2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)

A Matrix-Decomposition-based Context Tensor Approach for Personalized Travel Time Estimation

Xiaopei Li, Fanzhang Li, Xin Dai, Helan Liang
School of Computer Science and Technology & Joint International Research Laboratory of Machine Learning and Neuromorphic Computing, Provincial Key Laboratory for Computer Information Processing Technology
Soochow University, Suzhou 215006, Jiangsu, China
Email: xpLiris@gmail.com, lfzh@suda.edu.cn, xdtydx@gmail.com, hlliang@suda.edu.cn

***Abstract*—In this paper, we use the matrix-decomposition-based context-aware tensor approach to estimate the user's personalized travel time. The model mainly has the following steps: firstly, by dealing with the GPS trajectories received in current time period to filter the appropriate data and the filtered data is expanded into a third-order tensor, corresponding to the driver, the road segment and the period of time. Sorting the GPS data of the past period into a third-order tensor of the same form. Each data in the tensor represents the travel time of a driver on a certain road during a certain period of time. Secondly, based on the definition of the third-order tensor, the third-order tensor can be considered as a combination of multiple second-order tensors (matrices). According to the slicing operation, the matrix of the invisible position is taken from the tensor, and the pre-processing is performed by using matrix decomposition to fill the vacancy. Thirdly, the feature matrix of different time space and geographic location, i.e. the context information, is extracted by the known GPS trajectory dataset to decompose the travel-time tensor, and then the time cost of a certain driver on a certain road in a certain period is obtained. Personalized travel time forecasts provide users with more choices of travel options and travel routes.**

Keywords—travel time estimation, tensor, matrix factorization, matrix-decomposition-based context-aware tensor decomposition

I. INTRODUCTION

As the population increases, travel issues are increasingly valued. The development of science and technology has led to the advancement of tourism. More and more people are willing to enjoy life. However, enjoying life means making the most of time and money. In the field of machine learning [1], a good travel-time prediction can not only provide individual user with a money-saving, labor-saving and time-saving out-of-office choice [23], but also provide a basic information for traffic management for agencies such as traffic management departments, for example, to avoid various traffic jams through travel-time predictions [16].

In the systems of intelligent transportation, one of the most important fundamental components is travel time prediction [12]. In recent years, there have been many methods for estimating travel time [6], [10], [15], [22], [24]. Besides, there have been many methods based on GPS trajectory datasets [8], but these models are limited, such as limited to limited datasets, GPS trajectories are limited to small towns, focus only on the average travel time of individuals and ignore the intrinsic relationship between roads and time and space, and so on [26]-[28], which means that they are not travel time prediction models applicable everywhere. In order to improve these limitations, this paper utilizes a method based on matrix decomposition preprocessing and context-aware tensor decomposition for travel time estimation. The method organizes a large and relatively sparse GPS trajectory dataset dynamically provided from the city into a third-order tensor to simulate the relationship between the driver, the trajectory and the time period, according to the tensor definition, preprocessing the driver-trajectory matrix over a certain period of time. And combining the current and historical trajectory data, the tensor is decomposed by using contextual information such as correlation between the extracted time and space.

II. RELATED WORK

A large and sparse tensor can use the Tucker decomposition to fill the values of unknown locations. Tucker decomposition is a common decomposition method for processing a tensor data. Due to the increasing relationship between time and space and individuals, Tucker decomposition does not predict road information correctly enough. In [4], which used the large and sparse trajectories of the city, represented by sparse tensors, using the traditional Tucker decomposition and the context information, to decompose the large and sparse tensor. Context information is used to extract the correlation between space-time and individuals. As far as the results are concerned, adding context information improves the accuracy of prediction. In the following methods, most of them are based on this method to join a certain constraint, such as the method in [13]. This method adds Block Term Decomposition to the context-aware tensor Tucker decomposition described above. BTD unifies two decompositions and approximates by a sum of low multi-linear rank terms [2], [19]. This decomposition method takes into account that the Tucker decomposition only estimates unknown entries based on its own non-zero entries and ignores other information. And compared with the above methods, the accuracy of the decomposition results is greatly improved .

