1000
1500
2000
2500
Number of Trees
OOB Error
Test Error

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

RF via Random Input Selection

Simplest RF (Forest-RI)

- At each internal node select at random input variables to split on.
- Grow the tree using CART method to grow tree, do not prune.
- The size F of the group is fixed: Two possibilities F=1, and log(M+1), where M is the number of inputs.

Data set	Train size	Test size	Inputs	Classes
Glass	214	–	9	6
Breast cancer	699	–	9	2
Diabetes	768	–	8	2
Sonar	208	–	60	2
Vowel	990	–	10	11
Ionosphere	351	–	34	2
Vehicle	846	–	18	4
Soybean	685	–	35	19
German credit	1000	–	24	2
Image	2310	–	19	7
Ecoli	336	–	7	8
Votes	435	–	16	2
Liver	345	–	6	2
Letters	15000	5000	16	26
Sat-images	4435	2000	36	6
Zip-code	7291	2007	256	10
Waveform	300	3000	21	3
Twonorm	300	3000	20	2
Threenorm	300	3000	20	2
Ringnorm	300	3000	20	2

Data from UCI repository

RF 100 trees
AdaBoost 50 iterations

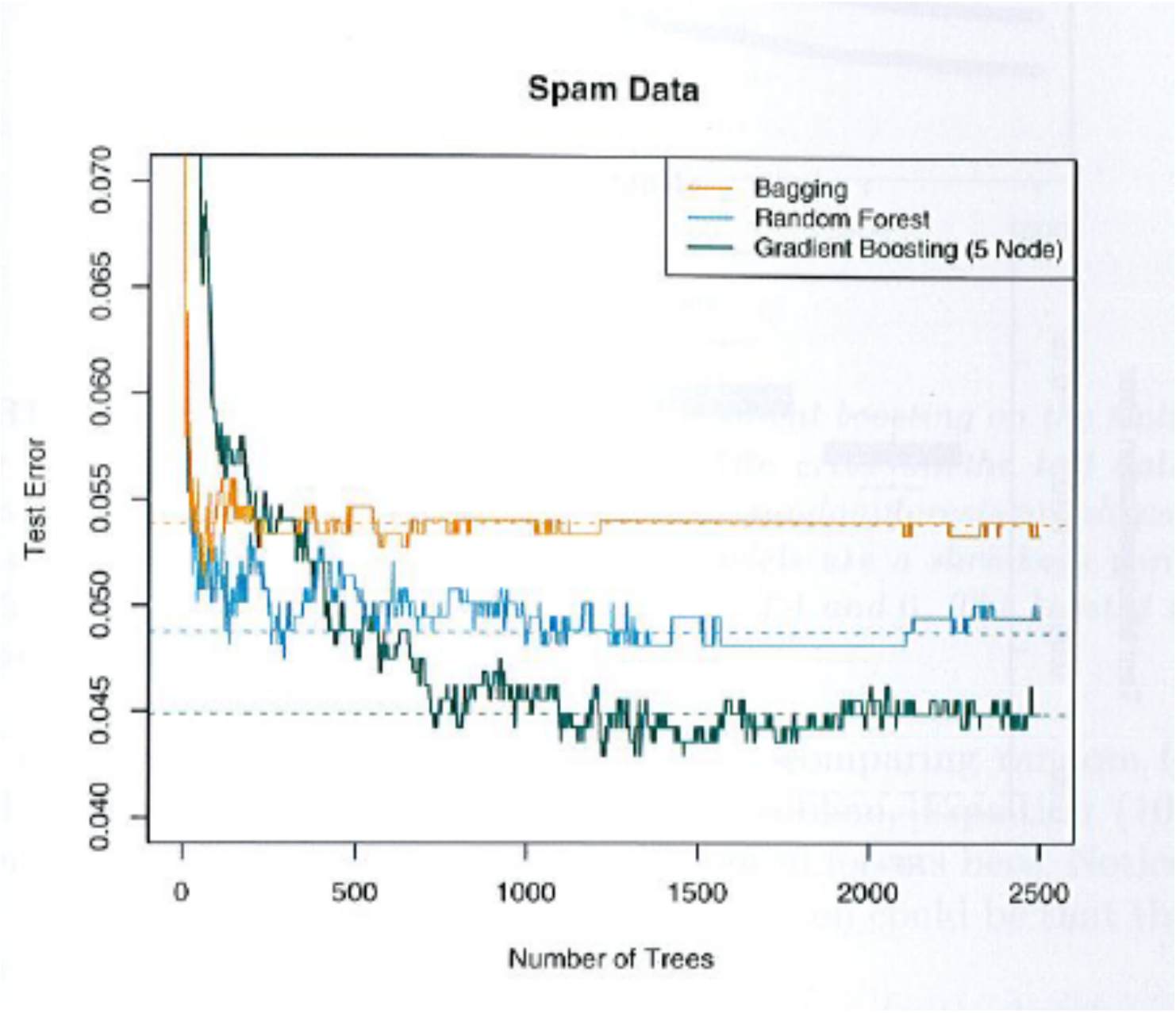
Try:
AdaBoost
Selection
Forest-RI single input
One tree

