Article

Machine learning algorithms for predicting roadside fine particulate matter concentration level in Hong Kong Central

Yin Zhao, **Yahya Abu Hasan**

School of Mathematical Sciences, Universiti Sains Malaysia (USM), Penang, Malaysia

E-mail: ameenzhao@gmail.com

Received 1 May 2013; *Accepted 5 June 2013*; *Published online 1 September 2013*

Abstract

Data mining is an approach to discover knowledge from large data. Pollutant forecasting is an important problem in the environmental sciences. This paper tries to use data mining methods to forecast fine particles ($PM_{2.5}$) concentration level in Hong Kong Central, which is a famous business centre in Asia. There are several classification algorithms available in data mining, such as Artificial Neural Network (ANN) and Support Vector Machine (SVM). ANN and SVM are both machine learning algorithm used in variant area. This paper builds $PM_{2.5}$ concentration level predictive models based on ANN and SVM by using R packages. The data set includes 2008-2011 period meteorological data and $PM_{2.5}$ data. The $PM_{2.5}$ concentration is divided into 2 levels: low and high. The critical point is 40μg/m^3 (24 hours mean), which is based on the standard of US Environmental Protection Agency (EPA). The parameters of both models are selected by multiple cross validation. According to 100 times 10-fold cross validation, the testing accuracy of SVM is around 0.803~0.820, which is much better than ANN whose accuracy is around 0.746~0.793.

Keywords Artificial Neural Network (ANN); Support Vector Machine (SVM); $PM_{2.5}$ prediction; data mining; machine learning.

Computational Ecology and Software
ISSN 2220-721X
URL: http://www.iaees.org/publications/journals/ces/online-version.asp
RSS: http://www.iaees.org/publications/journals/ces/rss.xml
E-mail: ces@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Air pollution is a major problem for some time. Various organic and inorganic pollutants from all aspects of human activities are added daily to the air. One of the most important pollutants is particulate matter. Particulate matter (PM) can be defined as a mixture of fine particles and droplets in the air and this can be characterized by their sizes. $PM_{2.5}$ refers to particulate matter whose size is 2.5 micrometers or smaller. Due to its effect on health, it is crucial to prevent the pollution getting worse in a long run. According to WHO's report, the mortality in cities with high levels of pollution exceeds that observed in relatively cleaner cities by 15–20% (WHO, 2011). Forecasting of air quality is much needed in a short term so that necessary preventive action can be taken during episodes of air pollution. Considering that our target data set is from a roadside

station of business centre which is in Hong Kong, so we try to find a moderate standard of $PM_{2.5}$ as the split criterion. The standard of US Environmental Protection Agency (EPA) divides the mean of $PM_{2.5}$ concentration in 24-hour level into 6 levels, either "Good" or "Moderate" means the concentration should be less than 40μg/m^3 (EPA, 2006). As a result, we use this critical value as our standard point. The number of particulate at a particular time is dependent on many environmental factors, especially the meteorological data, say, air pressure, rainfall, humidity, air temperature, wind speed, etc.

In this paper, we try to build models for predicting next day's $PM_{2.5}$ mean concentration level by using two popular machine learning algorithms, which are, Artificial Neural Network (ANN) and Support Vector Machine (SVM). ANN is inspired by attempts to simulate biological neural system. It may contain many intermediary layers between its input and output layers and may use types of activation functions (Rojas, 1996; Nagendra and Khare, 2006; Zhang, 2010). SVM has its root in statistical learning theory. It requires only a dozen examples for training and is insensitive to the number of dimensions (Tan et al., 2006).

An important issue in data mining is not only to analyse data but also to see them, so we choose R (Ihaka and Gentleman, 1996) as our analysis tool in this paper. R is an open source programming language and software environment for statistical computing and graphics. It is widely used for data analysis and statistical computing projects. In this paper, we will use some R packages as our analysis tools, namely "nnet" package (Ripley, 2013) and "e1071" package (Meyer et al., 2013). Moreover, we also use some packages for plotting figures, such as "reshape2" package (Wickham, 2013) and "ggplot2" package (Wickham, 2013).

The structure of the paper is: Section 2 reviews some basic concept of these two algorithms. Section 3 and 4 will describe the data and the experiments. At last, the conclusion will be given in Section 5.

2 Methodology

2.1 Artificial Neural Network (ANN)

ANN is formed by a set of computing units (i.e. the neurons) linked to each other. Each neuron executes two consecutive calculations: a linear combination of its inputs, followed by a nonlinear computation of the result to obtain its output value that is then fed to other neurons in the network. Each of the neuron connections has an associated weight. Constructing an artificial neural network consists of establishing architecture for the network and then using an algorithm to find the weights of the connections between the neurons. The network may contain many intermediary layers between its input and output layers. Such intermediary layers are called hidden layers and the nodes embedded in these layers are called hidden nodes. The network may use types of activation functions, such as linear, sigmoid (logistic), and hyperbolic tangent functions, etc. In R "nnet" package, the sigmoid function is default for classification model. Its expression is shown below:

$$f(x) = \frac{e^x}{1+e^x}$$

The back-propagation (i.e. BP) algorithm is used in layered feed-forward ANN (Hagan et al., 1996; Viotti et al., 2002; Zhang and Barrion, 2006; Zhang and Zhang, 2008). The BP algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error is calculated.

