Predicting PM2.5 Value in Future

Kaushal Thaker

kaushalthaker145@gmail.com

Abstract. PM2.5 is a critical environmental issue, with high levels harming human health and the environment. Predicting PM2.5 levels can help implement preventive measures. This paper uses data mining techniques to address two problems: predicting the PM2.5 value for the next time session and determining the PM2.5 pollution level category for the next day. Neural networks and other models are used, showing good experimental results. The paper concludes with potential refinements to the prediction process.

1 Introduction

I am living on an unhealthy Mother Earth. Industrialization and modernization provide us with a better life while failing to provide a healthier environment. Particle Pollution(PM) is a rising problem that has drawn increasing attention nowadays. Many websites, like Baidu, put real-time PM values on their front pages. The general public and the government are aware of this insidious air pollution and are trying to find ways to solve this environmental puzzle.

PM is a mixture of solids and liquid droplets floating in the air[4]. Some particles are released directly from a specific source, while others form in complicated chemical reactions in the atmosphere. Particles come in a wide range of sizes. Particles less than or equal to 10 micrometers in diameter are so small that they can get into the lungs, potentially causing serious health problems. Ten micrometers is less than the width of a single human hair.

More specifically, this paper focuses on fine particles. Fine particles (PM2.5) are 2.5 micrometers in diameter or smaller and can only be seen with an electron microscope. Fine particles are produced from all types of combustion, including motor vehicles, power plants, residential wood burning, forest fires, agricultural burning, and some industrial processes. PM2.5 will trigger both negative health effects and environmental effects. For example, it will irritate the eyes, cause building damage, or even cause heart attacks.

Considering the destructive effect of PM2.5, predicting the value of PM2.5 becomes a meaningful task. If I can use the past historical data to predict the PM2.5 value of the next hour or even the next day, the public and the departments concerned will be more prompt to make some corresponding measures. For example, if I know that tomorrow will be heavy with PM2.5, I might cancel the football match held outdoors.

In this paper, several machine learning methods are deployed to predict the value or level of PM2.5 in the future. The dataset I downloaded from FTP is

provided by the U.S. Department of State[2]. The dataset consists of various PM2.5 data from different cities in China for several years, and a more detailed description of the data will be presented in Section 3.1.

The paper is organized as follows: In Section 2, I give a sound description of the problem. In Section 3, I present some basic processing and analysis of the original raw data. In Section 4, multiple machine learning methods are proposed to solve the problems using the data I get. All the experimental results are shown in Section 5, after which I elaborate on possible future work in Section 6. Finally, I conclude this paper in Section 7.

2 Problem Description

I develop two different problems to solve in this paper.

Theorem 1. *Given historic data* $[X, Y]$*, in which* $X = [x_1^T, x_2^T, ..., x_n^T]^T$*, and* $Y = [y_1, y_2, ..., y_n]$*, where* x_i *is the historic data series corresponds to the observed data* y_i*. Our goal is to predict the number* y_t *when I have observed* x_t*. Therefore, this is a regression problem.*

Next, I consider a more practical problem. Considering the scenario that you want to plan for tomorrow, the PM level of that day should be taken into account, just like the traditional information, such as temperature or rain. So, the next problem is to predict the level of the next day's average PM2.5. The categories here is set according to Air Quality Guide.[1] There are seven categories: Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy, Hazardous, and Beyond Index; I use 1 to 7 to represent them, respectively.

Theorem 2. *Given historic data* $[X, Y]$*, in which* $X = [x_1^T, x_2^T, ..., x_n^T]^T$*, and* $Y = [y_1, y_2, ..., y_n]$*, where* x_i *is the historical data series corresponds to the average daily PM2.5 pollution category* $y_i \in \{1, 2, 3, 4, 5, 6, 7\}$*. Our goal is to us predict the average daily PM2.5 level category* y_t *when I have observed* x_t*. Therefore, this is a multi-classification problem.*

After this, I found that the training data was insufficient, so I modified that a little bit, changing seven categories into 3, that is, healthy(0-50), moderate(51-100), and unhealthy(¿100), respectively. In other words, $y_i \in \{1, 2, 3\}$

3 Data Analysis and Processing

After I had defined the scope of this paper and the specific problem description, it was high time that I introduced the dataset and the ways to handle the data. This may highlight the properties of the information I get and provide insight into the approaches I implement to achieve the goal and solve the problems.

