Random Forests

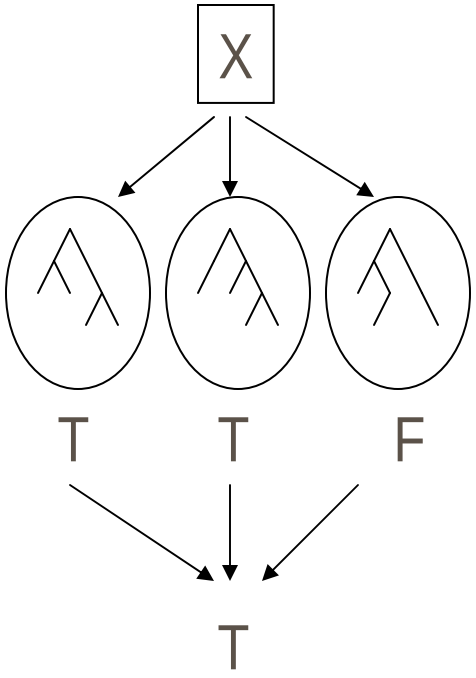
Adrian Barbu

Beyond Decision Trees

- Decision trees:
  - Can overfit
  - Need validation set
- Improvement: Random Forests (Breiman 2001)
  - Better generalization power
  - More robustness to noise
  - No need for validation dataset
  - Know what to expect on unseen data
- We don’t put all our money in one tree
- Might be slower

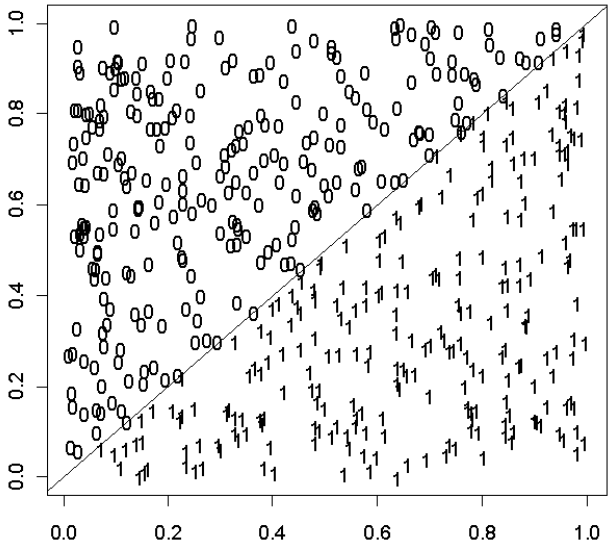
Overview

- A random forest is a collection of decision trees
- Given an observation:
  - Each decision tree votes for a class
  - The class with most votes is the final result

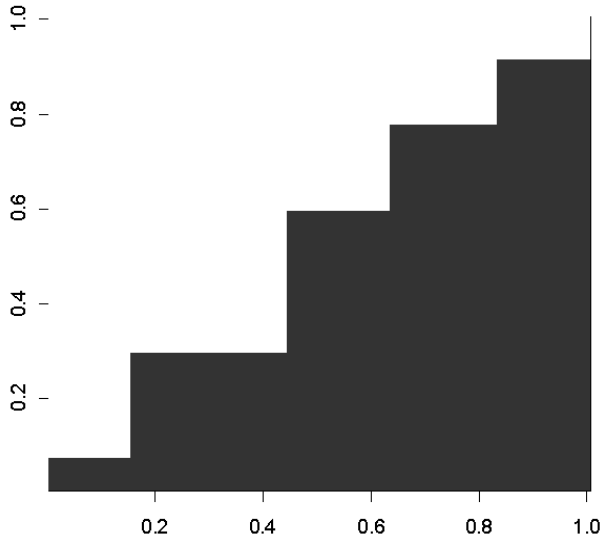


- Growing the trees:
  - All trees are binary trees
  - Each tree is fully grown
  - Each tree has a degree of randomness
  - There should be little correlation between the trees

