Homework 1

PSTAT 131/231, Winter 2018

Due on Feburary 4, 2018 at 11:00 pm

Note: *If your compiled pdf does not look right, number of spaces in* `indent1` *variable*

Predicting Algae Blooms

Background High concentrations of certain harmful algae in rivers constitute a serious ecological problem with a strong impact not only on river lifeforms, but also on water quality. Being able to monitor and perform an early forecast of algae blooms is essential to improving the quality of rivers.

With the goal of addressing this prediction problem, several water samples were collected in different European rivers at different times during a period of approximately 1 year. For each water sample, different chemical properties were measured as well as the frequency of occurrence of seven harmful algae. Some other characteristics of the water collection process were also stored, such as the season of the year, the river size, and the river speed.

Goal We want to understand how these frequencies are related to certain chemical attributes of water samples as well as other characteristics of the samples (like season of the year, type of river, etc.)

Data Description The data set consists of data for 180 water samples and each observation in the available datasets is in effect an aggregation of several water samples collected from the same river over a period of 3 months, during the same season of the year. Each observation contains information on 11 variables. Three of these variables are nominal and describe the season of the year when the water samples to be aggregated were collected, as well as the size and speed of the river in question. The eight remaining variables are values of different chemical parameters measured in the water samples forming the aggregation, namely: Maximum pH value, Minimum value of O_2 (oxygen), Mean value of Cl (chloride), Mean value of NO_3^- (nitrates), Mean value of NH_4^+ (ammonium), Mean of PO_4^3 (orthophosphate), Mean of total PO_4 (phosphate) and Mean of chlorophyll.

Associated with each of these parameters are seven frequency numbers of different harmful algae found in the respective water samples. No information is given regarding the names of the algae that were identified.

We can start the analysis by loading into R the data from the "algae.txt" file (the training data, i.e. the data that will be used to obtain the predictive models). To read the data from the file it is sufficient to issue the following command:

```
algae = read.table("algae.txt", header=T, na.strings="NA")
```

1. ***Descriptive summary statistics*** Given the lack of further information on the problem domain, it is wise to investigate some of the statistical properties of the data, so as to get a better grasp of the problem. It is always a good idea to start our analysis with some kind of exploratory data analysis. A first idea of the statistical properties of the data can be obtained through a summary of its descriptive statistics.

    (a) Count the number of observations in each level of speed using `summarise()` in `dplyr`.

    (b) Read the help files of `summarise()`, `summarise_all()` and `summarise_at()` and choose your favorite one to calculate the mean and variance of each chemical (Ignore a_1). What do you notice about the magnitude of the two quantities for different chemicals?

(c) Mean and Variance is one measure of central tendency and spread of data. Median and Median Absolute Deviation are alternative measures of central tendency and spread.

For a univariate data set $X_1, X_2, ..., X_n$, the Median Absolute Deviation (MAD) is defined as the median of the absolute deviations from the data's median:

$$\text{MAD} = \text{median}(|X_i - \text{median}(X)|)$$

    i. Compute median and MAD of each chemical and compare the two sets of quantities (i.e., mean vs. median and variance vs. MAD). What do you notice? Hint: use R built-in function `mad()`.

    ii. Read the help file of `mad()` and explain what the `constant` does.

2. ***Data visualization*** Most of the time, the information in the data set is also well captured graphically. Histogram, scatter plot, boxplot, Q-Q plot are frequently used tools for data visualization. Use ggplot for all of these visualizations.

(a) Produce a histogram of *Chla* (ignoring missing values) with the title 'Histogram of Chla' based on algae data set. Use an appropriate argument to show the probability instead of the frequency as the vertical axis. (Hint: look at the examples in the help file for function `geom_histogram()`). Is the distribution skewed?

(b) Add a density curve using `geom_density()` and rug plots using `rug()`to above histogram.

(c) Create a boxplot with the title 'A conditioned Boxplot of Algae a_1' for a_1 grouped by *season*. Color each box by *season*. (Refer to help page for `geom_boxplot()`).

(d) Are there any outliers for *mnO2* and *Cl*? How many observations would you consider as outliers? How did you arrive at this conclusion?

(e) Compare mean vs. median, variance vs. MAD for *oPO4* and *PO4*. What do you notice? Can you conclude which set of measures is more robust when outliers are present?