3.1 Data Description

The dataset I get is provided by the U.S. Department of State[2]. In the dataset, I have the PM2.5 data in Beijing from 2008 to 2014, Chengdu from 2012 to 2014, Guangzhou from 2011 to 2014, Shanghai from 2011 to 2014, and Shenyang from 2013 to 2014. Each CSV sheet contains the time of each record and the corresponding PM value of that time. Since Beijing has the most PM2.5 data records, I mostly train the model using Beijing data.

Unfortunately, some data are missing for several reasons. That makes the time series incomplete and triggers several difficulties in generating a training set, which will be elaborated on in Section 3.2. I wrote a simple Python program to get the statistics of all the data.

A summary of the data can be found in Table 1.

Table 1. Data Description

City	Year	Missing Records	All Records	Missing Rate
Beijing	2008	266	5087	5.2%
Beijing	2009	1981	8760	22.6%
Beijing	2010	669	8760	7.6%
Beijing	2011	727	8760	8.3%
Beijing	2012	489	8784	5.6%
Beijing	2013	82	8760	9.4%
Beijing	2014	99	8760	11.3%
Chengdu	2012	4372	8784	49.8%
Chengdu	2013	1393	8760	15.9%
Chengdu	2014	285	8760	3.3%
Guangzhou	2011	7863	8760	89.8%
Guangzhou	2012	2249	8784	25.6%
Guangzhou	2013	384	8760	4.4%
Guangzhou	2014	669	8760	7.6%
Shanghai	2011	8683	8760	99.1%
Shanghai	2012	283	8784	3.2%
Shanghai	2013	184	8760	2.1%
Shanghai	2014	136	8760	1.6%
Shenyang	2013	3374	8760	38.5%
Shenyang	2014	357	8760	4.1%

3.2 Training Data Acquisition

Now that I have the raw data, I shall convert it into a format recognized by MATLAB. I wrote a Python script to do the transformation.

Recall that in the last section, I illustrated that much raw data is missing in the original CSV. So, in the Python script, I carefully select the recordings that are available for training, which means I choose the records with complete prior PM data instead of some missing ones.

Considering that every city has a disparate environment and thus the model should be trained within different cities, I henceforth only choose data from Beijing to do this Beijing. Without the loss of generality, the same approaches can be used in other cities as long as I have sufficient data from that city.

For problem 1, I assume that the PM level will only be related to recent historical data; the data that is far earlier than the predicted time will have little or nothing to do with that time's value. I collected all the records, ensuring that their past 24-hour data was complete, and I have a total of 47407 records altogether in Beijing. I separate them into my training, cross-validation, and testing data. Basically, here, I harness the AR model, where the dimension of the AR model is set to 24, to train the data and predict the PM value.

Alternatively, I select another set of features to experiment with. Instead of choosing only the past hours' data, I add the same hours' data in the past days into the feature space, taking into account the fact that the PM value is probably related to the time of the day. So, the total dimension is two more than the first one, namely 26.

For problem 2, I calculate the daily average PM2.5 value in Beijing and collect all the records that their past 3-day data are complete. Finally, I have a total of 1505 records of data. I separate them to be my training, cross-validation, and testing data as well.

4 Learning Approaches

Two problems are regression problems and classification problems, respectively. Different methods are employed to solve the problems.

Here are several models that I can train to solve the two problems:

For Problem 1:

- Linear Regression
- Polynomial Regression
- Neural Network
- ...

For Problem 2:

- Logistic Regression
- Neural Network
- Linear Regression
- Gaussian Mixture Model
- ...

I am quite familiar with these methods, so I do not bother introducing these approaches from scratch. Considering the time and my poor laptop configuration, I tried some of these methods and then showed the corresponding results in the next section.

5 Experimental Results

My data processing tasks are finished on a 4G memory MacOS X computer, which utilizes Python 2.7 scripts. My machine learning problems are solved using a 4G memory Windows 7. The version of MATLAB is MATLAB 2013b.

5.1 Results for Problem 1

As stated before, due to the constraints of time and other resources, I performed one method to solve Problem 1, which is the Neural Network, because Neural Network fits the data in a polynomial way and thus is a better way. Also, the neural network will not take much time, for the amount of data is not that much. Thanks to the friendly MATLAB machine learning toolkits[3], I avoid writing a bunch of codes and debugging for a long time. I set 70% as the training set, 15% as the cross-validation set, and the rest 15% as the testing set. The neurons in the hidden layer are set to be 50.

