HADOOP-BASED DISTRIBUTED SYSTEM FOR ONLINE PREDICTION OF AIR POLLUTION BASED ON SUPPORT VECTOR MACHINE

Z. Ghaemi[a]*, M. Farnaghi[b], A. Alimohammadi[b]

[a] Dep of Geodesy and Geomatics, K.N.Toosi University of Technology - zghaemi@mail.Kntu.ac.ir
[b] Dep of Geodesy and Geomatics, K.N.Toosi University of Technology - (Farnaghi, Alimoh_abb)@Kntu.ac.ir

KEY WORDS: Urban Air Pollution, Online Prediction, Big Data, Spatial Analysis, Distributed Computing, Support Vector Machine

ABSTRACT:

The critical impact of air pollution on human health and environment in one hand and the complexity of pollutant concentration behavior in the other hand lead the scientists to look for advance techniques for monitoring and predicting the urban air quality. Additionally, recent developments in data measurement techniques have led to collection of various types of data about air quality. Such data is extremely voluminous and to be useful it must be processed at high velocity. Due to the complexity of big data analysis especially for dynamic applications, online forecasting of pollutant concentration trends within a reasonable processing time is still an open problem. The purpose of this paper is to present an online forecasting approach based on Support Vector Machine (SVM) to predict the air quality one day in advance. In order to overcome the computational requirements for large-scale data analysis, distributed computing based on the Hadoop platform has been employed to leverage the processing power of multiple processing units. The MapReduce programming model is adopted for massive parallel processing in this study. Based on the online algorithm and Hadoop framework, an online forecasting system is designed to predict the air pollution of Tehran for the next 24 hours. The results have been assessed on the basis of Processing Time and Efficiency. Quite accurate predictions of air pollutant indicator levels within an acceptable processing time prove that the presented approach is very suitable to tackle large scale air pollution prediction problems.

1. INTRODUCTION

By developing data measurement techniques, the world has witnessed explosive, exponential growth of data generation in recent years which leads the data collection outpace data processing capabilities. Analyzing and processing vast amounts of information in a reasonable time is one of the biggest scientist's challenges in facing with such big amount of data (Zhai, Ong et al. 2014).

One of the crucial phenomenon requiring big data processing is the air pollution. Affecting by various factors and dynamic behavior of air pollution lead the scientists to encounter big amount of data for prediction of the air quality (Zheng, Liu et al. 2013). In the other hand, urban air pollution poses a significant threat to human health (García Nieto, Combarro et al. 2013) which turns it to a vital problem that requires immediate actions. So, there is a strong need to find efficient solutions to deal with different aspects of such complex big data problems especially data storage and real-time processing.

In the last years, numerous studies on the air-pollution problems using statistical methods have been published. Among them, machine learning algorithms allow scientists to solve hugely complex problems (García Nieto, Combarro et al. 2013). SVM, as a newly presented machine learning algorithm, has been proven as a powerful method to deal with complex and nonlinear phenomenon especially air pollution prediction (Ip, Vong et al. 2010, García Nieto, Combarro et al. 2013). But in real-life, machine learning algorithms including SVM, face with some limitations such as computational complexity and computation time (Çatak and Balaban 2013). SVM cannot deal with big streaming data as it requires to be retrained by adding each newly gathered training sample (Wang, Men et al. 2008).The computation time and storage space of SVM algorithm are very largely because of large scale kernel matrices which need to be recomputed several times (Bottou, Weston et al. 2004). In order to overcome the computational problems of conventional methods when facing with big and streaming data, some new techniques have been developed by researches. One useful solution is presenting online algorithms based on the conventional ones to speed up the process in dynamic applications (Wang, Men et al. 2008). Although the online approaches have overcome the deficiencies of conventional methods in dealing with streaming data and enhance the processing time, lack of required memory and demands of fast processing for big data problems still poses challenges.