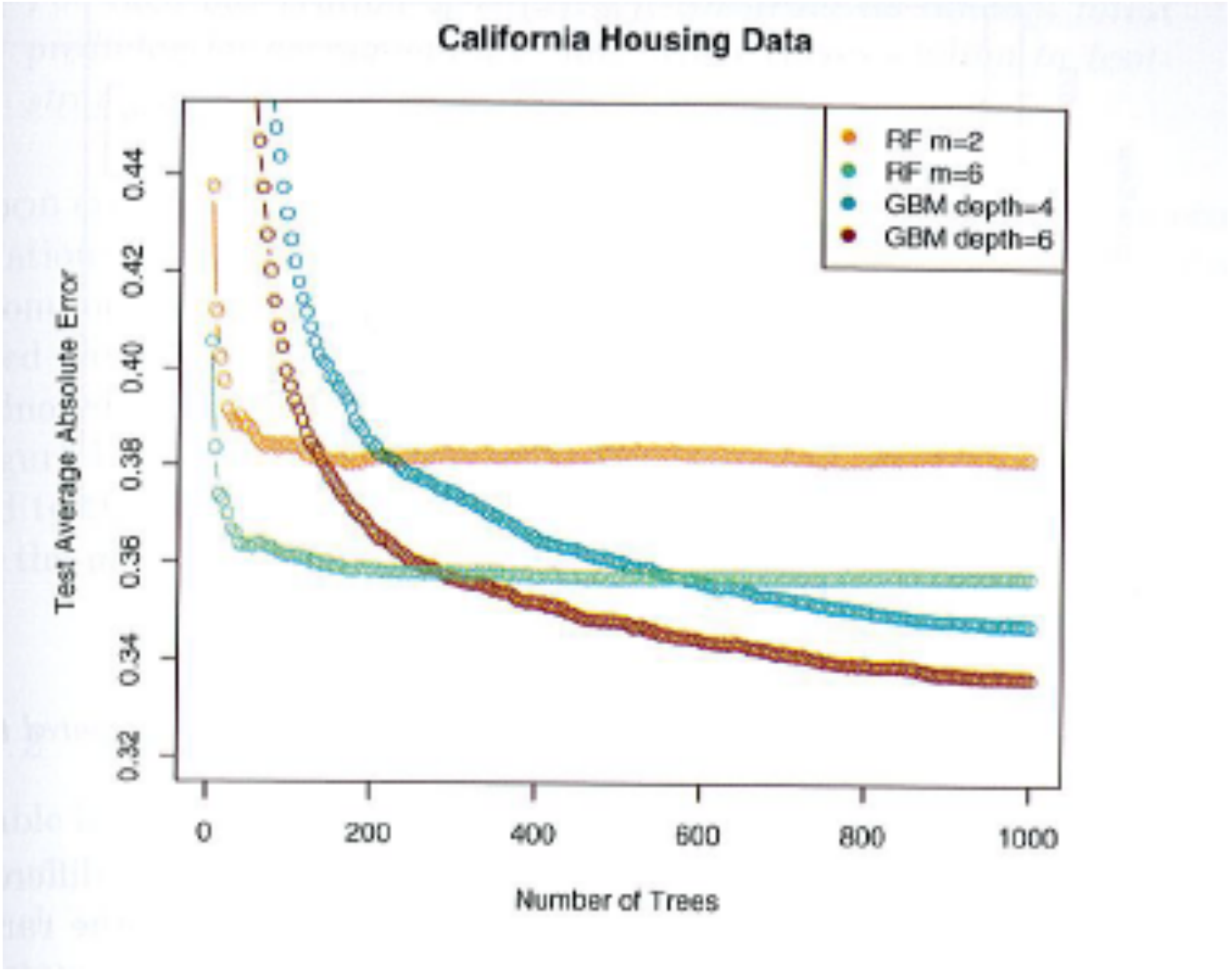
Compare favorably with Adaboost

Table 2. Test set errors (%).