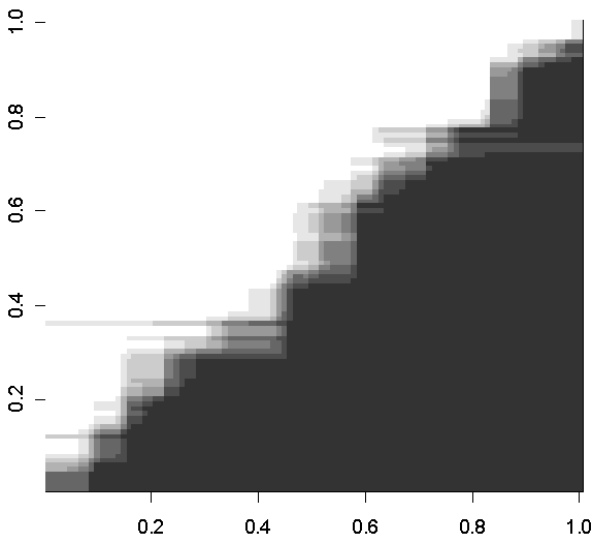
Example



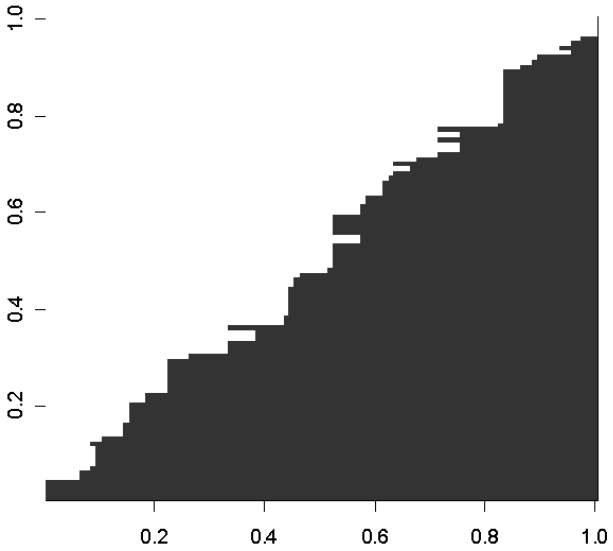
Original data



One decision tree



Avg of 25 decision trees



Voted result of 25 decision trees

Growing the trees

Dataset: N samples, each having M attributes (features)

A value m<M is chosen, $m \sim \sqrt{M}$ or $m \sim \log M$

Growing one tree:

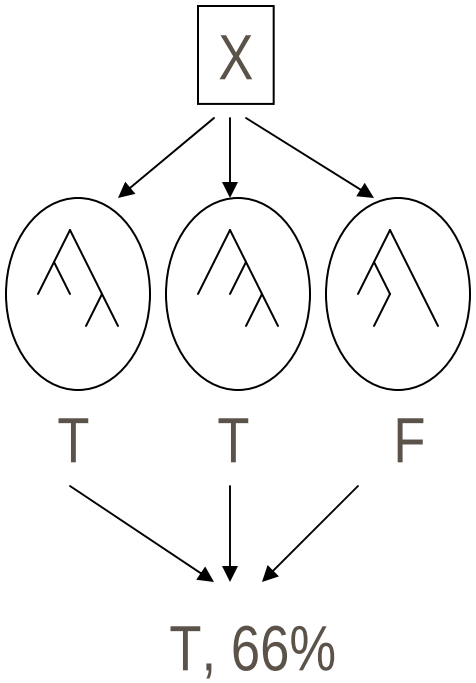
- Select N samples randomly with replacement (bootstrap)
- At each node, m attributes are selected randomly from the M
- The best binary split from the m attributes (based on information gain) is chosen
- The tree is fully grown, no pruning

Growing One Tree

- Best binary split
  - Best threshold on a continuous or discrete attribute
  - Best subset if the number of values is small
- After the best split
  - For each child, the m attributes are resampled
  - This way, the set of attributes is very different at each node
  - Important variables will eventually be selected
  - Trees will be different
- The tree must be fully grown (one sample per leaf).

Combining the Trees

- Grow many trees.
  - Recommended about 500
  - For large data sets about 150 may be sufficient
- Each tree casts a vote at its terminal nodes. For a binary target the vote will be T or F
- Count up the T votes. This is the RF score and the percent T votes received is the predicted probability
- Very simple mechanism

