Stats 315B

Homework 2

Due March 4, 2009

For this homework, as in Homework 1, you will be using the R software for Windows or Linux. Most of this homework involves becoming familiar with MART. MART is covered in Sections 10.6 – 10.14.2 of the text. Further information can be found in the papers: Greedy function approximation: a gradient boosting machine and Stochastic gradient boosting. Both of these papers are available at

http://www-stat.stanford.edu/~hastie/#reports

The first step is to download the software from

http://www-stat.stanford.edu/~hastie/R_MART.html

Please make sure that you execute EXACTLY the installation instructions found at the above URL and the readme.txt file that will be generated in the installation process in the specified order.

The second step is to go through the MART tutorial from

http://www-stat.stanford.edu/~hastie/R_MART.html#tutorial

and become familiar with MART. The data sets xdata, ydata, classlab used in the tutorial can be downloaded from the same URL. If you use read.table instead of scan, please read the HINTS provided at the end of the homework. You do not need to turn in any of the output form the tutorial. We strongly advise you to complete the tutorial. It is intended to get you ready for the work you have to perform for this homework.

Note when you follow the tutorial that the R commands use out-of-date syntax. You should replace underscores (“_”) with equals (“=”) as assignment operators throughout.

The data sets spam.data, income.data, california.data and occupation.data along with documentation files spam.info, spam.names, income.info, california.info can be found in the class web page in the Data section. Note that the results from the computational problems may not look exactly like the ones in the textbook due to plotting options and slight possible differences in the data sets used.

Adaboost (Freund and Shapire 1996) was the first successful boosting algorithm. Read sections 10.1 – 10.5 of the text for problems 1 and 2.

1. Text: Problem 10.1
2. Text: problem 10.2
3. Text: problem 10.3.

4. Regression: California Housing. The data set california.data consists of aggregated data from 20,640 California census blocks (from the 1990 census). The goal is to predict the
median house value in each neighborhood from the other attributes described in california.info.
Fit a MART model to the data and write a short report that should include at least
(a) The prediction accuracy of MART on the data set.
(b) Identification of the most important variables.
(c) Comments on the dependence of the response on the most important variables (you may
want to consider singleplots, pairplots, etc.).

5. Regression: Marketing data. The data set income.data was already used in Homework 1. Review income.info for the information about order of attributes etc. Fit a MART
model for predicting income form the other demographic attributes and compare the accuracy
with the accuracy of your best tree from Homework 1.

6. Multiclass classification: marketing data. The data set occupation.data comes
from the same marketing database used in Homework 1. The description of the attributes can
be found in occupation.info. The goal in this problem is to fit a MART model to predict the
type of occupation from the 13 other demographic variables.
(a) Report the test set misclassification error for MART on the data set, and also the
misclassification error for each class.
(b) Identify the most important variables.
(c) Comment on the dependence of the response on the most important variables (use
singleplots, pairplots, etc.).
(d) Report which variables are the most important in discriminating each class from all the
others as well as the classes to which they are misclassified.

7. Binary classification: Spam Email. The data set for this problem is spam.data, with
documentation files spam.info and spam.names. The data set is a collection of 4601 emails
of which 1813 were considered “spam”, i.e. unsolicited commercial email. The data set consists of
58 attributes of which 57 are continuous predictors and one is a class label that indicates
whether the email was considered spam (1) or not (0). Among the 57 predictor attributes are:
percentage of the word “free” in the email, percentage of exclamation marks in the email, etc.
See file spam.names for the full list of attributes. The goal is, of course, to predict whether or
not an email is “spam”.
(a) Based on these data, fit a MART model for predicting whether or not an email is “spam”.
What is your estimate of the misclassification rate? Of all the spam emails of the test set what
percentage was misclassified, and of all the non-spam emails in the test set what percentage was
misclassified?
(b) Your classifier in part (a) can be used as a spam filter. One of the possible disadvantages
of such a spam filter is that it might filter out too many good (non-spam) emails. Therefore, a
better spam filter might be the one which penalizes misclassifying non-spam emails more heavily
than the spam ones. Suppose that you want to build a spam filter which “throws out” no more
that 0.3% of the good (non-spam) emails. You have to find an “optimal” cost matrix that
penalizes misclassifying “good” emails as “spam” more than misclassifying “spam” emails as
“good” by the method of trial and error. Once you have constructed your final spam filter with
the property described above, answer the following questions:
(i) What is the overall misclassification error of your final filter and what is the percentage
of good emails and spam emails that were misclassified respectively?
(ii) What are the important variables in discriminating good emails from spam for your
spam filter?
(iii) Using the interpreting tools provided by MART, describe the dependence of the
response on the most important attributes.
Hints:

1. In order to read help files for MART functions, type `marthelp()` at the R prompt. For example, typing

   ```r
   > marthelp('varimp')
   ```

gives information about the function `varimp` which displays the relative importance of the predictors used in building a MART model. To read about all the MART functions type

   ```r
   > marthelp()
   ```

2. Since some of the datasets contain strings (e.g. NA) we advise that you use the function `read.table` as you did in Homework 1 instead of `scan`. Do NOT change categorical variables with `as.factor` as you did for RPART. The type of each variable (e.g. continuous, categorical, etc.) is specified in the vector `lx` as explained in the tutorial. The predictors should be stored in a matrix and NOT in a data.frame (as returned by `read.table`), otherwise MART will give you an error. Below is an example for `california.data`:

   ```r
   > california = read.table('/usr/class/stats315b/WWW/DATA/california.data',sep=',')
   > names(california) = c('MedVal','MedIncome','HouseAge','AveRooms','AveBdrms','Population','AveOccup','Latitude','Longitude')
   > y=california[,1]
   > x=california[,2:9]
   > lx = ... you set this...
   > mart(x,y,lx)
   ```

3. If a dataset contains missing values MART will issue a warning, and then replace the NA’s with an internal code.

4. Suppose that you need to solve a binary classification problem. The R command:

   ```r
   > mart(x,y,lx,martmode='class')
   ```

fits a MART model with equal costs for misclassifying one class into the other. An example with unequal costs is shown below:

   ```r
   > cost = c(0,1,10,0)
   > cost =matrix(cost,nrow=2,byrow=T)
   > mart(x,y,lx,martmode='class',cost mtx=cost)
   ```

The example with equal costs would use `cost = c(0,1,1,0)` in the first line above.