Data set	Adaboost	Selection	Forest-RI single input	One tree
Glass	22.0	20.6	21.2	36.9
Breast cancer	3.2	2.9	2.7	6.3
Diabetes	26.6	24.2	24.3	33.1
Sonar	15.6	15.9	18.0	31.7
Vowel	4.1	3.4	3.3	30.4
Ionosphere	6.4	7.1	7.5	12.7
Vehicle	23.2	25.8	26.4	33.1
German credit	23.5	24.4	26.2	33.3
Image	1.6	2.1	2.7	6.4
Ecoli	14.8	12.8	13.0	24.5
Votes	4.8	4.1	4.6	7.4
Liver	30.7	25.1	24.7	40.6
Letters	3.4	3.5	4.7	19.8
Sat-images	8.8	8.6	10.5	17.2
Zip-code	6.2	6.3	7.8	20.6
Waveform	17.8	17.2	17.3	34.0
Twonorm	4.9	3.9	3.9	24.7
Threenorm	18.8	17.5	17.5	38.4
Ringnorm	6.9	4.9	4.9	25.7

AdaBoost 3 hrs.
RF 4 min


Spam Data
Bagging
Random Forest
Gradient Boosting (5 Node)
Test Error
0.040
0.045
0.050
0.055
0.060
0.065
0.070
0
500
1000
1500
2000
2500
Number of Trees


California Housing Data
Test Average Absolute Error
0.32
0.34
0.36
0.38
0.40
0.42
0.44
0
200
400
600
800
1000
Number of Trees
RF m=2
RF m=6
GBM depth=4
GBM depth=6

RF via Linear Combination

More Complex RF (Forest –RC)

- Take random linear combinations of the input variables.
- A feature is generated by specifying L, the number of variables to be combined. At each given node, the L variables are randomly selected and added together with coefficients that are uniform random numbers in [-1,1].
- F linear combinations are generated, and then searched over for the best split.

- On synthetic data, Forest-RC does very well.
- Performance closer to Adaboost than Forest-RI.

Table 3. Test set errors (%).

		Forest-RC		
Data set	Adaboost	Selection	Two features	One tree
Glass	22.0	24.4	23.5	42.4
Breast cancer	3.2	3.1	2.9	5.8
Diabetes	26.6	23.0	23.1	32.1
Sonar	15.6	13.6	13.8	31.7
Vowel	4.1	3.3	3.3	30.4
Ionosphere	6.4	5.5	5.7	14.2
Vehicle	23.2	23.1	22.8	39.1
German credit	23.5	22.8	23.8	32.6
Image	1.6	1.6	1.8	6.0
Ecoli	14.8	12.9	12.4	25.3
Votes	4.8	4.1	4.0	8.6
Liver	30.7	27.3	27.2	40.3
Letters	3.4	3.4	4.1	23.8
Sat-images	8.8	9.1	10.2	17.3
Zip-code	6.2	6.2	7.2	22.7
Waveform	17.8	16.0	16.1	33.2
Twonorm	4.9	3.8	3.9	20.9
Threenorm	18.8	16.8	16.9	34.8
Ringnorm	6.9	4.8	4.6	24.6

Categorical Variables

- Categorical values need to be dealt with in order to combined for numerical variables.
- Each time a categorical variable is selected to split on at a node, select at random a subset of categories of the variable, and define a substitute variable that is one when the subset is true and zero otherwise.
- Use an indicator matrix (N x K-1)
- F must be increased.
- Computationally efficient.

Example: Soybean data N=685, 35 variables, 19 classes, and 15 categorical variables, for F=12, 5.3% error rate (5.8% for Adaboost). F = 8 gives error rate of 5.5%.

Strength and Correlation

What is the effect of strength and correlation on the generalization error.