Storing ever-increasing data of air pollution on a single machine and processing such big amount of data using single processor is a limiting factor in online applications. Distributed computing is a recent solution to overcome the large amount of memory and computation power requirements for training large scale dataset such as air pollution (Alham, Li et al. 2013). Dividing the workload between more than one processor can optimize the processing time to have higher performance especially for online applications. One of the distributed frameworks which provides scalable platform for

* Corresponding author

storage and analysis big data is Hadoop (Bhandarkar 2010). Hadoop consists of a processing part which is benefited MapReduce (Gao, Li et al.). MapReduce can process exceedingly large amounts of data by breaking down the overall problem into some parallel sub-problems (Krämer and Senner 2015). The integration of MapReduce and SVM has been examined in some studies to speed up the processing time.

In a recent work, a MapReduce-based distributed SVM ensemble algorithm has been utilized for image classification. Each SVM is trained in parallel using a cluster of computers. According to the results, the combination of ensemble SVMs with MapReduce achieved high accuracy and reduced the processing time significantly (Zhai, Ong et al. 2014) . In order to overcome the high computational time required for large datasets processing, Collobert, et al. split the dataset into several parts and have utilized a mixture of several SVMs. Each SVM trained one part of data. Finally, the results of all classifiers were integrated to fine final solution. The results proved that using parallel SVMs was much faster than training a single SVM (Collobert, Bengio et al. 2002). Based on the idea of exchanging support vectors among the connected networks, Lu, et al. proposed a distributed SVM algorithm. Several sites had considered in each one partition of data was located. Each subset of data was trained locally via SVM on each site and the SVs were exchanged with the other sites (Lu, Roychowdhury et al. 2008). Zanghirati and Zanni have decomposed the main SVM problem into smaller sub-problems. The outcome of sub-problems were then combined. The results showed that the proposed technique could be useful for training SVMs on multiprocessor systems (Zanghirati and Zanni 2003).

In addition to big data problems such as memory usage and computation time, there exists another problem in air pollution prediction which is insufficient number of air quality monitoring stations (Zheng, Liu et al. 2013). Lack of enough stations prevent scientists from monitoring spatial distribution of pollutant concentrations thoroughly. In order to overcome this shortcoming, geographical parameters and spatial analysis using Geographical Information System (GIS) can be employed to monitor the spatial distribution of air pollutants more accurately. In this study, along with pollutant concentrations and meteorological data some spatial parameters including local height, surface curvature and distance to the roads have been utilized.

The purpose of this study is to propose an online system to predict the air quality of Tehran one day in advance. First, an online algorithm based on SVM is developed to accurately predict the air quality. Following that, the parallel version of the online algorithm using MapReduce is provided, which can solve both the large-scale air pollution problem and the online prediction at the same time. The obtained results have shown that distributing the process can effectively enhance the processing time and provide much faster processing to analyze big air pollution data.

The rest of this paper is organized as follows. Section 2 briefly introduces SVM and online SVM techniques. It is followed by introducing MapReduce and Hadoop. Section 3 describes the study area and the design of the distributed online SVM algorithm in more detail. Section 4 evaluates the performance of MapReduce online SVM. Section 5 concludes the paper and points out some future work.

2. MATERIAL AND METODOLOGY

2.1 Support Vector Machine

A Support Vector Machine (SVM) is a binary classifier which is developed by Vapnik (Vapnik 1998). The goal of SVM is to perform classification by constructing the optimal hyperplanes that maximize the distance between the two classes (Wang, Men et al. 2008). The input samples are called vectors and the vectors near the hyperplane are the support vectors (SV). In nonlinear cases, kernel functions are employed to map the original data into a higher dimensional feature space through a linear or non-linear mapping (Haifeng, Jun et al. 2009).

