RF

A MULTI-USE ENSEMBLE TOOL FOR DATA

Leo Breiman
Statistics Department, UCB
leo@stat.berkeley.edu

I Salute Italy

Outline

1. What is RF?

2. Properties as a classification machine.

    a) excellent accuracy

    b) scales up

    b) handles
        thousands of variables
        many valued categoricals (v.4.0)
        extensive missing values (v4.0)
        badly unbalanced data sets (v4.0)

    c) gives internal unbiased estimate of test set error as trees are added to ensemble

    d) cannot overfit

2) facilities for looking inside the black box

    a) variable importance

    b) outlier detection

    c) data views via scaling

    d) application to unsupervised data

What is RF?= Random Forests

A random forest (RF) is a collection of tree predictors

$$f(\mathbf{x}, \mathbf{T}, \Theta_k), k = 1, 2, \ldots, K)$$

where the Θ_k *are i.i.d random vectors.*

The forest prediction is the unweighted plurality of class votes The LLN insures convergence as $k \to \infty$ and the test set error rates (modulo a little noise) are monotonically decreasing.

The key to accuracy is low correlation and bias. To keep bias low, trees are grown to maximum depth. To keep correlation low, the current version uses this randomization.

1) Each tree is grown on a bootstrap sample of the training set.

2) A number m is specified much smaller than the total number of variables M. At each node, m variables are selected at random out of the M, and the split is the best split on these m variables

The only adjustable parameter in RF is m. The default value for m is $\sqrt{M}$. But RF is not sensitive to the value of m over a wide range.

Two Key Byproducts

The out-of-bag test set

For every tree grown, about one-third of the cases are out-of-bag (out of the bootstrap sample). Called *oob*.

The oob samples can serve as a test set for the tree grown on the non-oob data. This is used to:

i) Form unbiased estimates of the forest test set error as the trees are added.
2) Form estimates of variable importance.

The node proximities

Since the trees are grown to maximum depth, the terminal nodes are small. For each tree grown, pour all the data down the tree. If two data points $\mathbf{x}_n$ and $\mathbf{x}_k$ occupy the same terminal node, then increase $prox(\mathbf{x}_n,\mathbf{x}_k)$ by one.

At the end of forest growing, these proximities, divided by the number of trees, form an intrinsic similarity measure between pairs of data vectors. This is used to:

1) Estimate missing values.
2) Give informative data views via metric scaling.
3) Locate outliers.

Properties as a classification machine.

a) excellent accuracy

b) scales up

\

c) handles
thousands of variables
many valued categoricals (v.4.0)
extensive missing values (v4.0)
badly unbalanced data sets (v4.0)

d) gives internal unbiased estimate of test set error as trees are added to ensemble

e) cannot overfit (already discussed)

Accuracy

My paper: Random Forests , Machine Learning(2001) 45 5-320 gives comparisons:

The test set accuracy of RF is compared to Adaboost on a number of benchmark data sets. RF is slightly better. Adaboost is very sensitive to noise in the labels. RF is not.

Compared to SVMs:

RF is not as accurate as SVMs on pixel image data.

It is superior in document classification.

On benchmark data sets commonly used in Machine Learning the SVM results I have seen show error rates comparable to RF.

Based on my present knowledge, RF is competitive in accuracy with the best classification algorithms that are out there now.

Scaling up to Large Data Sets

The analysis of RF shows that it's compute time is

$$cN_T\sqrt{M}\ N\log(N)$$

N_T = number of trees
M = number of variables
N = number of instances

The constant was estimated with a run on a data set with 15,000 instances and 16 variables.

Using this value leads to the estimate that to grow a forest of 100 trees for a data set with 100,000 instances and 1000 variables would take three hours on my 800Mhz machine.

Parallelizing is Trivial

Each tree is grown independently of the outcomes of the other trees grown. If each of J processors is given the job of growing K trees, there is no need for interprocessor communication until all have finished their runs and the results are aggregated.

Number of Variables

RF has been run on genetic data with thousands of variables and no variable selection and given excellent results.

But there are limits. If the number of noisy variables becomes too large, variable selection will have to be used.

But the threshold for RF is much higher than for non-ensemble methods.

Handling Categorical Variables

Handling categorical values has always been difficult in many classification algorithms.

For instance, given a categorical variable with 20,000 values, how is it to be dealt with? A customary way is to code it into 20000 0-1, variables, a nasty procedure which substitutes 20,000 variables for one.

This occurs in practice--in document classification, one of the variables may be a list of 20,000 words.

RFD handles categorical in the efficient way that CART does--with a fast O(N) algorithm to find the best split of the categoricals at each node.

T

Replacing Missing Values

RF has two ways of replacing missing values.

The Cheap Way

Replace every missing value in the mth coordinate by the median of the non-missing values of that coordinate or by the most frequent value if it is categorical.

The Right Way

This is an iterative process. If the mth coordinate in instance $\mathbf{x}_n$ is missing then it is estimated by a weighted average over the instances $\mathbf{x}_k$ with non-missing mth coordinate where the weight is $prox(\mathbf{x}_n,\mathbf{x}_k)$.

The replaced values are used in the next iteration of the forest which computes new proximities.

The process it automatically stopped when no more improvement is possible or when five iterations are reached.

An Example

The training set for the satellite data has 4434 instances, 36 variables and 6 classes. A test set is available with 2000 instances.

With 200 trees in the forest, the test set error is 9.8% Then 20%, 40%, 60% and 80% of the data were deleted as missing (randomly). Both the cheap fix and the right fix were applied, and the test error computed.

Test Set Error (%)

Missing %	20%	40%	60%	80%
cheap	11.8	13.4	15.7	20.7
right	10.7	11.3	12.5	13.5

It's surprising that with 80% missing data the error rate only rises from 9.8% to 13.5%

I've gotten similar results on other data sets.

Unbalanced Data Sets

A data set is unbalanced if one or more classes--often the classes of interest, are severely underrepresented. This occurs in QSAR data, document classification, and in genetics data.

The algorithm for the 3 class, 21 variable, wave form data was altered to produce both an unbalanced training and test set. The class populations numbers are:

training:	1002	1006	51
test	1521	1404	75

In RF the user can set targets for the relative error rate of each class. Setting 1,1,1, means make all the error rates equal. This is the output of a run with this setting:

oob error:	32.5	32.4	31.4
test set error	31.3	30.3	38.7

The other way is to set a capture rate for the target class. Setting it at 90% for class 3 gives these results.

	capture%	%true 3	% false 3
oob	90.2	18.3	87.8
test set	92.0	23.7	90.3

The OOB Test Set Error Estimate

For every tree grown, about one-third of the cases are oob (out-of-bag --out of the bootstrap sample).

Put these oob cases down the corresponding tree and get a predicted classification for them.

For each case n, pluralize the predicted classification over all the trees that n was oob to get a test set estimate $\hat{y}_n$ for y_n.

Averaging the loss over all n give the oob test set estimate of prediction error.

Runs on many data sets have shown it to be unbiased with error on the order of using a test set of the same size as the training set.

It is computed at user set intervals in the forest construction process and outputted to the monitor.

