World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:2, 2016

Comparison of Different *k*-NN Models for Speed Prediction in an Urban Traffic Network

Seyoung Kim, Jeongmin Kim, Kwang Ryel Ryu

Open Science Index, Computer and Information Engineering Vol:10, No:2, 2016 publications.waset.org/10006804.pdf

***Abstract*—**A database that records average traffic speeds measured at five-minute intervals for all the links in the traffic network of a metropolitan city. While learning from this data the models that can predict future traffic speed would be beneficial for the applications such as the car navigation system, building predictive models for every link becomes a nontrivial job if the number of links in a given network is huge. An advantage of adopting k-nearest neighbor (*k*-NN) as predictive models is that it does not require any explicit model building. Instead, *k*-NN takes a long time to make a prediction because it needs to search for the k-nearest neighbors in the database at prediction time. In this paper, we investigate how much we can speed up *k*-NN in making traffic speed predictions by reducing the amount of data to be searched for without a significant sacrifice of prediction accuracy. The rationale behind this is that we had a better look at only the recent data because the traffic patterns not only repeat daily or weekly but also change over time. In our experiments, we build several different *k*-NN models employing different sets of features which are the current and past traffic speeds of the target link and the neighbor links in its up/down-stream. The performances of these models are compared by measuring the average prediction accuracy and the average time taken to make a prediction using various amounts of data.

***Keywords*—**Big data, *k*-NN, machine learning, traffic speed prediction.

I. Introduction

MOST traveling time prediction systems, like car navigation systems and transport information services that are provided by public institutions help users to find the quickest route to a destination. However, the advertised traveling time is predicted by using only the current rather than the future traffic speeds. The traffic situation can rapidly shift between normal and congested states, especially in transitional periods like rush hours. This is not sufficiently taken into account when the traveling time is predicted with only the current state of traffic. If we could predict future traffic speeds, we might be able to predict traveling time more precisely.

In this paper, we compare the performance of several different traffic speed prediction models. For speed prediction, we use a traffic-speed database that continuously updates itself by recording average traffic speed at every link in a traffic network at 5 min intervals. With such data, one can build speed prediction models of different future horizons for each link by using a variety of learning algorithms [1]-[5]. However, the number of prediction models to be learned is enormous if the size of the traffic network is large. Maintaining that many models are also a nontrivial burden because the models should be regularly updated by relearning from an ever-changing traffic database to keep up with varying traffic trends. Prediction by the *k*-nearest neighbor (*k*-NN) method, in contrast, does not require any model building. Instead, *k*-NN needs to look up training instances at the time of prediction to sort out the *k* most similar instances based on which to make a prediction. Notice that maintaining training instances will be much easier than maintaining models, while the computation by *k*-NN can be much heavier than those by other methods using prebuilt models.

The required time for the prediction can be reduced by using not all observed speed data but the traffic data of only the most recent few days or weeks, as traffic patterns repeat daily or weekly. In this paper, we compare the required time costs of making traffic speed predictions and the prediction accuracy between *k*-NN models by reducing the amount of data to be searched for. To further save the time required for prediction, we consider the maintenance of a related data table to make the training example set for the *k*-NN models. Our prediction system includes two kinds of tables. One is a data table of entries that are shifted whenever newly observed traffic data are added. The other is an index table of each entry that is an index of the data table, which shows the location of each value required to make the training instances and target instances.

The rest of the paper is organized as follows. Section II introduces the method for predicting future traffic speeds, which is originated from the method by Rasyidi [4]. Section III describes the method of managing an observed traffic data table. Section IV reports the performance of prediction models from the point of view of not only the average prediction accuracy but also the average time taken to make a prediction depending on the amount of data for prediction. Finally, Section V gives a summary and some concluding remarks.

II. Prediction of Future Traffic Speeds

A. Speed Prediction Model

A speed prediction model is a relation between one or more values that have an effect on traffic speed at the prediction time and predictive speed. Let t be the current time; $\mathbf{X}_t$ be the vector of the values that are used for predicting speed, i.e., the vector of relevant features; and the relation f^* be the speed prediction model. Then $f^*(\mathbf{X}_t)$ is the predictive speed at the time $(t+m)$, denoted by $y^*(t+m)$.

We can obtain various types of f^* by using different learning methods or relevant features. The following parts introduce the features and the learning algorithm.

Seyoung Kim, Jeongmin Kim, and Kwang Ryel Ryu are with the Department of Electrical and Computer Engineering, Pusan National University (phone: +82-51-510-3645, +82-51-510-3531, +82-51-510-2453; e-mail: birdzero@pusan.ac.kr, jeongminkim.islab@gmail.ac.kr, krryu @pusan.ac.kr).

Open Science Index, Computer and Information Engineering Vol:10, No:2, 2016 publications.waset.org/10006804.pdf

B. Features

To predict with high accuracy, it is important to select features that are related to the target predictive value. Basically, the future traffic condition of the target link depends on the current and recent traffic conditions. The traffic conditions of neighbor links connected to the target link are also related to the future traffic state of the target link. For example, if neighbor links toward which the vehicle is flowing—i.e., upstream links—are congested, downstream links are also gradually congested. Similarly, if the traffic congestion of downstream links is relaxed, that of upstream links will also be gradually resolved.

Considering these characteristics of traffic conditions, Rasyidi selected not only the current and recent observed speed data of the target link but also the observed speed data of neighbor links of the target link as features [4], [10]. The neighbor links of the target link are defined by the depth metric. Therefore, the number of neighbor links differs depending on the target link. An arrow in Fig. 1 signifies one link of the road network, and the direction of the arrows indicates the flow of vehicles. The dashed arrows are neighbor links of the target link, which are in depth 2 from the target link.

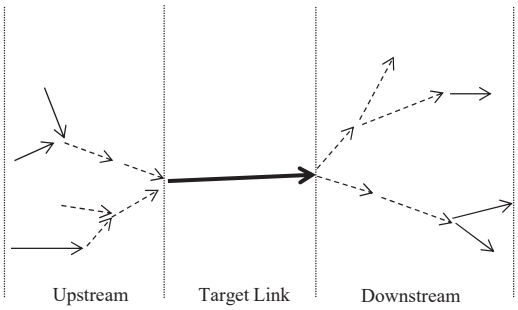

Fig. 1 A sample road network

Another characteristic is that the traffic situation depends on the time of day. The traffic congestion on weekdays usually occurs during rush hours. Considering this point attributes related to time, such as day of the month, the day of the week, hour, minute, and so on, can be used as features.

Some of the gathered features may be able to degrade the prediction performance because they are redundant or useless. Therefore, we need to select features that optimize the performance in advance. There are two typical kinds of feature subset selection methods: wrapper and filter methods [7], [8]. The wrapper method usually gives a better performing set of features than the filter method [4]. However, in the wrapper method, the set of features is evaluated with a machine learning algorithm that is employed to build a model. To evaluate each set of features, we need to build each model using the set to be evaluated. Thus, feature subset selection using the wrapper method requires heavy computation time. In the filter method, on the other hand, features are selected with just general characteristics of the data regardless of the model learned with the machine learning algorithm.

In this paper, we compare the performance of various prediction models. Some of them are learned with features that consist of the current and recent traffic data of the target link and others with features that include the traffic situations of the neighbor links of the target link and other factors related to time. We use correlation-based feature selection (CFS), a filter method for feature subset selection, in order to reduce the burden of the cost of feature subset selection [9].

C. Prediction with k-Nearest Neighbors Algorithm

Prediction by the *k*-nearest neighbor (*k*-NN) method does not require any model building. Instead, a model learned with the *k*-nearest neighbor (*k*-NN) algorithm has a set of training instances because it is a kind of nonparametric instance-based learning algorithm. It predicts the target label value by measuring the distances between the training instances and the target instance and taking the average or weighted average of the label values of the training instances, which are selected as the *k* most similar instances based on the distance measure [6].

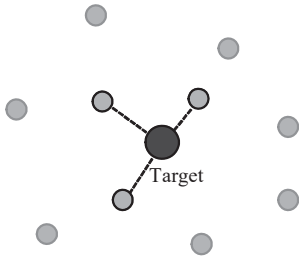

Fig. 2 An example of *k*-NN with $k = 3$

According to Rasyidi, the performance of the model learned with an ensemble of model trees is better than that of one with the *k*-NN algorithm. However, building a model with an ensemble of model trees is expensive, and it is impossible to update the model whenever newly observed traffic data are added. On the other hand, to update a model based on the *k*-NN method means just to add the new instances into the set of training instances. Therefore, the *k*-NN model is easier to maintain than a prebuilt parametric model.

Even so, because of adding new traffic data and increasing the number of training instances continuously over time, the cost of calculating the distance between the target instance and the training instances is increasing. Namely, the required cost of prediction with the *k*-NN-based model is expensive. Therefore, we need to reduce the number of training instances that are distances calculated with the target instance in order to decrease the time necessary for prediction without a significant sacrifice of prediction accuracy.

In this paper, we propose a data reduction method and compare the accuracy of prediction. As previously mentioned, the state of traffic shows a cyclic tendency over time. Considering this trait, we use data generated in a time zone near the prediction time, e.g., in this paper, we focus on only the most recent few days or weeks from the prediction time to 30 minutes before and after.

III. Data Management

When models based on *k*-NN predict the future traffic speed,

Open Science Index, Computer and Information Engineering Vol:10, No:2, 2016 publications.waset.org/10006804.pdf

time is required not only to make a prediction but also to read the features and label the values of the training instances. The cost of reading training examples from text files every time is too expensive. In contrast, keeping training examples in the main memory instead of reading them from text files requires a significant amount of memory and results in raising the cost of updating the set of training examples for each link and horizon if new traffic data are added over time. Therefore, this paper suggests a data table that stores data only for the required period and provides a method of updating the data table.

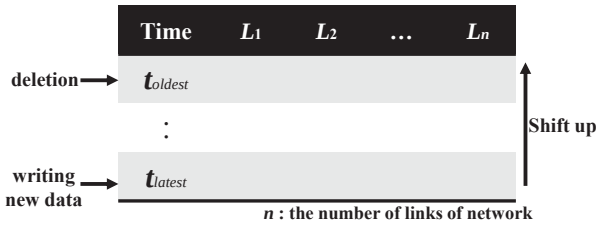

Fig. 3 Proposed data table and updating method

For updating, the data table needs to be shifted up and new data are written whenever they are generated. Then the oldest data must be deleted from the table. Namely, the location of the entry that we need to read is constant even as time passes. Therefore, we can make an index table of entries that refer to the feature values in the data table. Each row of the index table means the location of feature values of one training instance.

IV. Experimental Results

We experiment with the traffic speed data observed from March to June 2014, which are provided by the Transportation Information Service Center of Busan Metropolitan City. Traffic data from May 2014 are used for validation, and the prediction models are tested with traffic data from June 2014.

To evaluate the performance of the proposed method, we select 10 paths that cover almost all main roads of the city and measure the mean absolute percentage error (MAPE).

The performance of the *k*-NN models dominates that of prediction based on the current speed at all times (07:00~20:00); in particular, the prediction of the *k*-NN models is more accurate during transition times, i.e., rush hours (07:30~09:00, 17:30~19:00), especially morning rush hours. However, the required cost of prediction with *k*-NN models is much more expensive. Table I shows that prediction with *k*-NN models requires much more time cost, from almost 3 times to 80 times more.

TABLE I
Comparison of Required Time for Prediction

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of Week	Week	
3	2.61	28.14	2.71	32.41	
5	4.14	56.23	7.11	53.34	1
8	6.71	52.94	11.39	83.06	

TABLE II
Performance of Prediction in All Day

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of week	Week	
3	4.23	3.98	4.19	3.91	
5	3.93	3.90	3.89	3.70	4.06
8	3.83	3.78	3.76	**3.63**	

TABLE III
Performance of Prediction in Week Day

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of Week	Week	
3	4.22	3.81	4.17	3.65	
5	3.89	3.74	3.82	3.43	4.09
8	3.80	3.61	3.71	**3.40**	

TABLE IV
Performance of Prediction in Weekend

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of Week	Week	
3	4.26	4.38	4.26	4.54	
5	4.03	4.26	4.04	4.30	3.97
8	3.92	4.17	**3.87**	4.17	

TABLE V
Performance of Prediction in Rush Hours

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of Week	Week	
3	5.01	4.64	4.84	4.26	
5	4.63	4.48	4.48	3.93	4.89
8	4.38	4.34	4.26	**3.83**	

TABLE VI
Performance of Prediction in Morning Rush Hours

Number of weeks	pattern		neighbor		current
	day of the week	week	day of the week	week	
3	5.05	4.49	4.87	3.90	
5	4.67	4.28	4.52	3.57	5.09
8	4.35	4.02	4.24	**3.45**	

TABLE VII
Performance of Prediction in Evening Rush Hours

Number of weeks	Pattern		Neighbor		Current
	Day of Week	Week	Day of Week	Week	
3	4.97	4.80	4.82	4.61	
5	4.59	4.67	4.43	4.30	4.70
8	4.41	4.66	4.27	**4.22**	

V. Conclusion

In this paper, we compare the performance of various traffic speed prediction models and prediction based on current speed. The *k*-nearest neighbor algorithm (*k*-NN) is employed for building models because it requires little time to build models and the models learned with *k*-NN are updated easily. However, the *k*-NN algorithm has a drawback in that it requires expensive prediction time cost because it searches for the k-nearest neighbors in the database at the prediction time. Thereby, we proposed a method of selecting the training instances and a method of managing the data table to decrease the amount of

time required for prediction.

Our experimental results show that the prediction performance of *k*-NN models is better than that of the current speed-based prediction during transition times, especially morning rush hours. Use of features such as neighbor links in addition to a target link also improves performance in comparison to a model that consists of just a target link. However; the former takes a longer time to make a prediction than the latter, because the number of features is larger. Therefore, we need to consider the trade-off between prediction accuracy and time cost when using prediction models.

REFERENCES

[1] X. Zhang and J. A. Rice, "Short-Term Travel Time Prediction Using A time-Varying Coefficient Linear Model," Transp. Res. C, vol. 11, no. 3, pp. 187-210, 2003.

[2] L. Vanajakshi and L. R. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," in Intelligent Vehicles Symposium, 2004 IEEE, 2004, pp. 194-199.

[3] M. A. Rasyidi, J. Kim, and K. R. Ryu, "Short-term Prediction of Vehicle Speed on Main City Roads using the k-Nearest Neighbor Algorithm," J. Intell. Inf. Syst., vol. 20, no. 1, pp. 121-131, 2014.

[4] M. A. Rasyidi, and K. R. Ryu, "Short-Term Speed Prediction on Urban Highways by Ensemble Learning with Feature Subset Selection," Database Systems for Advanced Applications, Springer Berlin Heidelberg, 2014, pp. 46-60.

[5] H. Sun, H. X. Liu, H. Xiao, and B. Ran, "Short Term Traffic Forecasting Using the Local Linear Regression Model," UC Irvine Cent. Traffic Simul. Stud., 2002

[6] S. J. Russell, and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed, Pearson Education, 2010.

[7] R. Kohavi, and G. H. John, "Wrappers for feature subset selection," Artificial intelligence, vol. 97, no. 1, 1997, pp. 273-324.

[8] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," Machine Learning: Proceedings of the Eleventh International Conference, 1994, pp. 121-129.

[9] M. A. Hall, Correlation-based Feature Selection for Machine Learning, Doctoral dissertation, The University of Waikato, 1999.

[10] M. A. Rasyidi, and K. R. Ryu, "Comparison of Traffic Speed and Travel Time Predictions on Urban Traffic Network," Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International conference on. IEEE, 2014, pp. 373-380.

Open Science Index, Computer and Information Engineering Vol:10, No:2, 2016 publications.waset.org/10006804.pdf