Predictive Modeling

Assignment #3

February 24, 2014

- Homeworks are due by 11am on Monday, March 17th, 2014.**NOTE the unusual deadline**.
- Your homework should be written using any word-processor and submitted through Canvas.
- Note: If any question requires writing R code, the code should be included in your report.

1. **(8+2=10 pts) SGD** - In this problem you will use SGD to estimate the parameters of an MLR problem. The dataset is derived from `https://archive.ics.uci.edu/ml/datasets/Forest+Fires`. However, the data is already partitioned into `forestfire-train.csv` and `forestfire-test.csv`. It contains various features, where the last column (`area`) is the target variable. You should load the data, standardize the training data (make each column including the target variable mean 0 and variance 1). This can be done as (for each column j :

   ```
   train[, j]= (train[, j] - mean(train[, j])/sd(train[,j])
   test[, j]= (test[, j] - mean(train[, j])/sd(train[,j])
   ```

   Note that using the above transformation, you make each column of training data 0 mean and 1 variance, and then apply the same transformation on the test data (test data will NOT be 0 mean, 1 variance). Consider the following different learning rates 0.00025, 0.00575, and 0.0065 for this problem and initialize all coefficients to 0.5.

   (a) For each learning rate, as you make multiple passes (epochs) over the training data, record the Root Mean Squared Error (RMSE) obtained for both training and test data, and plot the number of epochs vs RMSE. What is a reasonable number of epochs (one answer per learning rate) after which you stop training?

   (b) How does your final model compare in terms of MSE on the test data to a batch solution (i.e. using MLR directly on the entire data)? You can use `lm`.

2. **(5 pts)** (See Exercise 4.1 of Bishop.) Prove that two sets of points in R^n are linearly separable in this n-dimensional space if and only if their convex hulls do not intersect.

3. **(4+2+4+4+6+2=22 pts) Regression Trees (RT) and Support Vector Regression (SVR)**. In this problem, you will use RT, MLR, and SVR to predict Published Relative Performance ("PRP"), an integer, given all the other variables as predictors in the dataset 'Machine1.csv'.

   (a) Build a RT using the `rpart` package. State how you attempted to avoid overfitting, and the quality (in terms of mean absolute error (MAE), mean squared error (MSE), and R-squared R^2) of your model using 5-fold cross validation (CV). Compare your results with those obtained by using MLR. Specify whether the metrics reported are empirical (i.e., obtained over the training data) or actually reflect the (estimated) true values, as measured, for example, across one or more holdout sets.
   (b) Plot the tree corresponding to your RT model.
   (c) Build a SVR with a linear kernel using the `kernlab` package and similar to part (a), report the quality in terms of MAE, MSE, and R^2 of your model using 5-fold CV.
   (d) Plot the predicted values (x-axis) against the target values (y-axis) from all three of your models, where each model is colored differently, on a single plot for all the data points in Machine1.csv. You

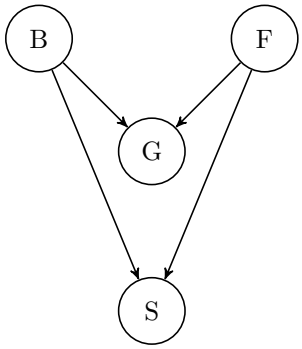
can do the single plot with multi-colors using `ggplot2`. Create a dataframe with 3 columns, Model, Target, and Prediction (e.g. "SVR", 1.5, 2) that contains the information for all 3 models. Then you can use the following R command:

```
ggplot(<dataframe>, aes(x = Pred, y = Target, color=Model, shape=Model)) + theme_bw() +
        geom_point() + geom_abline(intercept=0, slope=1, color="grey") +
        scale_colour_brewer(palette="Set1") + xlab("Predicted PRP") +
         ylab("Target PRP") + theme(legend.position = "top")
```

Comment on your model quality results from (a), (c), and how they relate to the plot (e.g. when is SVR better, when is MLR better, etc)
(e) Dataset 'Machine2.csv' is the same as 'Machine1.csv', except that a small number of the target values have been changed so that they act as outliers. Repeat parts (a), (c), and (d) using this "distorted" dataset (Machine2.csv). Now predict the performance on 'Machinetest.csv' for SVR, MLR and RT and report the performance on this held-out set w.r.t. MAE, MSE and R-squared as before. Also evaluate the same metrics for the models trained using 'Machine1.csv'. Can you see any difference in the effect of outliers on the performance of the three models (MLR, RT, SVR) vs when trained with outliers? What can you conclude about the regression models based on this?
(f) Produce an ensemble version of your solution using the `randomForest` package and compare the quality metrics obtained (R^2) with the values obtained in part (a). Set the value of ntree parameter to ten while building the model.

