CSL7020 Assignment 1

Topic: Linear Regression

Wilfred Kisku (P19EE003)

15/09/2019

Problem

For the linear regression task, we need to predict the values of the air pollutants: Ozone, Sulphur dioxide and particulate matter PM2.5 using the past data for air pollution and meteorological data. The air pollutant data for three pollutants (output value Y) - O_3, $PM_{2.5}$ and SO_2 hourly data from the year 2010 to 2018 can be downloaded from the following link:

https://aqs.epa.gov/aqsweb/airdata/download_files.html#main-content

The above website of the EPA (Environment Protection agency), being part of the AQS (Air Quality System) database would be relied on heavily for our machine learning model for predicting the above mentioned values. The data from the AQS being very sparse with only hourly concentrations of pollutant data as numerical values, should be coupled with data from other source that can be related with the pollutants. This leads to formulation of a collective dataset from the meteorological dataset from :

http://mesowest.utah.edu/

We need to model a predictive learning model to estimate the future dated values of the pollutants such as O_3, $PM_{2.5}$ and SO_2.

1 Dataset

Thee files in the AQS dataset contain the hourly data (sometimes called measurements, samples, etc). These files are separated by parameter (or parameter group) to make the sizes more manageable.

EPA does not get sample data reported at duration other than hourly, so each of the sampled data points have been taken for each site at an hourly basis. If a particular file is empty (record count = 0) that means that no hourly data was collected for that parameter or group. The important fields that need to be taken into account are the `Local Date` and `Local Time` from this dataset. For getting a Linear Regression model we would be taking a small portion of the dataset that includes all then states and different counties in the state. Our focus would be the in a suburban residential area in the southwestern Cook County, in the State of Illinois. The two locations picked are Aspin Village (O_3 and $PM_{2.5}$ measurement) and Lemont Village (SO_2 measurement)

From the MesoWest dataset the two locations that are picked which is very close to the AQS collection site are situated at Lansing Municipal Airport, near to Aspin Village and Lewis University site which is closer to Lemont Village site. The features that would be helping us model a regression model would be

Temperature, relative humidity, wind speed and direction, wind gust, precipitation accumulation, visibility, dew point, wind cardinal direction, pressure and weather conditions. The data picked from Aspin Village is Ozone and PM2.5 concentration and the values picked from the Lemont Village is sulphur dioxide.

2 Approach

The most important aspect is to get an idea of the dataset and extract useful information that can be processed by the system for its training. This makes it imperative to pre-process and remove redundant information from the dataset. Also, the dataset should be free from inconsistencies such as missing data points, multiple data points, incoherent data points between the AQS dataset and the MesoWest dataset.

The simplified idea to remove multiple data point that I have used is to pick up the first data point for that particular hour being closest to the data point associated with the other dataset (since Mesowest has multiple points during a particular hour).

Other scheme to make data computational friendly, so as to train a linear model, is to assign numeric values to fields such as wind directions (16 values) and weather conditions (31 values). The values can be normalized to fall within a range of [0,1].

2.1 Pre-processing

The most difficult problem was to clean the dataset as these were the issues that was related to the dataset:

- There were two datasets that has to be accessed and consolidated, the MesoWest dataset for the features and the AQS dataset to obtain the particulate matter concentration in the air.
- The Datasets had a missing and uncomputable values such as `'NA'`, `'N/A'`, `'na'`, `'n/a'`, `'--'`, `'-'`, `Null` being some of the most common missing values in the dataset.
- The features that have been used in the dataset are listed as `hour`, `altimeter`, `air_temperature`, `relative_humidity`, `wind_speed`, `wind_direction`, `visibility`, `visibility`, `dew_point_temperature`, `pressure`.
- The range for the pressure was in `Pascals` and the values were in the range of thousands so it had to be normalized in the range of [0,1]. Other values also needed to be normalized as the squared value would grow in the `np.square` function and `np.sum` functions.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- The cleaning of the dataset is done using the `Pandas` library in python, the DataFrame is easier to handle and process the `csv` files.
- The complete matrix contain the `X` matrix (D features + 1 bias) concatinated with the output vector `Y`. So the matrix would be of the dimension `m` $\times$ `D+2`, where `m` is the total number of samples (or) rows in the matrix (data points).