Example:

Forest-RI run on the sonar data

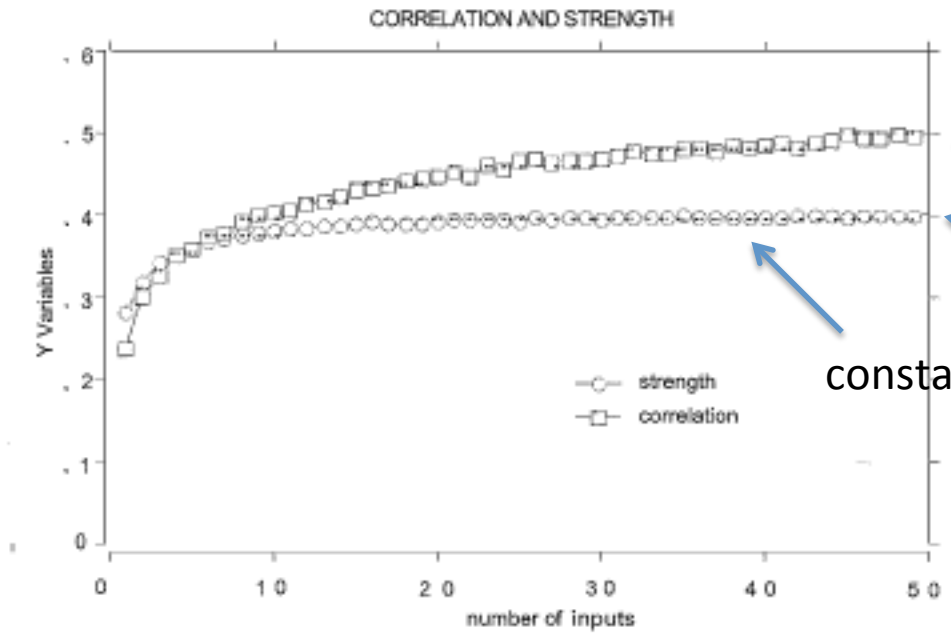
60 inputs, 208 observations

Using inputs ranging from 1 to 50.

For each F, 100 trees are grown to form a RF and the terminal values of test error, strength, correlation are recorded.

This is done 80 times.

