SIAM WORKSHOP MAY 2003

RF / tools

A Class of Two-eyed Algorithms

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

The Data Avalanches and the Genesis of Smart Algorithms

Its hard to construct a terabyte data
base if each item has to be
entered by hand.

The avalanche was started by the
advent of high speed computing
and electronic entering of data.

Aided and assisted by technology
growth such as

high resolution and more band
width in satellite pictures.

more weather stations gathering more
detailed information.

centralized data warehousing for
commercial firms.

The Data Avalanche--A Small Example

San Francisco Chronicle-
October 24,2002

Since Neurome was founded three years ago, its appetite for powerful computers and more data storage has grown so rapidly that it now uses over 100 gigabytes a week.

"And that's just taking it easy" said Warren Young, a co-founder of the biotech company that processes data from digital images of the brain for pharmaceutical research firms. "otherwise we'll run out of space"

Current Avalanches and Problems

Multi-terabyte data bases are becoming common. The interesting data bases are ones used to answer significant questions.

Astronomy data base of one billion stellar objects- soon to be two billion.

what kinds of objects are there in this data base? dwarf stars, quasars...? find objects dissimilar to anything seen before.

NSA data bases of millions of documents.

which of these are relevant to terrorism in Macedonia?

Merck data base of molecular bond structure of millions of chemicals.

which of these are biologically active?

Satellite Photos terabyte data base

has anything changed since the last fly over--of what nature

El Nino data base of hundreds of gigabytes over 30-40 years.

how accurately can El Nino activity be predicted?

Smart Algorithms

Trying to get answers to interesting
questions from a large data base cannot be done
by human inspection.

It requires algorithms initially trained by humans.

They have to be fast and accurate.
Mistakes are often costly.

In document recognition, if the algorithm does not
recognize a relevant documents, important
information may be missed.

In drug searches, incorrectly labeling too many
compounds as being potentially active leads to
high laboratory costs.

How Algorithms are Constructed

Algorithms don't start off being smart. Then have to be trained, by human judgment.

The human has to offer the algorithm examples of what the algorithm needs to distinguish between.

Astronomers have to offer examples of novas, dwarf stars, sun-like stars, etc.

To distinguish between relevant and irrelevant documents, the algorithm has to be shown examples of both as judged by human readers.

In the language of machine learning,
a training set has to offered to the algorithm.

Training Sets and Prediction

The most common situation is that there is a single response variable y to be predicted, i.e.

relevant, non relevant
dwarf star, super nova, etc.

The training set consists of a number of examples, each assigned a y-value by human judgment , plus a set of m measurements on each example denoted $x(1), x(2), \ldots, x(M)$.

For instance, in document recognition, the measurements are the word counts in the document of a specified list of words.

Each stellar object recorded in the sky has measurements on its spectrum, shape, extent, etc.

The algorithm then proceeds to make predictions for the objects it has not seen using only the measurements **x** on the object to form a predicted y.

Its error rate on the objects whose identity (y) is unknown is called: *the generalization error (use error for short)*

The lower the generalization error, the smarter the algorithm

Two-eyed Data Algorithms

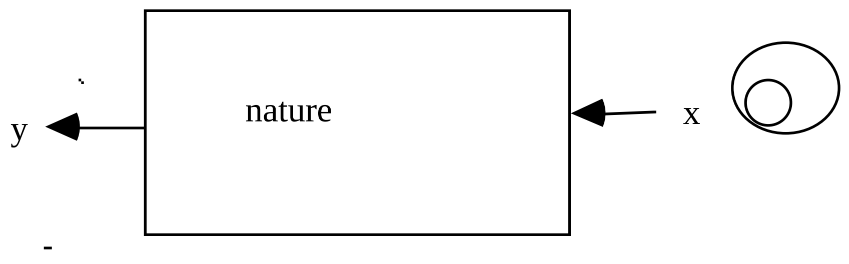
Two algorithm currently give the most accurate predictions in classification and regression

1. Support Vector Machines

2. Random Forests

They have their eye on accurate prediction.

Think of nature as a black box:



In prediction the eye is not concerned with the inside of the black box.

Given an input **x** the goal of the algorithm is to produce a prediction $\hat{y}$ as close as possible to the **y** that "nature" produces.

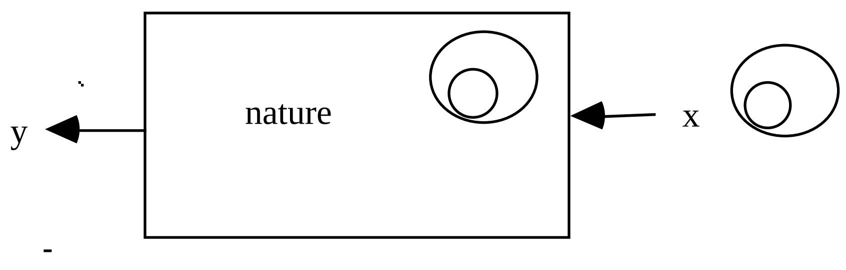
"nature"= nature +humans that classify what nature produces.

The Second Eye

In most scientific applications it is not enough to have accurate predictions--more information is necessary.

Essential information are answers to:

What's going on inside the black box?



For Instance

In analyzing microarray data having thousands of variables and sample size usually less that 100, it is often put as a classification problem.

But the goal in the analysis is to find the genes that are important in discriminating between the classes.

A Statistical Principle and Quandary

Inferring a mechanism for the black box is a highly risky and ambiguous venture. Nature's mechanisms are generally complex and cannot summarized by a relatively simple stochastic model, even as a first approximation.

An important principle

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

Suppose there is an algorithm $f(\mathbf{x})$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 $\mathbf{y}$. i.e. by the size of the prediction error (PE)

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

Quandary: The most accurate prediction algorithms are also the most complex and inscrutable.

i.e. compare trying to understand 100 trees in an ensemble as compared to a single tree structure as in CART.

Two Eyed RF/tools

I am developing a class of algorithms called RF/tools that have both:

1. Excellent accuracy--the best currently available.

2. Gives good insights into the inside of the box.

There is a:

classification version (RF/cl)

regression version (RF/rg)

Soon to be joined by a multiple dependent output version (RF/my)

These tools are free open source (f77) and available at:

www.stat.berkeley.edu/RFtools

with manuals for their use and interfaces to R and S+.

We estimate that there have been almost 3000 downloads since the algorithms were put on the web site.

Outline of Where I'm Going

What is RF?

A) The Right Eye (classification machine)

i) excellent accuracy

ii) scales up

iii) handles
- thousands of variables
- many valued categoricals
- extensive missing values
- badly unbalanced data sets

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

v) cannot overfit

B) ***The Left Eye (inside the black box)***

i) variable importance

ii) outlier detection

iii) data views via scaling

<u>From Supervised to Unsupervised</u>

How about data with no class labels,
or regression y-values.