Let $D = \{(x_i, y_i)) \mid i=1, 2, \ldots, N\}$ be the set of training examples. The goal of the ANN learning algorithm is to determine a set of weights w that minimize the total sum of squared errors.

$$E(w) = \frac{1}{2} \cdot \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the output value by performing a weighted sum on its input.

And the weight update formula used by the gradient descent method:

$$w_j \leftarrow w_j - \lambda \frac{\partial E(w)}{\partial w_j}$$

where λ is the learning rate.

There are two phases in each iteration of the BP algorithm:

♦ The forward phase. During the forward phase, outputs of the neurons at level *k* are computed prior to computing the outputs at level *k*+1.

♦ The backward phase. During the backward phase, the weights at level *k*+1 are updated before the weights at level *k* are updated.

This BP algorithm allows us to use the error for neurons at layer *k*+1 to estimate the errors for neurons at layer *k*.

2.2 Support Vector Machine (SVM)

SVM is a relative learning algorithm used for binary classification. The basic idea is to find a hyper-plane which separates the d-dimensional data perfectly into two classes. However, since the data is often not linearly separable, SVM casts the example data into a higher dimensional space where the data is separable. The mapping of the original data into this new space is carried out with the help of the so-called kernel functions. In practice, it does not involve any computations in that high-dimensional space. Thus, SVM is linear machines operating on this dual representation induced by kernel functions (Karatzoglou et al., 2006).

We are given *t* training examples $\{x_i, y_i\}$, $i = 1, 2, \ldots, t$. All hyper-plans are parameterized by a vector *w* and a constant *b*, expressed in the equation:

$$w \cdot x + b = 0$$

To obtain the geometric distance from the hyper-plane to a data point, we must normalize by the magnitude of *w*. This distance is simply:

$$d = \frac{y_i(w \cdot x_i + b)}{\|w\|} \geq \frac{1}{\|w\|}$$

From the equation above we see this is accomplished by minimizing ||*w*|| (i.e. maximizing *d*). The main method of doing this is with Lagrange multipliers (Burges, 1998). The problem is eventually transformed into:

Maximize: $\sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$

Subject to: $\sum_{i=1}^{t} \alpha_i y_i = 0$, $0 \leq \alpha_i \leq C$.

where α is the vector of *t* non-negative Lagrange multipliers to be determined, *C* is cost parameter, and *k* represents a kernel function.

Kernels functions return the inner product between two points in a suitable feature space. Generally, the Gaussian Radial Basis Function (RBF) kernel is used when there is no prior knowledge about the data, the expression is:

$$k(x, x') = \exp(-\gamma \|x - x'\|^2)$$

There are also some other kernel functions in SVM algorithm, such as Linear kernel which is useful in text categorization, Polynomial kernel which is popular in image processing, etc.

3 Data Preparation

One of the most important steps in data mining is how to collect the data. In another word, we have to choose proper data sets as our target. Firstly, all of data for the 2000-2011 periods were obtained from Hong Kong Environmental Protection Department (HKEPD) and Hong Kong Met-online. The air monitoring station is Central Air Monitoring which is a roadside station in the centre of Hong Kong, and the meteorological monitoring station is King's Park Weather Station which is one of the major stations of Hong Kong Observatory (HKO). As mentioned in Section 1, accurately predicting high $PM_{2.5}$ concentration is of most value from a public health standpoint, thus, the response variable has two classes, which are, "Low" indicating the daily mean concentration of $PM_{2.5}$ is below 40μg/m^3, and "High" representing above it. Fig. 1 shows that the days of two levels in 2000-2011. We learn that the air quality becomes better in recent years since the "Low" level increases and "High" level decreases. The best situation is in 2009 which has the most "Low" days, while the worst is in 2004. Our task is to build a suitable prediction model, that is, the data should be in accordance to the trend of air pollutant (i.e. "Low" is more than "High"). Thus, we choose 2008-2011 as our data target. Additionally, the only meteorological data cannot exactly describe the air pollution, Fig. 2 and Fig. 3 shows the mean pressure and the mean air temperature, respectively. As we learn from both figures, either the mean pressure or mean air temperature waves slightly during 12 years period. But compare with Fig. 1, which shows the air pollution varies seriously. That means we have to add more predictor variables to classify the difference even if under the same meteorological conditions. Hence, we add the previous day $PM_{2.5}$ concentration and two time variables.