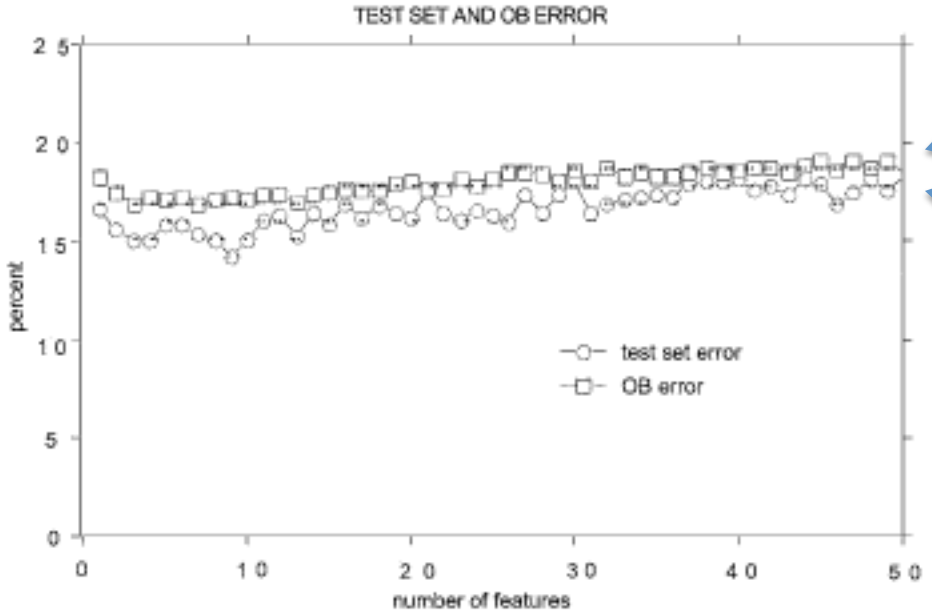
Effect of Inputs
On Sonar Data



correlation

strength

constant

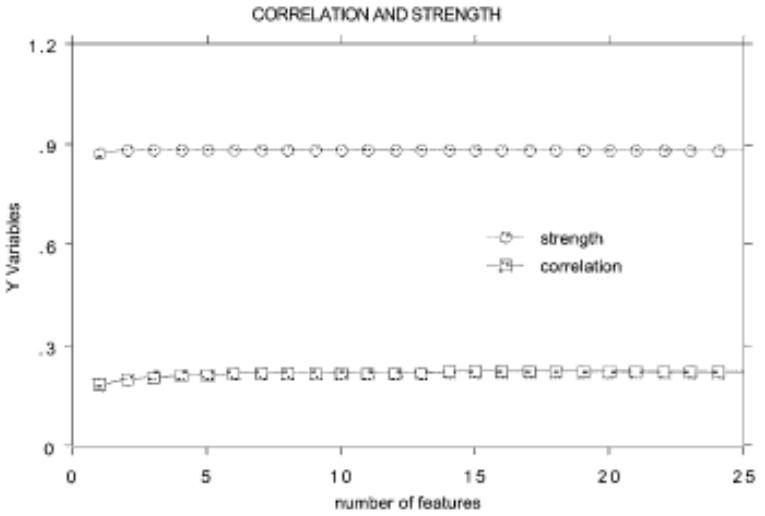


OOB error

Test set error

Figure 1. Effect of number of inputs on sonar data.

Breast Cancer Data



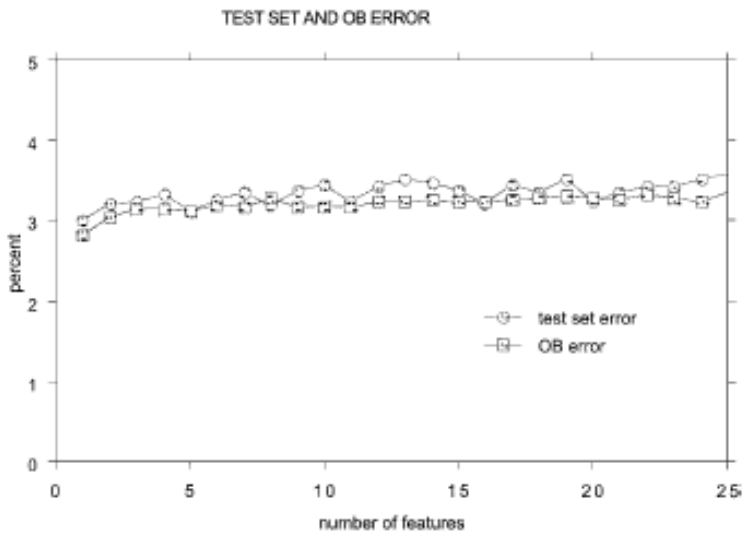
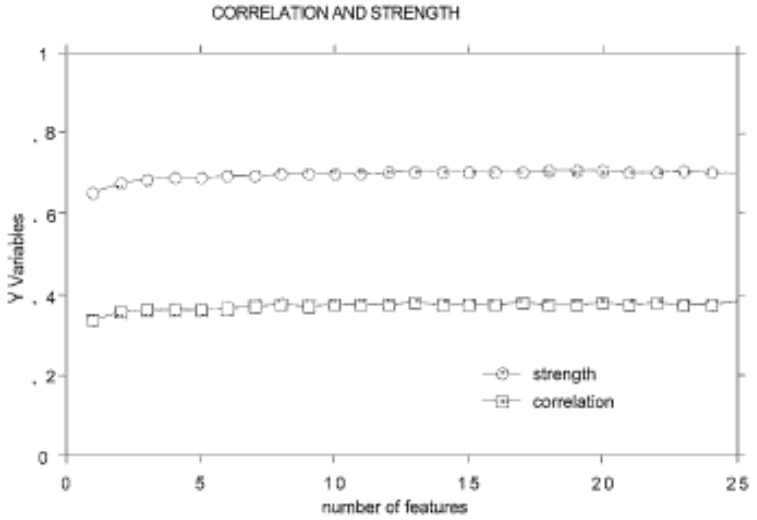


Figure 2. Effect of the number of features on the breast data set.