Science Uses Data To Explore Problems.

Think of the data as being generated by a black box .

A vector of input variables **x** (independent variables) go into one side.

Response variables **y** come out on the other side.

All we see are a sample of data

$$(\mathbf{y}_n, \mathbf{x}_n)\, n = 1, \ldots, N)$$

From this, scientists want to draw conclusions about the mechanism operating inside the black box.

Two important principle:

1) The inferences made about the inside of the black box come from the model(algorithm) you use to fit the data--and not from nature

2) The better the model fits the data, the more sound the inferences about the black box are.

Criterion for How Well the Model Fits

Suppose there is a model $f(\mathbf{x})$ that outputs an estimate $\hat{\mathbf{y}}$ of the true $\mathbf{y}$ for each value of $\mathbf{x}$.

Then a measure of how well f fits the data is given by how close $\hat{\mathbf{y}}$ is to $\mathbf{y}$. This can be measured as follows: given an independent test set

$$(\mathbf{y}'_n, \mathbf{x}'_n)\ n = 1, \ldots, N')$$

and a loss function $L(\mathbf{y}, \hat{\mathbf{y}})$, define the estimated prediction error as

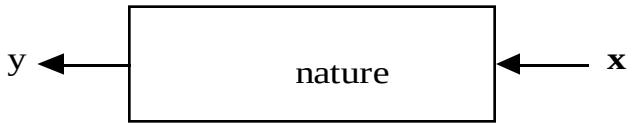
$$PE = av_{n.'} L(\mathbf{y}'_n, f(\mathbf{x}'_{n'}))$$

If there is no test set, use cross-validation to estimate PE.

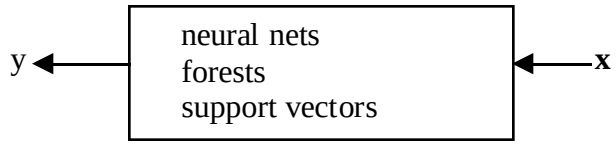
The lower the PE, the better the fit to the data

Information From Our Black Box

Nature forms the outputs **y** from the inputs **x** by means of a black box with complex and unknown interior.



Current most accurate prediction methods are also complex black boxes.



Two black boxes, of which ours seems only slightly less inscrutable than nature's.

My biostatisticians friends tell me, "Doctors can interpret logistic regression." There is no way they can interpret a black box containing fifty trees hooked together. In a choice between accuracy and interpretability, they'll go for interpretability. "

Accuracy vs. Interpretability

Framing the question as the choice between accuracy and interpretability is an incorrect interpretation of what the goal of a statistical analysis is.

The point of a model is to get useful information about the relation between the response and predictor variables.

Interpretability is a way of getting information.

But a model does not have to be simple to provide reliable information about the relation between predictor and response variables.

- *The goal is not interpretability, but accurate information*

RF can supply more and better information about the inside of the black box than any current "interpretable" models.

This will be illustrated by examples.

Tools for Black Box Inspection

i) Estimating variable importance

i.e. which variables are instrumental in the classification.

i) Data views via proximities and metric scaling.

iii) Outlier detection via proximities

iv) A device that makes i)-iii) applicable to unlabeled data

Variable Importance.

Because of the need to know which variables are important in the classification, RF has three different ways of looking at variable importance.

Sometimes influential variables are hard to spot--using these three measures provides more information.

Measure 1

To estimate the importance of the mth variable, in the oob cases for the kth tree, randomly permute all values of the mth variable

Put these altered oob **x**-values down the tree and get classifications.

Proceed as though computing a new internal error rate.

The amount by which this new error exceeds the original test set error is defined as the importance of the mth variable.

Measures 2 and 3

For the nth case in the data, its margin at the end of a run is the proportion of votes for its true class minus the maximum of the proportion of votes for each of the other classes.

The 2nd measure of importance of the mth variable is the average lowering of the margin across all cases when the mth variable is randomly permuted as in method 1.

The third measure is the count of how many margins are lowered minus the number of margins raised.

We illustrate the use of this information by some examples.

An Example--Hepatitis Data

Data: survival or non survival of 155 hepatitis patients with 19 covariates.

Analyzed by Diaconis and Efron in 1983 Scientific American.

The original Stanford Medical School analysis concluded that the important variables were numbers 6, 12, 14, 19.

Efron and Diaconis drew 500 bootstrap samples from the original data set and used a similar procedure, including logistic regression, to isolate the important variables in each bootstrapped data set.

Their conclusion , "Of the four variables originally selected not one was selected in more than 60 percent of the samples.

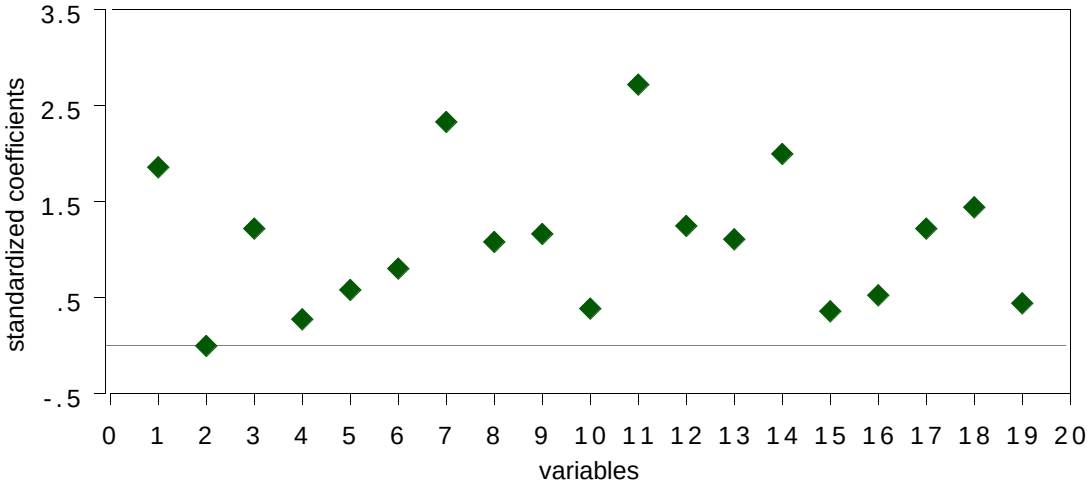
Hence the variables identified in the original analysis cannot be taken too seriously."

Logistic Regression Analysis

Error rate for logistic regression is 17.4%.

Variables importance is based on absolute values of the coefficients of the variables divided by their standard deviations.

