Learning Air Pollution with Bidirectional LSTM RNN

Weitian Tong[1], Lixin Li[1], Xiaolu Zhou[2], Andrew Hamilton[1]

Emails: {wtong; lli; xzhou, ah09029}@georgiasouthern.edu

[1] Department of Computer Science,
[2] Department of Geology and Geography,
Georgia Southern University,
Statesboro, GA 30460, USA

Abstract

An accurate understanding of air pollutants in a continuous space-time domain by spatiotemporal interpolation is critical for meaningful assessment of the quantitative relationship between the public health and perennial environmental exposures. Existing spatiotemporal interpolation algorithms are usually based on unrealistic assumptions by restricting the interpolation models to the ones with explicit and simple mathematical descriptions, thus neglecting plenty of hidden yet critical influence factors. We developed an efficient deep-learning-based spatiotemporal interpolation algorithm which can generate more accurate estimation for air pollution on a large geographic scale and over a long time period. The experimental results demonstrate the efficacy and efficiency of our novel algorithm.

Keywords: Spatiotemporal interpolation; Air pollution; Deep learning; Bidirectional LSTM RNN.

1 Introduction

With the advancement of geo-spatial technologies, especially Geographic Information Systems (GIS), environmental exposure analysis has made significant progress. An accurate understanding of $PM_{2.5}$ in a continuous space-time domain is critical for meaningful assessment of the quantitative relationship between the risk of lung cancer and the concentrations of $PM_{2.5}$, which can help to identify, monitor, and evaluate interventions, such as establishment and enforcement of air quality standards, reduction of industry or auto emissions, and local community-based efforts. Since air pollution data is commonly recorded at scattered or localized sampling locations, it is often necessary to predict or estimate air pollution concentrations at new data points within the range of a discrete set of known data points, which is known as *interpolation* in numerical analysis.

Compared with the traditional spatial interpolation, an additional time dimension needs to be considered in the spatiotemporal interpolation. Besides, plenty of hidden factors such as meteorlogy, land use, traffic flow, human activities *etc.* also affect the concentration of air pollutants. What's more, most existing spatiotemporal interpolation methods restrict the interpolation models to explicit and simple mathematical descriptions. On the other hand, the real relationship between the air pollutants and the influential factors is unknown, and can be so complicated that no explicit mathematical model fits for it. Black box approaches are preferred in this situation as alternatives to traditional models for input–output mathematical models.

Deep learning algorithms can extract high-level, complex abstractions as data representations through a hierarchical learning process [10]. The hierarchical learning architecture is motivated by the artificial intelligence emulating the deep, layered learning process of the primary sensorial areas of the neocortex in the human brain, which automatically extracts features and abstractions from the underlying data [1, 2, 3]. Because the air quality process is inherently complicated, they are perfect candidates as black-box approaches to automatically consider the hidden factors and build the model for air pollution data. Among various deep learning methodologies, *deep recurrent neural network* (DRNN) [9] is particularly suitable for time series forecasting and modeling because it not only considers the current input but also takes into account a trace of previously acquired information via recurrent connections, which allows a direct processing of temporal dependencies and other hidden correlations.

Our main contribution of this paper is to develop an efficient DRNN-based spatiotemporal interpolation algorithm which allow for the generation of more accurate estimation of air pollution on a large geographic scale and over a long-time period. Section 2 briefly describes our methods. The data sets, experiments settings and results are shown in Section 3. We conclude in Section 4.