4. **(5+2 = 7 pts)**. (a) Suppose points in R^2 are being obtained from two classes, C1 and C2, both of which are normally distributed with means at (2,0) and (0,2) respectively. The covariance matrix for both classes is the (2x2) identity matrix. If the priors of C1 and C2 are 1/5 and 4/5 respectively, what is the ideal (i.e. Bayes Optimal) decision boundary (derive the equation for this boundary)?
(b) Suppose the cost of misclassifying an input actually belonging to C1 is twice as expensive as misclassifying an input belonging to C2. Correct classification does not incur any cost. If the objective is to minimize the expected cost rather than expected misclassification rate, what would be the best decision boundary? (obtain the equation describing this boundary).

5. **(1+1+2 = 4 pts)** In the Bayesian network shown below, B stands for "Battery", F for "Fuel", G for "Gauge", and S for "Start". Compute the following probabilities:
(a) P(B = good, F = empty, G = empty, S = yes).
(b) P(B = bad, F = empty, G = not empty, S = no).
(c) Given that the battery is bad, compute the probability that the car will start.



P(B = bad) = 0.1
P(F = empty) = 0.2

P(G = empty | B = good, F = not empty) = 0.1
P(G = empty | B = good, F = empty) = 0.8
P(G = empty | B = bad, F = not empty) = 0.2
P(G = empty | B = bad, F = empty) = 0.9

P(S = no | B = good, F = not empty) = 0.1
P(S = no | B = good, F = empty) = 0.8
P(S = no | B = bad, F = not empty) = 0.9
P(S = no | B = bad, F = empty) = 1.0

6. **(2+ 2+ 3 + 3 + 2 = 12 pts) Multi-Level Modeling**. In this problem, you will explore multilevel modeling using an R package, "nlme". A dataset ("oxboys.csv") is provided. This dataset contains three features from 26 students in Oxford: student id, (measured) height, and (measurement) year.

(a) Plot the relationship between height and year, and draw a linearly regressed line ignoring the id variable.

(b) Plot the relationship between height and year, but this time, fit a different linear regression for each individual. In ggplot, you can do it by specifying the group variable:

```
ggplot(oxboys, aes(x=year, y=height, group=id)) + stat_smooth(method="lm")
```

Other plotting packages also should have some such facility. Comment on differences between (a) and (b).

(c) Divide the dataset into training and test sets. The training set contains the first two years of the measurements, and the test set contains the rest of the measurements. In R, this can be done by:

```
oxboys.train <- subset(oxboys, year < 3)
oxboys.test <- subset(oxboys, year >= 3)
```

Build three different linear models:

- Global model: a linear model ignoring the id variable
- Local model: a different linear model for each individual i.e., 26 different linear regressions. Template R code is provided:

```
for( key in unique(oxboys.train$id)){
oxboys.train.ind <- subset(oxboys.train, id==key)
oxboys.test.ind <- subset(oxboys.test, id==key)
# build a linear model with oxboys.train.ind
# predict on oxboys.test.ind, and measure MSE
}
```

Note that this is just a template, not the answer.

- Multilevel model: Use the nlme package to fit a multilevel model specified as follows:

$$
\begin{aligned}
\text{height}_{it} &= \beta_{0i} + \beta_{1i}\text{year}_{it} + \epsilon_{it} \\
\beta_{0i} &= \beta_{00} + \eta_{0i} \\
\beta_{1i} &= \beta_{10} + \eta_{1i} \\
\begin{bmatrix}\eta_{0i} \\ \eta_{1i}\end{bmatrix} &\sim \text{Bivariate Normal}(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{bmatrix}) \\
\epsilon_{it} &\sim \text{Normal}(0, \sigma^2)
\end{aligned}
$$

Predict the heights for the next 8 years, and calculate the mean squared errors from the three models.

```
mlm.obj <- lme(height~year, data=oxboys.train, random=list(id=pdDiag(~year)))
predict(mlm.obj, newdata=oxboys.test, level=1)
# calculate MSE
```

(d) Repeat (c), but this time, the training set contains the first 6 years, and the test set has the rest.

(e) Discuss the results from (c) and (d). When does the multilevel perform better, and when does it not?