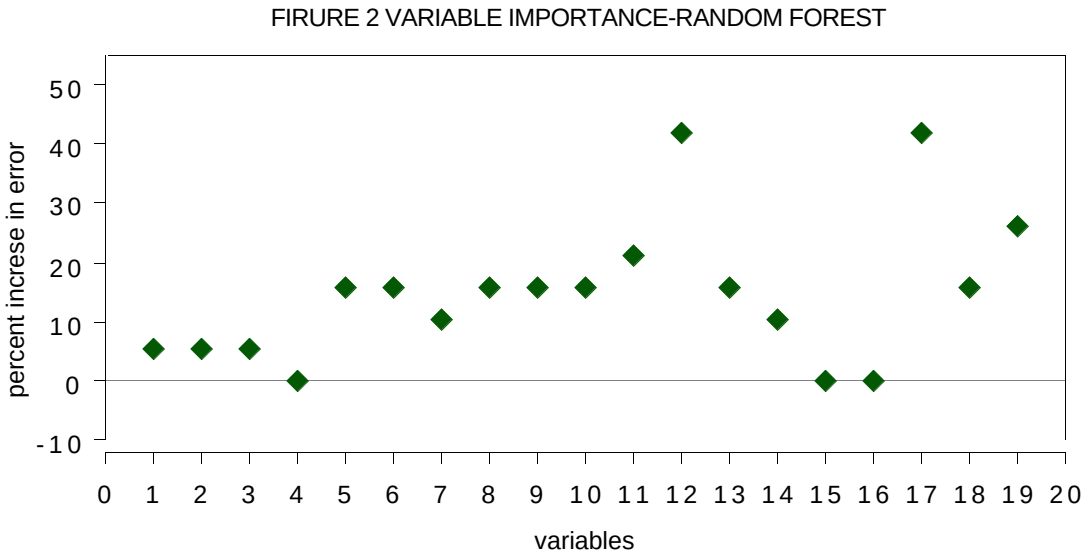
FIGURE 1 STANDARDIZED COEFFICIENTS-LOGISTIC REGRESSION



The conclusion is that variables 7 and 11 are the most important covariates. When logistic regression is run using only these two variables, the cross-validated error rate rises to 22.9% .

Analysis Using RF

The error rate is 12.3%--30% reduction from the logistic regression error. Variable importances (measure 1) are graphed below:



Two variables are singled out--the 12th and the 17th The test set error rates running 12 and 17 alone were 14.3% each.

Running both together did no better. Virtually all of the predictive capability is provided by a single variable, either 12 or 17. (they are highly correlated)

Remarks

There are 32 deaths and 123 survivors in the hepatitis data set. Calling everyone a survivor gives a baseline error rate of 20.6%.

Logistic regression lowers this to 17.4%. It is not extracting much useful information from the data, which may explain its inability to find the important variables.

Its weakness might have been unknown and the variable importances accepted at face value if its predictive accuracy is not evaluated.

The standard procedure when fitting data models such as logistic regression is to delete variables.

Diaconis and Efron (1983) state , "...statistical experience suggests that it is unwise to fit a model that depends on 19 variables with only 155 data points available."

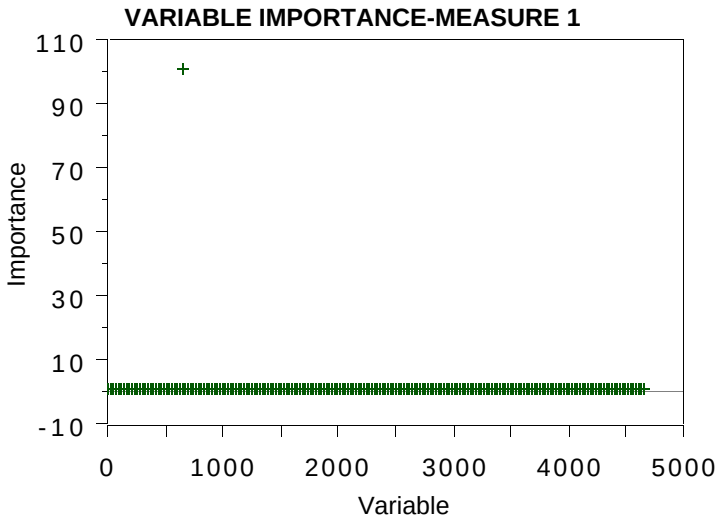
RF thrives on variables--the more the better. There is no need for variable selection ,On a sonar data set with 208 cases and 60 variables, the RF error rate is 14%. Logistic Regression has a 50% error rate.

Microarray Analysis

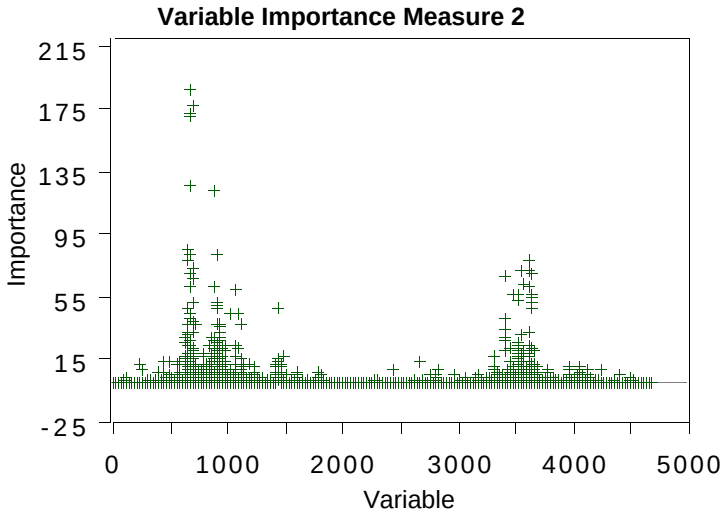
RF was run on a microarray lymphoma data set with three classes, sample size of 81 and 4682 variables (genes) without any variable selection. The error rate was low (1.2%).

What was also interesting from a scientific viewpoint was an estimate of the importance of each of the 4682 gene expressions.

RF was run and the measures of importance computed. Here are the results for the first measure of importance.



Next are the results for the second measure



The graphs show that measure 1 has the least sensitivity, showing only one significant variable.

Measure 2 has more, showing not only the activity around the gene singled ol by measure 1 but also a secondary burst of activity higher up.

Measure 3 (not shown) has too much sensitivity, fingering too many variables.

Using The Proximity Measure To Cluster

bupa liver disorders

This is a two-class biomedical data set consisting of six covariates, the last being alcohol consumption per day.

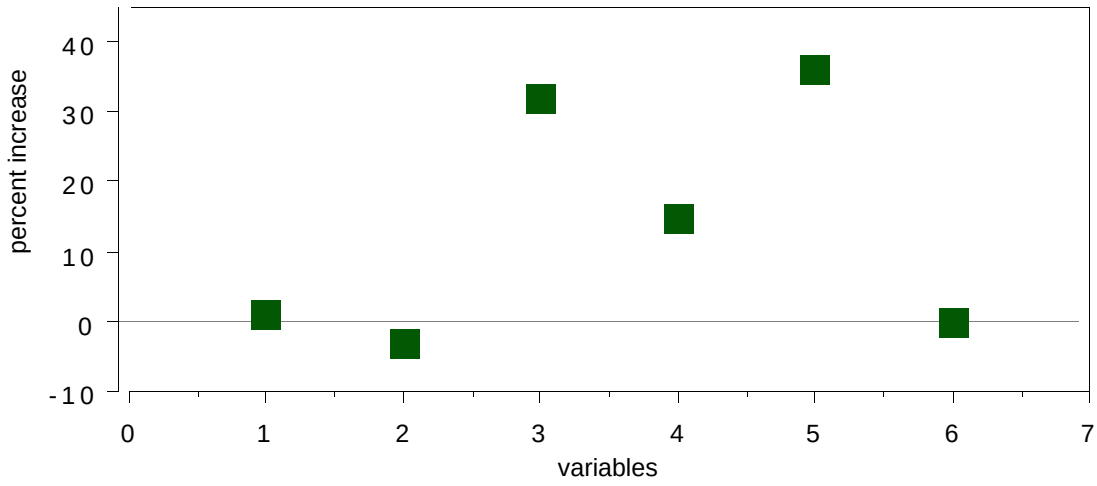
The first five attributes are the results of blood tests thought to be related to liver functioning. The 345 patients are classified into two classes by the severity of their liver disorders.