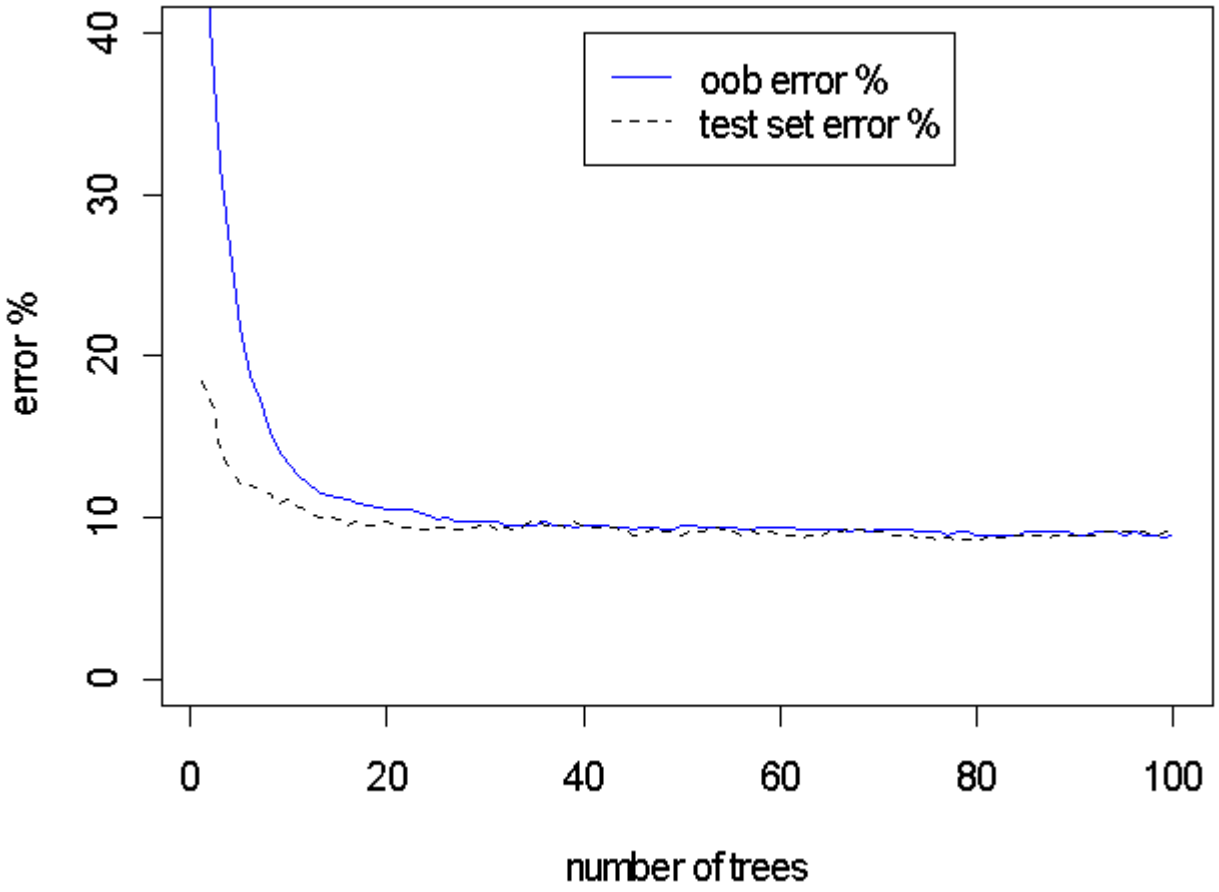


RF Self Testing

- Each tree is grown on about 63% of the original training data (due to the sampling with replacement)
- The remaining 37% of the data is available for testing
- A different 37% for each tree, named “Out of Bag” or OOB
- Use the OOB data to calibrate performance of each tree
- Also keep track how often each sample is classified correctly when it belongs to OOB
- All performance statistics reported by RF are based on OOB calculations

Example – Satellite Dataset

- Training set: 4435 observations.
- Test set: 2000 observations.
- 36 attributes.
- 100 trees.



Comparison with Decision Tree

- Datasets

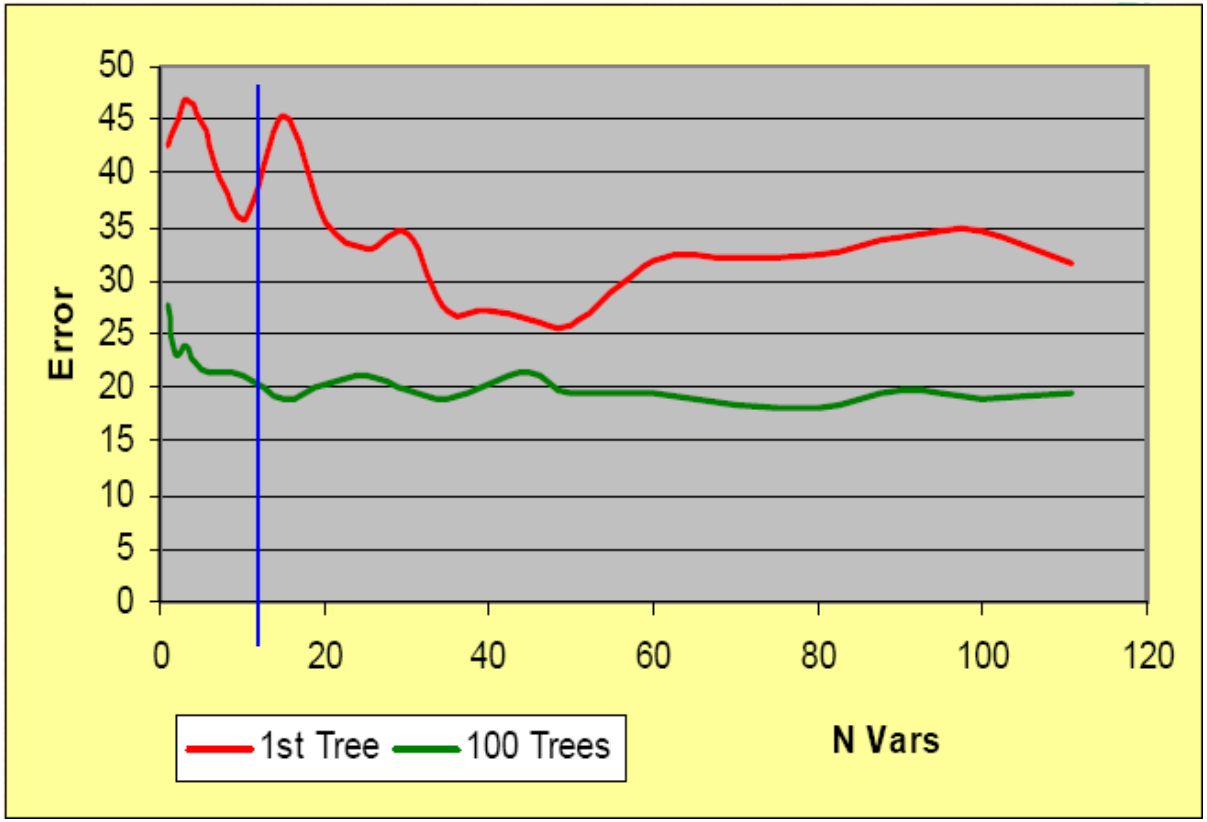
Data Set	Variables	Classes	Training	Test Set
Letters	16	26	15,000	5,000
Satellite	36	6	4,435	2,000
Shuttle	9	7	43,500	14,500
DNA	60	3	2,000	6,186