Outliers?
Clustering?
Missing values replacement?

By a slick device RF/cl can be turned
into an algorithm for analyzing 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 Law of Large Numbers insures convergence as $k \to \infty$

The test set error rates (modulo a little noise) are monotonically decreasing and converge to a limit.

That is: *there is no overfitting as the number of trees increases*

The key to accuracy is low correlation and bias.

To keep bias low, trees are grown to maximum depth.

Randomization for Low Correlation

To keep correlation low, the current version uses this randomization.

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

ii) A number **m** is specified much smaller than the total number of variables M.

iii) At each node, **m** variables are selected at random out of the M.

iv) The split used 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). A

bbreviated *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.

ii) 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,

increase $prox(\mathbf{x}_n,\mathbf{x}_k)$ by one.

At the end of forest growing, these proximities, form an intrinsic similarity measure between pairs of data vectors.

This is used to:

i) Estimate missing values.

ii) Give informative data views via metric scaling.

iii) Locate outliers.

Properties as a classification machine.

a) excellent accuracy

b) scales up

\

c) handles
thousands of variables
many valued categoricals
extensive missing values
badly unbalanced data sets

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 SVMs have 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.

Ditto regression

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

Unlike other algorithms where variable selection has to be used if there are more than a few hundred variables, RF thrives on variables, the more the merrier.

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.

In two class problems, RF handles categoricals in the efficient way that CART does--with a fast O(N) algorithm to find the best split of the categoricals at each node.

If there are more than two classes, a fast iterative algorithm is used.

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.

Class Weights

A data set is unbalanced if one or more classes--often the classes of interest, are severely underrepresented.

These will tend to have higher misclassification rates than the larger classes.

In some data sets, even though the data set is not badly unbalanced, some classes may have a larger error rate than the others.

In RF it is possible to set class weights that act "as if" they are increasing the size of a class that gets weight larger that one.

The result is that the class weights can be set to get any desired distribution of errors among the classes.

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.

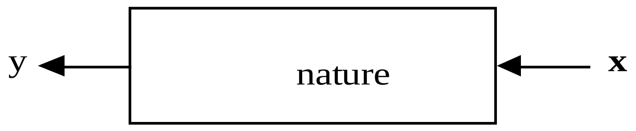
Averaging the loss over all n give the oob test set estimate of prediction error.

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.

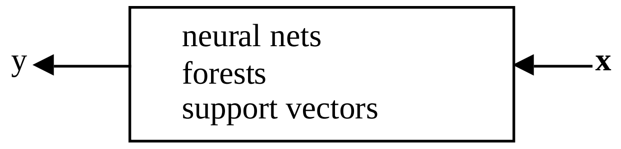
It is computed at user set intervals in the forest construction process and outputted to the monitor.

Accuracy vs Interpretability

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 A False Canard

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 as well as other information about the data structure.

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.

The Left Eye-
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 for doing similar analyses on unsupervised data

Variable Importance.

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.

To estimated the importance of the mth variable:

i) In the oob cases for the kth tree, randomly permute all values of the mth variable

ii) Put these new variable values down the kth tree and get classifications.

iii) Compute the margin.

The measure of importance of the mth variable is the average lowering of the margin across all cases when the mth variable is randomly permuted.

An Example--Hepatitis Data

Data: Survival (123) or non-survival (32) 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.

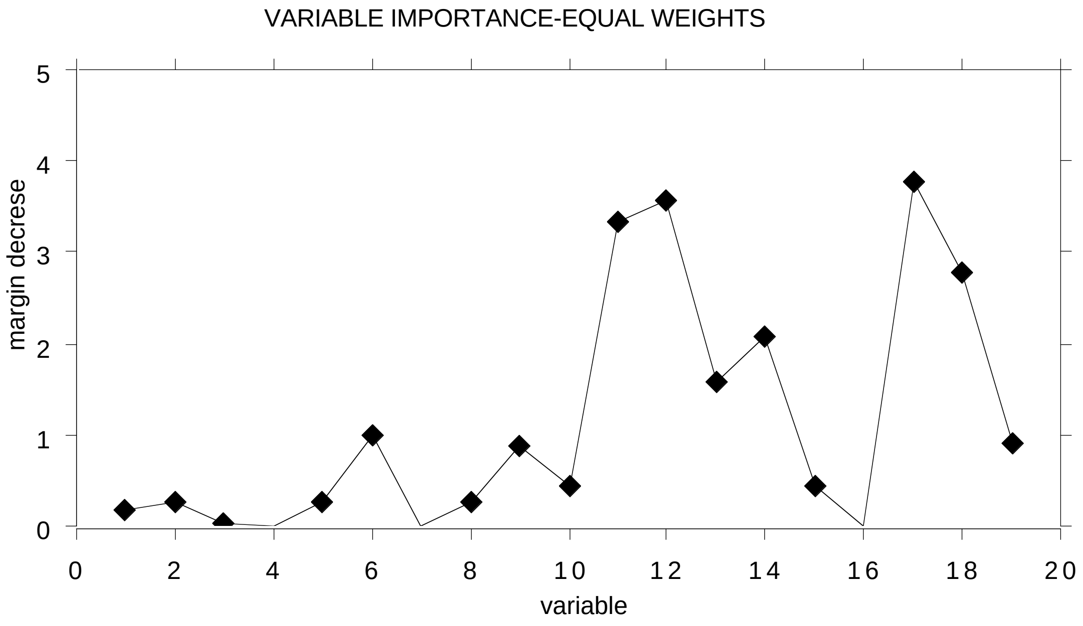
Error rate for logistic regression is 17.4%.

Efron and Diaconis drew 500 bootstrap samples from the original data set and looked for 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."

Analysis Using RF

The overall error rate is 14.2%. There is a 53% error in class 1, and 4% in class 2. The variable importances are graphed below:

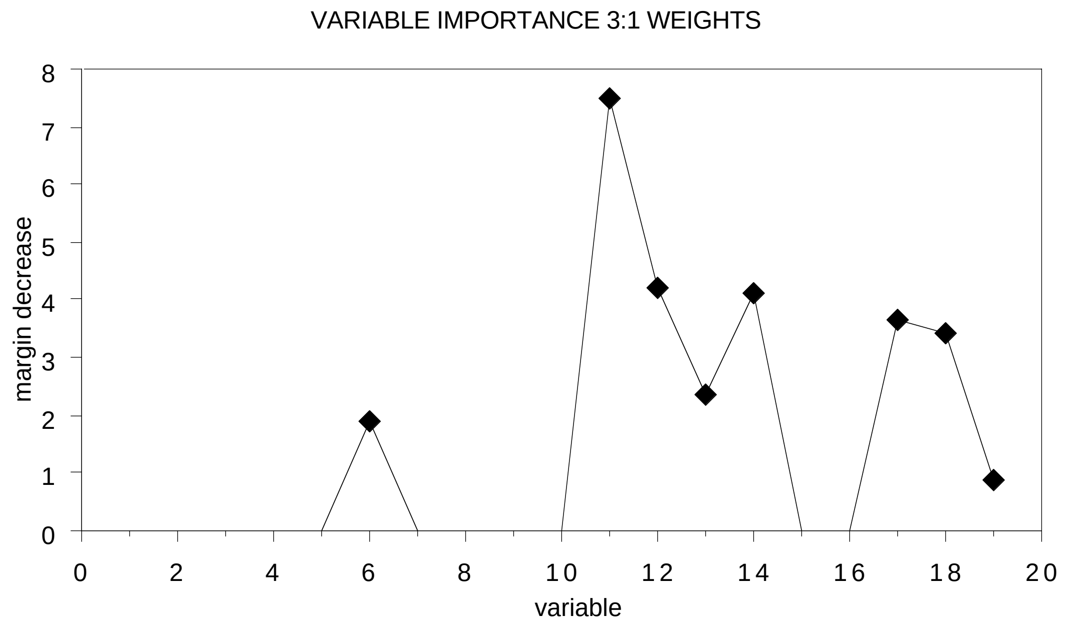


The three most important variables are 11,12,17.

Since the class of interest is non-survival which, with equal weights, has a high error rate, the classweight of class 1 was increased to 3.

The run gave an overall error rate of 22%, ,with class 1 error 19% and 23% for class 2.

The variable importances for this run are graphed below:



Variable 11 is the most important variable in separating non-survival from survival.

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."

Newer methods in Machine Learning thrive on variables--the more the better. The next example is an illustration.

Microarray Analysis

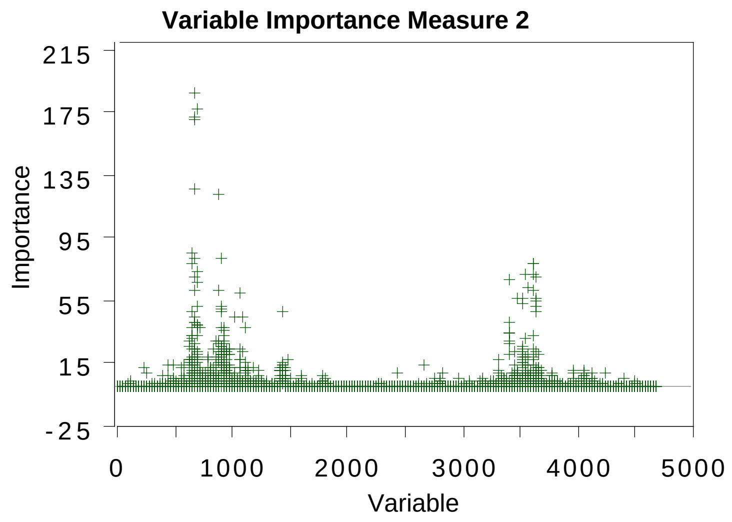
Random forests was run on a microarray lymphoma data set with three classes,

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 genes.

The graph below were produced by a run of random forests.



An Intrinsic Proximity Measure and Clustering

Since an individual tree is unpruned, the terminal nodes will contain only a small number of instances.

Run all cases in the training set down the tree. I

If case i and case j both land in the same terminal node increase the proximity between i and j by one.

At the end of the run, the proximities are normalized by dividing by twice the number of trees in the forest.

To cluster-use the proximity measures.

Example-Bupa Liver Disorders

This is a two-class biomedical data set consisting of the covariates

1. mcv	mean corpuscular volume
2. alkphos	alkaline phosphotase
3. sgpt	alamine aminotransferase
4. sgot	aspartate aminotransferase
5. gammagt	gamma-glutamyl transpeptidase
6. drinks	number of half-pint equivalents of alcoholic beverage drunk 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.

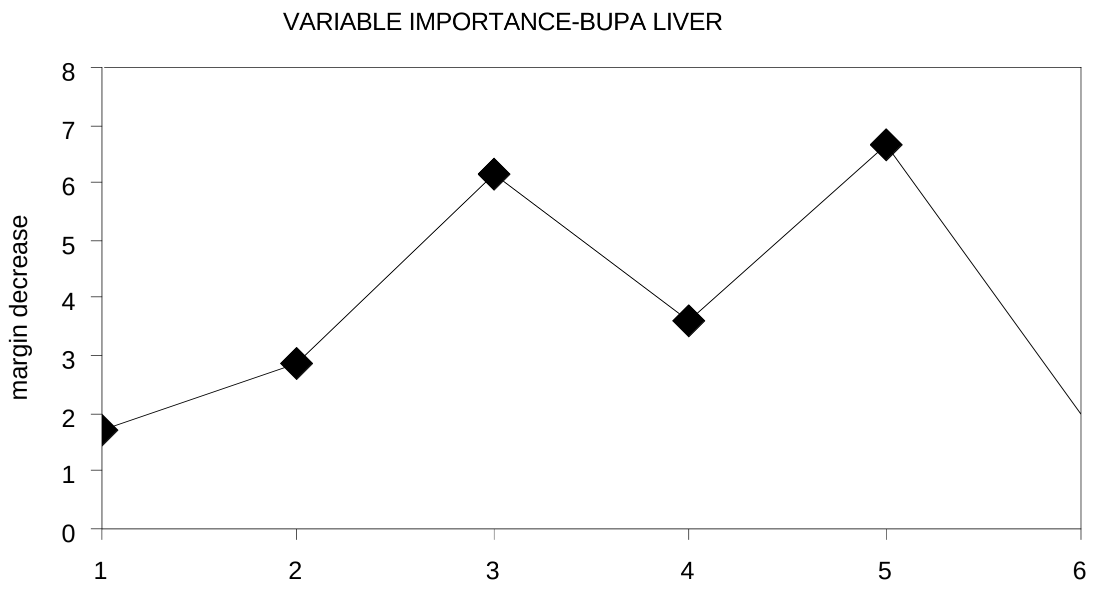
The class populations are 145 and 200(severe).

The misclassification error rate is 28% in an RF run.

Class 1 has a 50% error rate with a rate of 12% for class 2.

Setting the weight of class 2 to 1.4 gives an overall rates 28% and 31% for classes 1 and 2.

