[23] Random Forests for Microarrays

By ADELE CUTLER and JOHN R. STEVENS

Abstract

Random Forests is a powerful multipurpose tool for predicting and understanding data. If gene expression data come from known groups or classes (e.g., tumor patients and controls), Random Forests can rank the genes in terms of their usefulness in separating the groups. When the groups are unknown, Random Forests uses an intrinsic measure of the similarity of the genes to extract useful multivariate structure, including clusters. This chapter summarizes the Random Forests methodology and illustrates its use on freely available data sets.

Introduction

Microarrays present new challenges for statistical methods because of the large numbers of genes and relatively small numbers of microarrays. Random Forests (Breiman, 2001; Breiman and Cutler, 2005) provide a general-purpose tool for predicting and understanding data. They are becoming popular for analyzing microarray data (see, e.g., Díaz-Uriarte and Alvarez de Andrés, 2006) in part because they can handle large numbers of genes without formal variable selection, they are robust to outliers, do not require data to follow the normal (or any other) distribution, can be used for badly unbalanced data sets, and can impute missing values intelligently. This chapter refers to gene expression microarrays, although the ideas transfer directly to tissue arrays or even mass spectrometry data. We assume that all gene expression data have been normalized appropriately using a preprocessing method such as RMA (Irizarry *et al.*, 2003). Because the Random Forest methods discussed are invariant under monotone transformations, data do not need to be log transformed, although it may be advisable for numerical reasons.

A Random Forest is a collection of classification trees generated by bootstrap sampling from data and randomly sampling predictor variables at each node. This chapter describes trees and forests in more detail and intuitively shows how the trees in a Random Forest combine to give more accurate results.

Random Forests do not perform formal statistical inference and do not do significance tests or give p values. They are not intended for small,

METHODS IN ENZYMOLOGY, VOL. 411

0076-6879/06 $35.00
DOI: 10.1016/S0076-6879(06)11023-X

carefully controlled experiments. However, results of a Random Forests analysis might *suggest* interesting experiments and might give insight that could be missed by formal procedures.

Two quite general applications are considered. In the first situation, we have two or more labeled groups of microarrays (e.g., tumor versus control) and want to classify new microarrays or determine which genes would be useful in classifying new microarrays. We refer to this first situation as *classification* and note that it is a form of *supervised learning* (see Gollub and Sherlock, 2006). The second situation in which Random Forests may be useful is when we have unlabeled microarrays and want to find clusters or other interesting multivariate structure. This situation is referred to as *unsupervised learning*. Classical statistical clustering methods (Gollub and Sherlock, 2006) are popular in this situation and can be used in conjunction with a Random Forests analysis.

This chapter focuses on classification, although the unsupervised learning approach is also discussed.

Classification

Classification deals with data comprising a number of observations, each of which is known to come from one of a number of distinct groups or classes. For each observation, we have a number of predictors. Our goal is to use the predictors to classify unlabeled observations and to learn which predictors are important or useful in the classification.

In the microarray context, the observations usually represent the microarrays themselves (or the observational units, such as patients, from which they are obtained) and the predictors represent the genes. For example, we may have microarrays for cancer patients and controls and we may want to classify a new person into one of these two groups based on their microarray results. Perhaps more importantly, we may also want to determine which genes on the microarray are useful in classifying the new person, with the idea of developing a more efficient diagnostic tool or giving useful information about the genetic basis of the disease itself.

One common approach to data like these is to treat the genes individually and perform something similar to a t test to decide which genes are "significantly different" between the two groups, presumably with an adjustment for the number of comparisons. Methods such as significance analysis of microarrays (Tusher *et al.*, 2001) have this flavor. If there are more than two groups, an ANOVA approach might be used (Ayroles and Gibson, 2006). We refer to these procedures as "significance testing."

Classification differs from the significance testing approach in several important ways. Perhaps the most fundamental difference is that *t* tests and ANOVAs test whether the population means for the groups are different. For overlapping groups, we may conclude that the population means differ, but the groups may not be distinct enough to make accurate predictions of group membership. A second difference is that classification is inherently multivariate, so instead of asking whether each gene is *individually* good at separating the groups, we are asking whether the gene expression information from *all* the genes is useful. The collective expression levels of groups of genes may capture higher order terms such as interactions between genes, which may allow us to separate the groups better than any single gene.

