Projects

Jose M. Barcelo Ordinas

Universidad Politècnica de Catalunya (UPC-BarcelonaTECH),
Computer Architecture Dept.
joseb@ac.upc.edu

April 29, 2021

1 Project 2. Calibration of sensors in uncontrolled environments in Air Pollution Sensor Monitoring Networks.

The objective of this project is to calibrate an air pollution sensor in an air pollution monitoring sensor network. We will take the data sampled of one node that accommodates three sensors: a MIC2614 O_3 sensor, a temperature sensor and a relative humidity sensor. The data set contains one thousand samples. You have to write a report with the main result of the calibration process. Write your feelings, plot curves, and discuss the results. Each calibration method has assigned a set of points for grading purposes.

2 Project Realization (Part I): observe the data (1 point)

The data consists on a CSV file called ”datos_TOML-17001.csv”, being the format the following:

```
date;RefSt;Sensor_O3;Temp;RelHum
21/06/2017 7:00;15.0;36.3637;21.77;53.97
21/06/2017 7:30;15.0;34.8593;25.5;42.43
```

where it can be seen that the first row is a header that describes the data:

- **date:** Timestamp (UTC) for each measurement,
- **RefSt:** Reference Station O_3 concentrations, in μgr/m^3,
- **Sensor_O3:** MOX sensor measurements, in KΩ,
- **Temp:** Temperature sensor, in °C,
- **RelHum:** Relativa humidity sensor, in %.

The first step consists on understanding the data. For that purpose, the best approach is to plot several curves to see dependencies of the data. I recommend using PANDAS as tool to handle the data.

1. The ozone sensor works as a voltage divisor. That means, as explained in class, that it is represented as a variable resistor. Thus the first step is to plot the ozone (KOhms) as function of time to observe the data. Moreover, it is interesting to plot the ozone reference data as a function of time. You can compare them and see that they follow similar patterns.

2. In order to observe the linear dependence between the reference data and the sensor data, draw a scatter-plot, a plot in which in the x-axes you have the ozone sensor data and in the y-axes you have the reference data. Ideally, the data should looks like linear with a tangent of 45 degrees. You will see that it is not a perfect line since the sensor is not perfect (thus the calibration). Sometimes the scatter-plot is difficult to observe due to the scale (ozone sensor is in Kohms and ozone concentration is in μgr/m^3), then you can normalize each data point with respect its mean and standard deviation. It is to say, for example to normalize the ozone sensor data: (i) Obtain the mean of the training set, μ_{sensor}, (ii) Obtain the standard deviation (std) of the training set, σ_{sensor}, and (iii) Normalize all the samples of the training: for j=1,..., K$_1$,

$$\bar{x}_{sensor_j} = \frac{x_{sensor_j} - \mu_{sensor}}{\sigma_{sensor}} \tag{1}$$

where, $\bar{x}_{sensor_j}$ are the normalized sensor data. Do the same for the reference data and plot again the scatter-plot. In our case, you should not have difficulties in plotting the scatterplot in the original scale, thus, it is not necessary to normalize. In any case, you can do it in order to practice and see that the normalization does not change the pattern.

3. It is also interesting to plot scatterplots of the sensor with respect temperature and with respect relative humidity. Do the same with the reference station with respect the temperature and with respect to the relative humidity.

3 Project Realization (Theoretical Part II): calibration (9 points)

Here you have a set of calibration methods to test and play. In some cases, it is necessary to normalize (e.g. when you use gradient descent methods), and in other cases it is necessary to optimize hyperparameters. Make plots to explain results. For example, in gradient descent, you can plot cost against number of steps or in some of the machine learning methods is useful to plot cost against hyperparameters. You can obtain R^2, RMSE and MAE to check performance (see definitions in wikipedia), and also there are libraries in SciPy to get them. For each method you can plot the reference concentrations (e.g. in red) and the estimated concentrations (e.g. in blue), and you can regress (linearly) the Refdata against the estimation to see how good is the estimated data. At the end of the report, you can compare all the performance parameters (e.g. using a table) with all methods to see which method performs best.

1. Multiple linear regression (MLR)
   (a) Multiple linear regression (MLR) with scipy solver and with normal equations **(1 point)**
   (b) Multiple linear regression (MLR) with gradient descent method (batch, stochastic and mini batch) **(1 point)**
2. K-nearest neighbor (KNN) **(1 points)**
3. Random forest (RF) **(1 points)**
4. Kernel regression with RBF kernel and with a polynomial kernel (e.g. degree 4) **(1 points)**
5. Gaussian Process (GP) **(1 points)**
6. Support Vector Regression (SVR) **(1.5 points)**
7. Neural Network (NN) **(1.5 points)**