Given a training dataset of $\{x_i ; y_i\}$, i = 1, 2,…, N with input data $x_i \in R^n$ and corresponding labels $y_i \in \{+1, -1\}$. The classification decision function is represented in equation (1) (Burges 1998):

$$f(x) = sign\{\sum_{i=1}^{l} \alpha_i y_i k(x_i, x_j) + b\} \qquad (1)$$

Subject to the constraints: $\sum_{i=1}^{l} \alpha_i y_i = 0$ and $0 \le \alpha_i \le C \quad \forall_i$

C is the regularization constant which highly influence the SVM results by controlling the tradeoff between the errors and maximizing the distance between classes (Yeganeh, Motlagh et al. 2012). The coefficients α_i are obtained by solving the following problem:

$$\text{Maximize } \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{i} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i x_j) \qquad (2)$$

Samples with non-zero coefficients are known as support vectors (Wang 2005).

Although SVM is proven as an appropriate algorithm to address complex and nonlinear problems (Lu and Wang 2005, Juhos, Makra et al. 2008), it is unable to overcome big streaming data. For some new coming training data, SVM needs to always retrain the algorithm using all training data (including old and new coming data) (Wang, Men et al. 2008). In order to deal with this problem, an online learning algorithm based on SVM (LaSVM) is utilized in this study, which can meet the requirement of online learning to update the existing algorithm (Bordes, Ertekin et al. 2005). LaSVM includes two more steps than typical SVM named PROCESS and REPROCESS. By adding each new training sample, PROCESS checks if it can be a new support vector. REPROCESS phase removes the non-support vectors from the current dataset. Then the coefficient and consequently the separating hyperplanes will be updated (Bordes, Ertekin et al. 2005). In this case, LaSVM utilizes only the extracted support vectors and the newly inserted sample for training rather than using all existing training data. This process speeds up the training step and reduces the required memory.

2.2 MapReduce Model

MapReduce, introduced by Google (Lämmel 2008), provides parallel computing power to process parallelizable problems across huge datasets. Each MapReduce process consists of two phases: the Map phase and the Reduce phase which their functions are defined by user (Alham, Li et al. 2013).

The Mapper receives input data and transforms each element to an output in a parallel manner. The output from the Mapper is a list of (key, value) pairs. Reduce phase reads key-value generated by Mappers and process the values using the reduce function. In other words, the reduce function accepts an intermediate key and a set of values for that key. It merges together these values to form a possibly smaller set of values. Finally, the reducer send the obtained results to the output (Cary, Sun et al. 2009).

A map/reduce job consists of some independent map tasks and some independent reduce tasks. The number of Mappers determines the level of parallelism. The input data is divided into some independent chunks which are processed by the map tasks in a parallel manner. The outputs of the Mappers are then sorted by the MapReduce framework and sent to the reduce tasks as inputs. Both the input and the output of the job are stored in a file system (Dean and Ghemawat 2008).

2.3 Hadoop

Hadoop is an open source software platform which can provide fast and reliable analysis to handle large data sets (Loebman, Nunley et al. 2009). Hadoop includes many subproject but the two main ones are Hadoop Distributed File System (HDFS) and MapReduce. MapReduce is applied to process data and HDFS is utilized for storing data (Jorgensen, Rowland-Jones et al. 2014).

MapReduce integrates tightly with HDFS. MapReduce tasks run directly on the HDFS nodes that hold the required data. When running a MapReduce job, some transformations from the source of data to the result data set are provided. The input data is fed to the map function and the result of map is sent to a reduce function. Hadoop's MapReduce jobs manage the process of how to apply these transformations to the data across the different nodes in parallel (Turkington 2013).

3. DEVELOPMENT

3.1 Case Study and Dataset

Tehran, the capital of Iran, has been chosen as case study as it suffers from severe air pollution especially in winters. The prediction is performed based on the observed data of NO_2, CO, SO_2, PM_{10} and O_3, for the past years and meteorological parameters including wind speed, temperature, relative humidity, pressure and cloud cover. Although this data is gathered dynamically, its availability is limited only at monitoring stations. Whereas the spatial distribution beyond these locations still remains uncertain as it is influenced by geographical factors. To address the effect of spatial autocorrelation, along with pollution concentrations and meteorological factors some geographical parameters such as distance to the roads, local height and topography are employed to model the effects of spatial distribution of air pollution. Figure 1 demonstrates the study area and distribution of air pollution stations.