2.2 Model Description

The model chosen is a simple linear regression model with training being carried out using gradient descent, the number of samples are quite large as it has been sampled from the year 2008 to 2018.
In regression we would model a hypothesis that would be used to predict the output of the next days concentration of SO_2, $PM_{2.5}$ and O_3. A linear hypothesis is selected as the prediction is to made over a range of real values $\mathbf{R}$. The hypothesis is $h_w(x)$. A cost function is to be calculated to penalize the hypothesis as to obtain the correct set of parameters w.

$$J(w) = \frac{1}{2m} \sum (h_w(x) - y)^2$$

The linear model $h_w(x) = w^T x = w_0 + w_1 x_1 + ... + w_d x_d$, the parameters are to be adjusted to minimize the cost $J(w)$. One way of doing it is by gradient descent, where each iteration performs an update:

$$w_i := w_i - \eta \frac{1}{m} \sum (h_w(x) - y) x$$

With each step of the gradient descent the parameters w comes closer to the optimal values that will achieve the lost cost $J(w)$.

3 Experiments

The experiments are divided into the the training and the test phase where the derived model is checked for an output for the given dataset. Furthermore the hyper-parameters such as η and $features$ can be changed so to deduce a more accurate model. The model that has been chosen in the assignment has a limited number of paramters as the weather conditions and the wind cardinal directions were removed as they were not computable, to model these two features they can be coded with a computable values. Also regularization can be added to reduce overfitting and generalizing the linear model.

3.1 Setup

The experimental setup can be divided into several phases:

3.1.1 Pre-processing of Dataset

The pro-processing is the most important stage of cleaning the data and obtaining the dataset that can be worked with. The major portion of the time was spent in gleaning through the dataset that has a lot of missing values and unknown values that was removed using the `Pandas` library.

3.1.2 Select a Learning Model

The majority of the work was carried out on the dataframe that was converted into a `numpy` array so as to carry out operations such as `np.dot`, `np.sum()`, `np.square()` and `transpose` operations on the dataframe. The concept of bordcasting makes it easier to operate on large datasets.

3.1.3 Gradient Descent for training

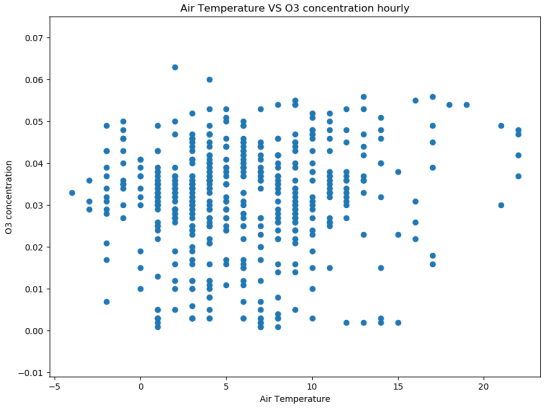
The Gradient descent is used to iterate through the dataset for `n` iters so that the values converge to a global minimum. The values that are obtained is the $J(w)$ essentially that is a convex function. The using gradient descent it would reach a global minimum value eventually are number of iteration. It is essential to keep the value of η relatively low so that the values not overshoot while training.