What can we learn about this data?

The misclassification error rate is 28% in a Random Forests run.

Variable Importance (method 1)

FIGURE 2 VARIABLE IMPORTANCE-BUPA LIVER

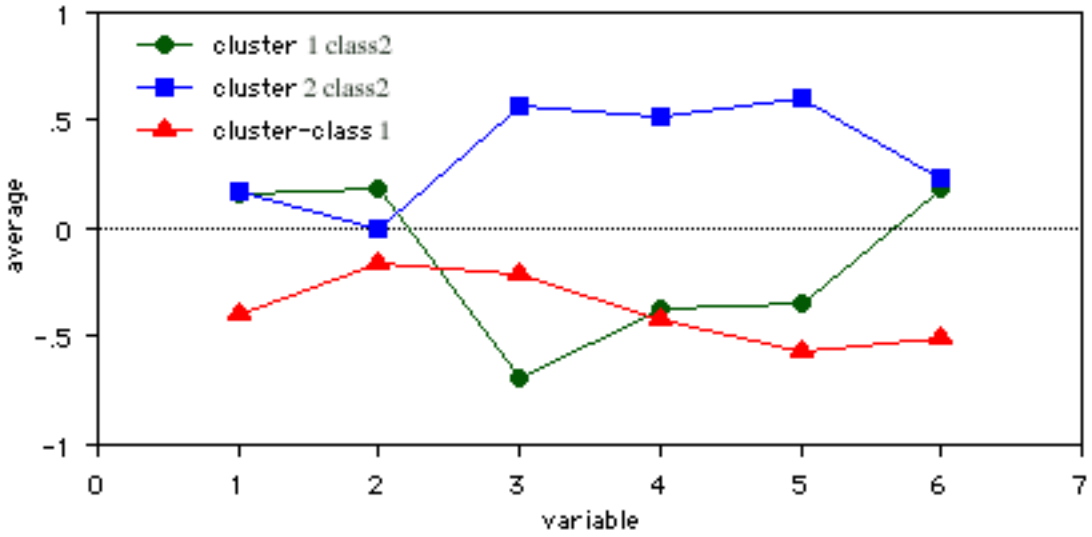


Clustering

Using the proximity measure outputted by RF, there are two class #2 clusters.

In each of these clusters, the average of each variable is computed and plotted:

Figure 3 Cluster Variable Averages



Something interesting emerges. The class two subjects consist of two distinct groups:

Those that have high scores on blood tests 3, 4, and 5 Those that have low scores on those tests.

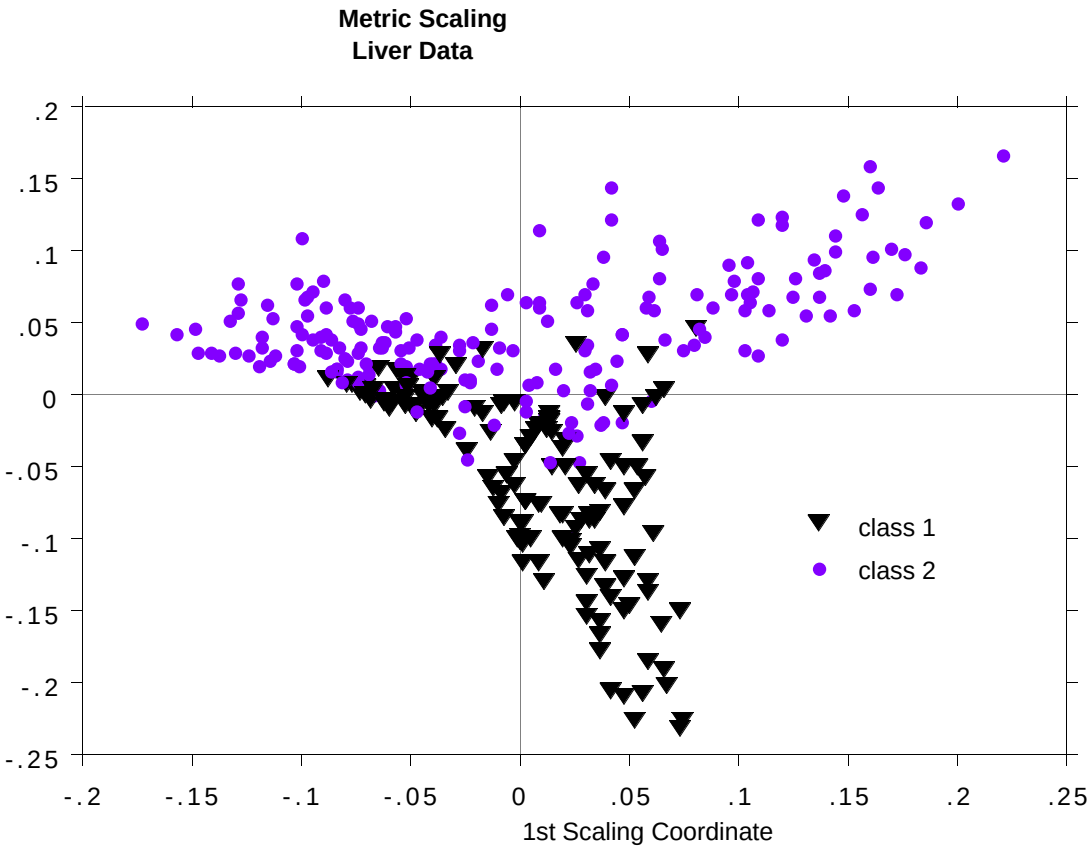
We will revisit this example below.

Scaling Coordinates

The proximities between instances k and n form a matrix {prox(n,k)}. The values 1-prox(n,k) are squared distances in a high-dimensional Euclidean space.

