Homework 1

PSTAT 131/231, Spring 2019

Eduardo Escoto (Perm: 7611817) / JunRong Ma (Perm: 1234567) / (PSTAT 131)

Due on April 20, 2019 at 11:59 pm

Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

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 200 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 "algaeBloom.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_table2("../algae_data/algaeBloom.txt", col_names=
                     c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
                       'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
                     na="XXXXXXX")

glimpse(algae)
```

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 season using `summarize()` in `dplyr`.

i. The numbers of observations for each season are calculated below and displayed in the `n_obs` column in the tibble below.

```
number_of_obs_by_season <- algae %>%
  # Group observations by season
  group_by(season) %>%
  # Counts the number of observations in the groups
  summarize(n_obs = n())

number_of_obs_by_season
```

```
## # A tibble: 4 x 2
##   season n_obs
##   <chr>  <int>
## 1 autumn    40
## 2 spring    53
## 3 summer    45
## 4 winter    62
```

(b) Are there missing values? Calculate the mean and variance of each chemical (Ignore a_1 through a_7). What do you notice about the magnitude of the two quantities for different chemicals?

i. There are indeed missing values! In the tibble below there is a preview of missing values.

```
algae_missing_values <- algae %>%
  # Filters on all columns on if any column contains an NA
  filter_all(any_vars(is.na(.)))

algae_missing_values
```

```
## # A tibble: 16 x 18
##    season size  speed  mxPH  mnO2     Cl    NO3   NH4   oPO4    PO4  Chla
##    <chr>  <chr> <chr> <dbl> <dbl>  <dbl>  <dbl> <dbl>  <dbl>  <dbl> <dbl>
##  1 autumn small high   6.8   11.1  9      0.63     20   4     NA      2.7
##  2 spring small high   8     NA    1.45   0.81     10   2.5    3      0.3
##  3 winter small low   NA     12.6  9      0.23     10   5      6      1.1
##  4 winter small high   6.6   10.8 NA      3.24     10   1      6.5   NA
##  5 spring small medi~  5.6   11.8 NA      2.22      5   1      1     NA
##  6 autumn small medi~  5.7   10.8 NA      2.55     10   1      4     NA
##  7 spring small high   6.6    9.5 NA      1.32     20   1      6     NA
##  8 summer small high   6.6   10.8 NA      2.64     10   2     11     NA
##  9 autumn small medi~  6.6   11.3 NA      4.17     10   1      6     NA
## 10 spring small medi~  6.5   10.4 NA      5.97     10   2     14     NA
## 11 summer small medi~  6.4   NA   NA     NA        NA  NA     14     NA
## 12 autumn small high   7.83  11.7  4.08   1.33     18   3.33   6.67  NA
## 13 winter medi~ high   9.7   10.8  0.222  0.406    10  22.4   10.1   NA
## 14 spring large low    9      5.8 NA      0.9     142 102    186     68.0
## 15 winter large high   8     10.9  9.06   0.825    40  21.1   56.1   NA
## 16 winter large medi~  8      7.6 NA     NA        NA  NA     NA     NA
## # ... with 7 more variables: a1 <dbl>, a2 <dbl>, a3 <dbl>, a4 <dbl>,
## #   a5 <dbl>, a6 <dbl>, a7 <dbl>
```

ii. The Means and Variances for each chemical are calculated below and displayed in the tibble.

```
m_v_by_chemical <- algae %>%
  # Selects All chemicals
```

```
  select(mnO2:Chla) %>%
  # Groups by key THEN removes NA's
  gather(chemical,factor_key = TRUE) %>% group_by(chemical) %>% na.omit() %>%
  # Calculates mean and variance
  summarize_all(list(mean = ~ mean(.),
                     variance = ~ var(.)))
m_v_by_chemical
```

```
## # A tibble: 7 x 3
##   chemical   mean   variance
##   <fct>     <dbl>      <dbl>
## 1 mnO2       9.12       5.72
## 2 Cl        43.6     2193.
## 3 NO3        3.28      14.3
## 4 NH4      501.   3851585.
## 5 oPO4      73.6     8306.
## 6 PO4      138.     16639.
## 7 Chla      14.0      420.
```

iii. As you can see in the tibble above, the magnitudes of thes Means for each chemical seem to vary highly and they all have large Variances with high magnitudes. That means that there is alot of spread between all of the data points for each chemical, except for mnO2 which has a variance of 5.7180895 which is relatively small compared to the variance of the other 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:

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

Compute median and MAD of each chemical and compare the two sets of quantities (i.e., mean & variance vs. median & MAD). What do you notice?