- Comparison with Decision Tree

Data Set 1	Decision Tree	RF	Decrease
Letters	12.6	6.4	49%
Satellite	14.8	10.3	30%
Shuttle	0.062	0.014	77%
DNA	6.2	5.0	19%

Number of Random Attributes m

N_vars	1st Tree	100 Trees
1	42.7	27.8
2	44.4	23.1
3	47	23.9
5	44.8	21.8
10	35.6	21.1
15	45.2	19
20	35.5	20.2
25	33.1	21.1
30	34.4	19.8
35	27.3	19
40	27.1	20.2
45	26.4	21.4
50	25.9	19.6
60	32	19.6
70	32.1	18.5
80	32.5	18.2
90	34.1	19.7
100	34.5	19
111	31.6	19.6



- Comparison with the first decision tree where m is varied.
- 111 attributes total
- Observe the error is insensitive to m

Comparison With AdaBoost

		RF	m=1	
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

Advantages

- Comparable in accuracy with SVM and Boosting
- Resistant to overfitting
- Can handle large data bases with thousands of attributes
- Can estimate what variables are important
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- Can handle large amounts of missing data.
- Can be extended to unlabeled data → unsupervised clustering.

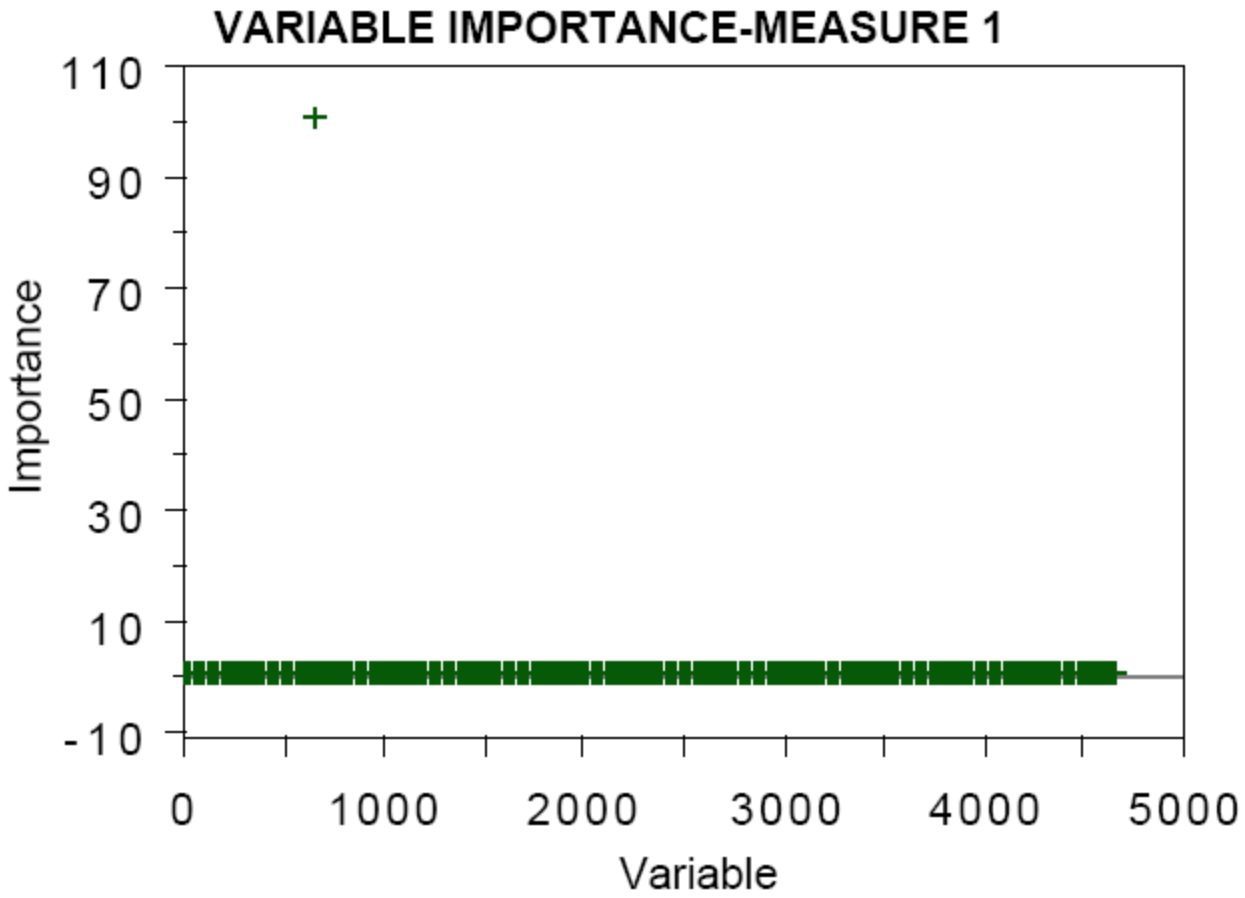
Variable Importance

To estimate the importance of variable m

- Compute test error E_0 on the OOB data
- For each tree in the forest.
  - Randomly permute all values of the m-th variable of the OOB observations
  - Take these altered OOB observations down the tree and get classifications.
- Obtain a new error rate E_1.
- Importance of variable m= E_1-E_0.

Example

- Microarray dataset



Breiman, 2002

Variable Importance II

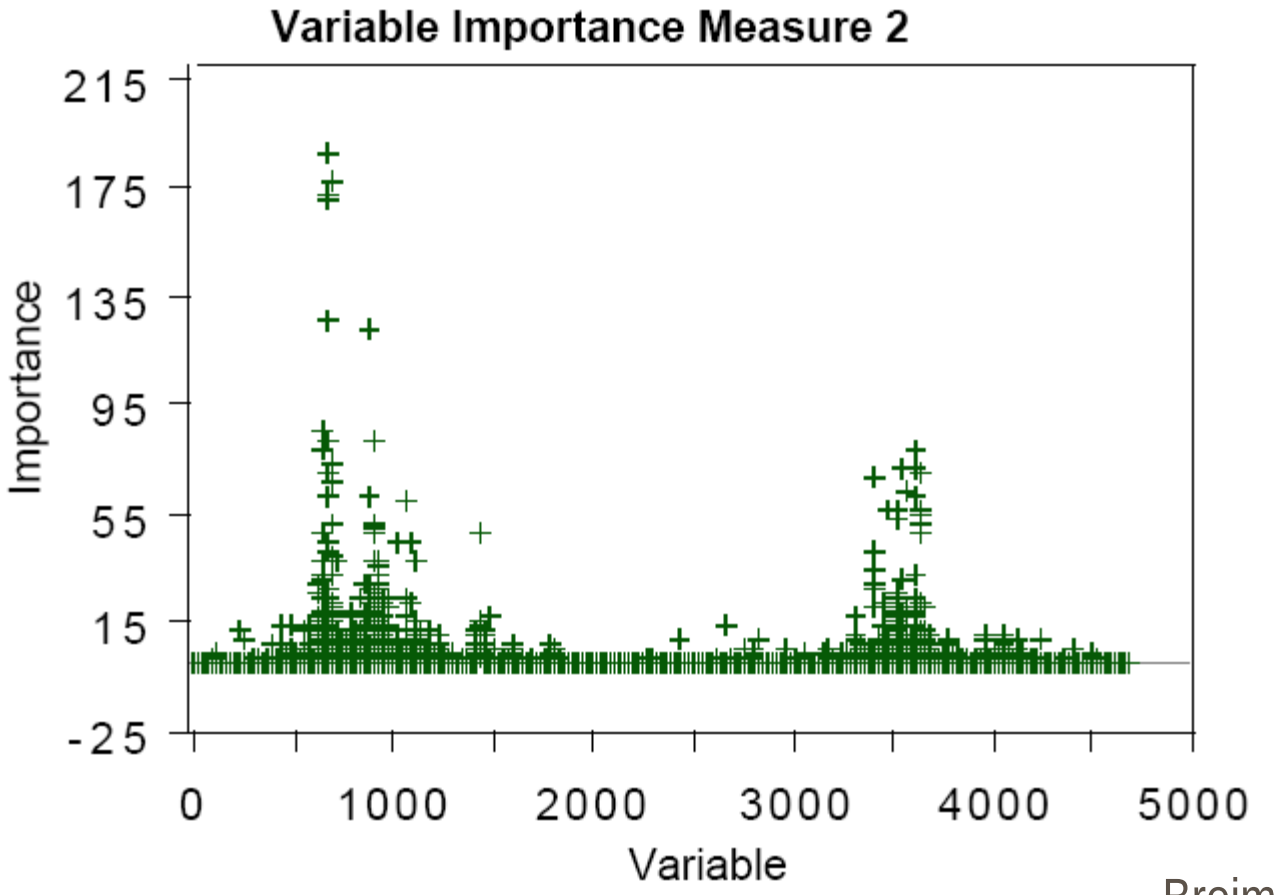
For each observation, **margin**=percent correct votes-max percent incorrect votes.