Variable Importance



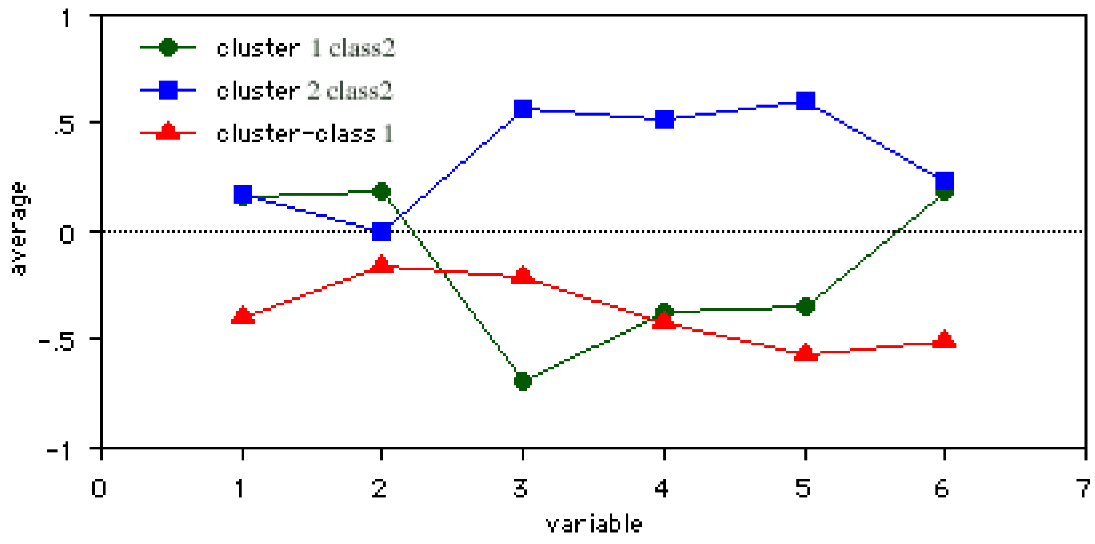
Blood tests 3 and 5 are the most important, followed by test 4.

Clustering

Using the proximity measure given by RF to cluster, 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.

Scaling Coordinates

The proximities between cases n and k form a matrix {prox(n,k)}.

The values 1-prox(n,k) are squared distances between vectors $\mathbf{x}(n), \mathbf{x}(k)$in a Euclidean space of dimension not greater than the number of cases.

The goal is to project the $\mathbf{x}(n)\}$ down into a low dimensional space while preserving the distances between them to the extent possible.

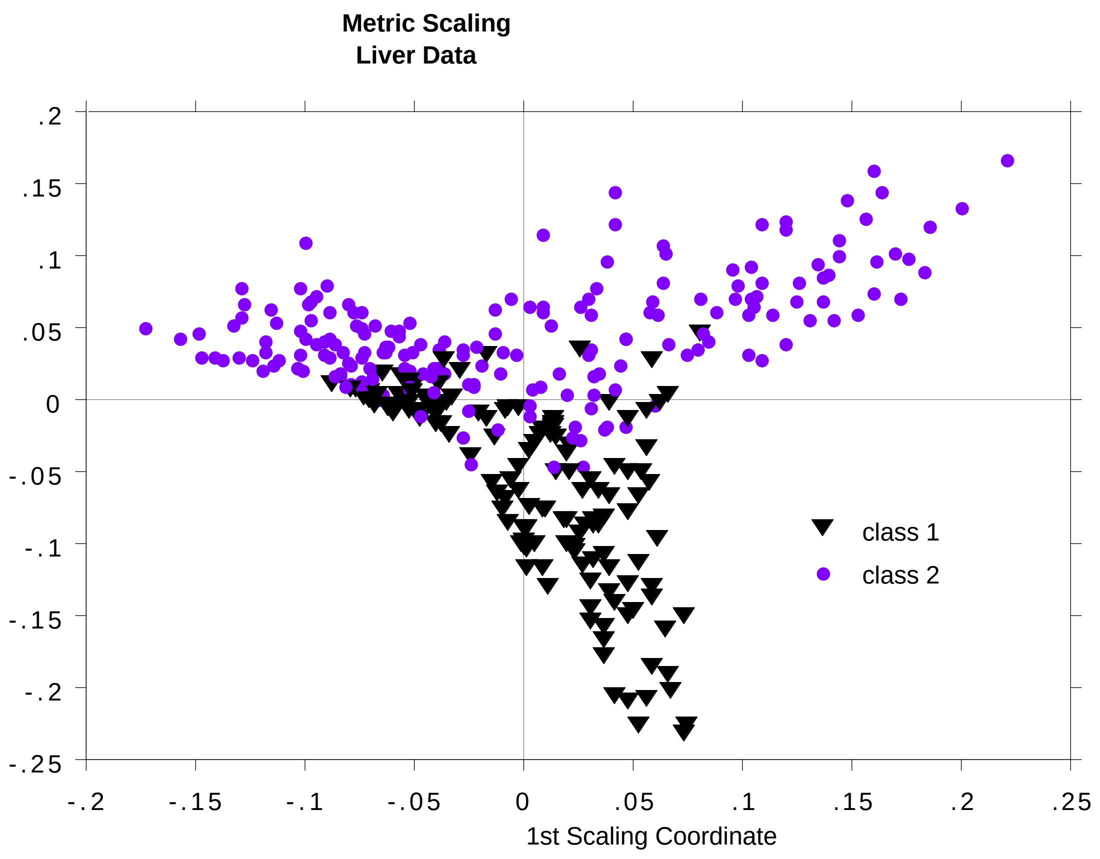
In metric scaling, the idea is to approximate the vectors **x**(n) by the first few scaling coordinates.

These correspond to eigenvectors of a modified prox matrix. T

The two dimensional plots of the ith scaling coordinate vs. the jth often gives useful information about the data.

The most useful is usually the graph of the 2nd vs. the 1st.