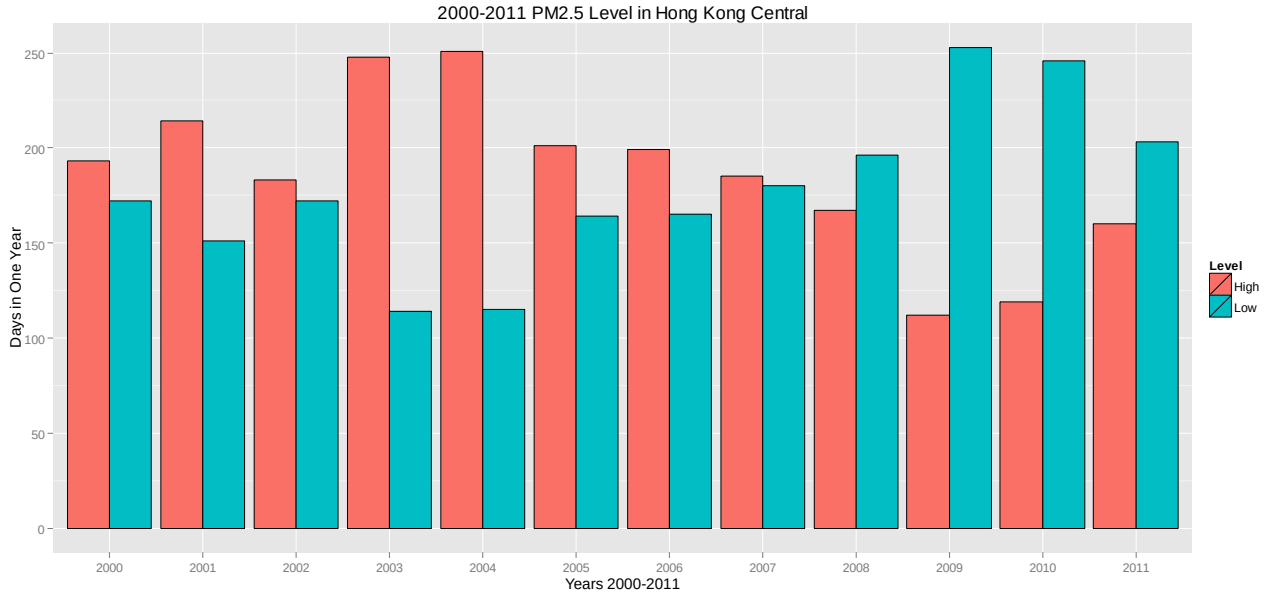

Fig. 1 $PM_{2.5}$ concentration levels in 2000-2011.

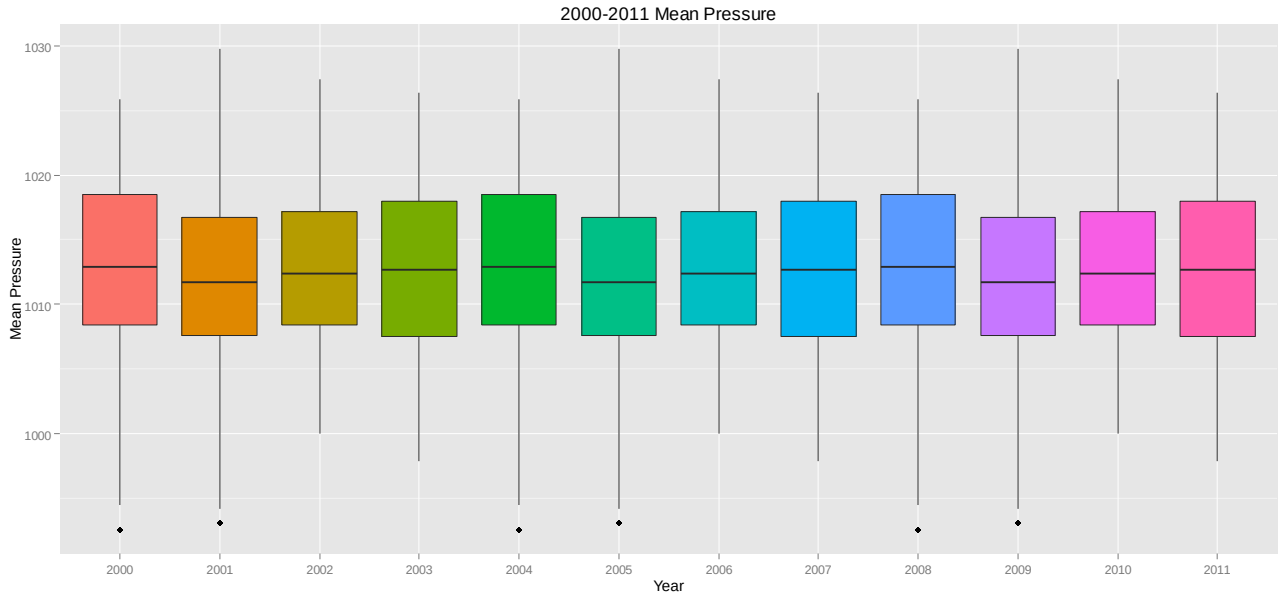

Fig. 2 2000-2011 mean pressure.