Traditional statistical methods for classification include linear discriminant analysis and logistic regression (see, e.g., Hastie *et al.*, 2001). For microarray data, these methods are not directly applicable because the number of genes is too large. One approach is to do some sort of gene filtering. For example, the significance testing approach may be used to determine a small set of genes that can then be used in a classification. However, as well as the distributional assumptions, this form of gene filtering ignores the multivariate structure of data and it is not clear whether valuable information may be lost. Another common approach is to use principal components analysis to reduce the dimensionality of data. One problem with this approach is that principal components analysis concentrates on finding combinations of genes with large variance and ignores the class labels. Genes with large variance dominate, and outliers can have a huge impact.

Random Forests can handle gene expression data sets without gene filtering and without assuming normality.

Random Forests for Classification

A Random Forest, as the name suggests, is made up of a collection of classification trees. This section briefly describes classification trees and the particular type of ''random'' classification tree used in the Random Forests method. It also explains how the trees are combined and how measures of variable importance and other useful quantities are obtained.

Classification Trees

Classification trees (Breiman *et al.*, 1984) are a binary decision. An example of a classification tree is given in Fig. 1. This tree was fit to data

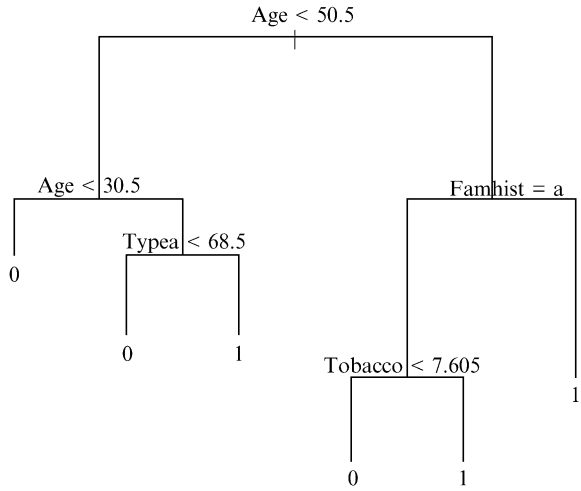

FIG. 1. Tree diagram for heart disease data.

from a South African study on coronary heart disease in men (Hastie *et al.*, 2001; Rousseauw *et al.*, 1983). The response variable was the absence of myocardial infarction (group = 0) or the presence of myocardial infarction (group = 1). Predictor variables were age, prevalence of "type a" behavior, tobacco use, and family history ("a" stands for "absent"). The tree comprises a collection of "nodes." At each node, we ask a question and if the answer is "yes" we move to the left, otherwise we move to the right. At the top node, we ask whether the man's age is less than 50.5. If it is, the man goes to the left; if not, he goes to the right. A similar procedure is followed for subsequent nodes until a stopping criterion is met, at which point the node is "terminal." Numbers at the bottom of the tree represent the class assigned to men who end up in the terminal nodes.

Each node involves a single predictor variable, and we say we "split" on that variable. To construct the tree, we need to split each node, which means we need to decide which variable to split on and at what value to split. To split a node, we look at every possible split on every available predictor. We choose the split that gives the best value of some criterion such as the gini index (Breiman *et al.*, 1984). Usually, the trees are grown to be quite large and are then "pruned" back to prevent overfitting (Breiman *et al.*, 1984). Classification trees are popular for a wide range of problems, in part because the tree diagrams are easily understood. More information on classification trees is given in Breiman *et al.* (1984). For microarray data, we often have thousands of genes, and finding the best possible split at each node can be

computationally expensive. Moreover, it has been suggested (Breiman, 2001; Dietterich, 2000) that we can get more accurate results by combining a variety of suitably chosen classification trees.

Trees in a Random Forest

A Random Forest combines a collection of classification trees that differ from each other in two key ways. First, each tree is fit to an independent bootstrap sample from the original data set. To get the bootstrap sample, we randomly sample microarrays *with replacement* from original data until our sample is as large as the original. Some microarrays appear once in the bootstrap sample, some twice, some more often, and some not at all. The microarrays that do not make it into the bootstrap sample are called ''out-of-bag'' data and form a natural test set for the tree that is fit to the bootstrap sample. The trees also differ because we do not choose the best possible split on all genes. Instead, we take a random sample of just a few genes, independently for each node, and find the best split on the selected genes. The number of genes to be selected at each node is usually chosen to be the square root of the total number of genes. More information about how to choose this value is available in Breiman and Cutler (2005) by looking at the parameter ''mtry.'' The trees are grown until each node contains microarrays from only one class (we say they are ''pure''), and the trees are not pruned.