(f) **(231 Only)** Read the code for built-in function `mad()` function and its help page. What role does the default `constant = 1.4826` play? Inspect Q-Q norm plots for *mxPH* and *oPO4* respectively. What are the implications of the default choice of the constant for these two cases when comparing measures of spread: i.e. sd() and mad()?

Predicting Algae Blooms

Some water samples contained unknown values in several chemicals. Missing data are very common in real-world problems, and may prevent the use of certain data mining techniques that are not able to handle missing values.

In this homework, we are going to introduce various ways to deal with missing values. After all the missing values have been taken care of, we will build a model to investigate the relationship between the variable `a1` and other 11 predictors (`season, size, speed, mxPH, mnO2, Cl, NO3, NH4, oPO4, PO4, Chla`) utilizing cross-validation in the next problem.

Dealing with missing values

3. (a) How many observations contain missing values? How many missing values are there in each variable?

(b) **Removing observations with missing values**: use `filter()` function in `dplyr` package to delete observations with any missing value, and save the resulting dataset (without missing values) as `algae.del`. Report how many observations are in `algae.del`.

Hint: `complete.cases()` may be useful.

(c) **Imputing unknowns with measures of central tendency**: the simplest and fastest way of filling in (imputing) missing values is to use some measures of central tendency such as mean, median and mode.

Use `mutate_if()` and `ifelse()` in `dplyr` to fill in missing values for each chemical with its mean, and save the imputed dataset as `algae.mean`. Report the number of observations in `algae.mean`. Display the values of each chemical for the 70th, 117th and 180th obsevation in `algae.mean`.

This simple strategy, although extremely fast and thus appealing for large datasets, imputed values may have large bias that can influence our model fitting. An alternative for decreasing bias of imputed values is to use relationships between variables.

(d) **Imputing unknowns using correlations**: another way to impute missing values is to use correlation with another variable. For a highly correlated pair of variables, we can fill in the unknown values by predicting one based on the other with a simple linear regression model, provided the two variables are not both unknown.

Compute pairwise correlation between all variables.

Then, fill in all the missing values in `Chla` based on `mxPH`. What are the values you obtain?

Hint: use `lm()` and `coef()` function.

Estimating the Test Error with Cross Validation (CV)

Using `algae.mean` dataset obtained in (3c), we will build a model to predict the levels of algae type `a1` based on 11 variables (`season`, `size`, `speed`, `mxPH`, `mnO2`, `Cl`, `NO3`, `NH4`, `oPO4`, `PO4`, `Chla`), and test generalization of model to data that have not been used for training.

4. **Cross-validation**: in class we talked about cross-validation (CV) as a way to select the best model by estimating "test error". In k-fold CV, each of k equally sized random partitions of data (chunks) are used in a heldout set (called validation set or test set). After k runs, we average the held-out error as our final estimate of the validation error. For this part, we will run cross-validation on only a single model, as a way to estimate our test error for future predictions (we are not using it here for model selection since we are considering only one model). Perform 6-fold cross-validation on this model to estimate the (average) test error.

   (a) First randomly partition data into 6 equal sized chunks. Use `set.seed(10)` before cut/sample.

   Hint: a simple way to randomly assign each observation to a chunk is to do the following. First, use `cut(..., label=FALSE)` to divide observation ids (1, 2, ...) into roughly equal numbers of chunk ids. Then, randomize output of `cut()` by using `sample()`.

   (b) Perform 6-fold cross-validation with training error and validation errors of each chunk determined from (4a).

   Since same computation is repeated 6 times, we can define the following function for simplicity.

```
do.chunk <- function(chunkid, chunkdef, dat){  # function argument

    train = (chunkdef != chunkid)
    Ytr = dat[train,]$a1 # get true response values in training set
    Yvl = dat[-train,]$a1 # get true response values in validation set

    lm.a1 <- lm(a1~., data = dat[train,])
    predYtr = predict(lm.a1)  # predict training response values
    predYvl = predict(lm.a1,dat[-train,]) # predict validation values

    data.frame(fold = chunkid,
               train.error = mean((predYtr - Ytr)^2), # compute and store training errors
               val.error = mean((predYvl - Yvl)^2))   # compute and store validation errors

}
```

3

First argument `chunkid` indicates which chunk to use as validation set (one of 1:6). Second argument `chunkdef` is chunk assignments from (4a). Third argument `dat` will be `algae.mean` dataset.