2　Problem Statement and Methods

2.1　Spatialtemporal Interpolation

Suppose there are n different monitoring stations $\{S_1, \ldots, S_n\}$ over an area $\mathcal{A}$. The observation from each station S_i at a specific time stamp t can be described as a tuple $\boldsymbol{x}_{i,t} = (log_i, lat_i, t, v_i)$, where v is the observed air pollutant concentration, log and lat desribe the longitude and latitude of the station S_i, respectively. Therefore, the input data set can be denoted as n time series, $\{ts_1, \ldots, ts_n\}$. Each time series $ts_i = (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,T})$ is the sequence of observed data from a single station S_i. Our target is to estimate v for any position in $\mathcal{A}$ at any time. In the spatial dimension, local air quality is usually influenced by the adjacent areas as air pollutants may disperse or transmit through the atmosphere with the wind. In the temporal dimension, historical states of the air pollutants can affect the current and future states. For instance, the air quality during the last day will effect the air quality during the next day. Another example is that the air quality tends to have similarities in the same season over recent years. In a word, the air quality data at given area has internal temporal correlation. Therefore, both spatial and temporal correlations should be taken into account when interpolating air pollutant's concentration spatiotemporally. Except all the above influence factors, lots of hidden factors such as meteorlogy, land use, traffic flow, human activities *etc.* can also trigger the change of air quality in both spatial and temporal dimensions and thus affect the concentration of air pollutants. Due to the lack of details in the data description, which only provides three influence factors, *i.e.* longitude, latitude and time, it is quite challenging to build a perfect mathematical model to estimate the concentration of air pollutants.

2.2　Bidirectional Long Short Term Memory RNN

Deep learning, also known as *artificial neural network* (ANN), enables the computer to extract high-level, complex abstractions as data representations through a hierarchical learning process. It can also avoid hand-crafted features that are usually expensive to create and require expert knowledge of the field. Typical architecture designs of deep learning includes *Convolutional Deep Neural Networks* (CDNN), *Deep Sparse Autoencoder* (DSA), *Deep Recurrent Neural Networks* (DRNN), *Multi-Layer Perceptions* (MLP), *Deep Restricted Boltzmann Machines* (DRBM), *etc.* [8]. Among them, *Recurrent neural network* (RNN) is particularly suitable for time series forecasting and modeling. RNN employs self-connected neurons to implement a cyclic structure in the network, which helps to "memorize" the historical input. In other words, RNN not only considers the current input but also takes into account a trace of previously acquired information via recurrent connections, which allows a direct processing of temporal dependencies.

3　Experiments

3.1　Data Set Description and Measure of Performance

To demonstrate the efficacy and efficiency of our new method, we explored the daily $PM_{2.5}$ data set in 2009 over the contiguous southeast region of the U.S., *i.e.* Georgia and Florida. This data set is measured by U.S. EPA's Air Quality System (AQS) monitoring sites and can be obtained from on the U.S. EPA website [5]. In this data set, each data entry is identified as a tuple (id, log, lat, t, v), where log and lat are the longitude and latitude coordinates of the monitoring stations, $t = (year, month, day)$ denotes the date when a $PM_{2.5}$ measurement is taken, and v is the measured $PM_{2.5}$ value. The original data set contains invalid entries, meaning no measurements are available at a particular site and on a particular day. After deleting all the invalid entries, there were $11,056$ daily measurements at 114 monitoring sites on all 365 days in the year 2009.

We adopted three performance indexes to evaluate the performance of the proposed model. These indexes

are the mean absolute error (MAE), the root-mean-square error (RMSE), and the mean absolute percentage error (MAPE); The previous two indexes are used to evaluate the absolute error, while the third one is used to measure the relative error. In other words, RMSE and MAE reflect the extremum effect and error range of the predicted values, and MAPE reflects the specificity of the average predicted value [4]. The optimal structure of our model was determined when the MAPE was minimized.

We adopted three performance indexes to evaluate the performance of the proposed model. These indexes are the mean absolute error (MAE), the root-mean-square error (RMSE), and the mean absolute percentage error (MAPE); The previous two indexes are used to evaluate the absolute error, while the third one is used to measure the relative error. In other words, RMSE and MAE reflect the extremum effect and error range of the predicted values, and MAPE reflects the specificity of the average predicted value [4]. The optimal structure of our model was determined when the MAPE was minimized.

3.2   Our Framework and Implementation Details

In the spatiotemporal interpolation, we assume that the local air quality is not only influenced by the adjacent areas but also correlated with the historical and future records at the adjacent areas. Our proposed framework employs the Bi-LSTM DRNN to capture the spatial and temporal correlations.