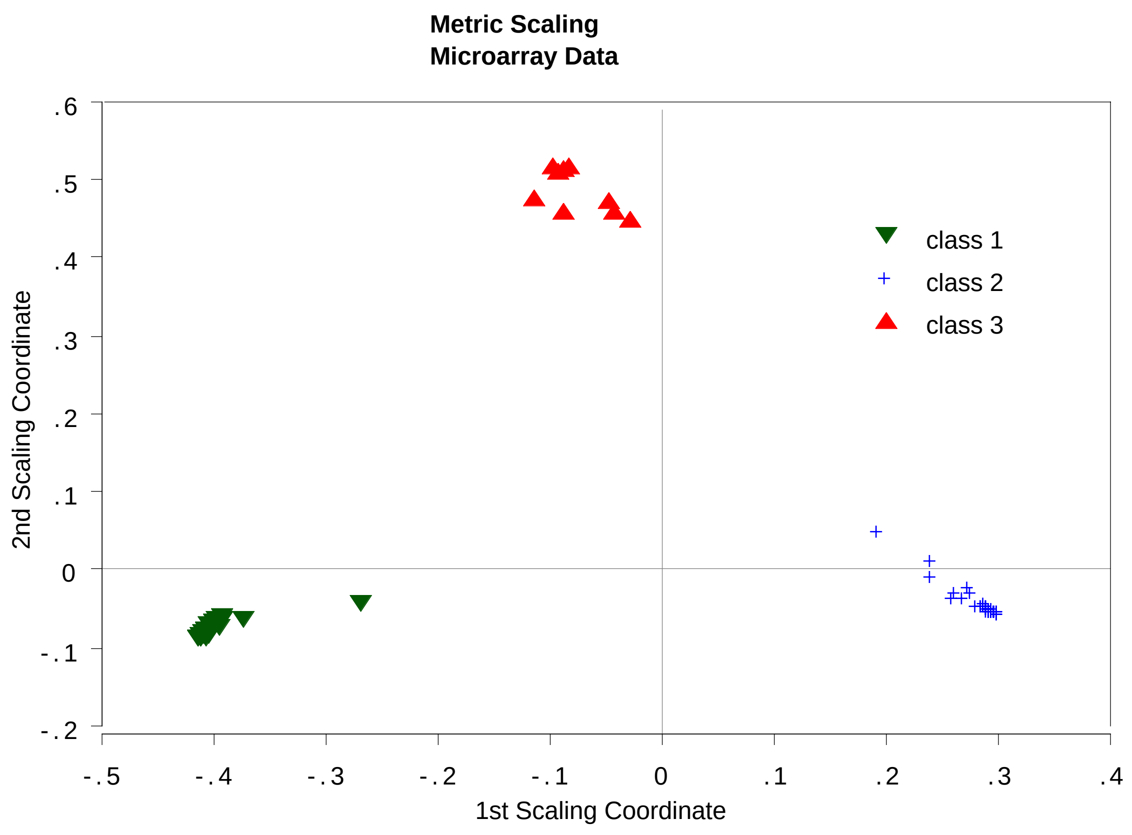
We illustrate with three examples. The first is the graph of 2nd vs. 1st scaling coordinates for the liver data



The two arms of the class #2 data in this picture correspond to the two clusters found and discussed above.

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:

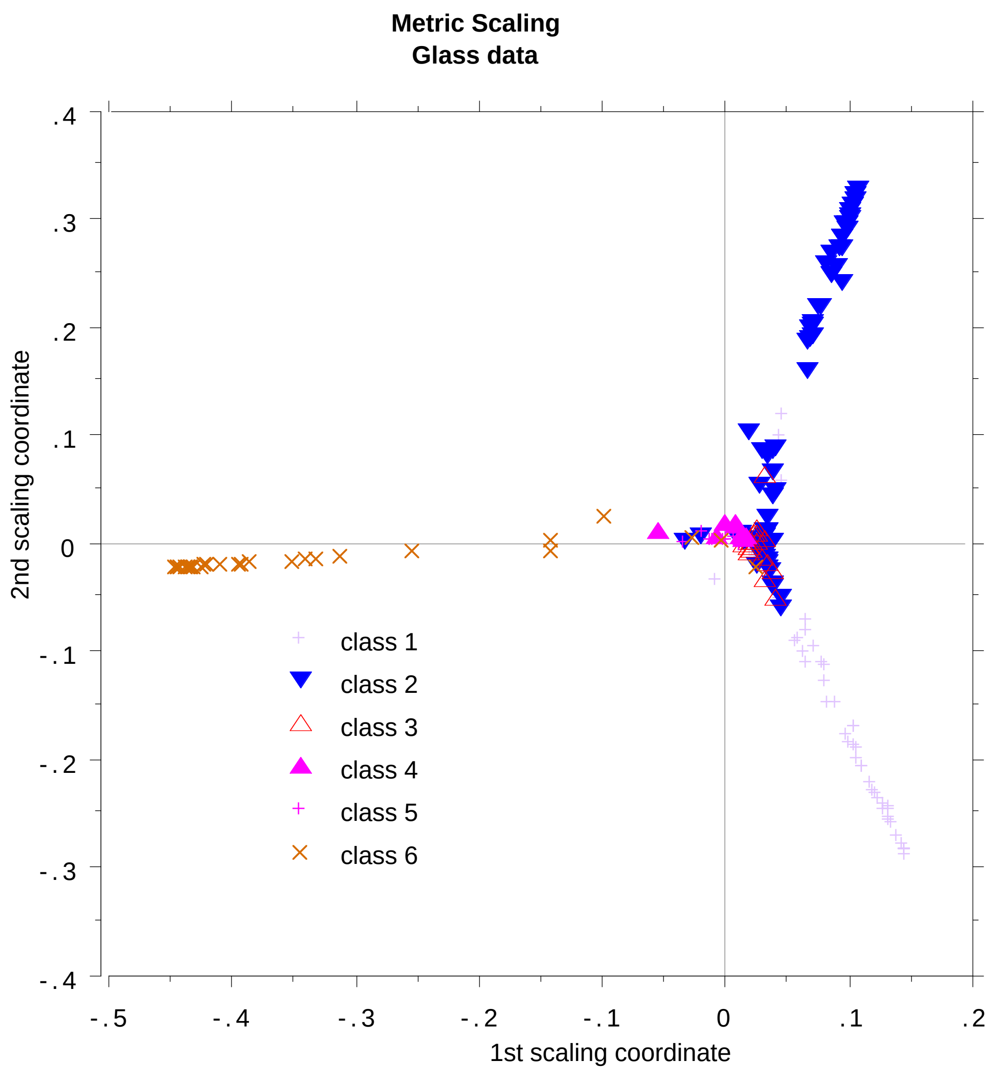


RF misclassifies one case.

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

The Glass Data

The third example is glass data with 214 cases, 9 variables and 6 classes. Here is a plot of the 2nd vs. the 1st scaling coordinates.:



None of the analyses to data have picked up this interesting and revealing structure of the data--compare the plots in Ripley's book.

Outlier Location

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

Since the data in some classes is more spread out 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,

i) Compute, for a case n, the sum of the squares of prox(n,k) for all k in the same class as case n.