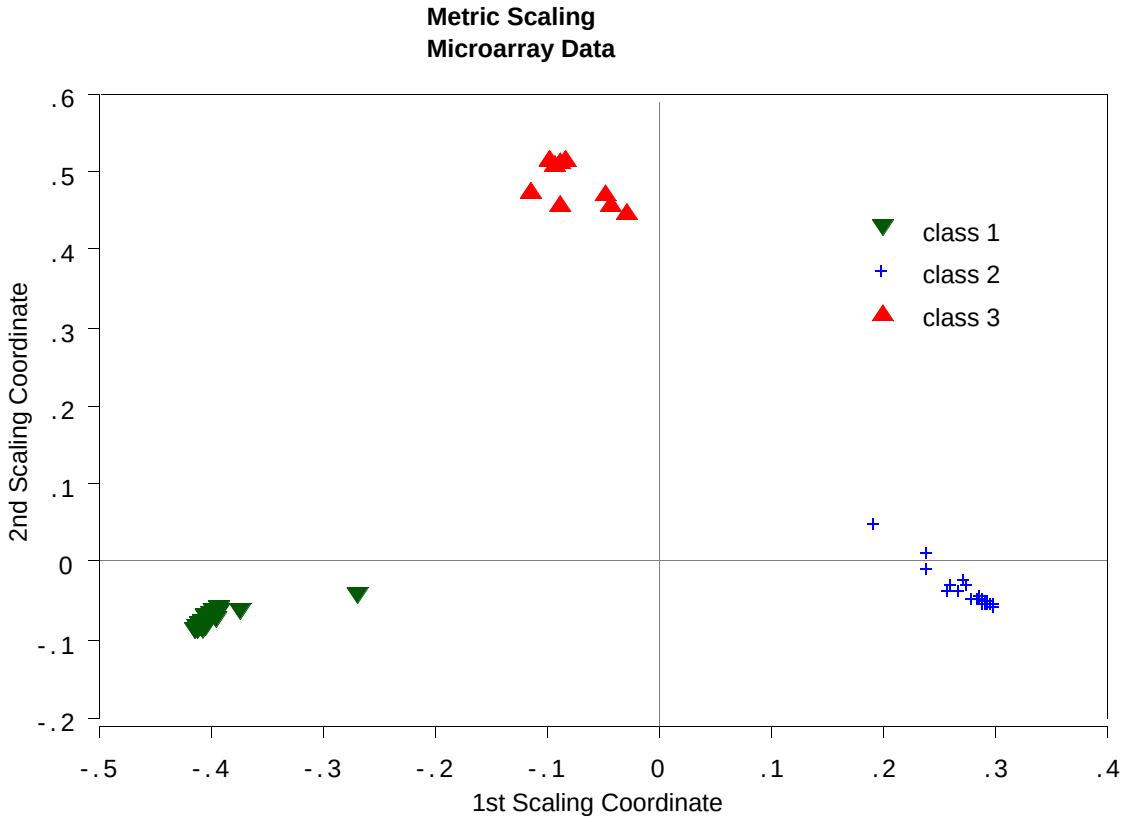
Then, there are *scaling coordinates* which project the data onto lower dimensional spaces while trying to preserving the distances between them.

This is the projection of the liver data.



Projecting the Microarray Data

The next example uses the microarray data. With 4682 variables, it is difficult to see how to cluster this data. Using proximities and the first two scaling coordinates gives this picture:

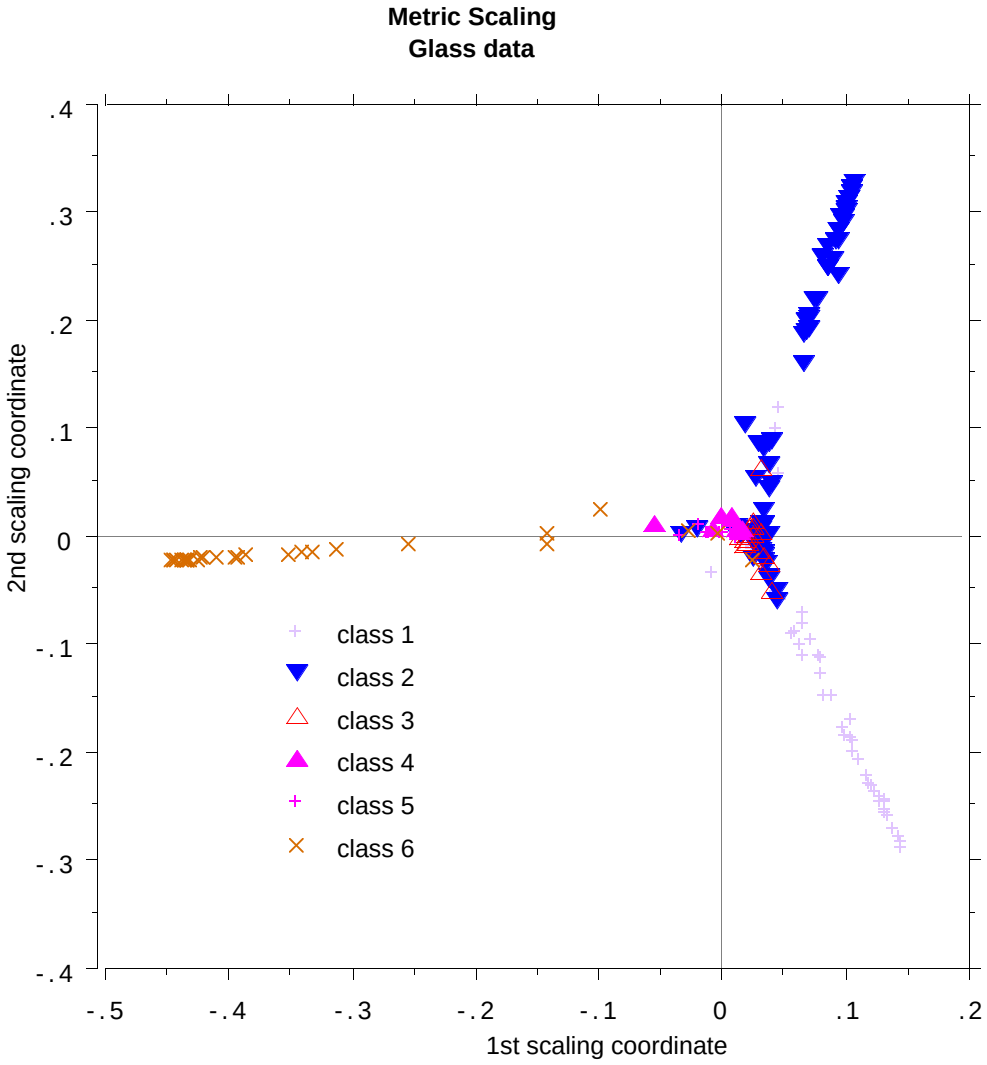


Random forests misclassifies one case. This case is represented by the isolated point in the lower left hand corner of the plot.

Projecting the Glass Data

The third example is glass data with 214 cases, 9 variables and 6 classes. This data set has been extensively analyzed (see Pattern recognition and Neural Networks-by B.D Ripley).

Here is a plot of the 2nd vs. the 1st scaling coordinates.:



Outlier Location

Outliers are defined as cases having small proximities to all other cases.

Since the data in some classes is more spread ol than others, outlyingness is defined only with respect to other data in the same class as the given case.

To define a measure of outlyingness, we first compute, for a case n, the sum of the squares of prox(n,k) for all k in the same class as case n.

Take the inverse of this sum--it will be large if the proximities prox(n,k) from n to the other cases k in the same class are generally small. Denote this quantity by ol(n).

