Bus arrival time prediction with real-time and historic data

Haitao Xu[1] · **Jing Ying**[1]

Received: 20 March 2017 / Revised: 18 June 2017 / Accepted: 20 June 2017 / Published online: 28 June 2017

Abstract Provision of accurate bus arrival information is vital to passengers for reducing their anxieties and waiting times at bus stop. GPS-equipped buses can be regarded as mobile sensors probing traffic flows on road surfaces. In this paper, we present an approach that predicts bus arrival time in terms of the knowledge learned from a large number of historical bus GPS trajectories. In our approach, we build time-dependent path-section graph, where a path-section is a road segment between two adjacent bus stops, to model the properties of dynamic road networks. Then, a clustering approach is designed to estimate the distribution of travel time on each path-section in different time slots. Finally, bus arrival time is predicted based on the path-section graph and real-time GPS information. Using a real-world trajectory dataset generated by 1000 buses in a period of 2 months, a bus arrival time prediction system is built. Then we evaluate the system with extensive experiments and realistic evaluations. Experiments show that our method is close to the actual value and better than some typical algorithms.

Keywords Bus arrival time prediction · Bus GPS trajectories · Path-section graph

1 Introduction

Trends over the past 10 years demonstrate that urban traffic congestion is increasing. Public transport, especially the bus transport, has been recognized as a valid method to alleviate traffic congestion. Public transport also can reduce the use of private car and fuel consumption. In order to attract more people to shift from personal vehicles to public transport, service quality should be improved. Nowadays most public transportation systems provide their timetables on the web or on the board of bus stops. Furthermore, with the development of smart phones, more and more apps are used for bus schedules. However, the provided information is always limited and not timely updated. Waiting a long time at bus stops may drive passengers away and make them reluctant to take buses. Meanwhile, accurate arrival time of the next bus will allow passengers to take alternative transport choices instead and improve their satisfaction. So providing accurate information about bus arrival to passengers is of great importance.

Bus arrivals at stops in road networks are stochastic because travel time on roads, dwell time at stops, and delays at intersections fluctuate spatially and temporally. And many unpredictable factors such as traffic conditions or harsh weather may cause buses to delay. Thus, developing such a model that can adapt to time varying traffic and demand conditions is a challenging task.

In this paper, we present a novel bus arrival time prediction approach based on real-time and real-world historical GPS trajectories of buses. In our approach, bus trajectories are firstly aggregated and mined in the cloud to discover some knowledge. Then bus arrival time is predicted based on the knowledge mined from the historical bus trajectories and real-time information. As the bus trajectories are constantly updated in the cloud, the prediction will be smarter and smarter.

The main contributions of the current work are summarized as follows.

- We propose a graph, which models the knowledge of routes based on the bus trajectories and reduces the online computation of arrival time prediction. We name the

✉ Haitao Xu
xuhaitaotao778@163.com

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, China

graph as path-section graph. This graph has not been used in the works of bus arrival time prediction.

- We present a clustering algorithm to learn the time-variant distributions of the travel times of path sections between any two bus stops. The algorithm includes two phases. The travel times of a path-section edge are clustered into several categories in the first phase. The proper time partition for each path-section edge is obtained in the second phase.
- We build a bus arrival time prediction system by using real-world trajectory dataset generated by 1000 buses in a period of 2 months. Experiments show that our method can predict more accurate arrival time with less online computation than competing methods.

The rest of the paper is organized as follows: the second part describes related work of the problem. In the third part, an overview of our approach is presented. Three major components of our approach: bus trajectory processing, path-section graph construction, and arrival time predicting are respectively introduced from the fourth section to the sixth section. Results obtained from experiments on real-world data of Hangzhou are reported in the seventh section. Concluding remarks are drawn in the last section.

2 Related work

Bus arrival time prediction has recently attracted interest of many researchers and practitioners in the area. The approaches for the prediction of bus arrival time can be mainly divided into the following categories: artificial neural network (ANN), Kalman filtering, support vector machines (SVM), regression analysis models and time series models.

ANN models [1–5] are the most commonly used approaches for arrival time prediction. These models are constructed with multiple layers of processing units named artificial neurons, emulating human intelligent data processing abilities. Using the learning process, they can automatically adjust the synaptic weights to map the input-output relationship. Public transport system is a very complex nonlinear system, and neural network model has the ability to deal with complex nonlinear systems, so many researchers use neural network model to predict the bus arrival time. In these works, it is considered that the current bus arrival time has a certain connection with that of buses in front of it. Meanwhile, bus arrival time is affected by road, environment and other factors.

The Kalman filter [6–8] is a model-based estimation technology that has the ability to accommodate traffic fluctuations with their time-dependent parameters adequately. It takes into account the stochastic nature of process disturbances and measurement noise. In the previous studies applying the Kalman filter to bus arrival time predictions, AVL, APC or both are employed. The algorithm needs real-time feeds when data is updated. At the same time, data fluctuations might cause difficulties in solving the time lag.

Support vector machine [9–12] is a specific type of learning algorithm developed on the basis of statistical learning theory, which is characterized by the capacity control of decision functions, the use of kernel functions and the sparsity of solutions. It has been proposed in time series forecasting. By integrating multiple sources of information, previous works proved that support vector machine can accurately predict bus arrival times. Although SVM also depends on the similarity between historic and real-time traffic patterns, it still shows many breakthroughs and plausible performance in forecasting of bus arrival time.

Regression analysis models [13] determine a dependent variable with a complex mathematical function formed by a set of independent variables. In general, the applicability of the regression models is limited because variables in transportation systems are highly inter-correlated. At the same time, their performance is undermined by long computational time because of their reliance on large amounts of historical data.

Time series models [14,15] are also used for predicting bus arrival time. It is assumed that all external factors of the system either are accounted for or are constant. Variation in the historical data patterns or a dissimilarity in real time data and historical data can lead to results being wrong and erroneous.

There are other methods: multiple data sources [16–18], metamodel approach [19], real-time data [20], agent-based approach [21], hybrid method [22–26], dynamic route [27]. Among these methods, some require special conditions and more processing time. Some fully depend on researchers' experience or knowledge to preprocess data in order to select control parameters. And the accuracy of prediction also needs to be improved.

3 Overview

As shown in Fig. 1, the architecture of our bus arrival time prediction system based on a hybrid model consists of three major components: Bus Trajectory Processing, Path-section Graph Construction, and Arrival Time Predicting. Bus Trajectory Processing and Path-section Graph Construction operate offline and Time Predicting is running online. The offline parts only need to be performed once unless the trajectory archive is updated.

3.1 Bus trajectory processing

In practice, a GPS log may record a bus's trajectory for several days, in which the buses always run the regular route.

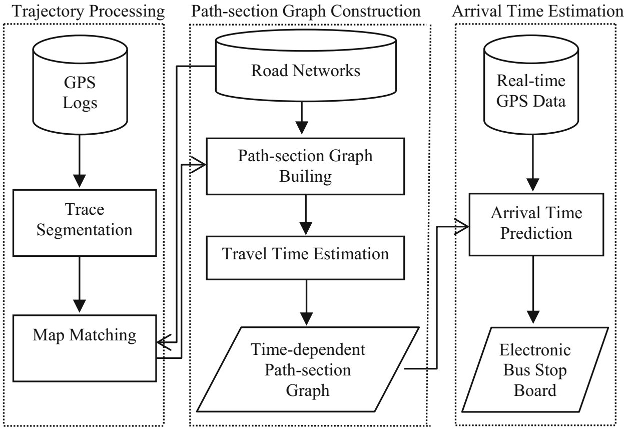

Fig. 1 System overview

Therefore, we partition a GPS log into some bus trajectories representing individual trips according to the longitude and latitude coordinates of origin stops and terminal stops. Then we employ a map matching algorithm to map each GPS point of a trip to the corresponding road segment. As a result, a bus trajectory is transformed into a sequence of road segments.

3.2 Path-section graph construction

Because traffic patterns of weekday are quite different from the patterns of weekend. We separate weekday trajectories from the weekend ones, and build a path-section graph for weekdays and weekends respectively. Then, we estimate the distribution of travel time of each path-section edge by using a clustering algorithm. In general, a path-section is the path segment between two adjacent bus stops. After the construction phase, a time-dependent path-section graph is available for online computation.

3.3 Arrival time predicting

In this phase, we present a method to compute the predicting arrival time of the bus at the queried stop. Given the bus ID, stop ID and the query time, this method computes the arrival time by using the path-section graph and real-time data. The whole travel time consists of four parts and will be computed section by section.

4 Bus trajectory processing

In this phase, bus GPS trajectories are segmented into trips, and each trip is matched against the road network. Thus a GPS log is partitioned into some bus trajectories representing individual trips. The partition is executed according to the longitude and latitude coordinates of original stations and terminal stations. Then a map matching algorithm is needed because of the well-known error of GPS data. The algorithm maps each GPS point of a trip to the corresponding road segment. The algorithm we employed is illustrated as follows.

```
Input:
geographical location $\varphi$
Output:
a meaningful geographical location $\varphi'$ on the road map
Steps:
{
  $\varphi' \leftarrow \infty$;
  z $\leftarrow \infty$;
  identify $G_k$ where $\varphi$ resides;
  identify R = $\{B_i | \mathrm{ResideG}(B_i, G_k) = \mathrm{true}\}$ as a subset of B-graphs which cross $G_k$;
  foreach $B_i$ in R do
  begin
    $\omega \leftarrow \mathrm{locate}(\varphi, B_i)$;
    if $d(\varphi, \omega) < z$ then
    begin
      z $\leftarrow d(\varphi, \omega)$;
      $\varphi' \leftarrow \omega$;
    end
    $\varphi'$ is the best remedy of $\varphi$ in terms of $d(\varphi, \varphi')$.
  end
  return $\varphi'$;
}
```

In the map matching algorithm, ResideG (B_i, G_k) is a Boolean function. ResideG (B_i, G_k) is true if link L_i lies or passes across the region of G_k. locate(ϕ, B_i) is a function from (ϕ, θ, B_i) into ϕ', such that the longitude and latitude coordinates of ϕ can be remedied to ϕ', which is located to polyline k associated with B_i. θ is the direction of ϕ, which represented by a geographical angle clockwise from the north. Function d (ϕ, ω) computes the geographical distance between ϕ and ω. Through experiments, we find that this map matching algorithm has a better performance when dealing with the low-sampling-rate bus trajectories.

5 Path-section graph construction

After the phase of bus trajectory processing, a path-section graph will be constructed. This consists of two parts of work: building path-section graph and travel time estimation. This section first describes the construction of the path-section graph, and then introduces the travel time estimation of path-section graph edges.

5.1 Building the path-section graph

For facilitating further description, several definitions and terms are listed below.

Definition 1 (*Path-section*) A path-section is the path segment between two adjacent bus stops.

Definition 2 (*Path-section Graph*) A path-section graph G =(V, E) is a directed graph that consists of a set of bus stops V and a set of path sections E.

According to the definition1, a path-section is a road segment between two adjacent bus stops. So it is not difficult to build the framework of path-section graph according to the road network and the longitude and latitude coordinates of bus stops. As we known that traffic patterns of weekdays are similar while those of weekdays and weekends are different. So we need to build two distinct path-section graphs for weekdays and weekends respectively. In other words, all the bus trajectories are divided into two parts. We put the weekday data part into a weekday path-section graph, and project the weekend data part into a weekend path-section graph. The vertexes of the path-section graph represent the bus stops. As a result, we convert each bus trajectory from a sequence of road segments to a path-section sequence. Then the distribution of travel time of each path-section edge should be estimated.

5.2 Travel time estimation

In this phase, we estimate the distribution of travel time of each path-section edge. The travel time of a path-section is dynamic at least in the following two aspects. Firstly, it is time-dependent. The travel time could be very long in rush hours while be quite short at other times. The time slots 8:00–9:00 and 14:00–15:00 have different distributions. Secondly, it is location-variant. Different path-sections have different time-variant traffic patterns. For example, some roads could be very crowded in the morning rush while be very fast in the evening rush. And some roads may be crowded most of the time. Therefore, we cannot apply the same or a predefined time partition method for all the path-sections.

Meanwhile, we find that the travel times of a path-section edge gather like a set of clusters rather than a typical Gaussian distribution. Some reasons can cause this. Firstly, different drivers may encounter different number of traffic lights. Secondly, different drivers have different personal behavior, skill and preferences. Therefore, we consider travel time of a path-section edge as a set of distributions corresponding to different time slots. This is distinct from some existing methods, which regard the travel time of a road as a single-valued function based on time of day. For example, some drivers need to spend 8–10 min to traverse a path-section in the time slot 8:00–10:00, while some drivers can accomplish this transition with a 5–18 min cost and some of them need less than 5 min.

Because the distribution will change over time of day. To learn different time partitions for different path-section edges based on the bus trajectories, we develop a clustering algorithm. The algorithm includes two phases. In the first phase, the travel times of a path-section edge are clustered into several categories according to the variance of travel times. In the second phase, a proper time partition for each path-section edge is obtained. As a result, the distributions of travel times in different time slots of each path-section edge can be estimated. The clustering algorithm is described as follows.

The first phase of clustering algorithm: x and y are two bus stops with leaving time t_x and arriving time t_y, s = (x,y;t_x, t_y) is a transition, let s_{xy} be the set of the transitions connecting (x, y) and $\tau_{xy} = \{(t_x, t_y)|(x, y; t_x, t_y) \in s_{xy}\}$. In this phase, we first sort τ_{xy} according to the values of travel time $\left(t_y - t_x\right)$, and then partition the sorted list L into several sub-lists in a recursive way. In each iteration, the variance of all the travel times in L is computed firstly. Later, the best split point having the minimal weighted average variance(WAV) is found. WAV is defined as Equation (1):

$$\text{WAV (i; L)} = \frac{|L_1(i)|}{|L|} Var\left(L_1(i)\right) + \frac{|L_2(i)|}{|L|} Var\left(L_2(i)\right) \tag{1}$$

In Eq. (1), $L_1(i)$ $L_2(i)$ are two sub-lists of L split at the i_{th} element and function Var() represents the variance. The best split point is the point which induces a maximum value of ΔV(i) as:

$$\Delta \text{V (i)} = \text{Val (L)} - \text{WAV (i; L)} \tag{2}$$

The first phase of algorithm terminates when max $max_i\{\Delta \text{V (i)}\}$ is less than a threshold δ. As a result, a set of split points which divide the whole list L into several clusters $C = \{c_1, c_2, \ldots, c_m\}$ are found. $c_i \in$ C represents a category of travel times.

The second phase of clustering algorithm: The main aim of this phase is to split the x-axis into several time slots such that the travel times have a relatively stable distribution in each slot. After the first phase, each travel time y_i can be represented with the category it belongs to $((y_i))$. Then the pair collection $S^{xc} = \{(x_i, c(y_i))\}_{i=1}^{n}$ are be sorted according to x_i, which is arriving time. The information entropy of S^{xc} is defined as:

$$\text{Ent}\left(S^{xc}\right) = -\sum_{i=1}^{m} p_i \log\left(p_i\right) \tag{3}$$

In Eq. (3), p_i. is the proportion of c_i. in the collection. The algorithm in the second phase runs iteratively to search a set of split points, which is similar to the algorithm in the first phase. However, we use the weighted average entropy(WAE) of S^{xc} instead of the WAV. WAE of S^{xc} is defined as:

$$\mathrm{WAE}\left(\mathrm{i};\, S^{xc}\right) = \frac{\left|s_1^{xc}\,(i)\right|}{\left|S^{xc}\right|} Ent\left(s_1^{xc}\,(i)\right) + \frac{\left|s_2^{xc}\,(i)\right|}{\left|S^{xc}\right|} Ent\left(s_2^{xc}\,(i)\right) \tag{4}$$

In Eq. (4), s_1^{xc} and s_2^{xc} are subsets of S^{xc} when split at the i_{th} pair. The best split point is the point whicleads to a maximum value of ΔE(i). as:

$$\Delta\mathrm{E}\,(\mathrm{i}) = \mathrm{Ent}\left(S^{xc}\right) - \mathrm{WAE}\left(\mathrm{i};\, S^{xc}\right) \tag{5}$$

As a result, the distribution of travel times in each time slot can be obtained after the second phase of the clustering algorithm. In other words, giving the time we can estimate the travel time of each path-section edge. These estimated travel times will be utilized in the next section for predicting arrival time.

6 Arrival time predicting

After the phase of path-section graph construction, bus arrival time will be predicted. In this section, we present the method to compute the predicting arrival time of the bus at the queried stop. The arrival time is then estimated as follows:

$$\mathrm{T} = T_{que} + T_1 + T_2 + \sum_{i=k}^{q-1} TW_i \tag{6}$$

In Eq. (6), T_{que} is the time when querying the arrival time. T_1 is the time for the bus to travel from its current location to the nearest next bus stop. Because the distance from the current bus location to the nearest next bus stop is in close, the possibility of traffic condition change is reduced greatly. So the bus speed won't change much and T_1 is defined as $T_1 = \frac{D}{V}$, in which D is the distance based on the road to the nearest next bus stop and V is the instantaneous velocity got through the GPS information.

T_2 represents the time for the bus to travel from its nearest next bus stop to the queried bus stop, which can be obtained from the path-section graph and travel time estimating described in Sect. 5. If there are i edges in the path-section graph from its nearest next bus stop to the queried bus stop, $T_2 = T_{21} + T_{22} + \ldots + T_{2i}$, T_{2i} is the travel time of i-th edge in the path-section graph. It can be computed using the method described in Sect. 5.

$\sum\limits_{i=k}^{q-1} TW_i$ is the dwelling time at stops from its nearest next bus stop $Stop_k$ to the queried bus stop $Stop_q$. The dwelling time of each stop is defined as the average dwelling time at this stop which can be computed from the bus trajectory data.

7 Practical examples

In this section, the proposed method for predicting bus arrival time at bus stops has been evaluated by real-world data in Hangzhou. Hangzhou is the capital of Zhejiang province and one of the most famous tourist cities in China. Hangzhou has a highly developed and sophisticated bus route network that comprises about 600 bus routes. Almost all the buses have been equipped with GPS devices. In Hangzhou, electronic bus stop board system provides the predicting arrival time information. We build a bus arrival time prediction system by using Hangzhou bus trajectory dataset generated by 1000 buses in a period of 2 months. The data set contains approximately one billion records. Each record contains the following information:

- *Bus ID* the unique bus ID.
- *Time* the sample timestamp "YYYY-MM-DDHH:MM:SS".
- *GPS position* the current longitude and latitude of the bus.
- *Speed* the current bus speed in km/h;
- *Orientation* the direction the bus is heading in, from 0^{circ} to 360^{circ}in clockwise.
- *GPS state* this value is set to 1 if the GPS data is incorrect, and 0 otherwise.

In our experiment, all the records with incorrect GPS data are removed.

In our tests, the prediction results are evaluated in terms of the performances of three measures: the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean square error (RMSE). The three terms can judge the difference between the observed and the predicted running time in different aspects. They are calculated as:

$$\mathrm{MAE} = \frac{\sum_{k=1}^{N} \left|t\,(k) - t_m(k)\right|}{N} \tag{7}$$

$$\mathrm{MAPE} = \left(\frac{1}{N}\sum_{k=1}^{N} \frac{\left|t\,(k) - t_m(k)\right|}{t_m(k)}\right) * 100\% \tag{8}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{k=1}^{N} \left((k) - t_m(k)\right)^2}{N-1}} \tag{9}$$

where $t(k)$ is the predicted running time of the test buses and $t_m(k)$ is the observed running time of the bus. And N is the test times.

We choose 6 routes including 2 test stops, as shown in Fig. 2. We first built the path-section graph by converting trajectory to a sequence. Then we use our method to estimate arrival time. Three of them were marked purple where the stop is Lung Bridge; three were marked as red where the stop is the martial arts small square.

Table 1 shows the number of valid observations of each route for each day, and RMSE statistics for the collected

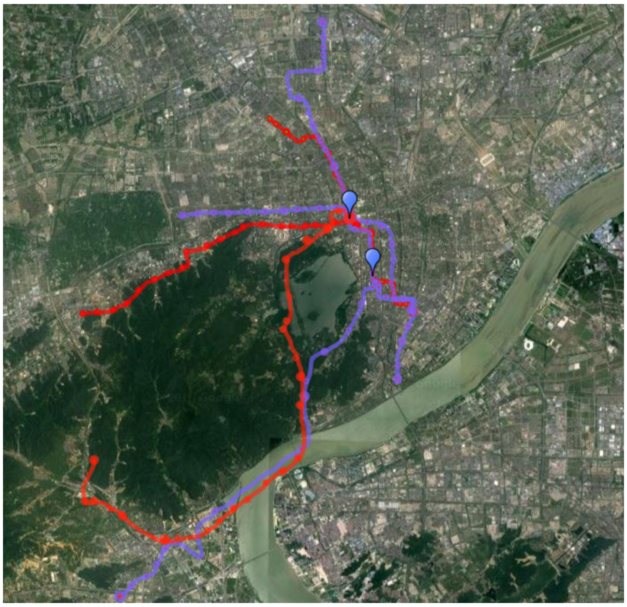

Fig. 2 Test route

travel time. In Table 1, we do the test on two bus stops from June 6 to June 10, which include route K10, K11, K41, K8, K19, K34. The RMSEs of our method are from 18.71 to 22.48. The average value of RMSE for 6 routes is 20.445.

The average value of the MAE, the MAPE and the RMSE of the two stops for all the bus routes are summarized in Fig. 3. MAE, RMSE and MAPE of stop1 is 28.3, 20.66 and 14.2%. The corresponding values of stop1 is 31.1, 20.23 and 16.3%.

We also test the proposed method on all bus stops of one route. The estimated arrival time and the measured arrival time of one bus for the 18 bus stops are compared and the results are presented in Fig. 4. On the whole, we can see that the prediction value is close to the actual value. As distance increases, the deviation increases.

To test the validity of the proposed method, we compare the prediction result of support vector machine (SVM) and our proposed method. The SVM is implemented with SVM MATLAB toolbox (University of Southampton, http://www.isis.ecs.soton.ac.uk/resources/svminfo/) on Microsoft Windows plat. Figure 5 shows the comparison results. We can see that our method exhibits some advantages over SVM model. All of the MAE, RMSE and MAPE values are lower than those of SVM.

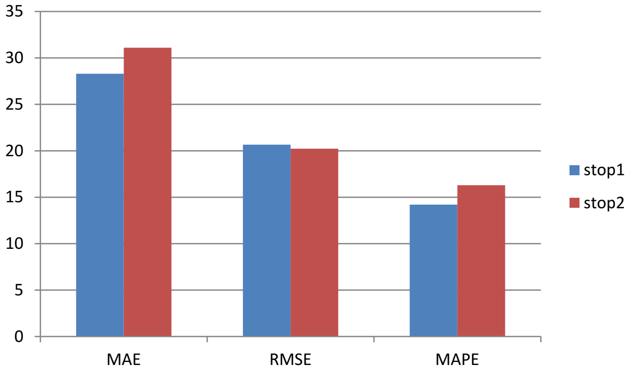

Fig. 3 Example of a feasible solution

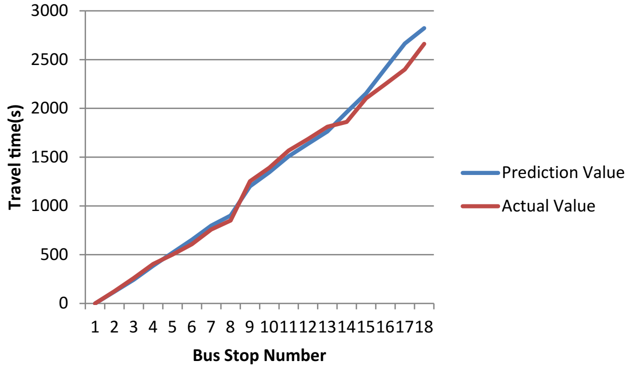

Fig. 4 Comparison of the predicted and measured time for 18 stops of one route

Moreover, we performed extensive practical experiments to compare the prediction result of three-layer artificial neural network (ANN) and our proposed method. The three-layer ANN consists of three hidden neurons. A scaled conjugate gradient algorithm is employed for training, and the hidden neurons are optimized by trial and error. Figure 6 shows the comparison results. MAE, RMSE and MAPE values prove that our method is a superior model.

Table 1 Parameter setting of improved algorithm

Stop No.	Route No.	Sample size (bus)					RMSE
		6-Jun	7-Jun	8-Jun	9-Jun	10-Jun	
1	K10	25	26	24	30	27	20.27
	K11	16	17	15	16	18	19.56
	K41	11	10	12	15	12	22.16
2	K8	17	16	15	16	20	19.49
	K19	10	11	11	12	10	22.48
	K34	12	10	12	11	10	18.71

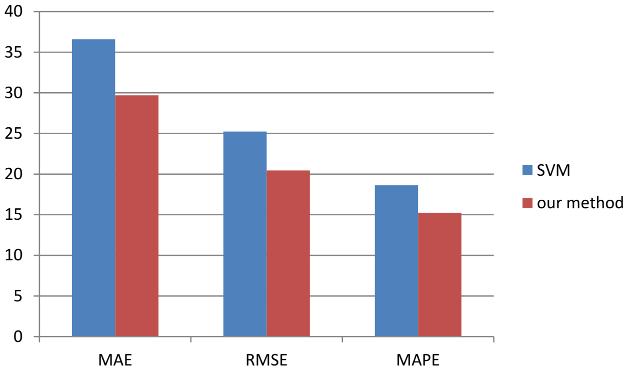

Fig. 5 Comparison of SVM and our method

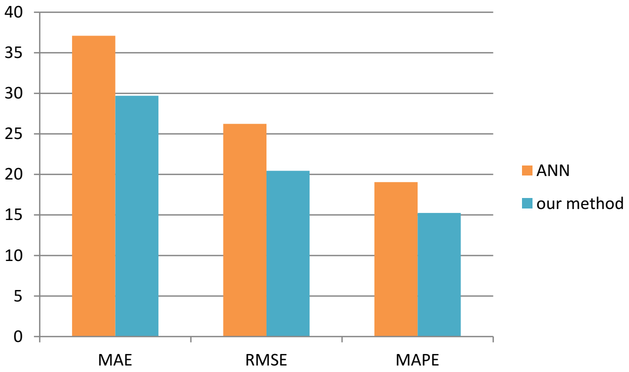

Fig. 6 Comparison of ANN and our method

Experiments show that the proposed method has higher predictive accuracy than SVM model and three-layer ANN model. The performance of our method for bus arrival time prediction at the stop is shown to be satisfactory. This can be attributed that our method makes full use of the real-time and historic data.

8 Conclusion

This paper presents an approach that predicts bus arrival time in terms of the knowledge learned from a large number of historical bus trajectories. In our method, a time-dependent path-section graph for weekdays and weekends is constructed respectively, and then arrival time is predicted based on this graph and real-time GPS information. The path-section graph models knowledge of routes based on the bus trajectories and reduces the online computation of arrival time prediction. This graph has not been employed in the previous works of bus arrival time prediction. Then a clustering algorithm is presented to learn the time-variant distributions of the travel times of path sections between any two bus stops. We build a bus arrival time prediction system by using a real-world trajectory dataset generated by 1000 buses in a period of 2 months, and evaluate the system with extensive experiments and in-the-field evaluations. Experiments show that our method can predict more accurate arrival time with less online computation than competing methods.

Acknowledgements This work was financially supported by Chinese National Natural Science Foundation (61572165) and Public Projects of Zhejiang Province (2015C33067).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this manuscript.

References

1. Chien, S.I.J., Ding, Y., Wei, C.: Dynamic bus arrival time prediction with artificial neural networks. J. Transp. Eng. **128**, 429–438 (2002)
2. Cathey, F.W., Dailey, D.J.: A prescription for transit arrival/departure prediction using automatic vehicle location data. J Transp. Res. Part C **11**(3), 241–264 (2003)
3. Hinsbergen, C.P.I.J., Van, Lint J.W.C., Van, Zuylen H.J.: Bayesian committee of neural networks to predict travel times with confidence intervals. J Transp. Res. Part C **17**(5), 498–509 (2009)
4. Liu, J., Yu, X., Xu, Z., et al.: A cloud-based taxi trace mining framework for smart city. Software (2016). doi:10.1002/spe.2435
5. Kumar, V., Kumar, B.A., Vanajakshi, L., et al.: Comparison of model based and machine learning approaches for bus arrival time prediction. In: Transportation Research Board 93rd Annual Meeting. 2014, 14–2518 (2014)
6. Steven, I., Chien, J., Kuchipudi, C.M.: Dynamic travel time prediction with real-time and historic data. J. Transp. Eng. **129**(6), 608–616 (2003)
7. Shalaby, A., Farhan, A.: Prediction model of bus arrival and departure times using AVL and APC data. J. Public Transp. **7**(1), 3 (2004)
8. Chen, M., Liu, X., Xia, J., et al.: A dynamic bus-arrival time prediction model based on APC data. J. Comput. Aided Civ. Infrastruct Eng. **19**(5), 364–376 (2004)
9. Bin, Y., Zhongzhen, Y., Baozhen, Y.: Bus arrival time prediction using support vector machines. J. Intell. Transp. Syst. **10**(4), 151–158 (2006)
10. Yu, B., Yang, Z.Z., Wang, J.: Bus travel-time prediction based on bus speed. J. Proc. ICE-Transp. **163**(1), 3–7 (2010)
11. Yu, B., Lam, W.H.K., Tam, M.L.: Bus arrival time prediction at bus stop with multiple routes. J. Transp. Res. Part C **19**(6), 1157–1170 (2011)
12. Yang, M., Chen, C., Wang, L., Yan, X., Zhou, J.: Bus arrival time prediction using support vector machine with genetic algorithm. J. Neural Netw. World **26**(3), 205–217 (2016)
13. Jeong, R., Rilett, L.: Prediction model of bus arrival time for real-time applications. J Transp. Res. Rec. **1927**, 195–204 (2005)
14. Sapankevych, N., Sankar, R.: Time series prediction using support vector machines: a survey. J. Comput. Intell. Mag. IEEE **4**(2), 24–38 (2009)
15. Zhu, T., Ma, F., Ma, T., et al.: The prediction of bus arrival time using global positioning system data and dynamic traffic information. In: Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP. IEEE, pp. 1–5 (2011)
16. Zhou, Y., Yao, L., Chen, Y., et al.: Bus arrival time calculation model based on smart card data. J. Transp. Res. Part C **74**(8), 1634–1638 (2016)
17. Min, W., Wynter, L.: Vehicle arrival prediction using multiple data sources including passenger bus arrival prediction. J. U.S. Patent 9,177,473 (2015)

18. Sayarshad, H.R., Chow, J.Y.J.: Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data. J. Adv. Transp. **50**(7), 1275–1294 (2016)
19. O'Sullivan, A., Pereira, F.C., Zhao, J., et al.: Uncertainty in bus arrival time predictions: treating heteroscedasticity with a metamodel approach. J IEEE Trans. Intell. Transp. Syst. **17**(11), 3286–3296 (2016)
20. Cats, O., Loutos, G.: Real-time bus arrival information system: an empirical evaluation. J. Intell. Transp. Syst. **2015**, 1–14 (2015)
21. Cats, O.: An agent-based approach for modeling real-time travel information in transit systems. J. Proc. Comput. Sci. **32**, 744–749 (2014)
22. Cats, O., Loutos, G.: Evaluating the added-value of online bus arrival prediction schemes. J. Transp. Res. Part A **86**, 35–55 (2016)
23. Cats, O.: Regularity-driven bus operation: principles, implementation and business models. J. Transp. Policy **36**, 223–230 (2014)
24. Hernandez, T.: Flex scheduling for bus arrival time prediction. J Transp. Res. Rec. **2418**, 110–115 (2014)
25. Balasubramanian, P., Rao, K.R.: An adaptive long-term bus arrival time prediction model with cyclic variations. J. Public Transp. **18**(1), 6 (2015)
26. Moreira-Matias, L., Mendes-Moreira, J., Sousa, J.F.D., et al.: Improving mass transit operations by using AVL-based systems: a survey. J. IEEE Trans. Intell. Transp. Syst. **16**(4), 1636–1653 (2015)
27. Silva, V.B.C.D., Sciammarella, T., Campista, M.E.M., et al.: Time of arrival prediction with dynamic route tracking for public transportation systems. In: Ad Hoc NETWORKING Workshop. IEEE, pp. 1–8 (2015)

Haitao Xu received the M.S. degree in computer software from Hangzhou Dianzi University, and now is a Ph.D. candidate of Zhejiang University. His research interests are in the areas of machine learning and data mining. As a member, he won the second prize of Zhejiang Science and Technology Award and the first prize of Zhejiang Higher scientific research achievement awards.

Jing Ying born in October 1971, Professor. He is deputy director of College of Computer Science and Technology, Zhejiang University. In 2003, he won the second prize of Zhejiang Province Award for excellence in the Eleventh teenagers (youth group). As a member of the chinese institute of computer software engineering committee, he mainly engaged in software development, software architecture, software engineering specifications. He has accomplished more than 20 scientific research projects at national and ministerial level. And he has published over 30 theses in important academic periodicals home and abroad.