Predicting PM2.5 Value in Future

Zhaowei Tan*
5120309701
IEEE Honor Class
Shanghai Jiao Tong University
Shanghai, China
tanzw94@gmail.com

ABSTRACT

PM2.5 is a critical environmental issue being discussed nowadays; when the amount of PM2.5 reaches a certain level, it does harm to both human beings and the whole environment. Therefore, predicting the value of PM2.5 becomes one of the critical problems, for this will allow citizens and the departments concerned to implement corresponding measures ahead of time. In this paper, I use data **********

Keywords

Data Mining, Regression, Classification PM2.5, Health and Environment

1. INTRODUCTION

[1]We are living on an unhealthy Mother Earth. The industrialization and modernization accommodate us with a better life, while failing to provide a healthier environment. The Partical Pollution(PM) is a rising problem which has drawn more and more attention nowadays. Many websites, like Baidu, put real-time PM value in their frontpages. The general public and the government have been aware of this insidious air pollution, 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 focus 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 cause irritation of the eyes, building damage, or even heart attacks.

Considering the destructive effect of PM2.5, predicting the value of PM2.5 becomes a meaningful task. If we can use the past historic data to predict the PM2.5 value of the next hour, or even 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 of 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 future. The dataset we downloaded from FTP is provided by U.S. Department of State[1]. The dataset consists of various PM2.5 data from different cities in China for several years, and more detailed description of data will be presented in Section 3.1.

The paper is organized as follows. I give the sound description of the problem in Section 2. In Section 3, I present some basic processing and analyzing of the original raw data. In Section 4, multiple machine learning methods are proposed to solve the problems using the data we get. All the experimental results are shown in Section 5, after which I elaborate 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.

PROMBLEM 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 us predict the number* y_t *when we 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 like temperature or raining or not. So the next problem is to predict to which level will the next day's average PM2.5 be. The categories here is set according

*This author is the one who did all the really hard work. All the codes, methods and approaches are written or designed by the author. I will EXPLICITLY point out when some work mentioned in the paper is not done by this author, for example, an open source program.

[1]This project does not ask us to hand in the source code, but I upload most of the source codes in a Github repository . You can find that in https://github.com/ZhaoweiTan/pm

Data Mining Course June, 2015, Shanghai, China.

to Air Quality Guide.[2] 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.

PROMBLEM 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 historic 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 we have observed* x_t*. Therefore, this is a multi-classification problem.*

3. DATA ANALYSIS AND PROCESSING

After I have defined the scope of this paper and the specific problem description, it is high time that I introduced the dataset and the approach handle the data. This may highlight the property of the information we get and provide a insight of the approaches we implement to achieve the goal and solve the problems.

3.1 Data Description

The dataset we get is provided by U.S. Department of State[1]. In the dataset, we 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. In each csv sheet, there contains the time of each record and corresponding PM value of that time. Since the Beijing has the most PM2.5 data records, I mostly train the model using Beijing data.

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

A brief summary of the data can be found in the 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 we have the raw data, we shall convert that into the format which can be 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 Python script I carefully select the recordings that are available for training, which means I choose the records that has a 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 lost of generality, the same approaches can be used in other cities, as long as we have sufficient data from that city.

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

Alternatively, I select another set of features to do the experiment. Instead of only choosing the past hours' data only, I select several past hour's data and the same hour's data in the past days, taking the fact that PM value is related to the time in a day into account.

For problem 2, I calculate the daily average PM2.5 value in Beijing and collect all the records that their past 7-dat data are complete. Now I have a total of ********** records of data. I separate them to be my training, cross validation and testing data.

4. LEARNING APPROACHES

Two problems are regression problem and classification problems, respectively. So different methods are employed to solve the problems.

Here are several models that we 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
- ...

We are quite familiar with these methods, so I do not bother introducing these approaches from scratch. Consider the time and my poor laptop configuration, I tried some of these methods, and then show the corresponding result in the next section.

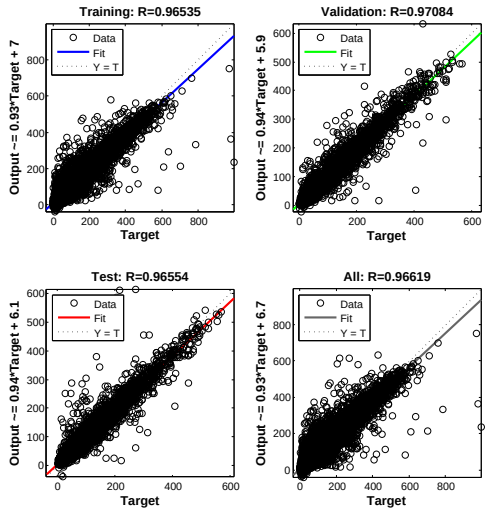

Figure 1: The Neural Network fitting using past hours features

5. EXPERIMENTAL RESULTS

My data processing tasks are finished on a 4G memory MacOS X computer. These tasks utilize Python 2.7 scripts. My machine learning problems are solved using a 4G memory Window 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 perform one method to solve the Problem 1, that is the Neural Network, because Neural Network fits the data in a polynomial and thus 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 bunch of codes and debugging for a long time.

For the model, as was stated in 3.2, we use two sets of models, and their performance is presented in the Figure 1 and Figure 2.

As we can see,

5.2 Results for Problem 2

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 considers the past PM value as the feature to train the model. Actually, more features can be added to train a better model. For example, we 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.
- We can define different problems and solve them. For example, my friend, Jiaming, aims to predict a long-term value series instead of

7. CONCLUSIONS

Speech recognition is really a tough task, especially in this project for it has a small sample number but a very high dimension feature space. The essence and the most dirty works lie in this part because the learning codes are built-in in MATLAB, otherwise this will be a HUGE project.

In this paper, I lead a roadmap from the beginning to the end. The Chapter 2 starts from getting the data to the processing (segmenting, truncating, etc.) of the data, making a compact, simple feature vector for those built in machine learning toolkits.

After these time-consuming work, some famous models which have been taught in class are imposed on the given feature vectors and thus generate different models. I run various models and disparate algorithms, namely MV and MP, to get the categories of the test data. The accuracy indicates more balanced data are needed to generate a better model.

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

8. ACKNOWLEDGMENTS

I learned bunch of knowledge out of this project, including the basics of several machine learning toolkits in MATLAB, the coding tricks, writing the paper, so on and so forth.[2] Everything just help me improve and let me become more prepared for my Ph.D career. Also, this long journey provides me with a better insight of data mining and machine learning, which is, and will continue to be, the hottest topic in computer science.

Cannot believe this semester is going to over. THANK YOU to TAs for a whole semester's great help. THANK YOU to Prof. Yuan for your wonderful instructions. This is a fruitful course and I do deeply feel the elegance of data mining. Hope that everything is alright!

9. REFERENCES

[1] Air quality monitoring program. http://www.stateair.net/web/historical/1/1.html.

[2]Actually, I never thought of that writing a paper is such an exhausting and skillful work. From now on I respect more of those 8-page, 10-page paper. It is almost desperate to type so much words.

Table 2: Experimental Results

	Testing Accuracy	English Testing Accuracy	Mandarin Testing Accuracy
LR with MV	65%	70%	60%
LR with MP	55%	60%	50%
LR (biased) with MV	87.5	100%	50%
LR (biased) with MP	87.5	100%	50%
K-means	51%	N/A	N/A
K-means (biased)	75.5%	N/A	N/A

[2] Air quality monitoring program. http://www.stateair.net/web/post/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.