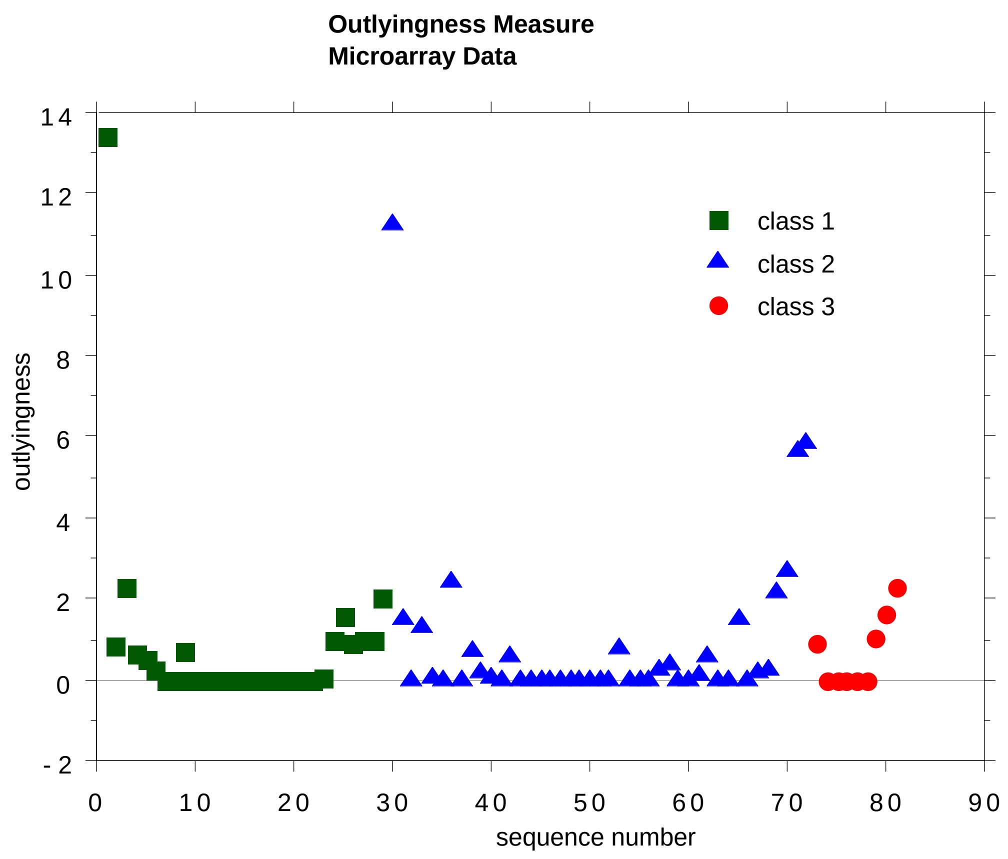
ii)Take the inverse of this sum--it will be large if the proximities prox(n,k) from n to the other cases k in the same class are generally small. Denote this quantity by out(n).

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

Subtract the median from each out(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.

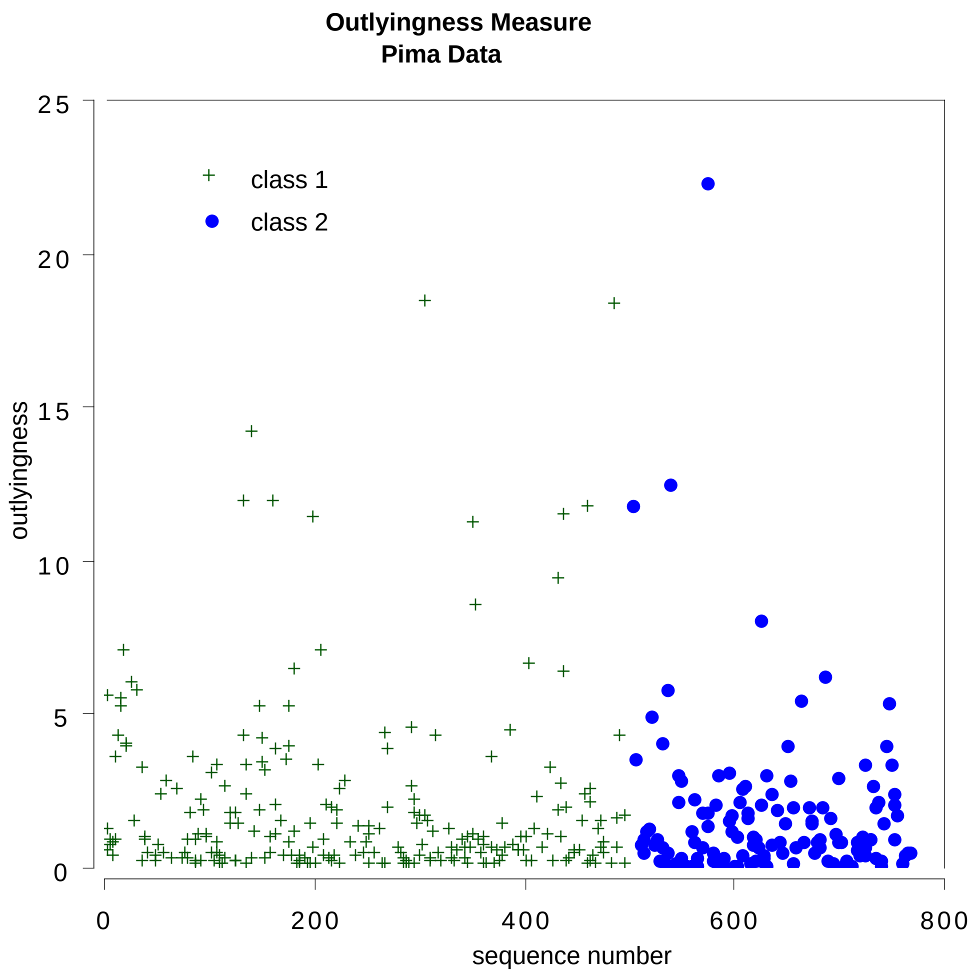
Outlyingness Graph--Microarray Data



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

Outlyingness Graph-Pima Indians Data.

This hepatitus data set has 768 cases, 8 variables and 2 classes. It has been used often as an example in Machine Learning research but is suspected of containing a substantial number of outliers.



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

From Supervised to Unsupervised

Unsupervised date consists of N vectors {**x**(n)} in M dimensions.

Desire:

i) Clustering

ii) Outlier detection

iii) Missing value replacement

All this can be done using RF

i) Give class label 1 to the original data.

ii) Create a synthetic class 2 by sampling independently from the one-dimensional marginal distributions of the original data.

The Synthetic Second Class

More explicitly:

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:

i) Its first coordinate is sampled at random from the N values x(1,n).

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

The synthetic data set has 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.

The Synthetic Two Class Problem

The are now two classes where the second class has been constructed by destroying all dependencies in the original unsupervised data.

When this two class data is run through RF 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 RF can be used to learn something about the structure of the data. Following are some examples.