```
med_mad_by_chemical <- algae %>%
  # Selects only the chemicals
  select(mnO2:Chla) %>%
  # Removes the NA's AFTER grouping
  gather(chemical,factor_key = TRUE) %>% group_by(chemical) %>% na.omit() %>%
  # Calculates the MAD and Medians
  summarize_all(list(median = ~ median(.),
                     mad = ~ mad(., constant = 1, low = F, high = F)))

med_mad_by_chemical
```

```
## # A tibble: 7 x 3
##   chemical median   mad
##   <fct>     <dbl> <dbl>
## 1 mnO2       9.8   1.38
## 2 Cl        32.7  22.4
## 3 NO3        2.68  1.46
## 4 NH4      103.   75.3
## 5 oPO4      40.2  29.7
## 6 PO4      103.   82.5
## 7 Chla       5.48  4.5
```

i. Between *Mean* and *Median* the values tend to be pretty close to eachother, except in the case of NH4 where the mean is about 400 more than the median.

ii. The more noticable difference is the difference between *MAD* and *Variance*. Where the MAD is significantly lower compared to the Variance for the same chemical. A prime example of this is again, NH4 where the MAD is 75.285005 which is 4 powers of 10 lower than the Variance of 3.8515847×10^6.

iii. These differences between Mean and Medain and MAD and Variacne might tell us that most values tend to be close together but there are some values for each chemical that are very large that are skewing the Means and Variances.

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 $mxPH$ with the title 'Histogram of mxPH' 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 `geom_rug()` to above histogram.

   (c) Create a boxplot with the title 'A conditioned Boxplot of Algal a_1' for a_1 grouped by *size*. (Refer to help page for `geom_boxplot()`).

   (d) Are there any outliers for $NO3$ and $NH4$? How many observations would you consider as outliers? How did you arrive at this conclusion?

   (e) Compare mean & variance vs. median & MAD for $NO3$ and $NH4$. What do you notice? Can you conclude which set of measures is more robust when outliers are present?

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?

```
n_missing_obs <- algae %>%
  filter_all(any_vars(is.na(.))) %>% nrow()

n_missing_obs
```

```
## [1] 16
```

   i. There are 16 observations with missing values.

```
n_missing_obs_by_variable <- algae %>%
  gather(variable,factor_key = TRUE) %>% group_by(variable) %>%
  summarize_all(list(n_missing_vals = ~ sum(is.na(.))))
```

```
n_missing_obs_by_variable
```

```
## # A tibble: 18 x 2
##    variable n_missing_vals
##    <fct>             <int>
##  1 season                0
##  2 size                  0
##  3 speed                 0
##  4 mxPH                  1
##  5 mnO2                  2
##  6 Cl                   10
##  7 NO3                   2
##  8 NH4                   2
##  9 oPO4                  2
## 10 PO4                   2
## 11 Chla                 12
## 12 a1                    0
## 13 a2                    0
## 14 a3                    0
## 15 a4                    0
## 16 a5                    0
## 17 a6                    0
## 18 a7                    0
```

i. This tibble displays the number of missing values per variable.

(b) **Removing observations with missing values**: use `filter()` function in `dplyr` package to 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.

```
algae.del <- algae %>%
  filter(complete.cases(.))

nrow(algae.del)
```

```
## [1] 184
```

i. There are 184 observations in `algae.del`

(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_at()` and `ifelse()` in `dplyr` to fill in missing values for each chemical with its median, and save the imputed dataset as `algae.med`. Report the number of observations in `algae.med`. Display the values of each chemical for the 48^{th}, 62^{th} and 199^{th} obsevation in `algae.med`.