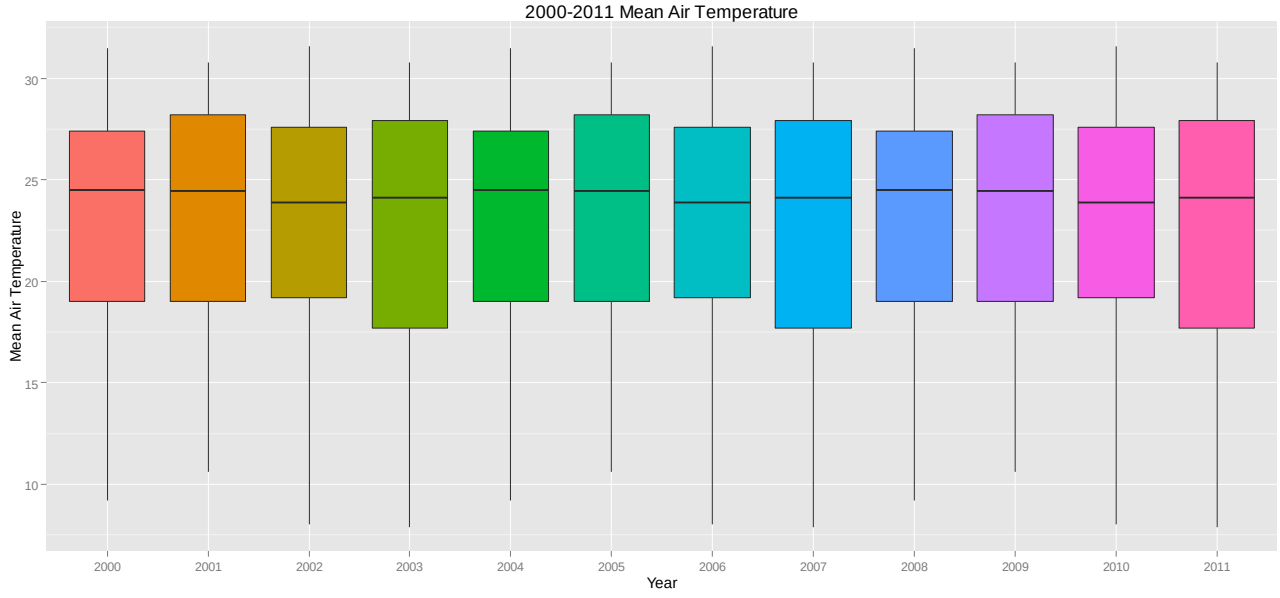

Fig. 3 2000-2011 mean air temperature.

We convert all hourly $PM_{2.5}$ data to daily mean values and the meteorological data is the original daily data. In addition, all of air data and meteorological data are numeric. We certainly cannot ignore the effects of seasonal changes and human activities; hence we add two time variables, namely the month (Fig. 4) and the day of week (Fig. 5). Fig. 4 clearly shows that $PM_{2.5}$ concentration reaches a low level from May to September, during which is the rainy season in Hong Kong. But from October to next April the pollutant is serious (though November seems better than others), especially in December. We should know that the rainfall may not be an important factor in the experiment as the response variable is the next day's $PM_{2.5}$, and it is easy to understand that rainy season includes variant meteorological factors. Fig. 5 presents the trends of people's activities in some sense. We learn that the air pollution is serious on Monday and Thursday, while the lowest level appears

on Saturday. This situation can be related to Hong Kong Central is a business area in Hong Kong and the air quality is also influenced by human activities.

At last, there are 1443 observations by deleting all NAs and 14 predictor variables (Table 1) and 1 response variable which is the next day's $PM_{2.5}$ concentration level. In summary, the percentage of "Low" level is around 61.4% and "High" level is around 38.3%, respectively (about 0.3% data are NAs).