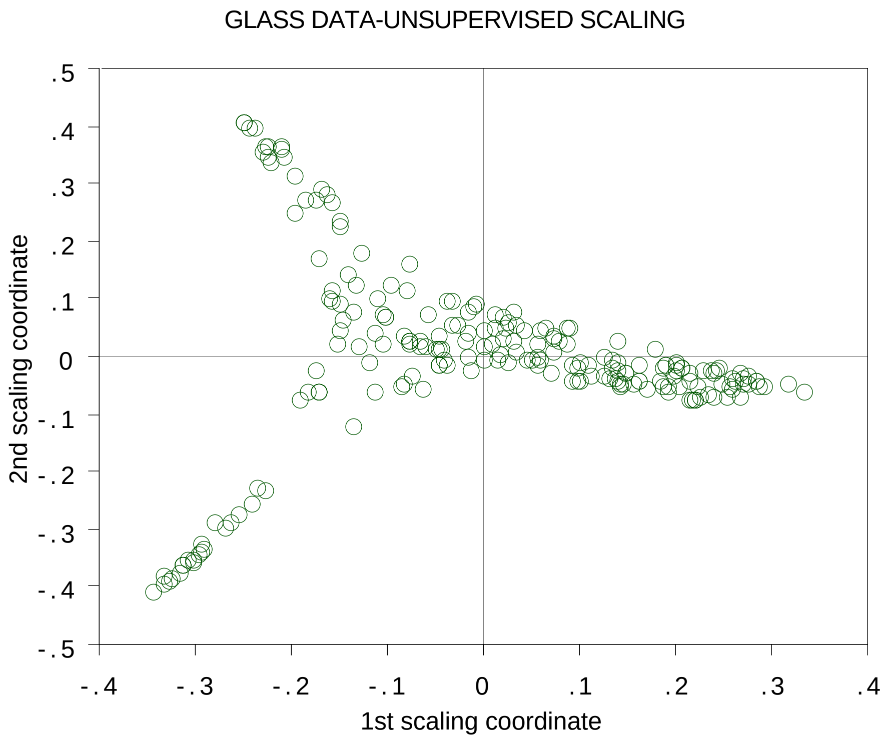
Clustering the Glass Data

All class labels were removed from the glass data.

Unsupervised data consisting of 214 nine-dimensional vectors that were labeled class 1. A second synthetic data set was constructed.

The two class problem was run through RF. Proximities for class 1 only and metric scaling projected the class 1 data onto two dimension.

Here is the outcome:



This is a good replica of the original starfish-like projection using the class data..

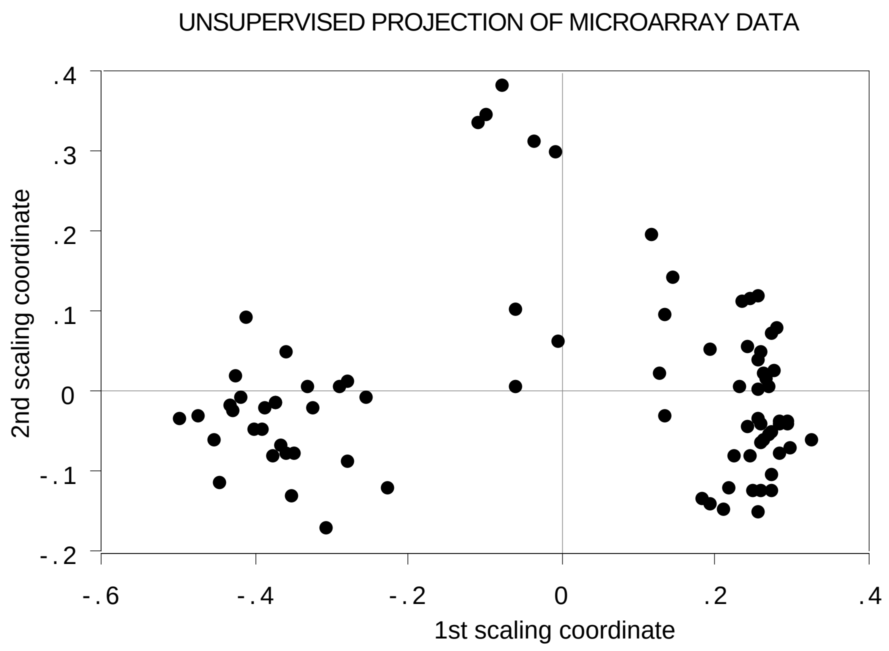
`

Application to the Microarray Data

Recall that the scaling plot of the microarray data showed three clusters--

two larger ones in the lower left hand and right hand corners and a smaller one in the top middle.

Again, we erased labels from the data and projected down an unsupervised view:



The three clusters are more diffuse but still apparent.

Finding Outliers

An Application to Chemical Spectra

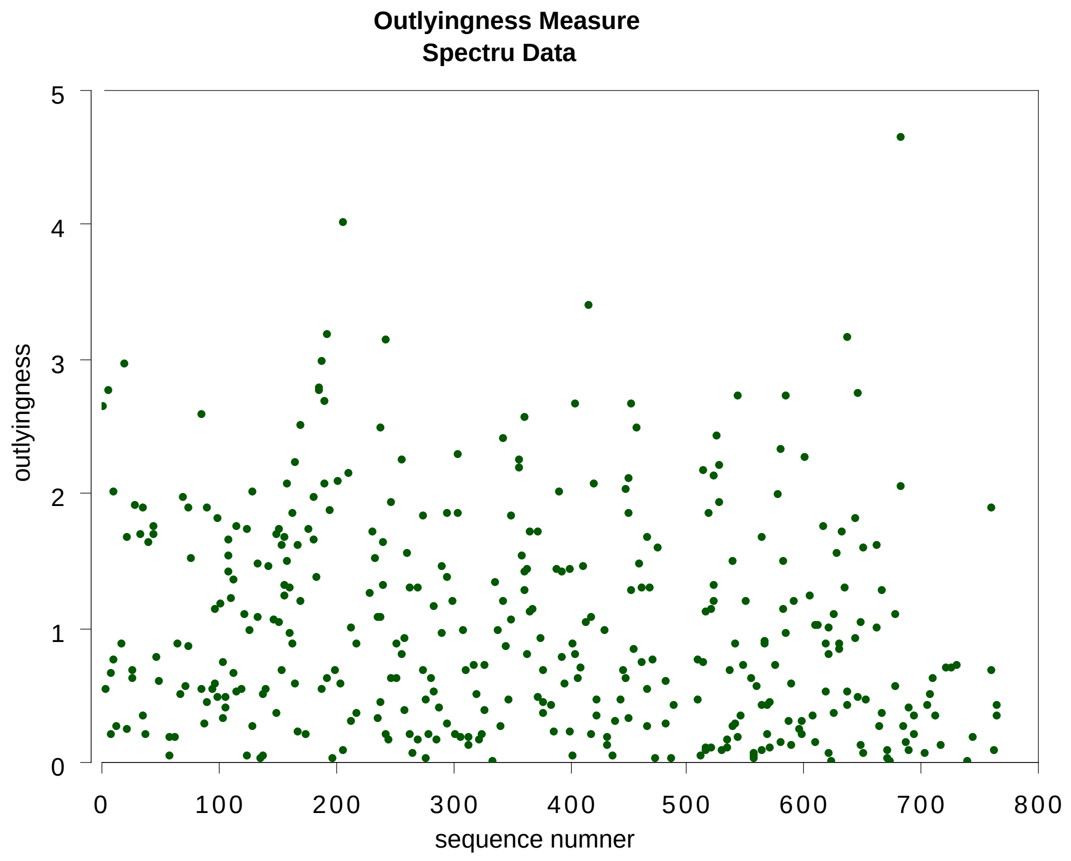
Data graciously supplied by Merck consists of the first 468 spectral intensities in the spectrums of 764 compounds.