To estimate the importance of variable m

- Use all trees in the forest.
  - Use the OOB observations, compute the average margin M_0.
  - Randomly permute the values of variable m in the OOB observations
  - Take these new observations down the tree.
  - Compute the new margin M_1.
- Variable m importance = M_0-M_1

Example

- Microarray data



Breiman, 2002

Variable Importance III

To estimate the importance of variable m

- Use all trees in the forest.
  - Use the OOB observations, count the number of votes for the correct class.
  - Randomly permute the values of variable m in the OOB observations
  - Take these new observations down the tree.
  - Subtract the number of votes for the correct class in the variable-m-permuted OOB data from the number of votes for the correct class in the untouched OOB data.
  - The average of this number over all trees in the forest is the raw importance score for variable m.
- z-score= raw score/(standard deviation across all trees)
- Assign a significance level to the z-score assuming normality.

Proximity Measure

- RF can be used to define proximity between any two observations:

1. Initialize a N×N matrix of proximities to zeroes
2. For any tree of the RF, apply it to all observations
3. If obs. i and j end up in the same final tree node, increase proximity prox(i,j) between i and j by one
4. Accumulate over all trees in RF and divide by twice the number of trees in the RF

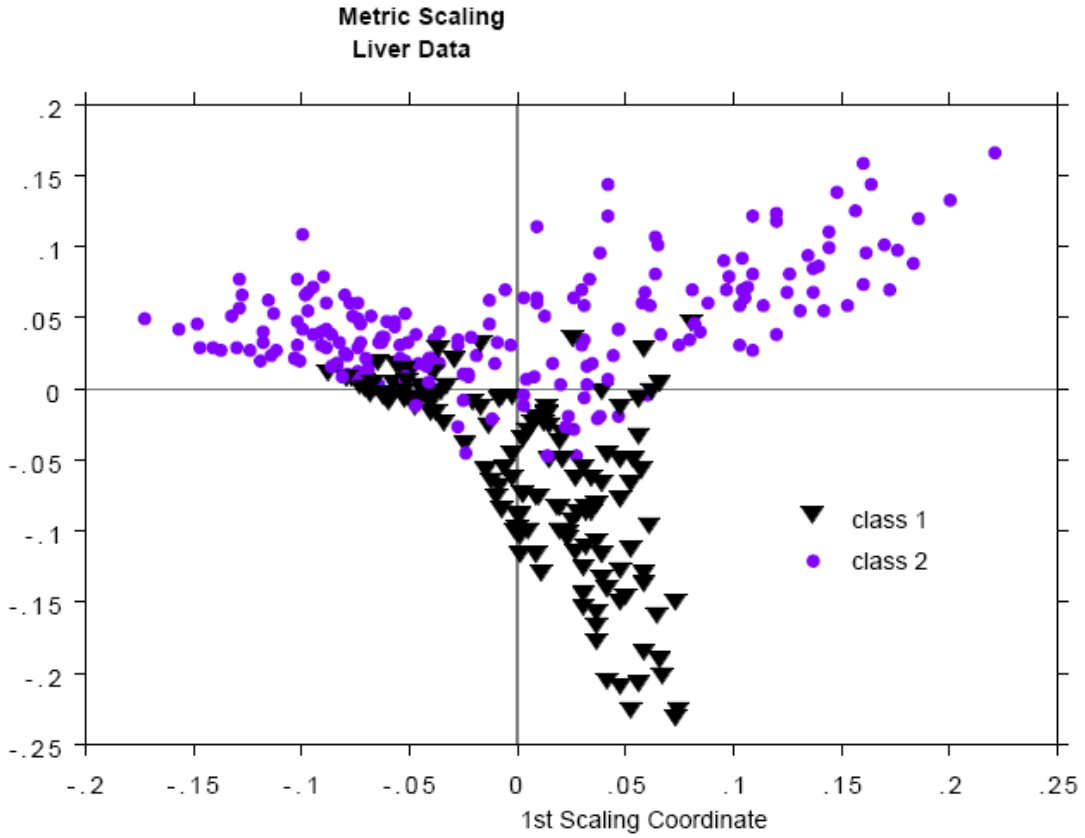
- Obtain an intrinsic measure of proximity
- Invariant to monotone transformations
- The measure is defined for any type of attributes, including categorical

Example

	A	B	C	D	E	F	G	H	I	J
1	RECORD	X0000001	X0000002	X0000003	X0000004	X0000005	X0000006	X0000007	X0000008	X0000009
2	1	1	0.488	0.198	0.13	0.092	0.12	0.076	0.106	0.026
3	2	0.488	1	0.146	0.11	0.086	0.104	0.074	0.086	0.01
4	3	0.198	0.146	1	0.252	0.1	0.208	0.032	0.062	0.032
5	4	0.13	0.11	0.252	1	0.046	0.194	0.028	0.068	0.038
6	5	0.092	0.086	0.1	0.046	1	0.332	0.076	0.094	0.058
7	6	0.12	0.104	0.208	0.194	0.332	1	0.052	0.08	0.064
8	7	0.076	0.074	0.032	0.028	0.076	0.052	1	0.514	0.04
9	8	0.106	0.086	0.062	0.068	0.094	0.08	0.514	1	0.048
10	9	0.026	0.01	0.032	0.038	0.058	0.064	0.04	0.048	1