```
# Gets Columns with missing data into vector
missing_keys <- algae %>%
  gather(factor_key = TRUE) %>% group_by(key) %>%
  filter_all(any_vars(is.na(.))) %>% tally() %>%
  pull(key)

# Imputes at all values in columns that have missing data
algae.med <- algae %>%
```

```
   mutate_at(.vars = vars(missing_keys),
             .funs = list(~ ifelse(is.na(.), median(., na.rm=TRUE), .)))
```

```
chemicals_algae.med = algae.med %>% select(mnO2:Chla)

displ_chem_imp <- rbind(chemicals_algae.med[48, ],
                        chemicals_algae.med[62, ],
                        chemicals_algae.med[199, ])
displ_chem_imp <- tribble(~ index, 48, 62, 199) %>% cbind(displ_chem_imp)

displ_chem_imp
```

```
##   index mnO2    Cl   NO3      NH4  oPO4      PO4  Chla
## 1    48 12.6  9.00 0.230  10.0000  5.00   6.0000 1.100
## 2    62  9.8 32.73 2.675 103.1665 40.15  14.0000 5.475
## 3   199  7.6 32.73 2.675 103.1665 40.15 103.2855 5.475
```

i. The Tibble above shows the Imputed Values at the index requested.

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 the continuous (chemical) variables.

```
algae.pairwise_corr <- algae %>% select(mnO2:Chla) %>%
  cor(use = 'complete.obs') %>% as_tibble(rownames = "id")

algae.pairwise_corr
```

```
## # A tibble: 7 x 8
##   id       mnO2      Cl   NO3     NH4   oPO4    PO4    Chla
##   <chr>   <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>   <dbl>
## 1 mnO2   1      -0.267  0.110 -0.0799 -0.397 -0.468 -0.135
## 2 Cl    -0.267   1      0.214  0.0669  0.381  0.448  0.145
## 3 NO3    0.110   0.214  1      0.724   0.136  0.161  0.148
## 4 NH4   -0.0799  0.0669 0.724  1       0.220  0.200  0.0920
## 5 oPO4  -0.397   0.381  0.136  0.220   1      0.912  0.109
## 6 PO4   -0.468   0.448  0.161  0.200   0.912  1      0.251
## 7 Chla  -0.135   0.145  0.148  0.0920  0.109  0.251  1
```

i. The Pairwise Correlations are computed and displayed above.

Then, fill in the missing value for `PO4` based on `oPO4` in the 28^{th} observation. What is the value you obtain?

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

```
PO4.lm = lm(PO4 ~ oPO4, algae)

algae.PO4.pred <- algae %>%
  mutate(PO4 = ifelse(is.na(PO4),
                      predict(PO4.lm, newdata = data.frame(oPO4)), PO4))
```