The challenge presented by Merck was to find small cohesive groups of outlying cases in this data.

There was excellent separation between the two classes.

An error rate of 0.5%, indicating strong dependencies in the original data.

We looked at outliers and generated this plot.

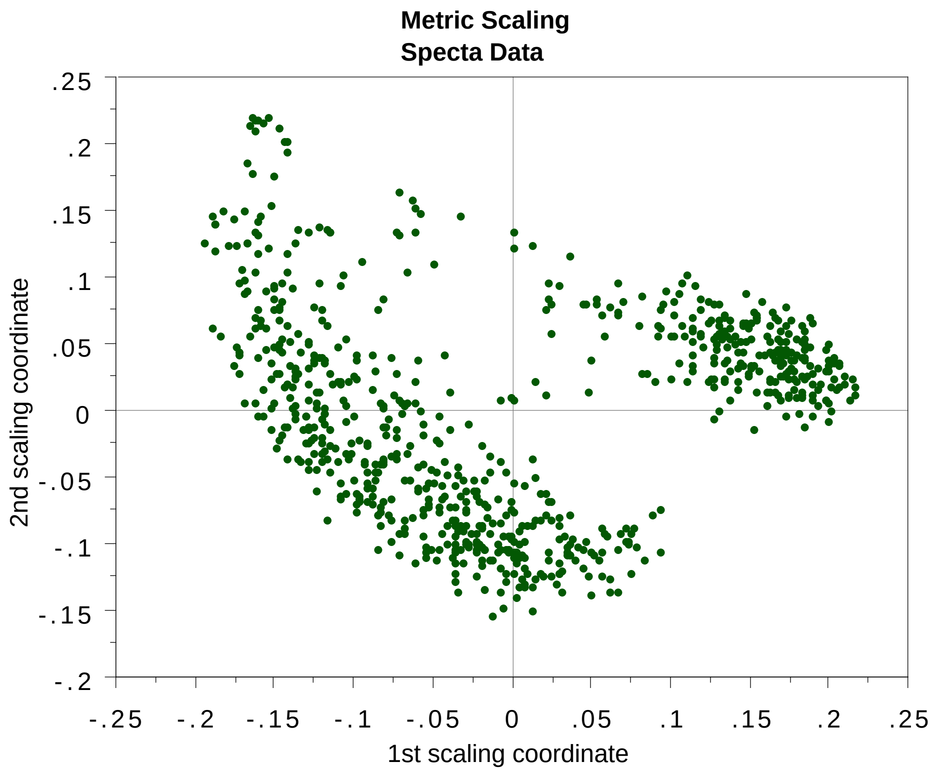


This plot gives no indication of outliers.

But outliers must be fairly isolated to show up in the outlier display.

Projecting Down

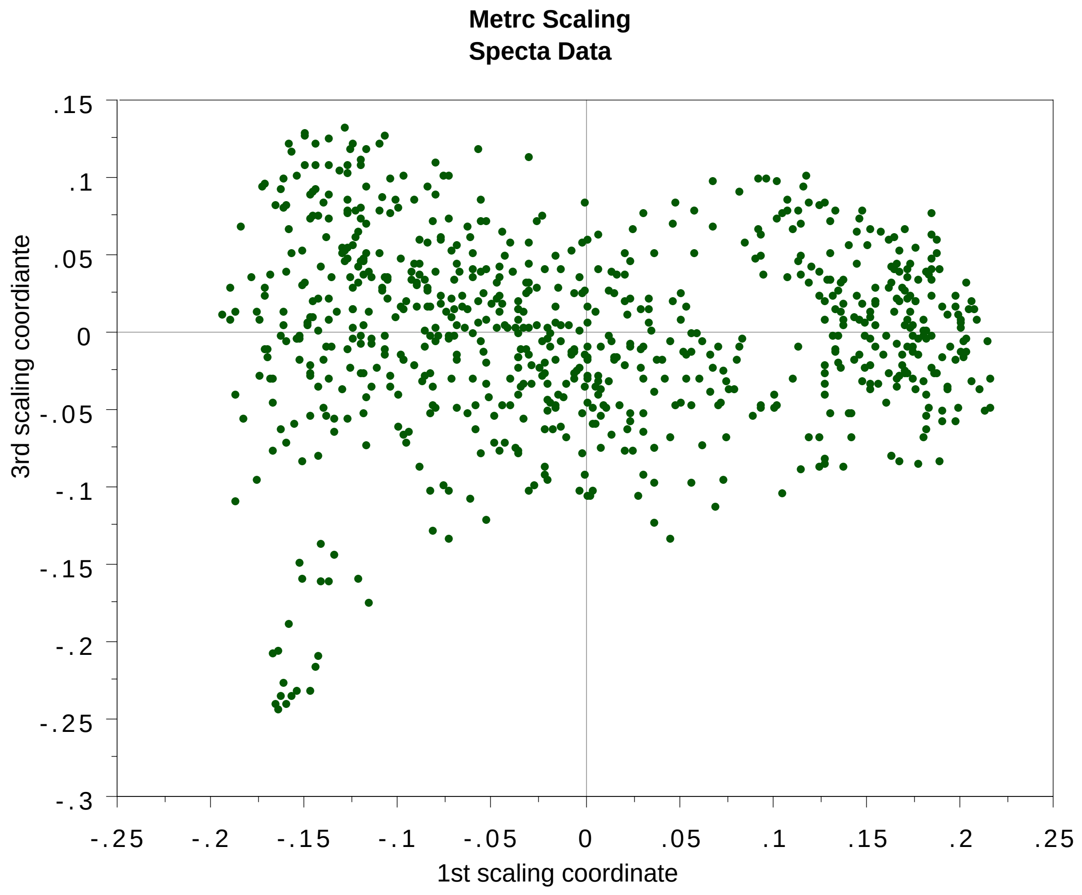
To search for outlying groups scaling coordinates were computed. The plot of the 2nd vs. the 1st is below:



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.



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.

RF/tools--An Ongoing Project

RF/tools are continually being upgraded.

My coworker Adele Cutler, is writing some very illuminating graphic displays for the information outputted by RF.

The next version of RF/cl will have the ability to detect interactions between variables--a capability that the microarray people lobbied for.

Plus other improvements, both major and minor.

RF/rg needs upgrading to the same level as RF/cl.

A new program RF/my that can use two eyes on situations with multiple dependent outputs will be added next month.

CONCLUSION

There is no data miniing program available that can provide the predictive accuracy and understanding of the data that RF/tools does.

and its free!!