As stated in 3.2, I use two sets of features for the model. Their performances are presented in Figure 1 and Figure 2.

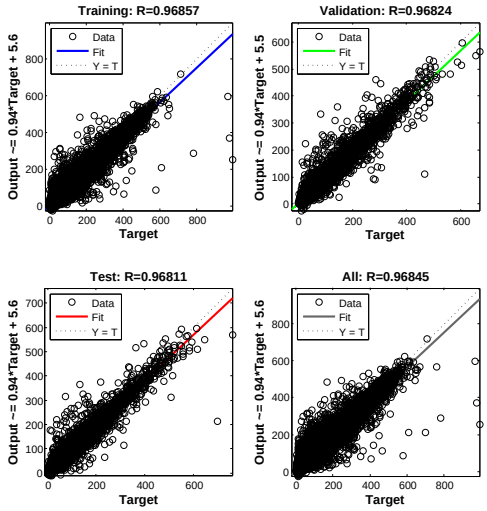

Fig. 1. The Neural Network fitting using past hours features

As I can see, the two different feature spaces perform well and have a negligible accuracy difference, both with R equal to more than 0.95. I can now safely state that using the local information is quite enough; the PM2.5 value of one time can be sufficiently told from its local information.

In the meantime, I performed a two-dimensional polynomial regression to see how many hours of past data are sufficient. I calculated the Mean Absolute

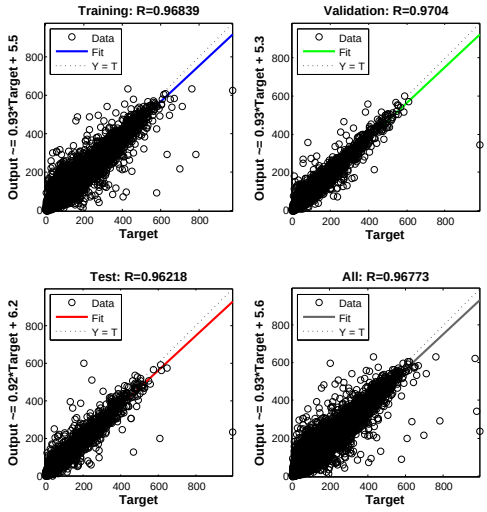

Fig. 2. The Neural Network fitting using past hours features and same-time past day features

Percentage Error(MAPE) to compare the different dimensions. The result is shown in Figure 3. I can see that using only past two-hour features can fit a decent result, further proving the locality property of the PM2.5 value.

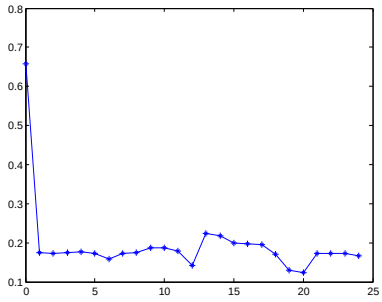

Fig. 3. The MAPE of using data of past x days

5.2 Results for Problem 2

I now try to solve problem 2 using a neural network. I also set 70% as the training set, 15% as the cross-validation set, and the rest 15% as the testing set. The neurons in the hidden layer are set to be 20. The experimental results are shown in Figures 4, 5, and 6.

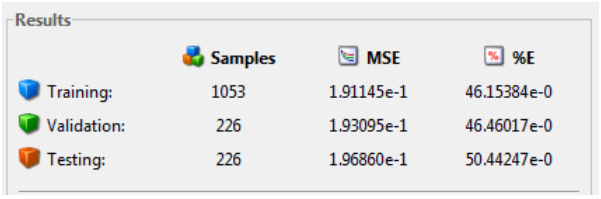

Fig. 4. Fitting Results of Problem 2

As I can tell, the result is not so good; although the determined data has around 50% accuracy, which is higher than a random guess, which is 33%, a lot of input cannot be classified. This is because the amount of training data is so small that I can not introduce a good model. If I have more data, this situation can be solved because I can judge from the result and the shape of the curve that this approach is promising.

6 Future Work

Retrospectively, considering all the defects in the whole process, there exist many ways to improve the performance.