Combining Trees

If we have a new microarray that has been suitably preprocessed to be on the same scale as original data, we pass it down each tree in the forest and each tree provides its best guess at the class. The most popular class, over all the trees in the forest, is the one we use as the Random Forest prediction. This procedure is called ''plurality voting.'' The votes themselves give an idea about which other classes are contenders. For example, suppose the Random Forest has 1000 trees of which 547 say ''class 1,'' 398 say ''class 2,'' and the rest say ''class 3.'' Then the Random Forest prediction is ''class 1,'' with ''class 2'' a possible contender and ''class 3'' out of the running. For a microarray that is part of the original data set, the procedure is modified by only voting the trees for which this particular microarray is out of bag.

Error Rate Estimates

Out-of-bag data are used to give an internal estimate of what the misclassification rate will be if the Random Forest is used to predict the classes for a new data set from the same population as the original (Breiman and

Cutler, 2005). To get this out-of-bag error rate, we use each tree to classify the corresponding out-of-bag data (those that did not make it into the bootstrap sample used to get that particular tree). For each tree, we compute error rates for each class and an overall error rate, and we average over all the trees in the forest.

Gene Importance

One interesting aspect of gene expression data is that genes with large expression values or those with highly variable expression values are not always the genes that are important for distinguishing the classes. Random Forests uses an unusual but intuitive measure of the importance of each gene in distinguishing the classes. Consider a single tree and think about the microarrays that are out of bag for this tree. When we pass the out-of-bag microarrays down the tree, we get the out-of-bag error rate for the tree. Now think about randomly permuting the expression values of a particular gene so that each out-of-bag microarray gets a random expression value for this particular gene and all the other genes are kept at their original values. Now we pass the *modified* out-of-bag data down the tree and compute its error rate. If the new error rate is about the same as before, the gene does not appear to be contributing to accurate classification. If, however, the new error rate is higher than before, the gene expression values were useful for accurate classification. The gene importance measure is obtained by averaging the increase in the error rate over all the trees in the forest and this average is used to rank the genes.

Unbalanced Data

Unbalanced data sets, where the class of interest is much smaller than the other classes, are becoming more frequent. A naive classifier will work on getting the large classes right while getting a high error rate on the small class. Random Forests has an effective way of weighting the classes to give balanced results in highly unbalanced data (Breiman and Cutler, 2005). One reason to do this is that the important genes may be different when we force the method to pay greater attention to the small class. Even in the balanced case, the weights can be adjusted to give lower error rates to decisions that have a high misclassification cost. For example, it is often more serious to conclude incorrectly that someone is healthy than it would be to conclude incorrectly that someone is sick.

Proximities

One of the difficult aspects of microarray data analysis is that with thousands of genes, it is not obvious how to get a good "feel" for data or a good impression of what is going on. Are there interesting patterns or

structures, such as subgroups within the known classes? Are there outliers? In a multiclass situation, are some of the groups separated while others overlap? Such questions are overwhelming if we try to examine them in obvious ways. Random Forests provides a way to look at data to give some insight into these questions and to show fascinating and unsuspected aspects of data. We do it by computing a measure of *proximity* or *similarity* between each pair of microarrays. We define the proximity between two microarrays as the proportion of the time that they end up in the same terminal node, where the proportion is taken over the trees in the forest. If two microarrays are always in the same terminal node, their proximity will be 1. If they are never in the same terminal node, their proximity will be 0. From these proximities, we derive a distance matrix and use a technique called ''classical multidimensional scaling'' (see, e.g., Cox and Cox, 2001). Multidimensional scaling takes any set of distances between microarrays and creates a set of points that can be plotted in two or three dimensions. Each point represents one of the microarrays and the distances between the points represent, as closely as possible, the distances between the corresponding microarrays. The resulting picture allows us to ''look'' at data in a new way.

A natural question at this point is whether it would be just as good to use multidimensional scaling on a conventional distance, such as Euclidean distance or one of the other distances used commonly in cluster analysis (Gollub and Sherlock, 2006). This can certainly be done, but one of the difficulties is that a conventional distance can be dominated by noisy and uninformative genes that can drown out the effects of the genes that are useful. In any case, it may be useful to have an additional view that may illuminate different features of the data.

Proximities are also used to detect outliers and provide a very effective method for filling in missing data (Breiman and Cutler, 2005).

Unsupervised Learning and Clustering

This section describes how Random Forests can be used for unsupervised learning. The presentation is much more brief because unsupervised learning is much more exploratory than classification and not as well understood.