All decomposition methods above were to decompose the tensor with unknown locations of zero-value. Based on the previous method, we made two improvements. First,

978-1-7281-5141-0/19/$31.00 ©2019 IEEE
DOI 10.1109/CBD.2019.00054

using the core tensor and three matrices decomposed by context-aware tensor decomposition instead of the random initialization in the original algorithm. Second, pre-filling the unknown positions of the tensor by using matrix decomposition. This paper uses the matrix decomposition method to fill the value of the unknown position [7]. And we added these two steps to the CATD method in [4] and the context-aware tensor method with BTD added in [13].

III. Overview

A. Tensor

For simplicity, the tensor can be regarded as a multi-dimensional array, the first-order tensor is a vector, the second-order tensor is a matrix, and the third-order tensor or even higher-order tensors are called high-order tensors, and the third-order tensor has three indexes for each value [9].

B. Time slot

A certain time interval. A certain period of time is divided into several time slots based on a specific time interval. In this paper, the time slot of one day is divided into 48, so each time slot is 30 minutes. (from 7:00 am to 9:00 am).

C. Slice operation

A slice operation is an operation of extracting a matrix from a tensor. If two dimensions are reserved in the tensor, other dimensional changes can result in a matrix, which is a tensor slice.

D. Tensor Tucker decomposition

Tucker decomposition is a high-order principal component analysis that decomposes a tensor into a core tensor and several matrices of tensors along each mode. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, its Tucker is decomposed into $\mathcal{X} \approx [\mathcal{G}; A, B, C] = \mathcal{G} \times_1 A \times_2 B \times_3 C$, as shown in Fig 1 [11].

E. Matrix decomposition

Refers to approximating the matrix M with $A * B$, then the element of $A * B$ can be used to estimate the element value corresponding to the invisible position in M, and $A * B$ can be regarded as the decomposition of M. This is the matrix decomposition, i.e. $M \approx A * B$ [7].

The framework of our model is shown in Fig 2. The map to get the GPS trajectories has been studied in many literatures, this article will not repeat them [14], [18], [25]. In this paper, the trajectory dataset is used to extract the contextual information [4], [5]. The obtained data is expanded into a third-order tensor according to the above method. The matrix in which the invisible elements locate is taken out by the tensor slicing operation just mentioned, and the matrix is subjected to matrix decomposition, and a value of the invisible position is filled into the original tensor. Then, using the contextual information, the processed tensor is subjected to Tucker decomposition to obtain predicted value of the unknown position.

IV. Methodology

A. Tensor modeling

In order to initially establish our model, the GPS trajectory dataset, that has been obtained by map matching, is processed and transformed into a third-order tensor $\mathcal{A}_r \in \mathbb{R}^{N \times M \times L}$, the three dimensions represent the driver, road segment and time period. Based on the known GPS trajectory dataset, the trajectory dataset of the latest L time segments is transformed into a third-order tensor, as shown in Fig 3. Each value of the tensor $\mathcal{A}_r(i, j, k) = a$ represents in the kth time period, the cost of the ith driver spending on the jth road is a. And L is the last time period of the statistics. It can be seen that such tensor is large and sparse, and it is very unfavorable for the prediction of the invisible position. In general, one of the method for dealing with higher dimensional matrices is dimensionality reduction.

Because high-dimensional matrices are not easy to process, they are broken down into some low-dimensional matrices processing. It is easier and more achievable to analyze high-dimensional matrices by using the relationship between low-dimensional matrices and high-dimensional matrices, so we take the method is to perform Tucker decomposition on the third-order tensor.