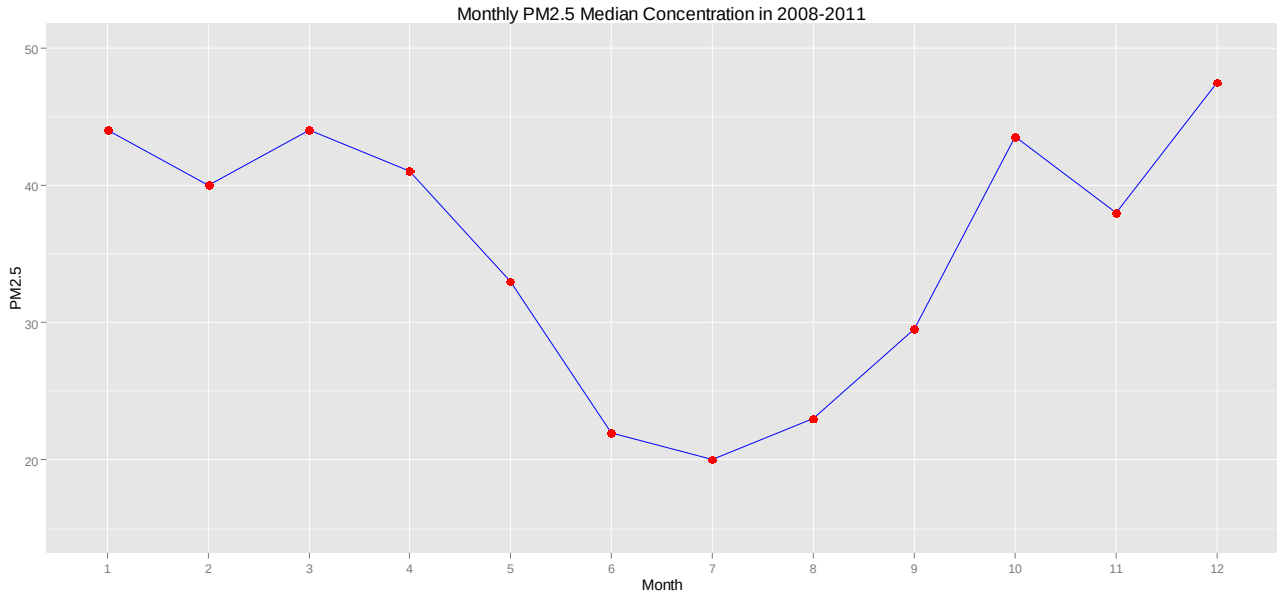

Fig. 4 Monthly $PM_{2.5}$ concentration in 2008-2011.

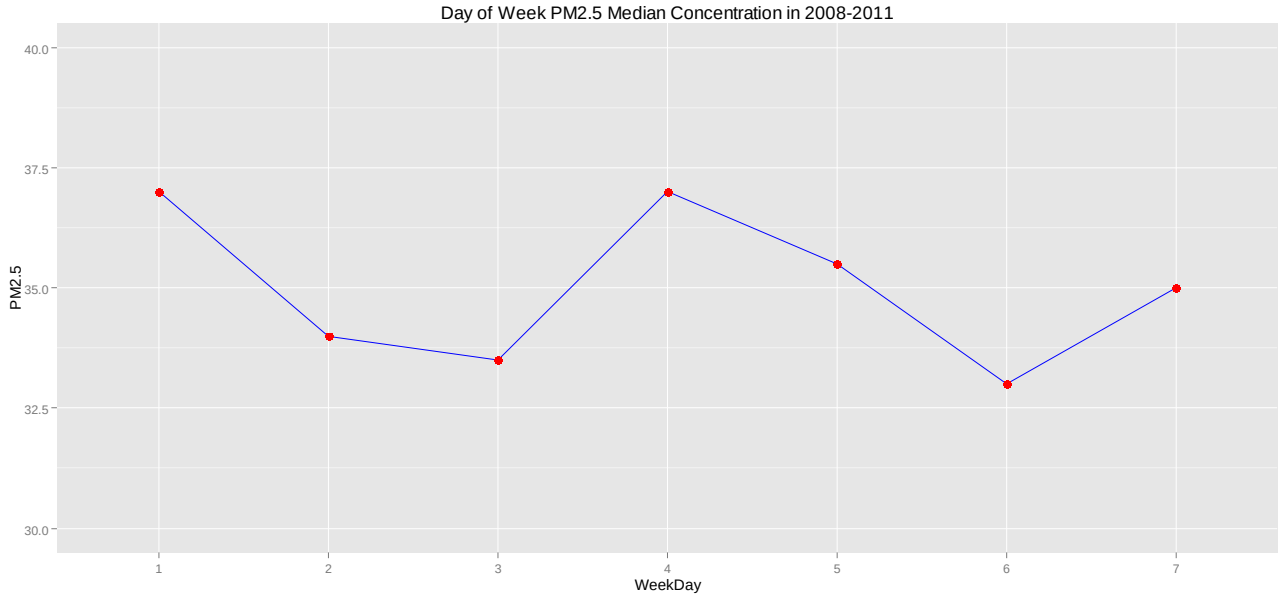

Fig. 5 Daily $PM_{2.5}$ concentration in 2008-2011.

Table 1 Variables list.

Notation	Description	Variable Class
MP	Mean Pressure	Numeric
AT1	Max Air Temperature	Numeric
AT2	Mean Air Temperature	Numeric
AT3	Min Air Temperature	Numeric
MDPT	Mean Dew Point Temperature	Numeric
RH1	Max Relative Humidity	Numeric
RH2	Mean Relative Humidity	Numeric
RH3	Min Relative Humidity	Numeric
TR	Total Rainfall	Numeric
PWD	Prevailing Wind Direction	Numeric
MWS	Mean Wind Speed	Numeric
$PM_{2.5}$	$PM_{2.5}$ concentration	Numeric
MONTH	Month	Nominal
WEEK	Day of week	Nominal

4 Experiments

The experiments include three parts: Section 4.1 and 4.2 will show how to train and test each model by 10-fold cross validation (10-fold CV), we will choose the best parameter using in Section 4.3 which will test the performance of each model by 100 times 10-fold CV.

