Assignment 1

Statistics and Data Science 365/565

Due: Monday February 3rd 11:59pm

This homework treats linear regression and classification, and gives you a chance to practice using R. If you have forgotten some definitions or terms from previous classes, see the file `notation.pdf` in `Files/assignments` tab on Canvas. It should provide all you need to know to do this assignment. Remember that you are allowed to collaborate on the homework with classmates, but you must write your final solutions by yourself and acknowledge any collaboration at the top of your homework.

Problem 1: Two views of linear regression (10 points)

Recall that in linear regression we model each response Y_i as a linear combination of input variables $X_{i,p}$ and noise. That is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + ... + \beta_p X_{i,p} + \epsilon_i$$

which can be written in matrix form as

$$Y = X\beta + \epsilon$$

where $Y \in \mathbb{R}^n$ is the vector of responses (outcomes), $X \in \mathbb{R}^{n \times (p+1)}$ is the design matrix, where each row is a data point, and $\beta \in \mathbb{R}^{p+1}$ is the vector of parameters, including the intercept, and $\epsilon \in \mathbb{R}^n$ is a noise vector. Assume throughout this problem that the matrix $X^T X$ is invertible.

View 1: $\widehat{\beta}$ minimizes the Euclidean distance between Y and $X\beta$.

Suppose we make no assumptions about ϵ. We simply want to find the β that minimizes the Euclidean distance between Y and $X\beta$, i.e., the ℓ_2 norm of $Y - X\beta$. That is, we seek

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|^2.$$

Derive an explicit form for the minimzer $\widehat{\beta}$. Your derivation should involve calculating the gradient of the objective function $f(\beta) = \|Y - X\beta\|^2$, and solving for the β that makes the gradient zero. Express your solution as a function of the matrix X and the vector Y. (If you get stuck, try to first find a clean way to write the gradient with respect to β of the ℓ_2 norm function $g(\beta) = \|\beta\|^2$.)

View 2: $\widehat{\beta}$ is the MLE in a normal model.

Suppose we assume the same linear regression model as above, but now we assume that the ϵ_i are uncorrelated and identically distributed as $N(0, \sigma^2)$. Therefore, we can write

$$Y \sim N(X\beta, \sigma^2 I_n),$$

meaning that Y has a multivariate normal distribution with mean $X\beta$ and diagonal covariance matrix $\sigma^2 I_n$. Recall that for a vector $X \sim N(\mu, \Sigma)$, the density is

$$f(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\Big(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\Big).$$

To derive the maximum likelihood estimator under this model, maximize the log density of Y as a function of β, assuming that σ^2 is known. Show that the maximizer is the same as that obtained under View 1.

Problem 2: Linear regression and classification (30 points)

Citi Bike is a public bicycle sharing system in New York City. There are hundreds of bike stations scattered throughout the city. Customers can check out a bike at any station and return it at any other station. Citi Bike caters to both commuters and tourists. Details on this program can be found at https://www.citibikenyc.com/

For this problem, you will build models to predict Citi Bike usage, in number of trips per day. The dataset consists of Citi Bike usage information and weather data recorded from Central Park.

In the citibike_*.csv files, we see:

1. date
2. trips: the total number of Citi Bike trips. This is the outcome variable.
3. n_stations: the total number of Citi Bike stations in service
4. holiday: whether or not the day is a work holiday
5. month: taken from the date variable
6. dayofweek: taken from the date variable

In the weather.csv file, we have:

1. date
2. PRCP: amount precipitation (i.e. rainfall amount) in inches
3. SNWD: snow depth in inches
4. SNOW: snowfall in inches
5. TMAX: maximum temperature for the day, in degrees F
6. TMIN: minimum temperature for the day, in degrees F
7. AWND: average windspeed

You are provided a training set consisting of data from 7/1/2013 to 3/31/2016, and a test set consisting of data after 4/1/2016. The weather file contains weather data for the entire year.

Part a: Read in and merge the data.

To read in the data, you can run, for example:

```
train <- read.csv("citibike_train.csv")
test <- read.csv("citibike_test.csv")
```

Merge the training and test data with the weather data, by date. Once you have successfully merged the data, you may drop the "date" variable; we will not need it for the rest of this assignment.

As always, before you start any modeling, you should look at the data. Make scatterplots of some of the numeric variables. Look for outliers and strange values. Comment on any steps you take to remove entries or otherwise process the data. Also comment on whether any predictors are strongly correlated with each other.

Comment

For the rest of this problem, you will train your models on the training data and evaluate them on the test data.