3.2 Parallelization of LaSVM based on MapReduce

This work involves two steps: first, training a data set using LaSVM to obtain a model and second, using the model to predict the air pollution for the next 24 hours. In order to share the workload, training and prediction phases are performed on two separated servers. To make LaSVM algorithm has a good scalability and run faster when the input data set is very large, a parallelization method for the LaSVM classifier is proposed and implement based on Hadoop's MapReduce framework.

The training phase is performed on the server on which Hadoop is running. A MapReduce program is designed to accelerate solving

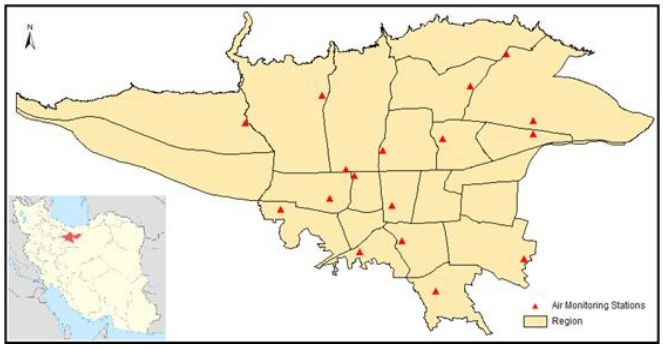

Figure 1. The study area and distribution of air pollution monitoring stations

LaSVM. In order to implement parallel LaSVM, the training data are retrieved from database in a form of primary key-value pairs. In this study, the keys are date and ID of pollution monitoring station. Values are array which includes label of air quality along with meteorology, pollution and geographical data. The value arrays are employed for training LaSVM. After training, the SVs are extracted and arranged as second key-value pairs.

In next step, the extracted SVs in the form of key-value from Mapper are sent to the reduce phase. The reduce task aggregates all SVs. The final SVs from reduce phase are saved in a file. When new training sample is available, the saved SVs along with the newly entered training samples are returned to the map phase to be used in training. The saved SVs are also sent to the second server as a request for air quality prediction is made. An overview of the idea is presented in Figure 2.

4. RESULTS

The algorithm for online prediction of air pollution is built on LaSVM and implemented using the Hadoop implementation of MapReduce and the Java programming language. Gaussian kernel was used for LaSVM. Optimal parameters (C, kernel specific parameter (γ)) were estimated by cross-validation method and set to 2 and 0.001 respectively.

The designed forecasting system can receive data in sequence, train LaSVM dynamically and predict the air quality one day in advance. Transferring data from one step to another is demonstrated in Figure 3. In order to test the online algorithm and assess performance of the system, one year data was set as testing data. The efficiency of the online algorithm is evaluated based on *Accuracy*, *RMSE* and *RSquered* estimators. The *Accuracy* of 0.7, *RMSE* of 0.6 and *RSquered* 0f 0.8 proved the feasibility of the online algorithm. Also, using the MapReduce model for distributing the process sped up the processing time. Therefore, the system could not only achieve promising results, but also possess a good prediction performance as well.