Satellite Data (smaller)



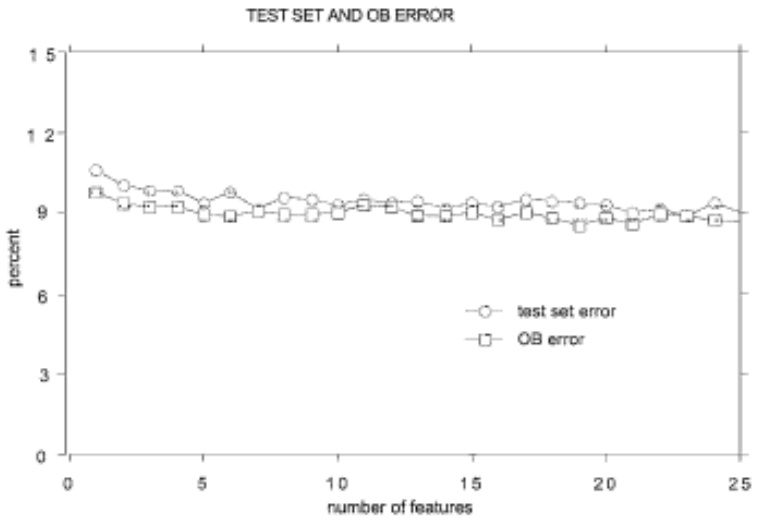


Figure 3. Effect of number of features on satellite data.

Is Adaboost a RF?

- Author shows Adaboost is similar in spirit to RF.
- The distribution of weights depends on the training set in Adaboost. Whereas, the distribution of random vectors for RF does not depend on the training set.
- Breiman proposes a conjecture, not a proof.
- Adaboost is sensitive to noise and outliers.
- Has been shown to overfit in only a few, potentially over simplified examples.

The effects of output noise

- Dieterich showed that when a fraction of output labels in the data are altered Adaboost performance degrades.
- RF, random selection, bagging, less sensitive to noise.
- Noise is prevalent in observational data.

Example:

For each data set, the 10% at random is split off as a test set. Two runs are made on the training, (1) as is, and (2) with noise in the form of changing 5% of the class labels randomly.

This is done 50 times.

Table 4. Increases in error rates due to noise (%).

Data set	Adaboost	Forest-RI	Forest-RC
Glass	1.6	.4	−.4
Breast cancer	43.2	1.8	11.1
Diabetes	6.8	1.7	2.8
Sonar	15.1	−6.6	4.2
Ionosphere	27.7	3.8	5.7
Soybean	26.9	3.2	8.5
Ecoli	7.5	7.9	7.8
Votes	48.9	6.3	4.6
Liver	10.3	−.2	4.8

2 class response
Least sensitive to
Noise.

* ”…Curiously Data Dependent”