Part b: Linear regression

Fit a linear regression model to predict the number of trips. Include all the covariates in the data. Print the summary of your model using the R `summary` command. Next, find the "best" linear model that uses only q variables (where including the intercept counts as one of the variables), for each $q = 1, 2, 3, 4, 5$. It is up to you to choose how to select the "best" subset of variables. (A categorical variable or factor such as "month" corresponds to a single variable.) Describe how you selected each model. Give the R^2 and the mean squared error (MSE) on the training and test set for each of the models. Which model gives the best fit to the data? Comment on your findings.

Part c: KNN Classification

Now we will transform the outcome variable to allow us to do classification. Create a new vector Y with entries:

$$Y[i] = \mathbf{1}\{trips[i] > median(trips)\}$$

Use the median of the variable from the full data (training and test combined). After computing the binary outcome variable Y, you should drop the original trips variable from the data.

Recall that in k-nearest neighbors classification, the predicted value $\widehat{Y}$ of X is the majority vote of the labels for the k nearest neighbors X_i to X. We will use the Euclidean distance as our measure of distance between points. Note that the Euclidean distance doesn't make much sense for factor variables, so just drop the predictors that are categorical for this problem. Standardize the numeric predictors so that they have mean zero and constant standard deviation—the R function `scale` can be used for this purpose.

Use the FNN library to perform k-nearest neighbor classification, using as the neighbors the labeled points in the training set. Fit a classifier for $k = 1 : 50$, and find the mis-classification rate on both the training and test sets for each k. On a single plot, show the training set error and the test set error as a function of k. How would you choose the optimal k? Comment on your findings, and in particular on the possibility of overfitting.

Problem 3: Classification for a Gaussian Mixture (25 points)

A Gaussian mixture model is a random combination of multiple Gaussians. Specifically, we can generate n data points from such a distribution in the following way. First generate labels $Y_1, \ldots, Y_n$ according to

$$Y_i = \begin{cases} 0 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2. \end{cases}$$

Then, generate the data $X_1, \ldots, X_n$ according to

$$X_i \sim \begin{cases} N(\mu_0, \sigma_0^2) & \text{if } Y_i = 0 \\ N(\mu_1, \sigma_1^2) & \text{if } Y_i = 1. \end{cases}$$

Given such data $\{X_i\}$, we may wish to recover the true labels Y_i, which is a classification task.

Part a.

Suppose we have a mixture of two Gaussians, $N(\mu_0, \sigma_0^2)$ and $N(\mu_1, \sigma_1^2)$, with $\mu_0 = 0, \mu_1 = 3$, and $\sigma_0^2 = \sigma_1^2 = 1$. Consider the loss function $\mathbf{1}\{f(X) \neq Y\}$. What is the classifier that minimizes the expected loss? Your classifier will be a function $f : \mathbb{R} \to \{0, 1\}$, so write it as an indicator function. Show your work, and simplify your answer as much as possible.

What is the Bayes error rate? Again, show your work.

Part b.

Suppose we have the same mixture as in Part a, but now $\sigma_0^2 \neq \sigma_1^2$. What classifier minimizes the expected loss in this case?

Part c.

Now generate $n = 2000$ data points from the mixture where $\mu_0 = 0, \mu_1 = 3$, and $\sigma_0^2 = 0.5, \sigma_1^2 = 1.5$. Plot a histogram of the X's. This histogram is meant to be a sanity check for you; it should help you verify that you've generated the data properly.

Set aside a randomly-selected test set of $n/5$ points. We will refer to the rest of the data as the training data. Use the labels of the training data to calculate the group means. That is, calculate the mean value of all the X_i's in the training data with label $Y_i = 0$. Call this sample mean $\widehat{\mu}_0$. Do the same thing to find $\widehat{\mu}_1$. To be explicit, let $C_j = \{i : Y_i = j\}$, and define

$$\widehat{\mu}_j = \frac{1}{|C_j|} \sum_{i \in C_j} X_i$$

Now classify the data in your test set. To do this, recall that your rule in Part b. depended on the true data means $\mu_0 = 0$ and $\mu_1 = 3$. Plug in the sample means $\widehat{\mu}_j$ instead. You should be able to do the classification in a single line of code, but there is no penalty for using more lines. Evaluate the estimator's performance using the loss:

$$\frac{1}{n} \sum_{i=1}^{n} 1\{\widehat{Y}_i \neq Y_i\}$$

Part d.

Now you train and evaluate classifiers for training sets of increasing size n, as specified below. For each n, you should

1. Generate a training set of size n from the mixture model in Part c.
2. Generate a test set of size 10,000. Note that the test set itself will change on each round, but the size will always be the same: 10,000.
3. Compute the sample means on the training data.
4. Classify the test data as described in Part c.
5. Compute the error rate.

Plot the error rate as a function of n. Comment on your findings. What is happening to the error rate as n grows?

```
seq.n <- seq(from = 2000, to = 20000, by = 20)
```