- The proximity matrix for the first 10 observations of the prostate dataset
- 1 on the main diagonal – “perfect” proximity to itself
- Similar observations have proximities close to one (green background)
- The closer proximity to 0, the more dissimilar observations i and j are (orange background)

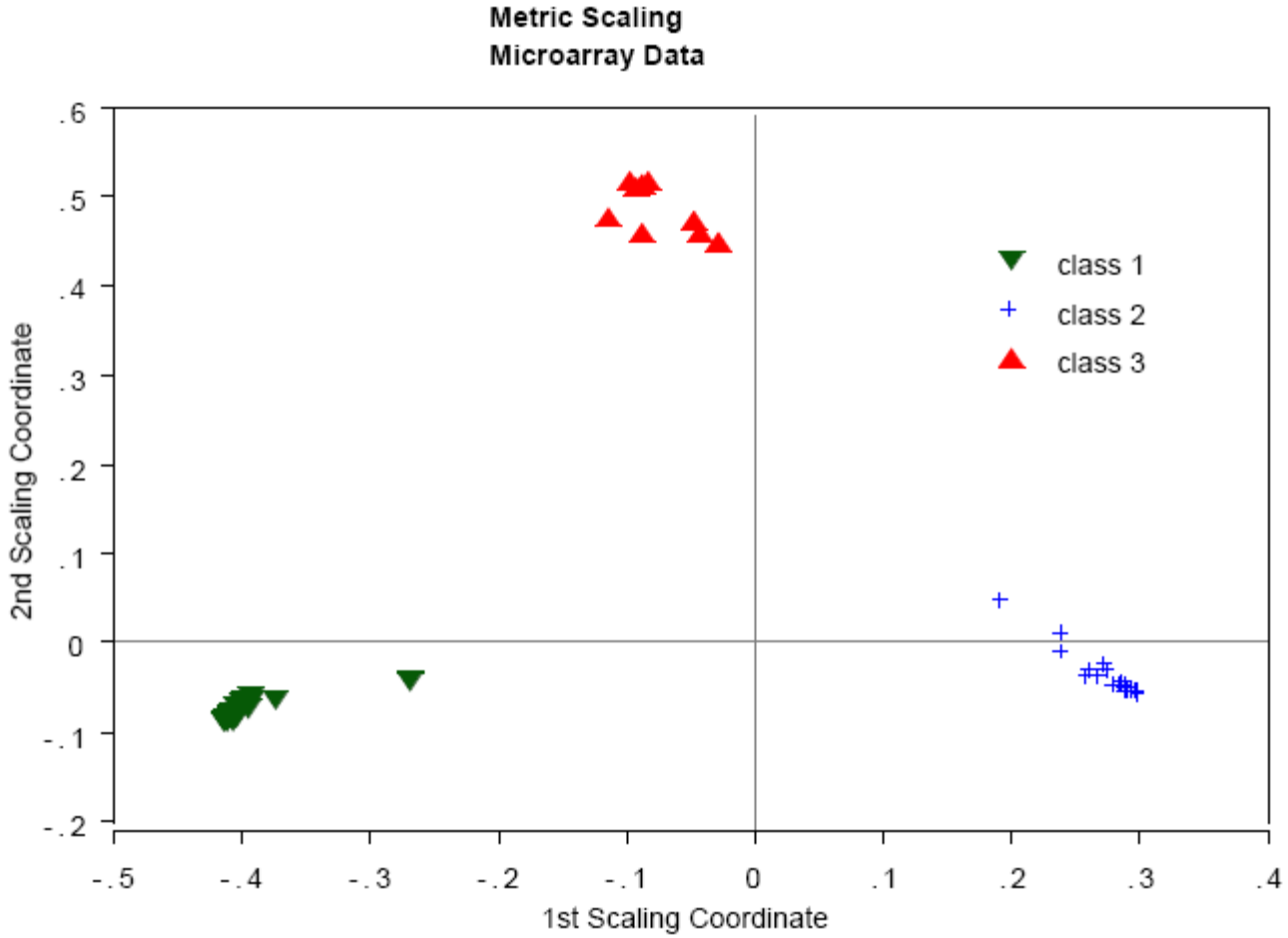
Application: Metric Scaling



Breiman, 2002

- Find a 2D representation of the data that preserves as much of the proximity as possible
- Done by energy minimization

Another example



- Here the classes are well separated
- The class label is not required for obtaining the graph
- In this case, the labels can be assigned automatically (unsupervised learning)

Breiman, 2002

Application: Outlier Detection

- Outliers can be defined as observations with small proximities to all other observations in the same target class
- The following algorithm is used:

1. For an observation i, compute

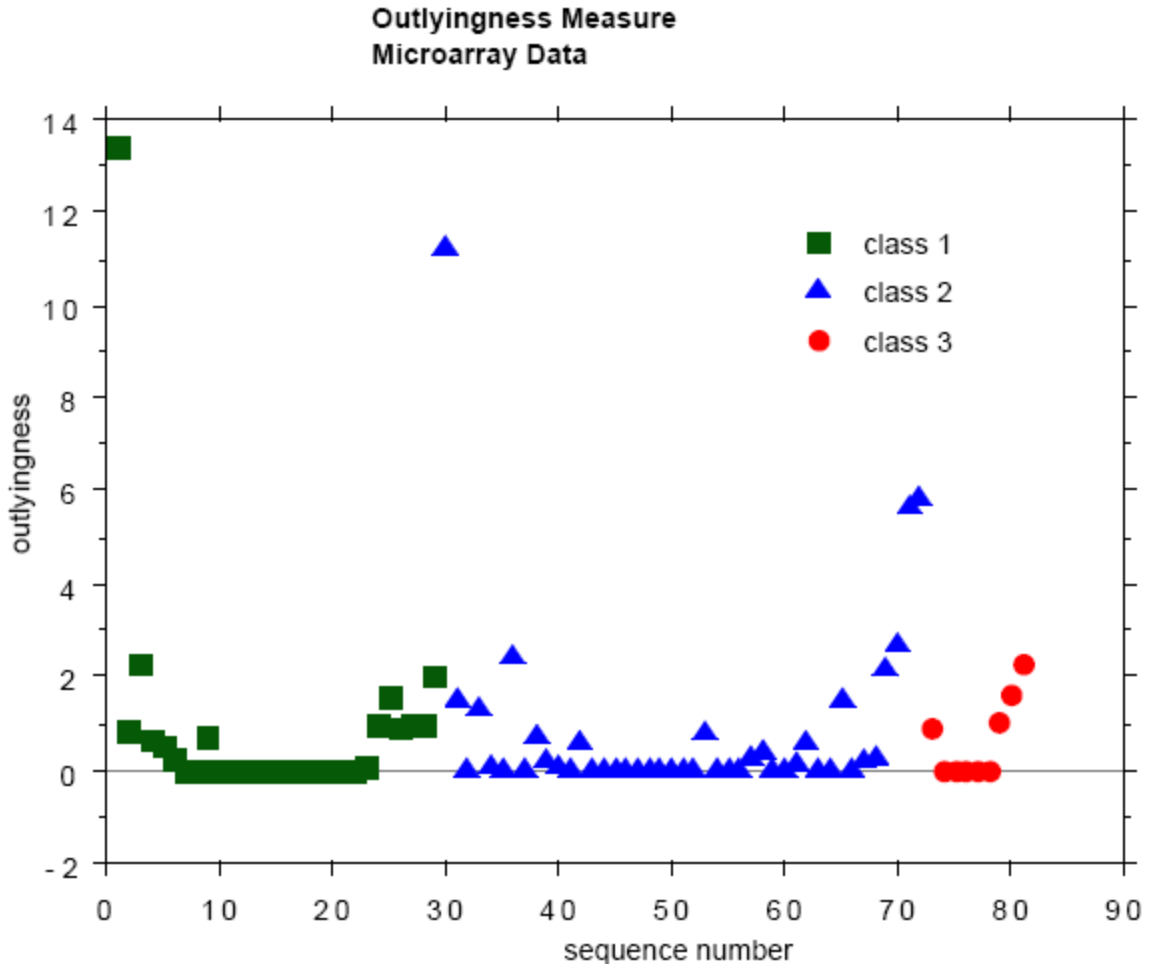
$$o(i) = \frac{1}{\sum_k prox(i,k)^2}$$

where the sum is over all k in the same class as i
o(i) is large if the observation is “far away” from the rest

2. Standardize using the median and standard deviation
3. The observations with the largest values are potential outliers

- Generally, a value above 10 is reason to suspect the observation of being an outlier

Example



- Two possible outliers

Breiman, 2002

Resistance to Label Noise

- 5% of the class labels were randomly altered.
- RF is more resistant to label noise than Adaboost.

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

RF for Regression

- Similar to RF for classification
- Variable selection
  - Instead of Information Gain evaluate change in variance of y
- Leaf nodes
  - Return mean value of y
  - Can also use the variance for scaling different leaves

Conclusion

Random Forest

- Great prediction power
- Comparable with Boosting and SVM
- No need for validation set, use OOB data
- Scalable to large datasets
- Variable importance measures

Proximity measure:

- Data visualization for high dimensional data (many attributes)
- Clustering
- Outlier detection

References

- Leo Breiman, *Random Forests*, Machine Learning, 45, 5-32, 2001
- Leo Breiman, "Looking Inside The Black Box". Wald Lecture II
- Website with code (Fortran)

  http://www.stat.berkeley.edu/users/breiman/RandomForests
- http://www.site.uottawa.ca/~nat/Courses/csi5388/Presentations/RandomForests.ppt
- http://nymetro.chapter.informs.org/prac_cor_pubs/RandomForest_SteinbergD.pdf