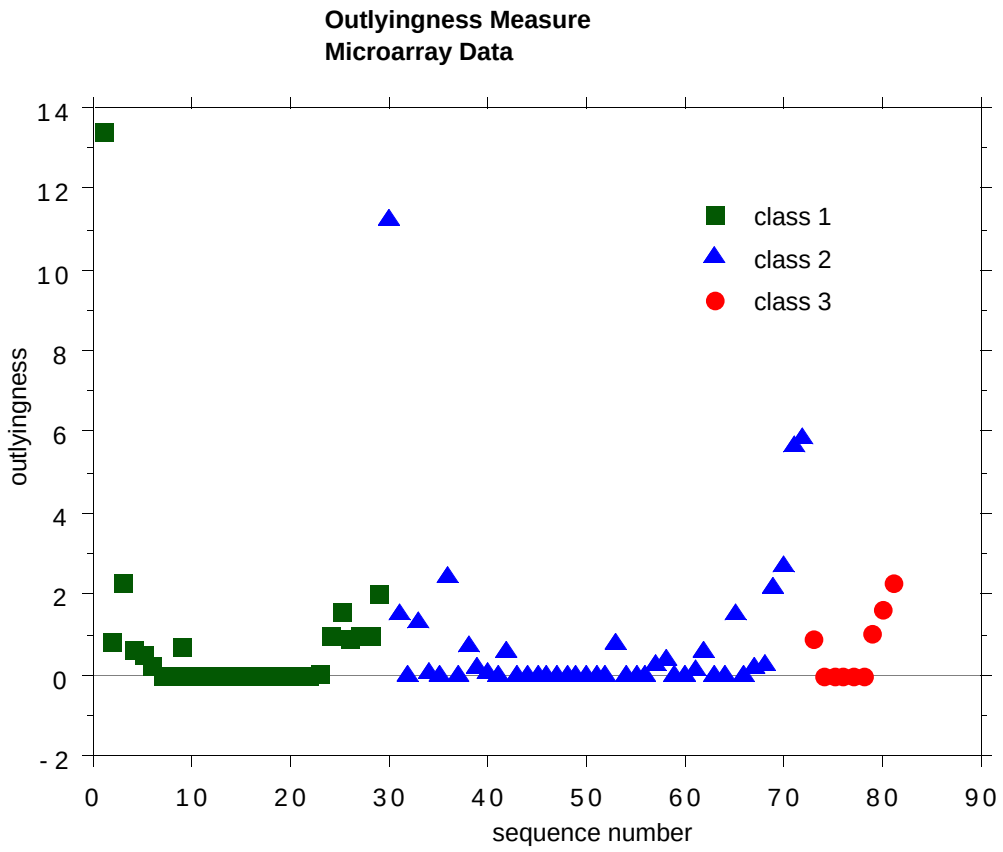
For all n in the same class, compute the median of the ol(n), and then the mean absolute deviation from the median.

Subtract the median from each ol(n) and divide by the deviation to give a normalized measure of outlyingness.

The values less than zero are set to zero. Generally, a value above 10 is reason to suspect the case of being outlying.

Outlyingness In Micorarray Data

Here is a graph of outlyingness for the microarray data

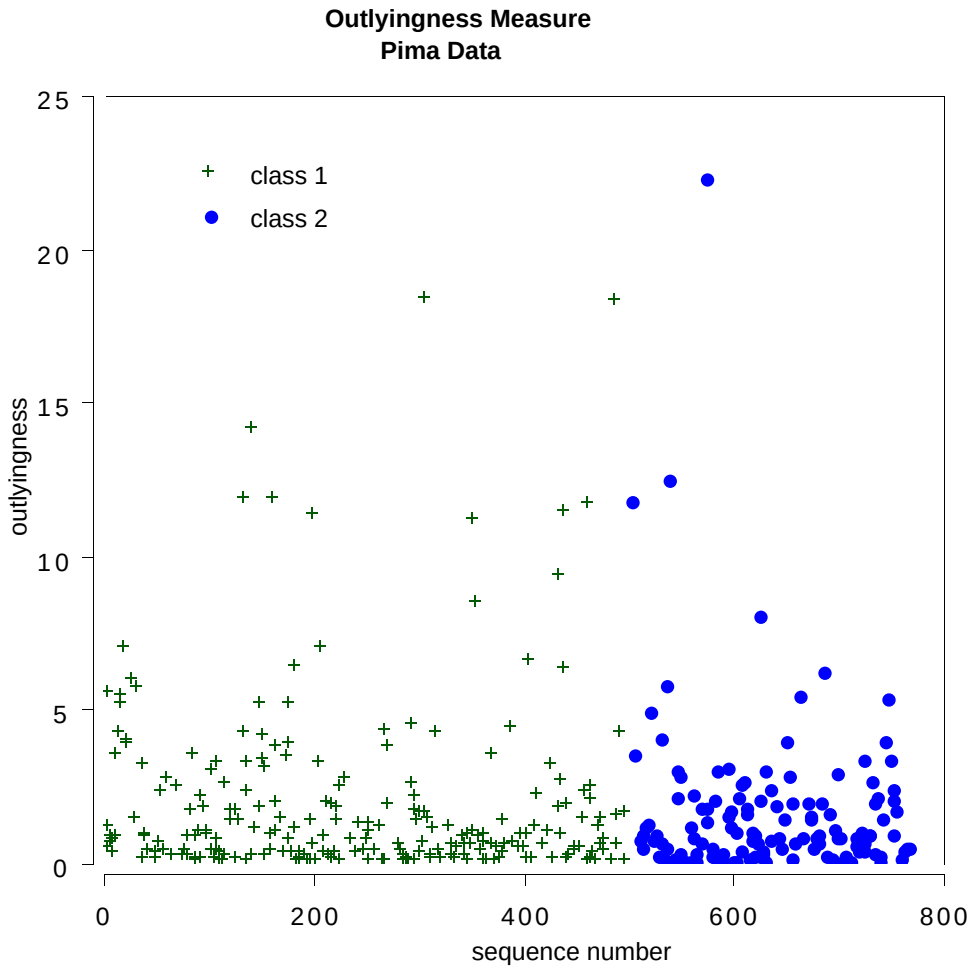


There are two possible outliers--one is the first case in class 1, the second is the first case in class 2.

Outlyingness In Pima Indian Data

As second example, we plot the outlyingness for the Pima Indians hepatitis data. This data set has 768 cases, 8 variables and 2 classes.

It has been used often as an example in Machine Learning research and is suspected of containing a number of outliers.



If 10 is used as a cutoff point, there are 12 cases suspected of being outliers.

Analyzing Unlabeled Data

Using an interesting device, it is possible to turn problems about the structure of unlabeled data (i.e. clusters, etc.) into a classification context.

Unlabeled date consists of N vectors {**x**(n)} in M dimensions. These vectors are assigned class label 1.

A synthetic set of N vectors is created and assigned class label 2.

The second synthetic set is created by independent sampling from the one-dimensional marginal distributions of the original data.

If the value of the mth coordinate of the original data for the nth case is $x(m,n)$, then a case in the synthetic data is constructed as follows:

Its first coordinate is sampled at random from the N values $x(1,n)$, its second coordinate is sampled at random from the N values $x(2,n)$, and so on.

Thus the synthetic data set can be considered to have the distribution of M independent variables where the distribution of the mth variable is the same as the univariate distribution of the mth variable in the original data.