4.1 ANN

"nnet" package is the R package we used in this paper. One of the most important parameters of ANN is to select the number of nodes in the hidden layer. We use 10 times 10-fold CV to calculate the training and testing accuracy when the nodes are from 1 to 30 and the number of iteration is 500. The result is shown in Table 2. We learn that the best number is *size* = 6, whose testing accuracy is 0.783. Figure 6 shows the trends of accuracy by changing the number of nodes in the hidden layer from 1 to 30 and the testing method is still 10 times 10-fold CV. We find that the testing accuracy is stable after *size* = 5, but the training accuracy waves more seriously than the testing. ANN may appear over-fitting when the hidden nodes is large, which means when the training accuracy increasing and the testing accuracy decreasing rapidly. But it does not appear over-fitting in our experiment, or say at least it has a proper performance within 30 hidden nodes. In summary, we will use *size* = 6 in the last section.

Table 2: Accuracy of Different Number of Hidden Nodes in ANN

Nodes	Training	Testing
1	0.616	0.616
2	0.616	0.616
3	0.616	0.616
4	0.801	0.759
5	0.829	0.781
6	**0.839**	**0.783**
7	0.844	0.782
8	0.849	0.779
9	0.849	0.774
10	0.852	0.777
11	0.857	0.779
12	0.851	0.772
13	0.861	0.777
14	0.837	0.778
15	0.853	0.777
16	0.829	0.777
17	0.861	0.776
18	0.835	0.779
19	0.845	0.781
20	0.841	0.776
21	0.838	0.779
22	0.835	0.775
23	0.838	0.779
24	0.831	0.780
25	0.831	0.774
26	0.853	0.775
27	0.837	0.781
28	0.839	0.773
29	0.832	0.781
30	0.852	0.776

Fig. 6 Trends of accuracy by changing nodes in the hidden layer.

4.2 SVM

We will use "e1071" package as SVM analysis tool in this paper. The kernel function we used is Gaussian Radial Basis Function (RBF) kernel, thus there are two important parameters have to be selected: *C* and γ (see Section 2.2). One can think of *C* as tunable parameter, that is, higher *C* corresponds to more importance on classifying all the training data correctly, lower *C* results in a more flexible hyper-plane that tries to minimize the margin error for each example. But if *C* is too much larger may appear over-fitting. So a proper cost *C* is one of the important parameters in SVM.

The optimal hyper-plane can be written as:

$$w = \sum_i \alpha_i y_i \exp(-\gamma \|x - x_i\|^2)$$

That is, the vector *w* is "support vectors". During the learning phase, the optimization adapts α_i to maximize the margin while retaining correct classification. There are two extreme cases for the choice of γ.

Firstly, imagine γ is very small, which means that the RBF kernel is very wide. Let us assume that it is so wide that the RBF kernel is still sufficiently positive for every data point of the dataset. This will probably give the optimizer a hard job since changing the value of a single α_i will change the decision function on all points because the kernel is too wide. The other extreme situation is when γ is large, which means that the RBF kernel is very narrow. When changing α_i for that point the decision function of SVM will basically change only for that point. This means that probably all training vectors will end up as support vectors. This is clearly not desirable. In summary, a certain value of γ determines a boundary for the RBF kernel in which the kernel will be larger than a certain value.

The R "e1071" package provides a tunable function for selecting both *C* and γ. In order to roughly find the suitable parameters, we set *C* as 10, 100, …, 1e+5. And γ is 1e-5, 1e-4, …, 0.1. The best result in our

experiment is C=10 and γ=0.01. Part of the result is shown in Table 3. The highest accuracy of testing is 0.816, which seems better than ANN. But this is only once 10-fold CV, which means it may "randomly" obtain such better result. Then we try to find more precise parameters C and γ among a certain range, which are, set C from 10 to 1000 and the step is 10 (i.e. 100 different *Cost*), while γ is 0.01, 0.001 and 0.0001. The testing method is still 10 times 10-fold CV in order to avoid "randomly" higher accuracy. Table 4 lists the maximum accuracy of each γ. Fig. 7 shows that the trends of accuracy by changing C and γ, respectively. We learn that the performance of γ = 0.001 is the best among these three curves, and the highest accuracy is around C = 740. Notice that when γ = 0.01 the highest result is at C = 10, which is the same result as previous experiment. Yet this curve decreases rapidly and it is not a satisfied parameter. Another γ = 0.0001 performs also stably but the accuracy is lower than γ = 0.001. Thus, we will use C = 740 and γ = 0.001 in the next section.

Table 3 Accuracy of different parameters in SVM.