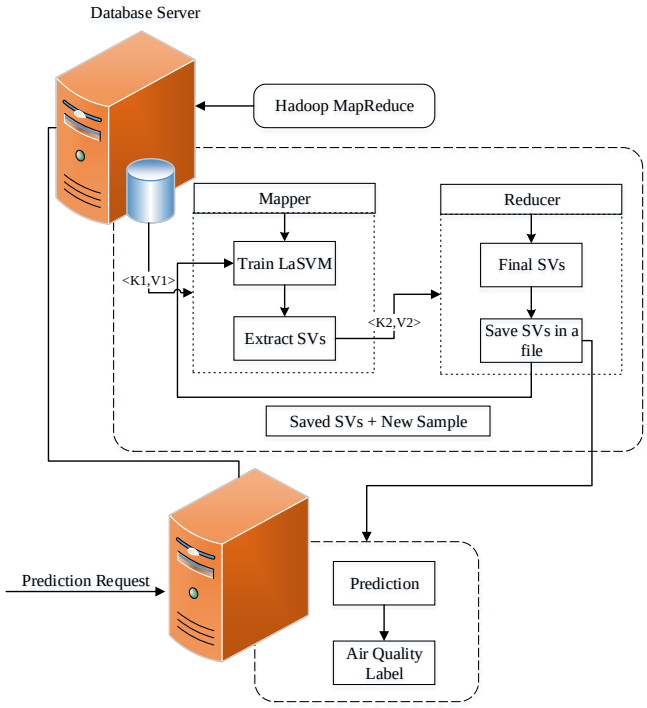

Figure 2. An overview of MapReduce based system for online air pollution prediction

5. CONCLUSION

To overcome the most challenging problems of big air pollution data, memory usage and computation time, a parallel LaSVM algorithm was developed based on Hadoop-MapReduce in this study. The proposed system had been utilized for online air pollution prediction of Tehran one day in advance. Geographical data was also applied to deal with coverage limitation of air pollution monitoring stations. The obtained results seemed extremely encouraging and suggested that the proposed system could allow training SVM-like models for very large scale data set in a reasonable time. Splitting the whole dataset over data nodes and training each subset of data in parallel will be the future research work. It is also planned to compare the obtained results by employing Hadoop and MapReduce programming with the online algorithm that had been executed on a single machine to evaluate the feasibility of the online distributed system.

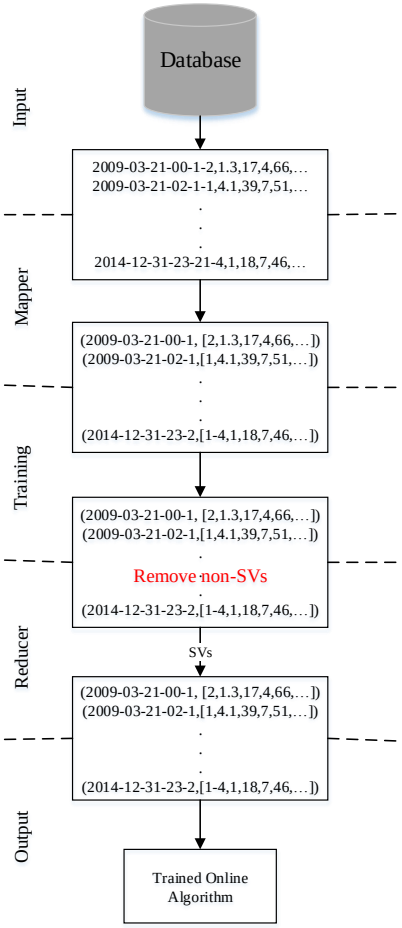

Figure 3. Transfer of data within map and reduce phases

REFERENCES

Alham, N. K., M. Li, Y. Liu and M. Qi (2013). "A MapReduce-based distributed SVM ensemble for scalable image classification and annotation." Computers & Mathematics with Applications 66(10): 1920-1934.

Bhandarkar, M. (2010). MapReduce programming with apache Hadoop. Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, IEEE.

Bordes, A., S. Ertekin, J. Weston and L. Bottou (2005). "Fast kernel classifiers with online and active learning." The Journal of Machine Learning Research 6: 1579-1619.

Bottou, L., J. Weston and G. H. Bakir (2004). Breaking SVM complexity with cross-training. Advances in neural information processing systems.

Burges, C. J. (1998). "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2(2): 121-167.

Cary, A., Z. Sun, V. Hristidis and N. Rishe (2009). Experiences on processing spatial data with mapreduce. Scientific and statistical database management, Springer.

Çatak, F. Ö. and M. E. Balaban (2013). "A MapReduce based distributed SVM algorithm for binary classification." Turkish Journal of Electrical Engineering & Computer Science.

Collobert, R., S. Bengio and Y. Bengio (2002). "A parallel mixture of SVMs for very large scale problems." Neural computation 14(5): 1105-1114.

Dean, J. and S. Ghemawat (2008). "MapReduce: simplified data processing on large clusters." Communications of the ACM 51(1): 107-113.

Gao, S., L. Li, W. Li, K. Janowicz and Y. Zhang "Constructing gazetteers from volunteered Big Geo-Data based on Hadoop." Computers, Environment and Urban Systems.

García Nieto, P. J., E. F. Combarro, J. J. del Coz Díaz and E. Montañés (2013). "A SVM-based regression model to study the air quality at local scale in Oviedo urban area (Northern Spain): A case study." Applied Mathematics and Computation 219(17): 8923-8937.

Haifeng, W., F. Jun and G. Chong (2009). Research on the assessment for air environment quality based on Support Vector Machine. Control and Decision Conference, 2009. CCDC'09. Chinese, IEEE.

Ip, W., C. Vong, J. Yang and P. Wong (2010). Forecasting daily ambient air pollution based on least squares support vector machines. Information and Automation (ICIA), 2010 IEEE International Conference on, IEEE.

Jorgensen, A., J. Rowland-Jones, J. Welch, D. Clark, C. Price and B. Mitchell (2014). Microsoft Big Data Solutions, John Wiley & Sons.

Juhos, I., L. Makra and B. Tóth (2008). "Forecasting of traffic origin NO and NO 2 concentrations by support vector machines and neural networks using principal component analysis." Simulation Modelling Practice and Theory 16(9): 1488-1502.

Krämer, M. and I. Senner (2015). "A modular software architecture for processing of big geospatial data in the cloud." Computers & Graphics 49: 69-81.

Lämmel, R. (2008). "Google's MapReduce programming model—Revisited." Science of computer programming 70(1): 1-30.

Loebman, S., D. Nunley, Y. Kwon, B. Howe, M. Balazinska and J. P. Gardner (2009). Analyzing massive astrophysical datasets: Can Pig/Hadoop or a relational DBMS help? Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, IEEE.

Lu, W.-Z. and W.-J. Wang (2005). "Potential assessment of the "support vector machine" method in forecasting ambient air pollutant trends." Chemosphere 59(5): 693-701.

Lu, Y., V. Roychowdhury and L. Vandenberghe (2008). "Distributed parallel support vector machines in strongly connected networks." Neural Networks, IEEE Transactions on 19(7): 1167-1178.

Turkington, G. (2013). Hadoop Beginner's Guide, Packt Publishing Ltd.

Vapnik, V. N. (1998). Statistical learning theory, Wiley New York.

Wang, L. (2005). Support Vector Machines: theory and applications, Springer.

Wang, W., C. Men and W. Lu (2008). "Online prediction model based on support vector machine." Neurocomputing 71(4): 550-558.

Yeganeh, B., M. Motlagh, Y. Rashidi and H. Kamalan (2012). "Prediction of CO concentrations based on a hybrid Partial Least Square and Support Vector Machine model." Atmospheric Environment 55: 357-365.

Zanghirati, G. and L. Zanni (2003). "A parallel solver for large quadratic programs in training support vector machines." Parallel computing 29(4): 535-551.

Zhai, Y., Y.-S. Ong and I. W. Tsang (2014). "The Emerging" Big Dimensionality"." Computational Intelligence Magazine, IEEE 9(3): 14-26.

Zheng, Y., F. Liu and H.-P. Hsieh (2013). U-Air: When urban air quality inference meets big data. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.