Run RF

When this two class data is run through random forests a high misclassification rate--say over 40%, implies that there is not much dependence structure in the original data.

That is, that its structure is largely that of M independent variables--not a very interesting distribution.

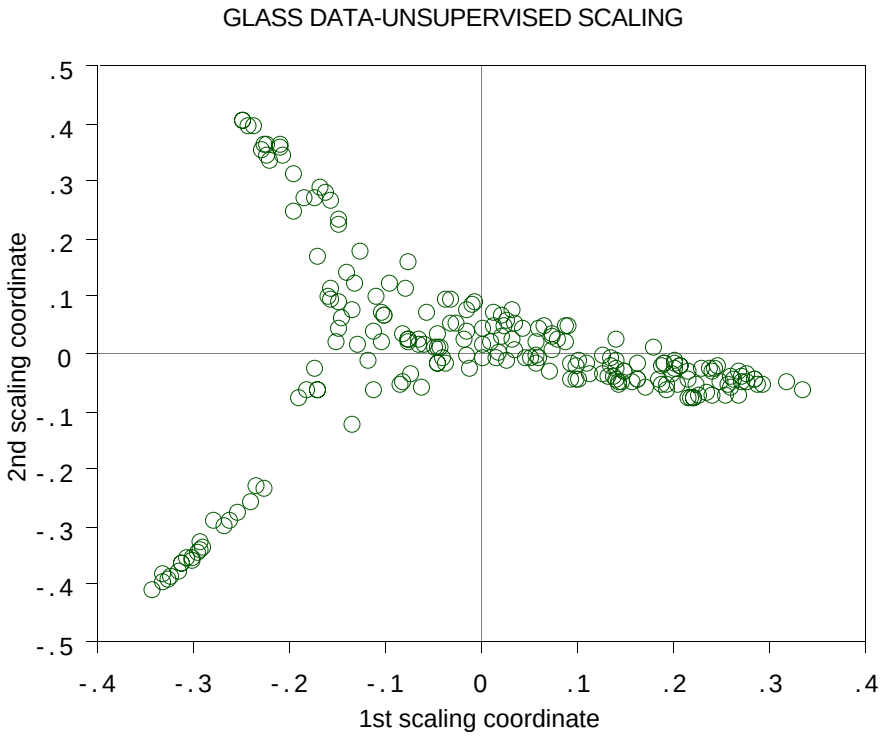
But if there is a strong dependence structure between the variables in the original data, the error rate will be low.

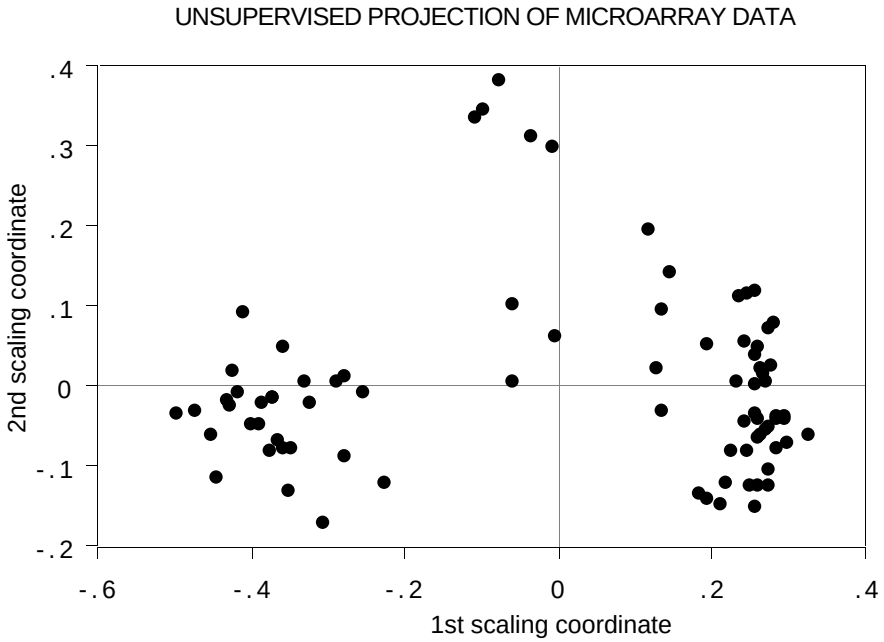
In this situation, the output of random forests can be used to learn something about the structure of the data.

Some examples follow:

Examples

These are the unsupervised projection graphs for the glass and microarray data.

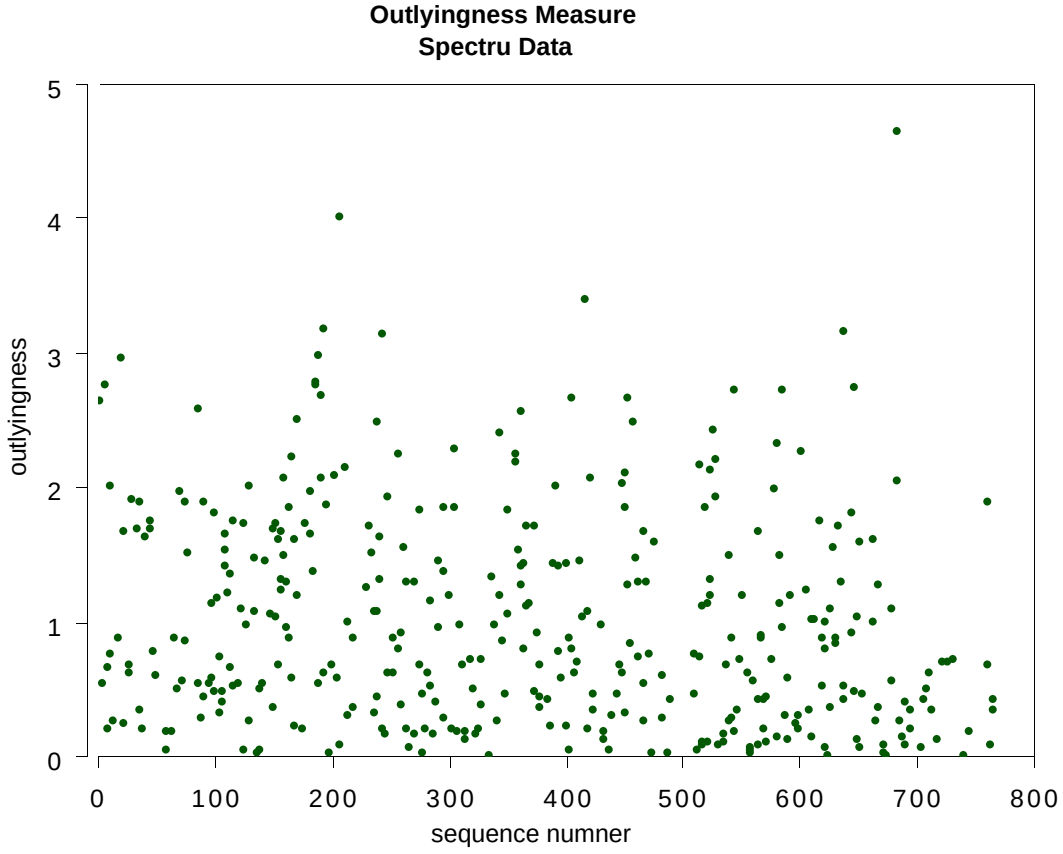




Challenge from Merck

Data supplied by Merck consists of the first 468 spectral intensities in the spectrums of 764 compounds. The challenge presented by Merck was to find small cohesive groups of outlying cases in this data.