In order to repeatedly call `do.chunk()` for each value of `chunkid`, use functions `lapply()` or `ldply()`. Note that `chunkdef` and `dat` should be passed in as optional arguments (refer to help pages).

Write the code and print out `train.error` and `val.error`.

5. **Test error on additional data**: now imagine that you actually get *new* data, in the sense that it wasn't available when you first fit the model. Additional data can be found in the file `algae-test.txt`.

```
algaeTest = read.table("algae-test.txt", header=T, na.strings="NA")
```

This data was not used within CV procedure for estimating the error. We can get a more accurate measure of true test error with this data. Using the model that you learned in part 4, calculate the "true" test error of your predictions based on the newly collected measurements in `algae-test.txt`. Is this roughly what you expected based on the CV estimated test error from part 4?

Cross Validation (CV) for Model Selection

In this problem, we will explore a dataset of sales of child car seats at 400 different stores. The goal in this part is to identify a relationship between sales and income.

6. ISLR package includes many of the datasets used in the ISLR textbook. Look at the variables defined in the Carseats dataset. We will be using the Sales and Income variables for this problem.

```
library(ISLR)
data(Carseats)
attach(Carseats)
```

(a) Plot Sales as a function of income using `ggplot`. Your plot should include the datapoints (`geom_point()`) as well as a smooth fit to the data (`geom_smooth()`). Based on your visualization, what is the general pattern of Sales as a function of Income? Does this match your intuition?

(b) In this part of the problem, we will find a polynomial function of Income that best fits the Carseats data. For each polynomial function between $p = 0, 1, 2, ...10$:

   i. Fit a linear regression to predict Sales as a function of $Income$, $Income^2$, ... $Income^p$ (you should include an intercept as well). Note that $p = 0$ model is an "intercept-only" model.

   ii. Use 6-fold cross validation to estimate the test error for this model. Save both the test error and the training error.

   Hint: `poly(Income, degree=p, raw=TRUE)` will return a matrix with p columns, where the p-th column is $Income^p$. For the predictors in your regression use `poly(Income, degree=p, raw=FALSE)`. The `raw=FALSE` option returns predictors which are numerically more stable (it returns a matrix of "orthogonal polynomials"). Numerical stability can be an issue because $Income^p$ can be very very large if p is large. The orthogonal polynomials returned when `raw=FALSE` are rescaled to account for this. A function similar to `do.chunk` from problem 4 will be helpful here as well.

   ```
   do.chunk <- function(chunkid, chunkdef, dat, p){  # function argument

       train = (chunkdef != chunkid)
   ```

```
        res = data.frame(degree=integer(), fold=integer(),
                         train.error=double(), val.error=double())


        if (p==0) {

          ## Your code here

          ## Fit an intercept only model to the data.
          ## Using poly(Income, degree=0) will cause an error

          ## Update residual

        } else {
          ## Your code here

          ## Fit a polynomial regression or order p.
          ## Use poly(Income, degree=p, raw=FALSE)

          ## Update residual
        }

      }
```

    iii. **(231 Only)** For $p = 4$, inspect the results given by `poly(Income, degree=p, raw=TRUE)` and `poly(Income, degree=p, raw=FALSE)`. How are they related? In addition, why does it increase the numerical stability of predictors using `poly(Income, degree=p, raw=FALSE)` ?

(c) Plot both the test error and training error (on the same plot) for each of the models estimated above as a function of p. What do you observe about the training error as p increases? What about the test error? Based on your results, which model should you select and why?

––––––––––––––––––––––––––––––

7. **(231 Only)** Prove L_∞ norm is supremum norm:

$$\|x\|_\infty = \lim_{r \to \infty} \left(\sum_{i=1}^{n} |x_i|^r \right)^{1/r} = \max_{1 \le i \le n} |x_i|$$

8. **(231 Only)** As we discussed in class, distance metrics satisfy the following properties:

- *Positivity*:
  - $d(x, y) \ge 0$
  - $d(x, y) = 0$ only if $x = y$

- *Symmetry*:
  - $d(x, y) = d(y, x)$ for all x and y

- *Triangle Inequality*:
  - $d(x, z) \le d(x, y) + d(y, z)$ for x, y, and z

Show that the following measures are distance metrics by showing the above properties hold:

1. $d(x, y) = \|x - y\|_2$
2. $d(x, y) = \|x - y\|_\infty$