- The approaches in this paper only consider the past PM value as the features to train the model. Actually, more features can be added to train a better model. For example, I can add the wind strength, precipitation, and other meteorological effects. These effects are highly related to the PM2.5 value.
- In the previous data selection, I treated every time in a day equally; I trained several uniform models for the whole data. Actually, there should have been some difference, and the resulting model for each time shall be different. I can thus choose some time in a day to see whether training separately could provide a better result.
- I can define other problems and solve them. For example, my friend, Jiaming, aims to predict a long-term value series instead of a single value. This mission, of course, is a far more difficult task, and as far as I know, his model does not perform well due to its complexity. I can try solving that conundrum in the future or even propose some more interesting and challenging problems.

7 Conclusions

The importance of the environment is beyond words. Environment governance is one of the most urgent contemporary issues. Before we can fully emancipate ourselves from pollution problems, I can use cutting-edge data mining techniques to predict the value of some pollution index and thus take corresponding measures.

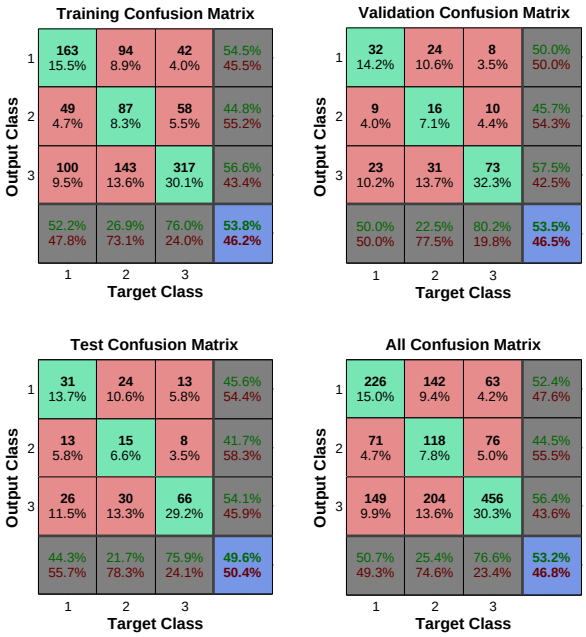

Fig. 5. The Confusion Matrix

In this paper, I lead a roadmap from the beginning to the end. I first give a full description of the PM2.5 problems to be solved in this paper. Next, Chapter 3 starts with getting the data and processing it, making a compact, simple feature vector for those built-in MATLAB machine-learning toolkits.

After these works, some famous models that have been taught in class and well built in MATLAB are imposed on the given feature vectors, generating different models. I use these disparate models, more concretely, polynomial regression and neural network, to solve the proposed problems, and the corresponding results are given and discussed.

Finally, I make some remarks on the whole approach, indicating some room for further study to improve. Some may be easier to realize than others, but each one will benefit the result to some extent after careful coding and tuning.

References

1. Air quality guide. http://www.stateair.net/web/post/1/1.html
2. Air quality monitoring program. http://www.stateair.net/web/historical/1/1.html
3. Fit data with a neural network. http://www.mathworks.com/help/nnet/gs/fit-data-with-a-neural-network.html
4. Particle pollution (pm). http://www.airnow.gov/index.cfm?action=aqibasics.particle

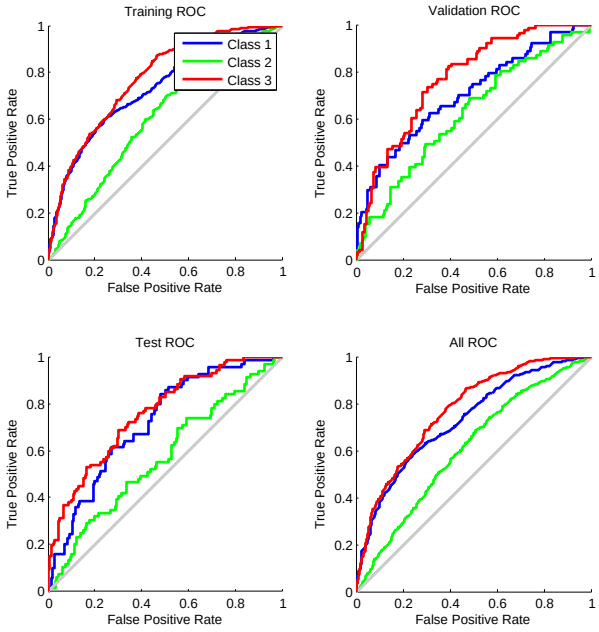

Fig. 6. The ROC Curve