When training the neural networks, it is common to encounter the overfitting problem, which means that the performance on the training set is much better than the performance on the testing set. In other words, the model is over-trained such that it "memorizes" the training data but does not "learn" to generalize from trend. We adopted the *k-cross validation* [7] and *dropout strategy* [11] to address the overfitting problem and improve the robustness of the our method.

According to the *Tobler's First Law of Geography (Page 236)* [12], which states "all places are related, but nearby places are more related than distant places", the similarity of two locations should decrease with the increasing distance. We assume the effect from the k nearest neighbors dominates the influence from the other neighbors. In our experiment, k-d tree was adopted to store the data set as k-d tree is an efficient data structure for searching the nearest neighbors. It has been proven that the average time complexity of searching k nearest points is $O(\log n)$ [6], where n is the number of points in the space and k is a constant. We also assume the current air quality is highly correlated with the air pollution concentrations in the past t days and future t days.

Next, three experiments were designed to demonstrate the effectiveness of our method.

Experiment 1: Our first experiment is to explore how the geographical neighbors effect the air quality of the interested point. In this experiment, the dropout rate is set to be 0 and $t \in \{1, 2\}$. The proposed algorithm ran for $k \in \{1, 3, 5, 7\}$ and the statistic measurements were collected in Table 1. A visualization of the MAPE values is shown in Figure 1. From the Figure 1, we can observe that the MAPE tends to decrease when the algorithm takes into account of more neighbors of the interested point.

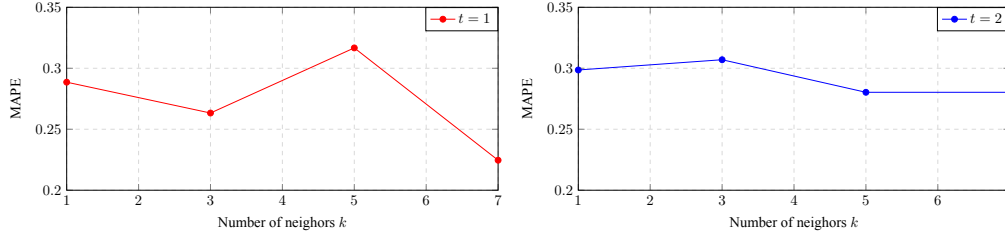Table 1: Measurements for our algorithm considering different number of neighbors.

(a) $t = 1$

k	MAE	RMSE	MAPE
1	2.0480	3.5485	0.2986
3	2.0383	3.5039	0.3070
5	1.8922	3.4682	0.2803
7	2.0542	3.8856	0.2803

(b) $t = 2$

k	MAE	RMSE	MAPE
1	2.0161	3.7223	0.2886
3	1.9338	3.8701	0.2633
5	2.3622	4.4404	0.3167
7	1.6352	3.5060	0.2246

Figure 1: MAPE for our algorithm considering different number of neighbors.

Experiment 2: Our second experiment is to investigate how the current state of the air pollutants affects the future states, and how many past states can be related with the current state. In this experiment, we fix the dropout rate as 0 and suppose the number of nearest neighbors is from $\{1, 2\}$. Our algorithm ran for $t \in \{1, 2, 4, 6, 8, 10\}$. The statistic measurements were collected in Table 2 and a visualization of the MAPE values is shown in Figure 2. From the Figure 2, we can observe that the MAPE tends to decrease when the algorithm takes into account of more past and future states.

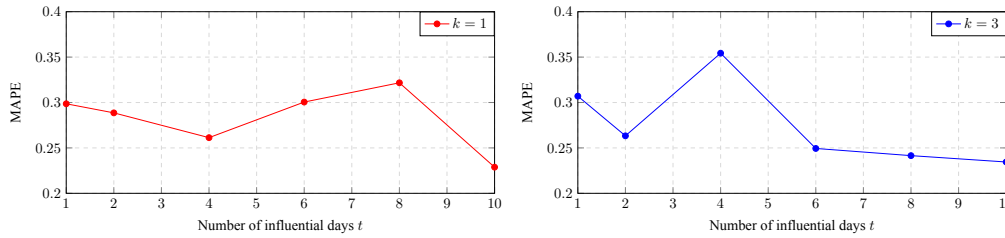Table 2: Measurements when considering different numbers of influential days.