Weak Inputs

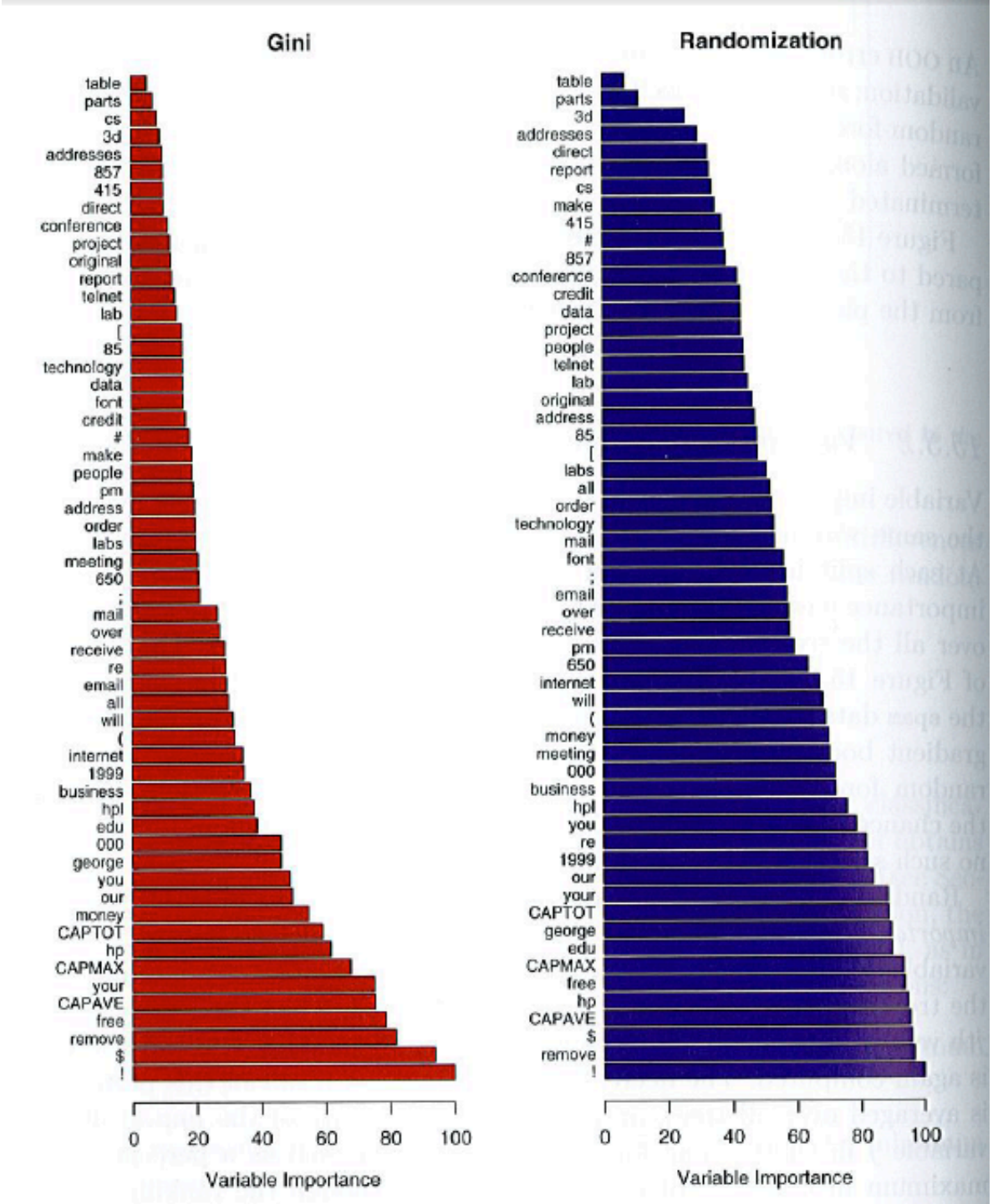
- Weak inputs are common in real world problems, e.g. medical diagnosis.
- No single input or small group of inputs can distinguish between the classes. Notably difficult for classifiers.
- RF do okay with this type of data, slow to converge.
- Adaboost may not work because the classifiers are too weak.

Variable Importance

- Often it is critical to understand the variables themselves and the interaction between variables in addition to the error and prediction aspects.

- **Classic/Ensemble Style**: At each split in the each tree, the improvement in the split-criteria is the importance measure attributed to the splitting variable, and is accumulated over all of the trees in the forest separately for each tree.

- **OOB style**: When the bth tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded. When the splits concerning the jth variable are removed (or permuted randomly), a drop in accuracy is observed.


Gini
table
parts
cs
3d
addresses
857
415
direct
conference
project
original
report
telnet
lab
[
85
technology
data
font
credit
#
make
people
pm
address
order
labs
meeting
650
;
mail
over
receive
re
email
all
will
(
internet
1999
business
hpl
edu
000
george
you
our
money
CAPTOT
hp
CAPMAX
your
CAPAVE
free
remove
$
!
0
20
40
60
80
100
Variable Importance
Randomization
table
parts
3d
addresses
direct
report
cs
make
415
#
857
conference
credit
data
project
people
telnet
lab
original
address
85
[
labs
all
order
technology
mail
font
;
email
over
receive
pm
650
internet
will
(
money
meeting
000
business
hpl
you
re
1999
our
your
CAPTOT
george
edu
CAPMAX
free
hp
CAPAVE
$
remove
!
0
20
40
60
80
100
Variable Importance

Regression with RF

- Jury is out…
- Same type of bounds and dependencies for the generalization error. However, in practice, far more features are needed than with classification.
- Seems to do well on the data sets considered.
- Improvements have been made (website).

Conclusions

- RF are effective tools for prediction
- Law of large numbers, the test error converges, and they do not overfit.
- The framework is intuitive in terms of correlation and strength.
- Few parameters to set… easy to use!

Three Questions

- Compare and contrast Random Forests to a couple of the other methods for classification discussed in class. Describe some situations or datasets, in which it may be more desirable to use RF?
- OOB error was demonstrated to be smoother, and, in most examples, an upper bound to test error. This would imply that we may not need to dedicate observations to a test set. How do you feel about this?
- In RF classification, could you use "importance" measure in connection with variable selection? For example, could you use the top 10, 11, or 12 predictors, and fit a RF with almost the same accuracy? How might you go about doing this?