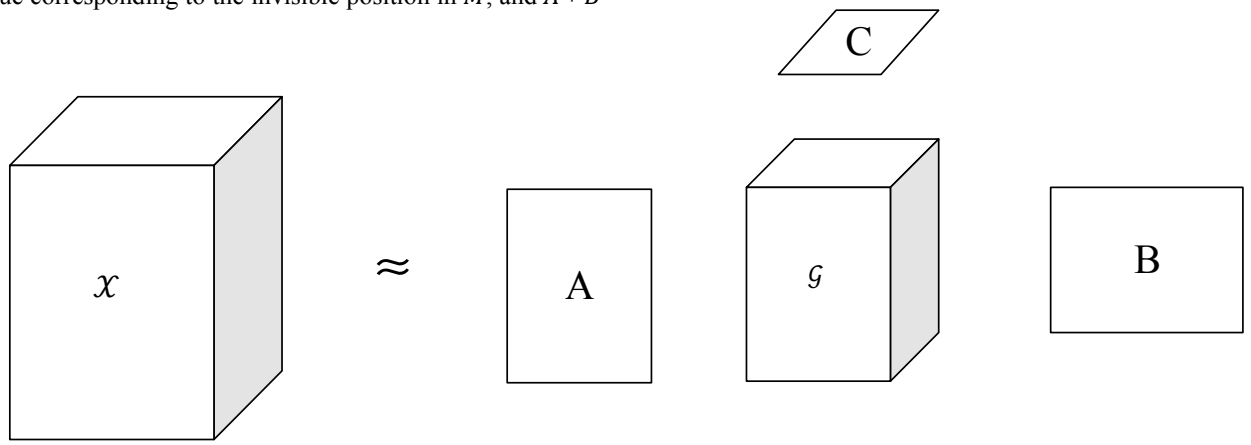

Fig. 1. Tensor Tucker decomposition

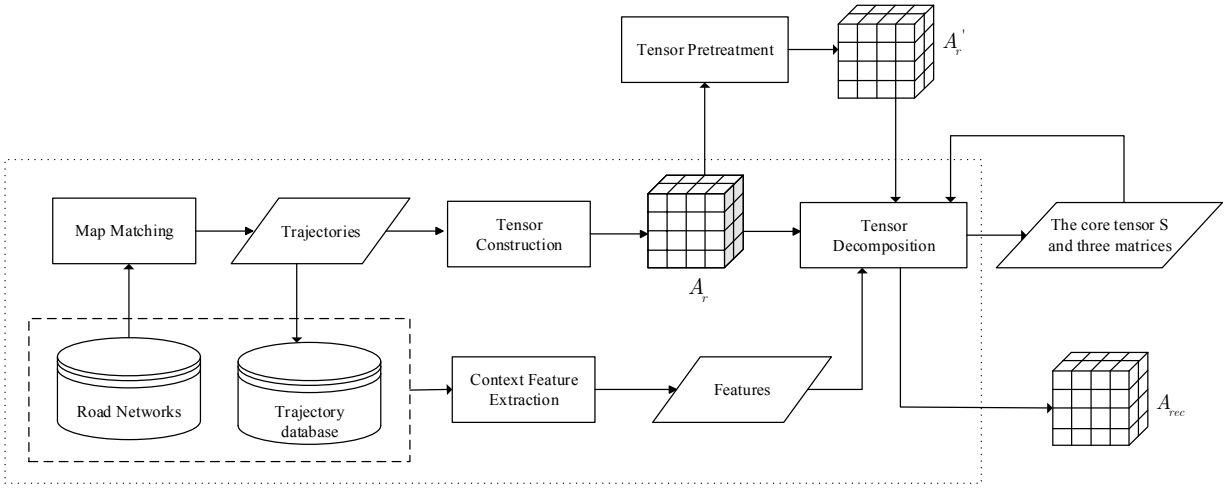

Fig. 2. Framework of our model

Because high-dimensional matrices are not easy to process, they are broken down into some low-dimensional matrices processing. It is easier and more achievable to analyze high-dimensional matrices by using the relationship between low-dimensional matrices and high-dimensional matrices, so we take the method is to perform Tucker decomposition on the third-order tensor. The tensor $\mathcal{A}_r$ is decomposed into a core tensor $S \in \mathbb{R}^{d_R \times d_U \times d_T}$, and three low-dimensional matrices: $R \in \mathbb{R}^{N \times d_R}$, $U \in \mathbb{R}^{M \times d_U}$, $T \in \mathbb{R}^{L \times d_T}$, and use (1) as the error function of the Tucker iteration, and the missing value of $\mathcal{A}_r$ can be supplemented by (2) to obtain the tensor $\mathcal{A}_{rec}$ [11].

