The Annals of Applied Statistics
2010, Vol. 4, No. 1, 266–298
DOI: 10.1214/09-AOAS285

BART: BAYESIAN ADDITIVE REGRESSION TREES

BY HUGH A. CHIPMAN, EDWARD I. GEORGE AND ROBERT E. MCCULLOCH

5.3. *Classification*: *A drug discovery application*. Our last example illustrates an application of the BART probit approach of Section 4 to a drug discovery classification problem. In such problems, the goal is to predict the "activity" of a compound using predictor variables that characterize the molecular structure of the compound. By "activity," one typically means the ability to effect a desired outcome against some biological target, such as inhibiting or killing a certain virus.

The data we consider describe $p = 266$ molecular characteristics of $n = 29{,}374$ compounds, of which 542 were classified as active. These predictors represent topological aspects of molecular structure. This data set was collected by the National Cancer Institute, and is described in Feng et al. (2003). Designating the activity of a compound by a binary variable ($Y = 1$ if active and $Y = 0$ otherwise), BART probit can be applied here to obtain posterior mean estimates of $P[Y = 1|x]$ for each x vector of the 266 molecular predictor values.

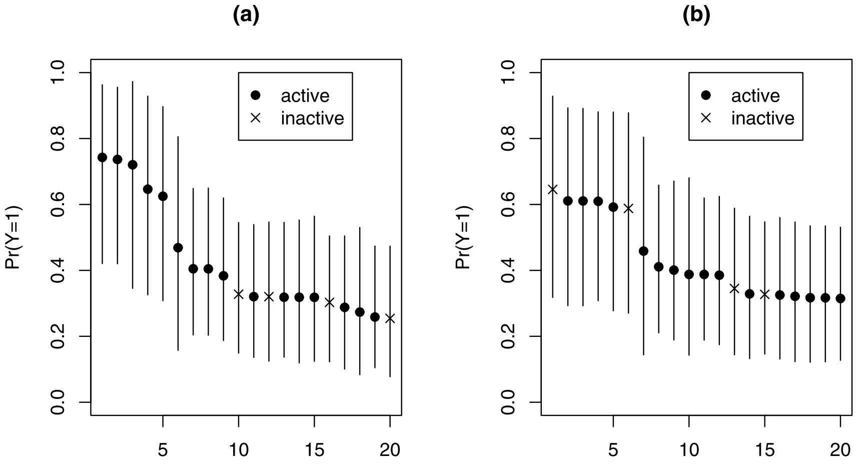

FIG. 9. *BART posterior intervals for the* 20 *compounds with highest predicted activity, using train* (a) *and test* (b) *sets.*

To get a feel for the extent to which BART's $P[Y = 1|x]$ estimates can be used to identify promising drugs, we randomly split the data into nonoverlapping train and test sets, each with 14,687 compounds of which 271 were active. We then applied BART probit to the training set with the default settings $m = 50$ trees and mean shrinkage $k = 2$ (recall ν and q have no meaning for the probit model). To gauge MCMC convergence, we performed four independent repetitions of 250,000 MCMC iterations and obtained essentially the same results each time.

Figure 9 plots the 20 largest $P[Y = 1|x]$ estimates for the train and the test sets. Also provided are the 90% posterior intervals which convey uncertainty and the identification whether the drug was in fact active ($y = 1$) or not ($y = 0$). The true positive rates in both the train and test sets for these 20 largest estimates are $16/20 = 80\%$ (there are 4 inactives in each plot), an impressive gain over the $271/14{,}687 = 1.85\%$ base rate. It may be of interest to note that the test set intervals are slightly wider, with an average width of 0.50 compared to 0.47 for the training intervals.

To gauge the predictive performance of BART probit on this data, we compared its out-of sample performance with boosted trees, neural networks and random forests (using `gbm`, `nnet` and `randomforest`, as in Section 5.1) and with support vector machines [using `svm` in the `e1071` package of Dimitriadou et al. (2008)]. L1-penalized logistic regression was excluded due to numeric difficulties. For this purpose, we randomly split the data into training and test sets, each containing 271 randomly selected active compounds. The remaining inactive compounds were then randomly allocated to create a training set of 1000 compounds and a test set of 28,374 observations. The training set was deliberately chosen smaller to make feasible a comparative experiment with 20 replications.