Gollub and Sherlock (2006) describe standard statistical methods for cluster analysis in the microarray context. In the microarray context, we might cluster either the microarrays or the genes. *Clustering the microarrays* involves separating the microarrays into groups or ''clusters'' so that microarrays in the same cluster have similar gene expression patterns, whereas those from different clusters have quite different expression patterns. For

example, we might cluster the microarrays in a medical example where we think there may be distinct types of people in the population. *Clustering the genes* involves finding groups of genes that have similar expression patterns across the microarrays, which in this context might represent different experimental conditions. In this case the goal might be to organize data to facilitate understanding or to identify coregulated genes.

Unsupervised learning is sometimes equated with clustering, but it can be viewed in the more general light of discovering multivariate structure. Cluster structure is one form of multivariate structure, but not the only one. One of the basic assumptions of all clustering methods is that there really *are* clusters. If there are not clusters, cluster analysis might not make sense but it might still make sense to ask whether there is an important multivariate structure.

Random Forests can be used for unsupervised learning without assuming a cluster structure (Breiman and Cutler, 2005). For simplicity, we describe the procedure for exploring structure in the genes and note that structure in the microarrays can be explored in an analogous way. To use Random Forests for unsupervised learning, we label real data ''class 1'' and generate synthetic data, which are labeled ''class 2.'' Then we use Random Forests to see if we can separate the two classes. Synthetic data are generated from real data by randomly permuting the expression values for each gene independently. In this way, we form a new data set that maintains the distributions of the individual expression values while destroying their multivariate structure. If the original expression values have no multivariate structure, synthetic data will look similar to original data and Random Forests will misclassify about half of the time. If, however, the misclassification rate is much lower than 50%, there is evidence of some interesting structure and we can use all the Random Forests tools (variable importance, proximities, and multidimensional scaling plots) to investigate the structure. In fact, the proximities can be used with a sensible clustering method to determine clusters if a cluster structure turns out to be present.

Case Study: Prostate Cancer Data Set

We illustrate Random Forests using a data set on prostate cancer (Singh *et al*., 2002). These data have 6033 gene expression values for 102 arrays (50 normal samples and 52 tumor samples). We used the normalization described by Dettling (2004). Random Forests was run with 500 trees and 100 randomly chosen genes at each node. Code is available upon request. The out-of-bag error rate was 7%, which is consistent with Dettling (2004), who cited a 9% cross-validation error rate. The four most important genes

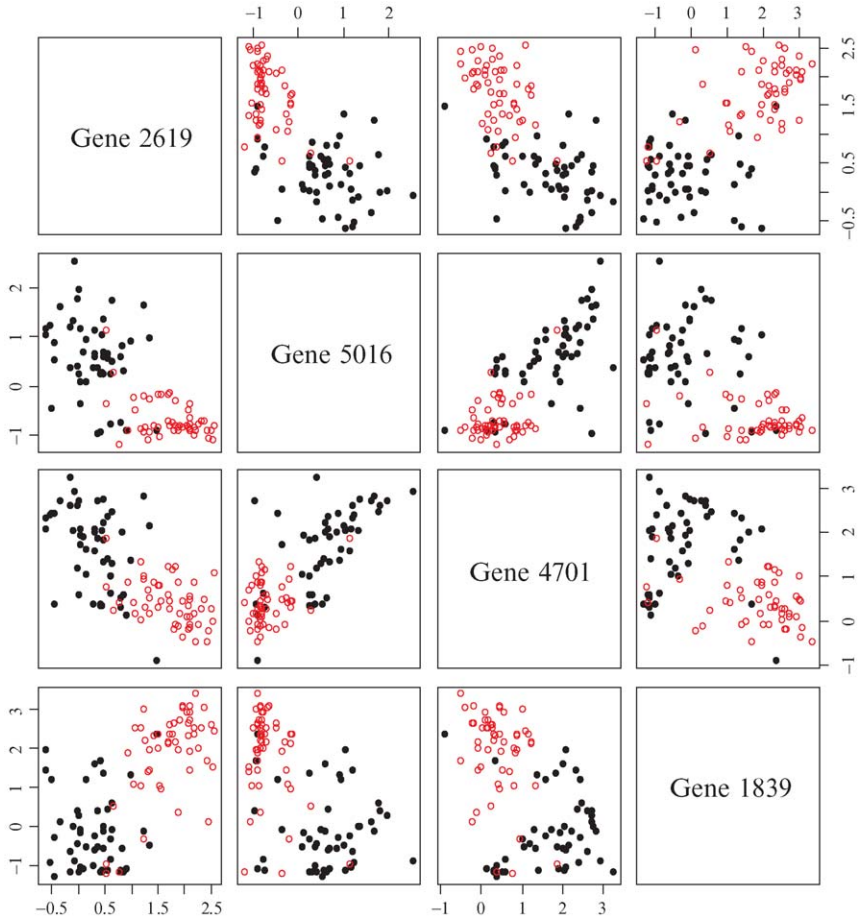

FIG. 2. Prostate cancer data: the four most important genes selected by Random Forests and their relationship to the groups. Solid black circles represent controls and open red circles represent tumors.

