CS425 Project1 (Report)
Multiple Linear Regression Analysis

Ksenia Burova

October 1^{st}, 2017

Abstract: In machine learning, multiple regression is a function if multiple inputs (discrete or numeric), that has one output that is a class code or continuous value.
In this project, the goal was to predict the gas milage of automobiles from their other multiple attributes by using multiple linear regression. We were given two files, one with data and another one with attribute information for that data. Six values for horsepower were missing, and had to be skipped or predicted. Data attributes contained both, discrete and numeric values. We had to compare performance of data with and without standardization. After calculating weights, we had to analyze which attribute was the best in predicting gas milage.

I. Project data

The following attributes are provided:

- mpg: continuous - **to be predicted**
- cylinders: multi-valued discrete
- displacement: continuous
- horsepower: continuous
- weight: continuous
- acceleration: continuous
- model year: multi-valued discrete
- origin: multi-valued discrete
- car name: string (unique for each instance) - **not used**

We represent data (all but milage and names) in a matrix form:

$$\mathbf{X} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & X_2^2 & \cdots & X_d^2 \\ \vdots & & & \\ X_1^N & X_2^N & \cdots & X_d^N \end{bmatrix}$$

Each column is a single attribute, each row is a single experiment or data sample (in our case, all params for one unique car).

II. Resolving missing values

There are six observations where horsepower attribute is not defined. Sometimes it is the best strategy to discard those observations. Another way is to estimate those values using *imputation* method. There is mean imputation and imputation by regression possible.
I've used two approaches: ignored observations with missing values and I tried mean imputation.

III. Calculating weights with Multiple Regression

(Equations are copied from the textbook)

Going back to multiple regression, our numeric output is vector r which represents milage attribute. We have a weighted sum of attributes $x_1, x_2 ... x_d$ and some noise.

Here is a multivariate linear model:

$$r^t = g(\mathbf{x}^t | w_0, w_1, \ldots, w_d) + \epsilon = w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t + \epsilon$$

We may assume error to be normal with mean 0 and constant variance. To maximize the likelihood we will minimize the sum of squared errors:

$$E(w_0, w_1, \ldots, w_d | \mathcal{X}) = \frac{1}{2} \sum_t (r^t - w_0 - w_1 x_1^t - w_2 x_2^t - \cdots - w_d x_d^t)^2$$

Taking derivative with respect to parameters we get the normal equations:

$$\begin{aligned}
\sum_t r^t &= N w_0 + w_1 \sum_t x_1^t + w_2 \sum_t x_2^t + \cdots + w_d \sum_t x_d^t \\
\sum_t x_1^t r^t &= w_0 \sum_t x_1^t + w_1 \sum_t (x_1^t)^2 + w_2 \sum_t x_1^t x_2^t + \cdots + w_d \sum_t x_1^t x_d^t \\
\sum_t x_2^t r^t &= w_0 \sum_t x_2^t + w_1 \sum_t x_1^t x_2^t + w_2 \sum_t (x_2^t)^2 + \cdots + w_d \sum_t x_2^t x_d^t \\
&\vdots \\
\sum_t x_d^t r^t &= w_0 \sum_t x_d^t + w_1 \sum_t x_d^t x_1^t + w_2 \sum_t x_d^t x_2^t + \cdots + w_d \sum_t (x_d^t)^2
\end{aligned}$$

We now can define the matrix of attributes (as before) but with first column being all ones, since in our model w_0 term is the same as $w_0 \cdot 1$. We can define a vector of all weights, and an output vector which is our milage values. So we get:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & & \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d, \end{bmatrix}, \boldsymbol{r} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^N \end{bmatrix}$$

Then the normal equations can be written this way now:

$$\mathbf{X}^T\mathbf{X}\boldsymbol{w} = \mathbf{X}^T\boldsymbol{r}$$

And we can derive weights:

$$\boldsymbol{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{r}$$

So the goal is to compute and analyze these weights.

IV. Program structure

I've written my application in Python as was suggested. I've used *numpy* library to use matrix operations on vectors.
I have two files. I've defined a class *MultRegression* in one and have written functions for calculating mean, standard deviation, data standardization and weights. My second one is basically like main, I create an instance of the class and pass parameters.
I pass 3 parameters to my class (I manually modify them in main function):

1. Data file name 'auto-mpg.data'
2. Boolean value to decide if i need to standardize my data. 1 - yes, 0 - no
3. Boolean value to decide if I need to skip data with missing attributes. 1 - yes, 0 - no

I've tested my functions on smaller data examples and matrices before running it on givven data.

V. Results

The first time I ran my program I skipped lines with missing attributes and I didn't standardize parameters. I received following values for weights:

```
[ -1.80586717e+01  -4.18254893e-01   1.88870416e-02  -1.13851962e-02
   -6.71865820e-03    1.02620868e-01   7.56755050e-01   1.41751561e+00]
```

It is really hard to analyze these values since attributes that were expected to have negative effect on milage have a corresponding weight with positive sign, like car displacement. More horsepower and cylinders do have a negative effect on milage, as expected. It is hard to analyze absolute values though.
Usually data standardization is needed when we use attributes of different types that have different value ranges. Here are results after standardization:

```
[ 20.37880831  -5.00325077  -2.37809451  -0.7357686
 2.30248253     0.4401979     3.09178799   6.02984611]
```

We can see that number of cylinders, car displacement and horsepower have negative effects on gas milage, as expected. Where other attributes have a more positive effect. it is possible, that newer car would have greater gas milage, so results look fine so far.
Now, I estimated horsepower for samples with missing values and added those:

```
[ 20.36175876  -4.70649111  -3.1073928   -0.05821537
   1.98385778   0.43863716   2.88210938   6.30068179]
```

The absolute values have changed, looking at absolute values of weights corresponding to horsepower we can assume now that that attribute is almost useless in predicting milage, which can't be really true. So sometimes, it is better to remove values from data than estimate values that may influence results in negative way. Or maybe better predicting algorithm should be used like imputation by regression to get better estimated values with less noise.

Now let's look at plots for given data: