Forecasting Traffic Flow in Big Cities Using Modified Tucker Decomposition

Manish Bhanu[1](✉), Shalini Priya[1], Sourav Kumar Dandapat[1], Joydeep Chandra[1], and João Mendes-Moreira[2]

[1] Indian Institute of Technology, Patna, India
{manish.pcs16,shalini.pcs16,sourav,joydeep}@iitp.ac.in
[2] LIAAD - INESC TEC, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal
jmoreira@fe.up.pt

Abstract. An efficient traffic-network is an essential demand for any smart city. Usually, city traffic forms a huge network with millions of locations and trips. Traffic flow prediction using such large data is a classical problem in intelligent transportation system (ITS). Many existing models such as ARIMA, SVR, ANN etc, are deployed to retrieve important characteristics of traffic-network and for forecasting mobility. However, these methods suffer from the inability to handle higher data dimensionality. The tensor-based approach has recently gained success over the existing methods due to its ability to decompose high dimension data into factor components. We present a modified Tucker decomposition method which predicts traffic mobility by approximating very large networks so as to handle the dimensionality problem. Our experiments on two big-city traffic-networks show that our method reduces the forecasting error, for up to 7 days, by around 80% as compared to the existing state of the art methods. Further, our method also efficiently handles the data dimensionality problem as compared to the existing methods.

Keywords: ODM · CP decomposition · Tucker · Time-series · CUR Traffic flow

1 Introduction

Over the last few decades, a large body of flow prediction techniques have been developed to facilitate the traffic organizations in controlling and improving the transportation efficiency—ranging from driver assistance to vehicle routing, forecasting and signal coordination [2,17]. Traffic-network is generally studied with Origin-Destination matrix (ODM). An ODM represents traffic volume in a given time duration. Many stacked ODMs for consecutive time duration represent time series of traffic volume of the sources and destinations. Given a large volume of origin-destination data (identified by the GPS locations), collected over a period of time, the traffic prediction problem is to forecast the traffic volume between locations at any given time in the future.

G. Gan et al. (Eds.): ADMA 2018, LNAI 11323, pp. 119–128, 2018.
https://doi.org/10.1007/978-3-030-05090-0_10

A large body of works exists that deals with traffic forecasting. Forecasting techniques can be broadly categorized as parametric and non-parametric techniques. Examples of parametric techniques include linear and nonlinear regression, smoothing techniques [16] and auto-regressive process by time-series analysis [5], to list a few. On the other hand the non-parametric techniques include non-parametric regression [3] and neural network based techniques [11,12]. Existing approaches for forecasting the traffic flow using time series approaches like, ARIMA [9], SARIMA [8] and Vector ARMA [7] are used to study the characteristics of the ODM and forecasting the same. Due to peak and off-peak time, traffic flow unveils a regular pattern, i.e. it presents daily seasonality. According to many previous studies, in order to predict better seasonal pattern SARIMA model is proved to perform better than the models constructed using support vector regression, historical average, and simple ARIMA [9]. Some machine learning techniques like ANN [6], SVR [1], KNN [18] and Bayesian network [14] have also been used. These methods generally perform well when modeling and prediction is done for each OD pair separately. Further, most of these approaches predict each time series independently. Forecasting each trip is considered isolated and prediction of thousands of trips goes separately. Thus these works do not capture the existing inter-relationship among the trips [4,19]. With widening popularity of deep learning techniques, Lv et al. [11] applied the same for traffic flow prediction using several traffic features. Experimental results indicate that these methods obtain much improved performance over traditional methods. However, these single point prediction methods are inefficient when dealing with larger traffic networks. For example, a traffic network containing $N \times N$ source-destination pair requires $O(n^2)$ time-series to predict, which is difficult to process when N exceeds thousands.

Tensor based methods on the other hand can deal with these problems better by factorizing the ODM into essential components. Tensor decomposition can yield dimensionality reduction; the reduced dimensional components are easy to be operated. This helps handling complete network as one unit. Tensors have found wide applications in forecasting traffic [13,15]. Most of these techniques aim to derive traffic approximations using low rank tensor decompositions. However, the major drawback of this approach is the requirement of huge historic data to predict the traffic flow from one source to a destination. There are two major scopes of improvement to these techniques, (a) improving the prediction accuracy and (b) maintaining higher accuracy in face of missing traffic data. Our paper is directed towards fulfilling these goals.

In this paper we propose a tensor decomposition based traffic forecasting technique that addresses both these requirements. The proposed approach uses three and four dimensional tensors to predict complete city traffic flow at once using modified Tucker tensor decomposition. Our modified Tucker decomposition technique uses CUR factorization, that gives the advantage of restoring the original values of matrix during approximation for prediction. The Frobenius norm of the difference between original and approximated matrix is very less when CUR is used over other matrix factorization methods like SVD, QR etc.

For our analysis, we use two datasets, one from New York city and the other from Thessaloniki city. Both contain traffic flow from many sources to destinations in the city and we predict the traffic flow (volume) of complete city. Experimental results indicate that the proposed approach yields significant forecasting accuracy even for sparse network data. The forecasting accuracy of our approach when compared to the recent techniques indicates an improvement of around 80%. Further, the proposed approach also effectively handles higher dimensionality of the traffic data.

The organization of the paper is as follows. We discuss the problem definition in Sect. 2. The proposed approach is discussed in Sect. 3 followed by the experimental procedure in Sect. 4. The experimental results are discussed in Sect. 5. Finally, we draw our conclusions in Sect. 6.

2 Problem Statement

For a big city, the traffic network consists of millions[1] of trips with GPS information of start and end locations along with the trip-timestamps. The ODM of these cities contains millions of time-series data. Using such a large ODM in any predictive model would incur huge computational cost. To tackle the problem of predicting traffic flows in large networks, considering the inter-dependence relationships of the time-series, we need to address the following research questions:-

RQ1: How to create an efficient ODM for a big city traffic network with the desired number of influencing factors (components) of mobility?
RQ2: How to handle the interdependence of the trip time-series to forecast the traffic flow in the entire network?
RQ3: How to handle higher dimensionality of the traffic data (influencing factors or components) that can affect the forecasting results?

We formulate the problem of large city traffic forecasting as a two-step process: (i) decomposing (unfolding) the traffic data $\mathcal{T}$ into components A_i (for i^{th} component) and (ii) approximating components with factorized matrices (F_i) for which the Frobenius norm of the difference is minimum under low-rank constraints [10,15]. The problem lies in finding approximated components A_i, F_i and reconstructing $\mathcal{T}$ after prediction.

$$\begin{gathered}\underset{A_i}{\operatorname{argmin}} \textstyle\sum_i \|A_i - F_i\|_F^2 \\ \text{subject to} : unfold(\mathcal{T}, i) = A_i \\ rank(A_i) = k < rank(\mathcal{T}_i)\end{gathered} \tag{1}$$

3 Methodology

In this section, we outline our methodology adopted to address each of the issues described in the previous section.

[1] in our examples: New York 2.6M, Thessaloniki 1.7M; M: million trips in 3 months.

***RQ1 (ODM creation)*:** The solution to the first question is the multidimensional ODM which can be easily represented with Tensor. The OD-tensor can be represented as a three-dimensional tensor $\mathcal{T} \in \mathbb{R}^{T \times S \times D}$, where T, S and D, represents the time periods, sources and destinations respectively. Likewise, we also use OD-tensor with four components $(T \times S \times D \times W)$ where W is the day of a week. Each entry of an OD-tensor cell is the volume of traffic from source S to destination D for given time T on a particular weekday W. We cannot create an ODM directly from GPS locations as the number of unique positions would be in millions. An efficient approach is to use clustering algorithms to group a set of nearby positions to represent a particular area of the city, that can act as an origin/destination in the ODM. We use a grid based strategy in our work as (a) it's faster, (b) density-based clustering techniques like DBSCAN caused uneven distributions of GPS locations resulting in few clusters to have approximately 90% of all locations while (c) hierarchical-based techniques need heavy calculations of Haversine distance between each location.

***RQ2 (Forecasting)*:** OD-tensor of the traffic network contains time-series data for each pair of origin-destination. The number of time-series for four components (c-4) will increase multiplicatively with the increase in the number of components in tensor. During decomposition we gain dimensionality reduction by decomposing the tensor with dimension $(T \times S \times D \times W)$ to smaller dimensional components, $T \times K_1$, $S \times K_2$, $D \times K_3$ and $W \times K_4$ respectively along with a tensor (core) with dimension $K_1 \times K_2 \times K_3$, where $K_1 < T$, $K_2 < S$, $K_3 < D$ and $K_4 < W$. We use a generalization of HOSVD decomposition called *Tucker decomposition*[2]. Tensor decomposition is also used in missing data problem rather than forecasting [10]. However, modelling the same for forecasting would require large historic traffic data. (we present details in Subsect. 5.2). However CP decomposition has no flexibility to assign different ranks to each components separately. Hence we use Tucker decomposition with CUR factorization owing to the advantages of CUR over SVD, (1) retaining real-valued data of time-series, (2) maintaining sparsity in components and (3) lower Frobenius norm difference than that of SVD.

Proposed Approach (Modified Tucker-CUR): Given an OD-tensor $\mathcal{T}$, we initially perform Tucker decomposition with CUR factorization to approximate the component matrices (Algorithm 1). The selection of best columns and rows in CUR is done on the basis of top higher L_1 values of columns and rows. We use ALS to approximate A_i (Algorithm 2).

[2] https://iksinc.online/2018/05/02/understanding-tensors-and-tensor-decompositions-part-3/.

Algorithm 1. Tucker Decomposition

procedure (Tucker) $(\mathcal{T}, R1, R2, ..., RN)$
  for `n=1,2,...,N` **do**
    $A^{(n)} \leftarrow C_n\ CUR(\mathcal{T}_n)$
  end for
  $G \leftarrow \mathcal{T} X_1 A^{(1)T} X_2 A^{(2)T} ... X_N A^{(N)T}$
  return $G, A^{(1)}, A^{(2)}, ..., A^{(N)}$
end procedure

Algorithm 2. Sampled CUR factorization

Input: $A(i)_{m \times n}$.
Output: CUR
for `i:n` **do**
  $r_{l2} = \|A_{(i)}(i, :)\|$
  $R \Leftarrow select\_top\_k_1\_rows$: C_{l2}
end for
for `i:n` **do**
  $c_{l2} = \|A_{(i)}(i, :)\|$
  $C \Leftarrow select\_top\_k_2\_columns$: r_{l2}
end for
$U \Leftarrow pseudo - inverse(R \cap C)$
$U \Leftarrow (U^{+}U)^{-1}U^{+}$
return C,U,R

For forecasting we use the approximated temporal component matrix, which would be of dimension $T \times K_1$, where T is the maximum days and K_1 is the number of columns in the low rank matrix (K_1 is 100 or less in our experiments). Thus this matrix produces K_1 time series with data for 1 to T days. We use these time series values to predict additional values up to δ additional days (we use a maximum value of $\delta = 7$), using ARIMA and LSTM. Applying prediction model on the temporal component (size $T \times K_1$) is computationally less expensive than applying on the entire tensor ($S \times D \times W$). We subsequently obtain a new matrix of size $(T + \delta) \times K_1$ by combining the predicted rows to the original temporal component matrix and then use the same to reconstruct the new tensor $\mathcal{T}'$ with dimension $(T + \delta) \times S \times D \times W$. The new entries in the $\mathcal{T}'$ for the δ additional days represent the forecasted values of the traffic. The steps of the proposed approach is outlined in Algorithm 3.

***RQ3* Higher Dimensionality:** The proposed approach in Algorithm 3 can effectively handle higher dimensionality. To investigate the same we conducted forecasting experiments with 3 and 4 components ($T \times S \times D$ and $T \times S \times D \times W$ respectively) to see the effect of more components on the predictive results.

We next discuss the dataset used for our experiments followed by the experimental results.

Algorithm 3. Tucker-CUR: Proposed algorithm

Input: $X_{old}, k_1, k_2, k_3, k_4, \delta.$
Output: $X_{predicted}$
$G, factors \leftarrow Tucker(X_{old}, k_1, k_2, k_3, k_4)$
$Factors : C_{T*K_1}, C_{S*K_2}, C_{D*K_3}, C_{W*K_4}$
$C_{(T+\delta)*k_1} \leftarrow predict(C_{T*K_1}, \delta)$
$X_{predicted} \leftarrow G_{K1\times K2\times K3} X_1 C_{(T+\delta)X_2K1} X_2 C_{S\times K2} X_3 C_{D\times K3} X_4 C_{W\times K4}$
return $X_{predicted}$

4 Experimental Procedure

For our experiments, we used New York City Green Taxi data(NYC) from January to March 2014 available online[3]. The other data is from Thessaloniki(THS) private taxi company for January to March 2015 which is provided in famous "taxi fare challenge kaggle competition, 2017". For Green Taxi data, each trip has GPS data of pickup location, drop-off location and time duration of the trip. The same information were obtained for the THS data, with an assumption that the startup time of the next trip being same as the drop-off time of the previous trip.

- For both the cities, the city GPS locations were clustered in grid size of 2 and 5 km^2. Then we created OD-tensor with components $T \times S \times D$ (c-3) and $T \times S \times D \times W$ (c-4) for all the grided city network. We obtained eight city traffic-networks: NYC02(c-3, c-4) & NYC05(c-3, c-4) and THS02(c-3, c-4) & THS05(c-3, c-4)[4]. Each grid is considered as a node of the OD-tensor. We only used the node-pairs whose traffic volume were equal or greater than 60 for this time duration.
- For 300 locations, the OD-tensor time series count is 90,000. We apply Algorithm 1 to obtain the component matrices and use the temporal component matrix as stated in Algorithm 3 to derive the predicted OD-tensor.
- During prediction we used 80% of temporal component data as train-set and rest as test-set.The best models are obtained at ARIMA(7,2,1) and at LSTM (nodes_per_Layer = 200, epoch = 100, batch size = 3, hidden layers = 4, optimizer = "adam"), validation_split = 0.1. Data were normalized before prediction by LSTM. Sigmoid activation and multivariate of window size 7 was used along with time steps up to 7. We tuned other parameters of the models to get the best results on networks using HOSVD (one of baseline approach) and then the same models were used for Tucker-CUR (Algorithm 3) during prediction.

[3] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[4] (Newyork/Thessaloniki city with grid size 02/05 and components 3/4, c stands for components).

5 Results and Discussions

We next outline the evaluation strategies to analyze the performance of the proposed method and subsequently outline the results obtained.

5.1 Evaluation Strategies

We compare our approach with existing methods proposed for traffic-network predictions [13,15]. We compare the efficiency of our proposed approach in terms of the data requirement and prediction accuracy. Dynamic Tensor Completion (DTC) [15] DTC is a recent state of the art approach proposed for traffic prediction, the prediction model of DTC is not directly comparable to our proposed method due to its large data requirement. We subsequently compare the prediction accuracy of our approach with two state-of-the-art tensor decomposition methods proposed in [13]:

- **CP Decomposition:** In [13], the authors have used CP decomposition which is a tensor decomposition method that uses a random assignment of initial decomposed matrices.
- **HOSVD:** We also compare our work with the HOSVD decomposition which uses SVD as factorization approach. We use the HOSVD[5] (Tucker SVD) approach to obtain the temporal component matrix for prediction.

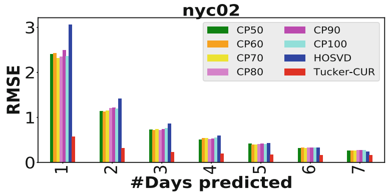
5.2 Data Requirement

In [15], the authors proposed a dynamic tensor completion technique for predicting short term traffic data. However the successful implementation of the technique requires satisfying two major conditions of the ODM:

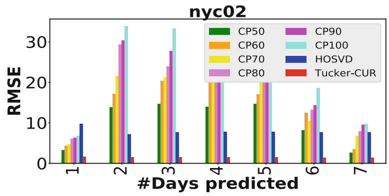
- A matrix with missing data must not have a complete row or column missing;
- A matrix must be a low rank matrix.

The second constraint, however, is applicable for almost every approach as very few factors (low-rank) are responsible for determining the traffic between locations. But the first constraint enforces heavy data requirement, i.e. a large volume of data is required to predict few data values. For an $M \times N$ matrix the predicted values cannot exceed max(M, N). We compared the maximum predicted values of our proposed approach with the DTC model. Giving a brief representation of data requirement in each approach, when ODM $M \times N$ is used, maximum predicted values in DTC is max(M-1, N-1) while it is cN (multiple columns) in our proposed approach where c is a positive integer. Both approaches used 90 days (3 months) for experimentation, hence "predicted values to data required" ratio for DTC is $1/M$ and for our approach it's c/M. We predicted up to 7 days, hence for c = 7 and M = 90, DTC and proposed approach have "predicted to data required" ratio as 0.01 and 0.07 respectively. This shows when modeling missing data problem as prediction problem like in DTC [15], the model demands more data requirement than proposed approach for same count of predicted values.

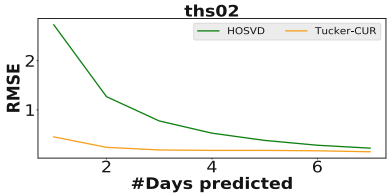
[5] k assignment at good prediction result: NYC-THS02 (50, 100, 100, 5) NYC-THS05 (50, 30, 30, 5) & same for Tucker-CUR.



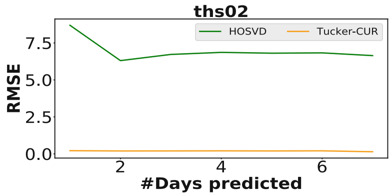
(a) RMSE by LSTM



(b) RMSE by ARIMA

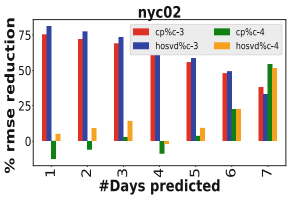


(c) RMSE by LSTM

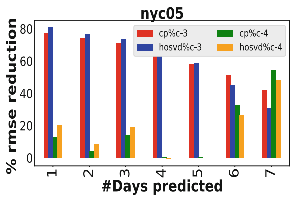


(d) RMSE by ARIMA

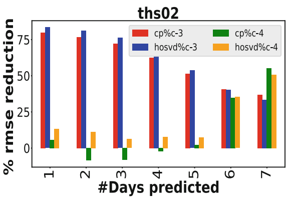
Fig. 1. LSTM performs much better than ARIMA. When the complete city network is predicted.



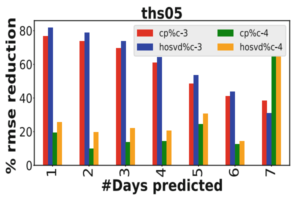
(a) % reduction of rmse in NYC02



(b) % reduction of rmse in NYC05



(c) % reduction rmse in THS02



(d) % reduction rmse in THS05

Fig. 2. % Reduction in RMSE using LSTM over both the city networks using (Eq. 2). Tucker-CUR is compared to CP70 (best among CP) and HOSVD in both (c-3, c-4).

5.3 Prediction Accuracy

Choosing the Time Series Model: We consider two major techniques for prediction of the time series: LSTM and ARIMA. Figure 1 shows the RMSE values of the predicted time series using LSTM and ARIMA. The range of values of RMSE in case of LSTM (Figs. 1a and c) is much lower as compared to

ARIMA (Figs. 1b and d). This is because traffic-network time series can be non-linear. LSTM captures the non-linearity of the data much better as compared to ARIMA. So, we use LSTM in our future experiments.

Comparison of Prediction Accuracy. We predicted eight traffic-networks using LSTM for 1 to 7 days using our approach as well as the CP and HOSVD based decomposition and calculate the RMSE values. The reduction in RMSE values is calculated using the following equation:

$$(OtherApproach_{RMSE} - TuckerCUR_{RMSE}) * 100/TuckerCUR_{RMSE} \quad (2)$$

It is observed that our approach has much lower RMSE as compared to CP (with different K = 50, 60, 70, 80, 90, 100) and HOSVD (with different k approximations) Fig. 2. In most cases, as shown in Fig. 2, our proposed method achieves around 80% reduction in RMSE, except in very few cases where CP(k = 70) performs better.

6 Conclusion

In this paper, we proposed an OD-tensor decomposition based method for forecasting large city traffic. The proposed method overcomes the problem of traditional time-series based prediction methods and allows addition of new dimensions if required. We have gained reduction in prediction error in terms of RMSE up to 80% compared to the state of the art approach. Our Tucker-CUR algorithm is efficient in sense of using real time-series data and reduced time complexity over CP, HOSVD. We also showed the effect of spatial and component changes in OD-tensor and this leaves scope for further analysis in this direction.

Acknowledgment. This work is funded by Project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020", a project financed by North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement.

References

1. Ahn, J.Y., Ko, E., Kim, E.: Predicting spatiotemporal traffic flow based on support vector regression and Bayesian classifier. In: 2015 IEEE Fifth International Conference on Big Data and Cloud Computing (BDCloud), pp. 125–130. IEEE (2015)
2. Bhanu, M., Chandra, J., Mendes-Moreira, J.: Enhancing traffic model of big cities: network skeleton & reciprocity. In: 2018 10th International Conference on Communication Systems & Networks (COMSNETS), pp. 121–128. IEEE (2018)
3. Davis, G.A., Nihan, N.L.: Nonparametric regression and short-term freeway traffic forecasting. J. Transp. Eng. **117**(2), 178–188 (1991)

4. Gabrielli, L., Rinzivillo, S., Ronzano, F., Villatoro, D.: From tweets to semantic trajectories: mining anomalous urban mobility patterns. In: Nin, J., Villatoro, D. (eds.) CitiSens 2013. LNCS (LNAI), vol. 8313, pp. 26–35. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04178-0_3
5. Ghosh, B., Basu, B., O'Mahony, M.: Bayesian time-series model for short-term traffic flow forecasting. J. Transp. Eng. **133**(3), 180–189 (2007)
6. Jun, M., Ying, M.: Research of traffic flow forecasting based on neural network. In: Second International Symposium on Intelligent Information Technology Application, IITA 2008, vol. 2, pp. 104–108. IEEE (2008)
7. Kamarianakis, Y., Prastacos, P.: Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches. Transp. Res. Rec.: J. Transp. Res. Board (1857), 74–84 (2003)
8. Kumar, S.V., Vanajakshi, L.: Short-term traffic flow prediction using seasonal arima model with limited input data. Eur. Transp. Res. Rev. **7**(3), 21 (2015)
9. Lee, S., Fambro, D.: Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transp. Res. Rec.: J. Transp. Res. Board (1678), 179–188 (1999)
10. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 208–220 (2013)
11. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
12. Priya, S., Bhanu, M., Dandapat, S.K., Ghosh, K., Chandra, J.: Characterizing infrastructure damage after earthquake: a split-query based IR approach. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 202–209. IEEE (2018)
13. Ren, J., Xie, Q.: Efficient OD trip matrix prediction based on tensor decomposition. In: 2017 18th IEEE International Conference on Mobile Data Management (MDM), pp. 180–185. IEEE (2017)
14. Sun, S., Zhang, C., Zhang, Y.: Traffic flow forecasting using a spatio-temporal bayesian network predictor. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005 Part II. LNCS, vol. 3697, pp. 273–278. Springer, Heidelberg (2005). https://doi.org/10.1007/11550907_43
15. Tan, H., Wu, Y., Shen, B., Jin, P.J., Ran, B.: Short-term traffic prediction based on dynamic tensor completion. IEEE Trans. Intell. Transp. Syst. **17**(8), 2123–2133 (2016)
16. Tan, M.-C., Wong, S.C., Xu, J.-M., Guan, Z.-R., Zhang, P.: An aggregation approach to short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **10**(1), 60–69 (2009)
17. Wang, J., Zhang, L., Zhang, D., Li, K.: An adaptive longitudinal driving assistance system based on driver characteristics. IEEE Trans. Intell. Transp. Syst. **14**(1), 1–12 (2013)
18. Xiaoyu, H., Yisheng, W., Siyu, H.: Short-term traffic flow forecasting based on two-tier k-nearest neighbor algorithm. Procedia-Soc. Behav. Sci. **96**, 2529–2536 (2013)
19. Yuan, Y., Raubal, M.: Extracting dynamic urban mobility patterns from mobile phone data. In: Xiao, N., Kwan, M.-P., Goodchild, M.F., Shekhar, S. (eds.) GIScience 2012. LNCS, vol. 7478, pp. 354–367. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33024-7_26