(a) $k = 1$

t	MAE	RMSE	MAPE
1	2.0480	3.5485	0.2986
2	2.0161	3.7223	0.2886
4	1.8494	3.5499	0.2613
6	2.1739	4.0725	0.3005
8	2.2621	4.0321	0.3217
10	1.6479	3.4898	0.2288

(b) $k = 3$

t	MAE	RMSE	MAPE
1	2.0383	3.5039	0.3070
2	1.9338	3.8701	0.2633
4	2.4623	4.0582	0.3543
6	1.7969	3.6649	0.2494
8	2.1296	2.5744	0.2415
10	1.7349	3.6652	0.2345

Figure 2: MAPE when considering different numbers of influential days.

Experiment 3: Our last experiment is comparing the Bi-LSTM DRNN with the LSTM DRNN in order to explore whether the current state of the air pollutants is correlated with the future states. In this experiment, we fix the dropout rate as 0. Suppose the number of nearest neighbors is 3 and the number of the influencing days is from $\{1, 2\}$. The statistic measurements were collected in Table 3. We can observe that the MAPE for the Bi-LSTM DRNN is smaller than the MAPE for the LSTM DRNN. Therefore, we can claim that taking the future states into account helps to learn more accurate air pollution concentrations.

Table 3: Measurements for Bi-LSTM DRNN and LSTM DRNN.

(a) $k = 3, t = 1$

Model	MAE	RMSE	MAPE
Bi-LSTM	2.0383	3.5039	0.3070
LSTM	2.6837	4.0523	0.4160

(b) $k = 3, t = 2$

Model	MAE	RMSE	MAPE
Bi-LSTM	1.9338	3.8701	0.2633
LSTM	2.1659	4.0645	0.3018

4 Conclusion

In this paper, we proposed a novel deep neural network to interpolate the spatiotemporal data. Our method follows the *Tobler's First Law of Geography*: "all places are related, but nearby places are more related than distant places". We consider the "close" points in spatial and temporal dimensions at the same time when interpolating an interested point. In particular, both past and future information are taken into account. We employed the bidirectional LSTM RNN to split the neurons of a regular LSTM RNN into two directions in order to memorize the past and future information. Our experiments demonstrate the effectiveness of our method. To the best of our knowledge, it is the first time to apply the bidirectional LSTM RNN in the spatiotemporal interpolation. Our experiments were conducted on the daily $PM_{2.5}$ data over the Georgia and Florida. In the future, we are going to explore larger data sets such as hourly air pollution data across a large area, say the whole U.S. There are many other factors effecting the air quality. However, our current data set only contains longitude, latitude and time information. We will study more complicated data set which contains other influential factors such as temperature, wind speed, wind direction, *etc*. A larger and more complicate data set will inevitably make our proposed method less efficient as more computations are needed. In the future, we will explore how to speed up the proposed method by deploying the experiments on the cluster computing frameworks.

References

[1] I. Arel, D. C. Rose, and T. P. Karnowski. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE Computational Intelligence Magazine*, 5(4):13–18, 2010.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[3] Y. Bengio and Y. LeCun. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.

[4] Y. Chen, R. Shi, S. Shu, and W. Gao. Ensemble and enhanced PM10 concentration forecast model based on stepwise regression and wavelet analysis. *Atmospheric Environment*, 74(Supplement C):346 − 359, 2013.

[5] EPA. Air quality system (aqs). available online: http://www3.epa.gov/pm, 2016.

[6] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3:209–226, 1977.

[7] S. Geisser. *Predictive inference*, volume 55. CRC press, 1993.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[12] W. R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46:234–240, 1970.