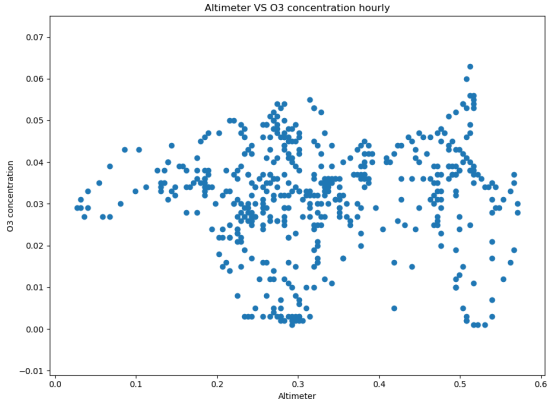
3.1.4 Predict using the trained model

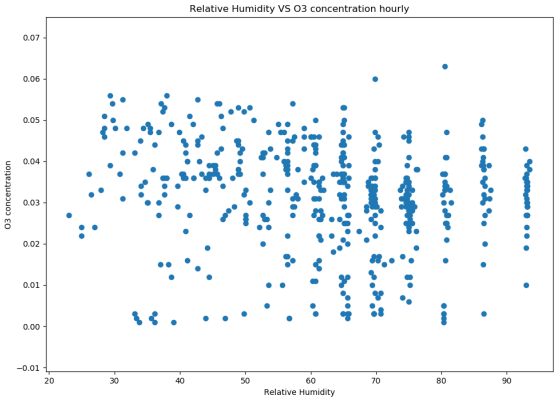
The final obtained parameters are tested using few input values so as to text the model accuracy.

3.2 Results

The plots are individual column value plots(`altimeter, air temperature etc.`) in relation with the concentrations of the pollutants. Since the models includes multiple features, it is multi-dimensional is nature making it hard to get a visual representation of the features in relation to the output concentration of the pollutants.
It can be seen that the it would be very difficult to get a linear relation between the concentration of the pollutants, here below is the concentration of O_3, with just a single meteorological data such as `pressure, temperature or humidity`. So including multiple features would make it easier to draw a relation between the output concentrations and the features.

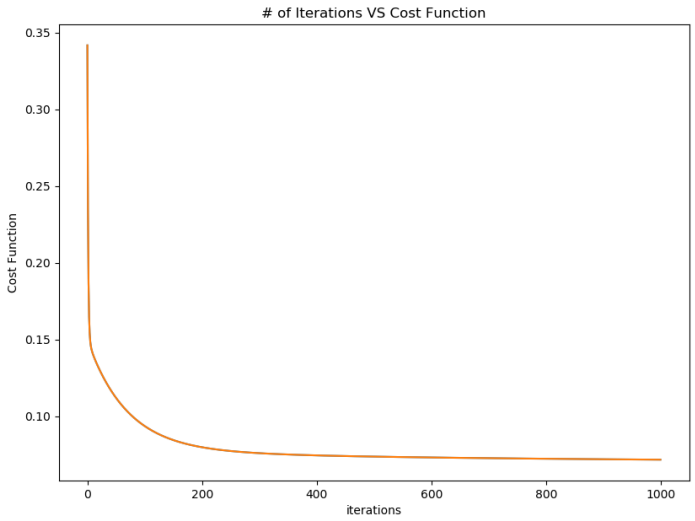






4 Analysis

The values that can be plotted is the cost function with respect to the number of iterations. The cost function needs to be minimized with respect to the parameters. The figure below shows the number of iterations reduces the cost function through gradient descent and it saturates at a certain value, approximates the minimum. Here for a test sample of 500 data points the cost function saturates at `0.07167366` for the number of iterations `1000`.



The weights obtained are `[[0.00935967],[0.00868393],[0.00331254],[0.00457893],[0.00512671],[0.00106384],[-0.00297324],[0.00652226],[-0.00056628],[0.00331254]]`. And the prediction values of a sample input `np.array([1.0,0.043,0.41,0.46,0.54,0,0,1,0.92,0.42])` results in `array([0.02335851])` and `np.array([1.0,0.52,0.52,0.23,0.82,0,0,1,-1.05,0.52])` results in `array([0.02969427])`. The operation between the weight vector and the inputs are supposed to be `predict = np.dot(inp,w)`.