For this experiment we considered both BART-default and BART-cv based on 10,000 MCMC iterations. For BART-default, we used the same default settings as above, namely, $m = 200$ trees and $k = 2$. For BART-cv, we used 5-fold cross-validation to choose from among $k = 0.25$, 0.5, 1, 2, 3 and $m = 100$, 200, 400 or 800. For all the competitors, we also used 5-fold cross-validation to select tuning parameters as in Section 5.1. However, the large number of predictors led to some different ranges of tuning parameters. Neural networks utilized a skip layer and 0, 1 or 2 hidden units, with possible decay values of 0.0001, 0.1, 0.5, 1, 2, 5, 10, 20 and 50. Even with 2 hidden units, the neural network model has over 800 weights. In random forests, we considered 2% variable sampling in addition to 10%, 25%, 50% and 100%. For support vector machines, two parameters, C, the cost of a constraint violation, and γ [Chang and Lin (2001)], were chosen by cross-validation, with possible values $C = 2^a, a = -6, -5, \dots, 0$ and $\gamma = 2^b, b = -7, -6, -5, -4$.

In each of 20 replicates, a different train/test split was generated. Test set performance for this classification problem was measured by area under the Receiver Operating Characteristic (ROC) curve, via the ROCR package of Sing et al. (2007). To generate a ROC curve, each method must produce a rank ordering of cases by predicted activity. All models considered generate a predicted probability of activity, though other rank orderings could be used. Larger AUC values indicate superior performance, with an AUC of 0.50 corresponding to the expected performance of a method that randomly orders observations by their predictions. A classifier's AUC value is the probability that it will rank a randomly chosen $y = 1$ example higher than a randomly chosen $y = 0$.

The area under curve (AUC) values in Table 5 indicate that for this data set, BART is very competitive with all the methods. Here random forests provides the best performance, followed closely by boosting, BART-cv and then support vector machines. The default version of BART and neural networks score slightly lower. Although the differences in AUC between these three groups are statistically significant (based on a 1-way ANOVA with a block effect for each replicate), the

TABLE 5

Classifier performance for the drug discovery problem, measured as AUC, the area under a ROC curve. Results are averages over 20 *replicates. The corresponding standard error is* 0.0040, *based on an ANOVA of AUC scores with a block effect for replicates*

Method	**AUC**
Random forests	0.7680
Boosting	0.7543
BART-cv	0.7483
Support vector	0.7417
BART	0.7245
Neural network	0.7205

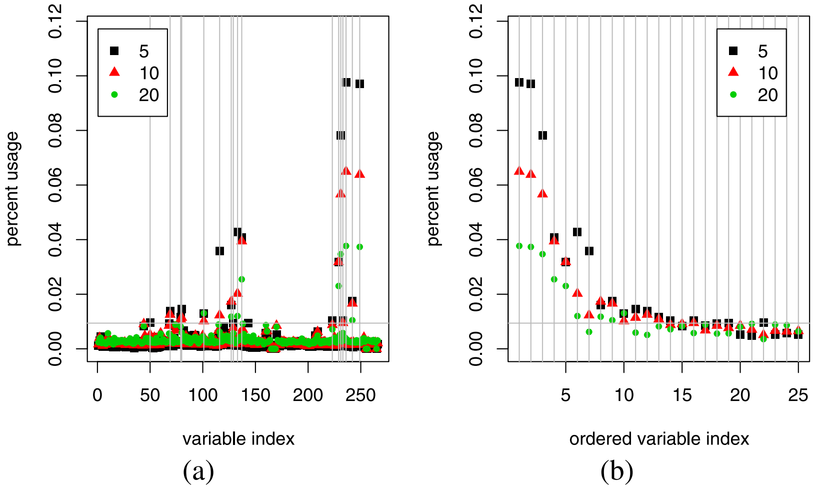

FIG. 10. *Variable importance measure, drug discovery example. Values are given for* 5, 10 *and* 20 *trees in the ensemble, for all* 266 *variables* (a) *and the* 25 *variables with the highest mean usage* (b). *Vertical lines in* (a) *indicate variables whose percent usage exceeds the* 95*th percentile. The* 95*th percentile is indicated by a horizontal line.*

practical differences are not appreciable. We remark again that by avoiding the cross-validated selection of tuning parameters, BART-default is much faster and easier to implement than the other methods here.

Finally, we turn to the issue of variable selection and demonstrate that by decreasing the number of trees m, BART probit can be used, just as BART in Section 5.2.1, to identify those predictors which have the most influence on the response. For this purpose, we modify the data setup as follows: instead of holding out a test set, all 542 active compounds and a subsample of 542 inactives were used to build a model. Four independent chains, each with 1,000,000 iterations, were used. The large number of iterations was used to ensure stability in the "percent usage" variable selection index (20). BART probit with $k = 2$ and with $m = 5, 10, 20$ trees were considered.

As Figure 10 shows, the same three variables are selected as most important for all three choices of m. Considering that $1/266 \approx 0.004$, percent usages of 0.050 to 0.100 are quite a bit larger than one would expect if all variables were equally important. As expected, variable usage is most concentrated in the case of a small ensemble (i.e., $m = 5$ trees).