```
algae.PO4.pred[28,"PO4"]
```

```
## # A tibble: 1 x 1
##     PO4
##   <dbl>
## 1  48.1
```

i. Above, the PO4 for the 28th observation is calculated using a linear model based on oPO4 due to the high pairwise correlation between them. In the 28th observation we have the value of oPO4 = 4 which gives us a predicted value of PO4 = 48.06929.

(e) **Questioning missing data assumptions**: When might imputation using only the observed data lead you to incorrect conclusions? In a couple of sentences, describe a scenario in which the imputed values of the chemical abundances in the algae data (imputed using either the median or correlation method) might be a poor substitute for the true missing values. Hint: look at the example from lecture 2.

i. Imputation might cause us to have incorrect conclusions becuase of relying too heavily on the observed data only. Lets say that in a scenario of the algae data we have new data that is far from the prediction based on oPO4 or far from the medians of each chemical, then we will have very high test error, and a model that is too overfitted on the training data.

ii. *In the context of Correlation Method:* In Lecture 2 we learned about *Survivorship Bias*. Many datesets have Survivorship Bias where the data that we have is insufficient in telling us about a certain variable. In the context of the algae data, using oPO4 to impute values of PO4 might be inducing Survivorship bias because oPO4 might actually not be sufficient in predicting PO4. Leading us to have high test error again

iii. *In the context of the Median Method:* this might lead us to the wrong conclusions by introducing alot of Bias because we are using only the observed data to impute. New data might be very far from the Median causing us to have values that have high test error.

Estimating the Test Error with Cross Validation (CV)

Using `algae.med` dataset obtained in (3c), we will build a linear regression 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 how to use cross-validation (CV) to estimate the "test error". In k-fold CV, each of k equally sized random partitions of the 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 5-fold cross-validation on this model to estimate the (average) test error.

(a) First randomly partition data into 5 equal sized chunks.

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 equal numbers of chunk ids. Then, randomize output of `cut()`by using `sample()`.

(b) Perform 5-fold cross-validation with training error and validation errors of each chunk determined from (4a).

Since same computation is repeated 5 times, we can define the following function for simplicity.

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

    train = (chunkdef != chunkid)
```

```
    Xtr = dat[train,1:11]  # get training set
    Ytr = dat[train,12]  # get true response values in trainig set

    Xvl = dat[!train,1:11]  # get validation set
    Yvl = dat[!train,12]  # get true response values in validation set

    lm.a1 <- lm(a1~., data = dat[train,1:12])
    predYtr = predict(lm.a1)  # predict training values
    predYvl = predict(lm.a1, Xvl)  # predict validation values

    data.frame(fold = chunkid,
               # compute and store training error
               train.error = mean((predYtr - Ytr$a1)^2),
                # compute and store test error
               val.error = mean((predYvl - Yvl$a1)^2))
}
```

First argument `chunkid` indicates which chunk to use as validation set (one of 1:5). Second argument `chunkdef` is chunk assignments from (4a). Third argument `dat` will be `algae.med` 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 the `train.error` and `val.error` five times (e.g. for each chunk).

5. BAD PROBLEM (test error is not similar to validation error) **Test error on additional data**: now imagine that you actually get *new* data that wasn't available when you first fit the model.

   (a) Additional data can be found in the file `algaeTest.txt`.

```
algae.Test <- read_table2('../algae_data/algaeTest.txt',
                    col_names=c('season','size','speed','mxPH','mnO2','Cl','NO3',
                                'NH4','oPO4','PO4','Chla','a1'),
                    na=c('XXXXXXX'))
```

   This data was not used to train the model and was not (e.g. wasn't used in the CV procedure to estimate the test error). We can get a more accurate measure of true test error by evaluating the model fit on this held out set of data. Using the same linear regression model from part 4 (fit to all of the training data), calculate the "true" test error of your predictions based on the newly collected measurements in `algaeTest.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 be exploring a dataset of wages from a group of 3000 workers. The goal in this part is to identify a relationship between wages and age.

6. First, install the `ISLR` package, which includes many of the datasets used in the ISLR textbook. Look at the variables defined in the `Wage` dataset. We will be using the `wage` and `age` variables for this problem.

```
library(ISLR)
head(Wage)
```

   (a) Plot wages as a function of age 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 wages as a function of age? Does this match what you expect?

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

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

ii. Use 5-fold cross validation to estimate the test error for this model. Save both the test error and the training error.

---

7. **(231 Only) The bias-variance tradeoff**. Prove that the mean squared error can be decomposed into the variance plus bias squared. That is, who $E[(\hat{\theta} - \theta)^2] = \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta})^2$ where $\text{Bias}(\hat{\theta}) = E[\hat{\theta} - \theta]$. Here $\hat{\theta}$ is an estimator (a random variable) of the fixed unknown constant θ. Hint: reogranize terms in the MSE by adding and subtracting $E[\hat{\theta}]$.

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

- *Positivity*:
  - $d(x, y) \geq 0$
  - $d(x, y) = 0$ only if $x = y$
- *Symmetry*:
  - $d(x, y) = d(y, x)$ for all x and y
- *Triangle Inequality*:
  - $d(x, z) \leq d(x, y) + d(y, z)$ 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. $d(x, y) = \|x - y\|_\infty$