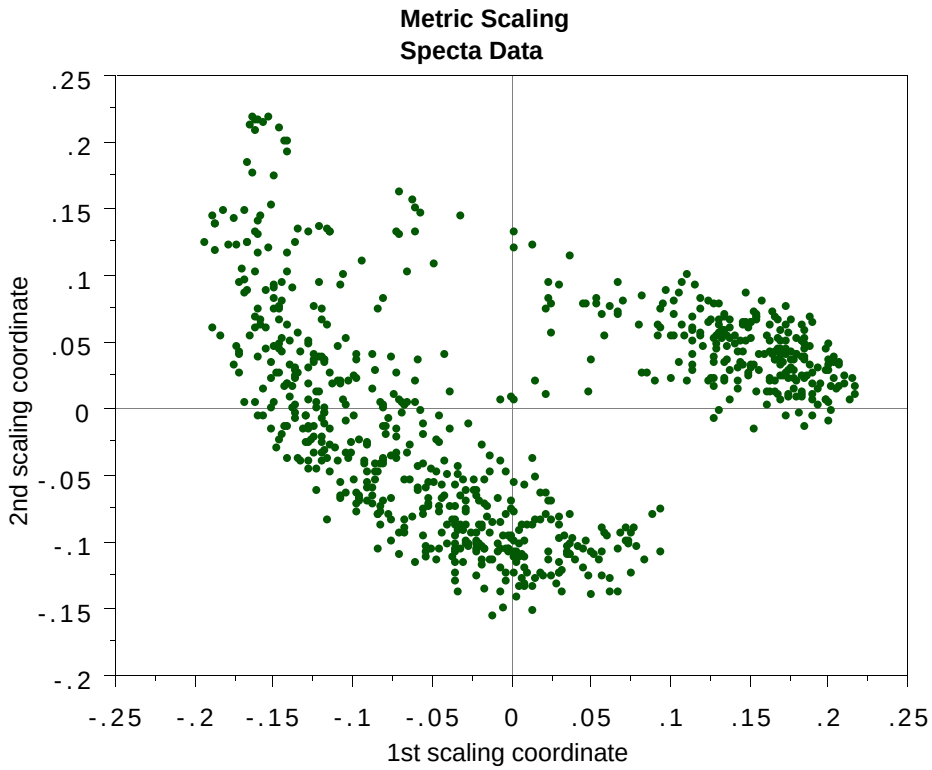
Creating the 2nd synthetic class there was excellent separation with an error rate of 0.5%, indicating strong dependencies in the original data. We looked at outliers and generated this plot.



Using Scaling

This plot gives no indication of outliers. But outliers must be fairly isolated to show up in the outlier display.

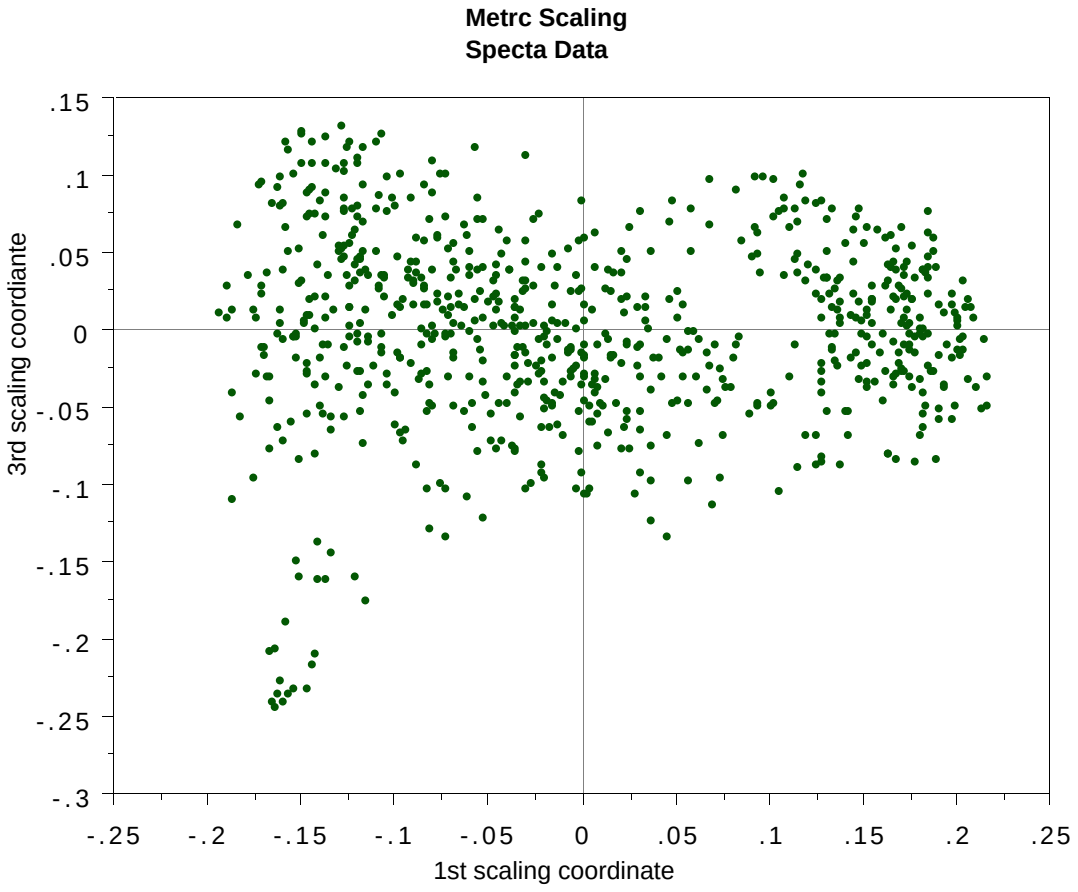
To search for outlying groups scaling coordinates were computed. The plot of the 2nd vs. the 1st is below:



his shows, first, that the spectra fall into two main clusters. There is a possibility of a small outlying group in the upper left hand corner.

To get another picture, the 3rd scaling coordinate is plotted vs. the 1st.

Another Picture



The group in question is now in the lower left hand corner and its separation from the body of the spectra has become more apparent.

To Summarize

i) With any model fit to data, the information extracted is about the model--not nature.

ii) The better the model emulates nature, the more reliable our information.

iii) A prime criterion as to how good the emulation is the error rate in predicting future outcomes.

iv) The most accurate current prediction algorithms can be applied to very high dimensional data, but are also complex.

v) But a complex predictor can yield a wealth of "interpretable" scientific information about the prediction mechanism and the data.

CURTAIN!

Curtain Call:

Random Forests is free software.

www.stat.berkeley,edu / users / breiman