C	***γ***	**Testing**
10	**0.01**	**0.816**
10	0.001	0.806
10	0.0001	0.787
100	0.01	0.808
100	0.001	0.811
100	0.0001	0.800
1000	0.01	0.773
1000	0.001	0.814
1000	0.0001	0.804
……	……	……

Table 4 Maximum accuracy of Gamma.

C	***γ***	**Maximum Accuracy**
10	0.01	0.812
740	**0.001**	**0.814**
820	0.0001	0.806

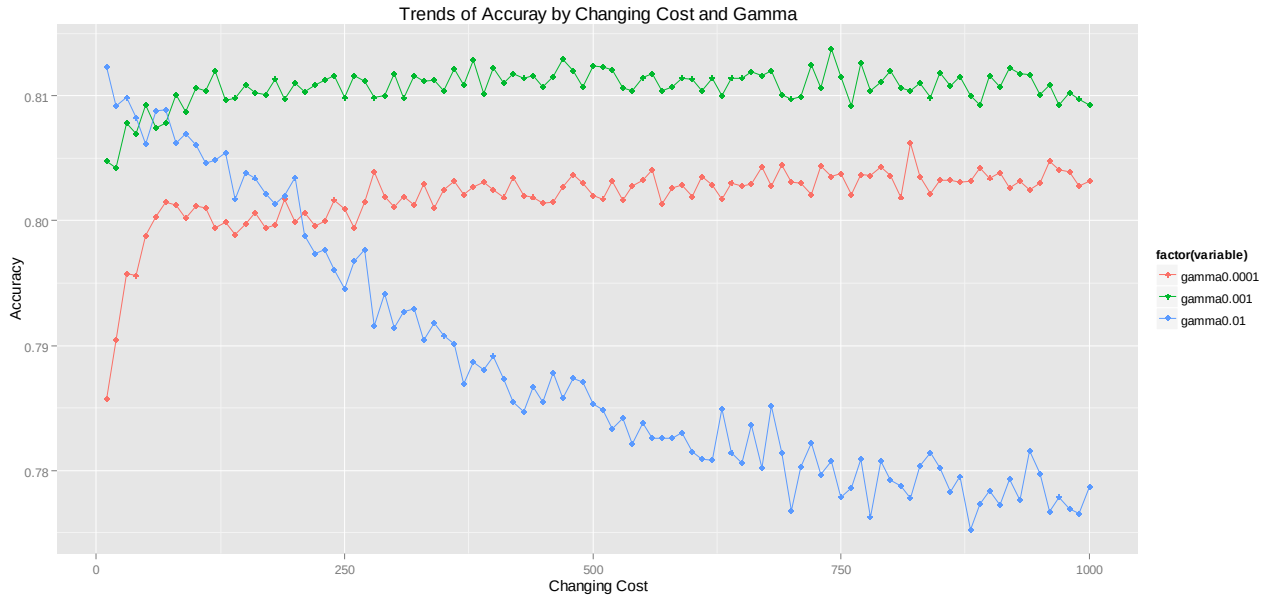

Fig. 7 Trends of accuracy by changing cost and Gamma.

4.3 Comparison

This section try to test the stability of two models, that is, a good algorithm should not only obtain high accuracy but also perform stable in process. We compare both algorithms by using 100 times 10-fold CV with the result shown in Table 5. We learn that SVM obtains the best result and its accuracy is around 0.803~0.816. More precisely, its lowest accuracy is better than the highest accuracy of ANN, whose accuracy is around 0.746~0.793. Fig. 8 shows the violin plot of this result. We can see ANN has a long tail which means its accuracy waves seriously. SVM is much more stable than ANN in this experiment.

Table 5 The result of 100 times 10-fold CV.

	Maximum	**Minimum**	**Median**
ANN	0.793	0.746	0.776
SVM	0.820	0.803	0.811

5 Conclusions

In this paper, we build $PM_{2.5}$ concentration level predictive models by using two popular machine learning algorithms, which are ANN and SVM. The dataset, which is from Hong Kong Central roadside station and King's Park station, includes 1443 rows and 15 columns by deleting all missing values. Based on all experiments, we have our conclusions as below.

(1) Either ANN or SVM needs to set proper parameters in the model. For ANN, we think the best method is to use multiple times 10-fold CV selecting the number of hidden nodes. While for SVM, it has to do single 10-fold CV to select an "approximate" range of C and γ (i.e. the default function in R "e1071" package). Then testing the grid value by using multiple times 10-fold CV, the final result may be different from the previous one but this is more accurate and stable.