identified by Random Forests are plotted in Fig. 2. Solid black circles represent controls and open red circles represent tumors. It is clear, from both the error rates and the pictures, that Random Forests is able to classify data very well and also to identify genes useful in this process. We compare to performing a principal components analysis on the same gene expression data. The first two principal components are shown in Fig. 3. It is apparent that the dimensions of greatest variability in these data have very little to do with the two groups.

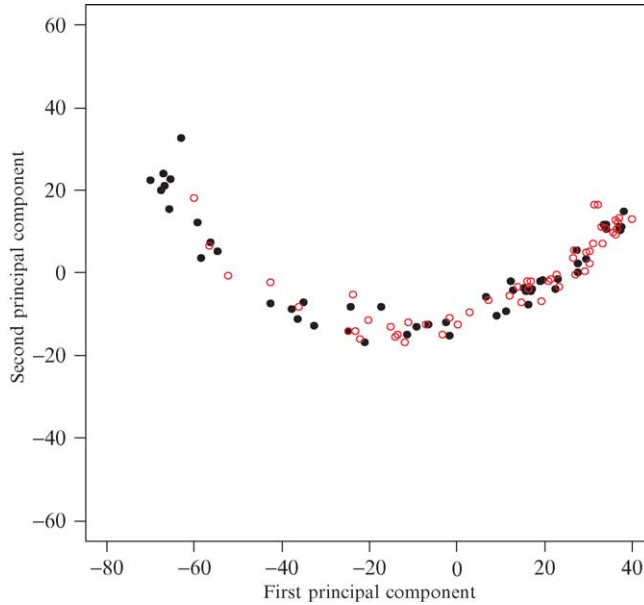

FIG. 3. Prostate cancer data: the first two principal components of data and their relationship to the groups. Solid black circles represent controls and open red circles represent tumors.

Conclusion

Random Forests provides a new and powerful approach to understanding gene expression data. According to Díaz-Uriarte and Alvarez de Andrés (2006): ''Because of its performance and features, random forest and gene selection using random forest should probably become part of the ''standard tool-box'' of methods for class prediction and gene selection with microarray data.''

Open source FORTRAN software for Random Forests is available from www.math.usu.edu/∼adele/forests. A commercial version, with an easy-to-use interface, is available from www.salford-systems.com. An R package is also available, written by Liaw and Wiener (2001).

References

Ayroles, J. F., and Gibson, G. (2006). Analysis of variance of microarray data. *Methods Enzymol.* **411,** 214–233.

Breiman, L. (2001). Random forests. *Machine Learn.* **45**(1), 5–32.

Breiman, L., and Cutler, A. (2005). www.math.usu.edu/~adele/forests.
Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). ''Classification and Regression Trees.'' Chapman and Hall, New York.
Cox, T. F., and Cox, M. A. A. (2001). ''Multidimensional Scaling,'' 2nd Ed. Chapman and Hall/CRC.
Dettling, M. (2004). BagBoosting for tumor classification with gene expression data. *Bioinformatics* **20**(18), 3583–3593.
Díaz-Uriarte, R., and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinform.* **7,** 3.
Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learn.* **40,** 139–157.
Gollub, J., and Sherlock, G. (2006). Clustering microarray data. *Methods Enzymol.* **411,** 194–213.
Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.'' Springer, New York.
Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4,** 249–264.
Liaw, A., and Wiener, M. (2002). Classification and regression by Random Forest. R News: The Newsletter of the R Project (http://cran.r-project.org/doc/Rnews/) **2**(3), 18–22.
Rousseauw, J., du Plessis, J., Benade, A., Jordann, P., Kotze, J., Jooste, P., and Ferreira, J. (1983). Coronary risk factor screening in three rural communities. *South Afr. Med. J.* **64,** 430–436.
Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., and Sellers, W. R. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1**(2), 203–209.
Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA* **98,** 5116–5121.

Further Reading

Dudoit, S., Fridlyand, J., and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* **97**(457), 77–87.
R Development Core Team (2004). R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna, Austria. ISBN 3-900051-07-0, URL http://www.r-project.org/.