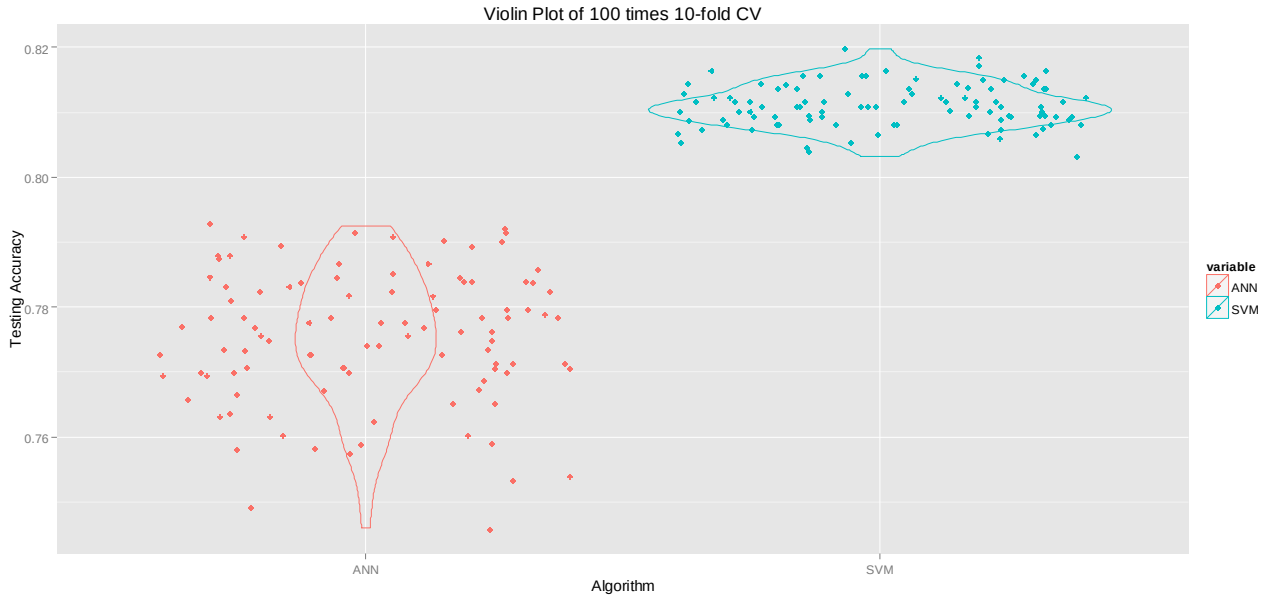

Fig. 8 Violin plot of 100 times 10-fold CV.

(2) According to 100 times 10-fold CV, the best result is from SVM. It not only obtains the higher accuracy but also performs much more stable than ANN. Thus, we think SVM is a better algorithm for $PM_{2.5}$ concentration predicting models.

Acknowledgement

Thanks to Hong Kong Environmental Protection Department (HKEPD) and Hong Kong Met-online provide all related data in this paper.

References

Burges CJ. 1998. A tutorial on support vector machines for pattern recognition. Data mining and Knowledge Discovery, 2(2): 121-167

EPA. 2006. Guideline for Reporting of Daily Air Quality - Air Quality Index (AQI). http://www.epa.gov/ttn/oarpg/t1/memoranda/rg701.pdf

Hagan MT, Demuth HB, Beale MH. 1996. Neural Network Design. Pws Pub, Boston, USA

Ihaka R, Gentleman R. 1996. R: A language for data analysis and graphics. Journal of computational and Graphical Statistics, 5(3): 299-314

Karatzoglou A, Meyer D, Hornik K. 2006. Support Vector Machines in R. Journal of Statistical Software, 15(i09)

Meyer D, Dimitriadou E, Hornik K, et al. 2013. "e1071" package, version 1.6.1. http://cran.r-project.org/web/packages/e1071/e1071.pdf

Nagendra SMS, Khare M. 2006. Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions. Ecological Modelling, 190(1-2): 99-115

Rojas R. 1996. Neural Networks: A Systematic Introduction. Springer

Ripley B. 2013. "nnet" package, version 7.3.6. http://cran.r-project.org/web/packages/nnet/nnet.pdf

Tan PN, Steinbach M, Kumar V. 2006. Introduction to Data Mining. Pearson Addison-Wesley, USA

Viotti P, Liuti G, Di Genova P. 2002. Atmospheric urban pollution: applications of an artificial neural network (ANN) to the city of Perugia. Ecological Modelling, 148(1): 27-46

WHO. 2011. Air Auality and Health. http://www.who.int/mediacentre/factsheets/fs313/en/index.html

Wickham H. 2013. "reshape2" package, version 1.2.2. http://cran.r-project.org/web/packages/reshape2/reshape2.pdf

Wickham H. 2013. "ggplot2" package, version 0.9.3.1. http://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf

Zhang WJ. 2010. Computational Ecology: Artificial Neural Networks and Their Applications. World Scientific, Singapore

Zhang WJ, Barrion AT. 2006. Function approximation and documentation of sampling data using artificial neural networks. Environmental Monitoring and Assessment, 122: 185-201

Zhang WJ, Zhang XY. 2008. Neural network modeling of survival dynamics of holometabolous insects: a case study. Ecological Modelling, 211: 433-443