$$\mathcal{L}(S,R,U,T) = \frac{1}{2} \| \mathcal{A}_r - S \times_R R \times_U U \times_T T \|^2 + \frac{\lambda}{2}(\| S \|^2 + \| R \|^2 + \| U \|^2 + \| T \|^2) \tag{1}$$

$$\mathcal{A}_{rec} = S \times_R R \times_U U \times_T T \tag{2}$$

B. Context information

In the process of dealing with the actual model, since the tensor $\mathcal{A}_r$ is too sparse, only analyzing the tensor itself will produce a larger error, and the decomposition of such a sparse tensor will make the result more inaccurate. In order to solve these problems, another third-order tensor, $\mathcal{A}_h$, is introduced as the historical tensor, which has the same structure as the tensor $\mathcal{A}_r$. The three dimensions represent the driver, the road segment and the time period. As shown in Fig. 3, and each value of the tensor $\mathcal{A}_h(i,j,k) = a'$ represents the ith driver in the kth time period, on the jth road, the cost of the historical average travel time is a'. In the sense of two tensors, $\mathcal{A}_h$ represents the trajectories of the past period of time, and the tensor $\mathcal{A}_r$ is sparser than the tensor $\mathcal{A}_h$. Increasing the historical travel time tensor, decomposing $\mathcal{A}_r$ and $\mathcal{A}_h$ together can reduce the error of only decomposing a sparse tensor $\mathcal{A}_r$.

In addition to the extracted historical travel time tensor, we also get two matrices X and Y through the GPS trajectories. As shown in Fig 4 A, Y represents a geospatial feature that captures the similarities between different sections of the geospatial, including the original geospatial features of each section, such as length, width, speed limit, etc. Y stores geographical features f_r of each road segment, as well as the distribution of Point of Interests f_p around r's terminals [4]. Matrix X, based on the grid divided for the city, as shown in Fig 4 B, extracted the temporal feature matrix. Dividing a city into disjoint and uniform grids, each consisting of many trajectories. The extracted temporal feature matrix consists of two parts, X_r and X_h. X_r is the time feature matrix based on real-time trajectories, the value in X_r represents the number of vehicles passing through a particular grid in a particular time period, and each row of X_r represents a coarse-grained traffic condition in the city over a particular time period. In short, X_r represents the correlation between different time periods based on coarse-grained traffic conditions. X_h has the same structure as X_r, representing the historical average number of vehicles passing through a particular grid over a period of time, as shown in Fig 3. In general, we build X_h of a whole day in advance, but only choose periods according to the current time, as shown in Fig 4 C, the rows from t_i to t_j will be retrieved from the prebuilt X_h to construct X with X_r [20].

C. Pretreatment

1) Tensor pretreatment: For the pretreatment of the tensor, the two steps of tensor slicing operation and matrix decomposition are selected. Firstly, using the tensor slicing operation, the experimental tensor is $\mathcal{A}_r \in \mathbb{R}^{N \times M \times L}$, which is divided into L matrices according to L time periods. Assuming that the invisible element is located in the last time period L, then the label is for the matrix of L, matrix $A_L \in \mathbb{R}^{N \times M}$, is taken out and processed. Then use

matrix decomposition to fill an element value in the invisible position in the tensor. Considering that each value of the matrix A_L has practical significance, it must be a non-negative value. A matrix decomposition method better than other methods mentioned in [7] discusses that for a common recommendation system, a scoring matrix R of size $m \times n$ is given, where r_{ij} represents the score of an user i on an item j. R can be written in the following form: $R = UV^T$. Where U is the user factor matrix of $m \times k$, and V is the item factor matrix of $n \times k$. This matrix decomposition process can be seen as an approximation process, $R \approx UV^T$. In order to make the error as small as possible, the optimization objective function is selected as: $min\, J = \frac{1}{2} \| R - UV^T \|$. But often the matrix R in the example is sparse, so our goal is to apply the data known already, in the matrix to recommend the user. The position of the value in the matrix R is denoted as (i, j), and all the observed position indices are denoted as S. The estimated value of the score for any position (i, j) is $r'_{ij} = \sum_{q=1}^{r} u_{iq} v_{jq}, i = 1, \dots, n, j = 1, \dots, m$, optimization objective function becomes (3), and the update formulas of u_{iq} and v_{jq} are (4) and (5), respectively, where $\alpha > 0$ represents the step size of the gradient drop. $j: (i, j) \in S$ and $i: (i, j) \in S$ represent a set of position indices of all non-zero elements on the vectors $R(i.:)$ and $R(:, j)$, respectively.

$$min\, J = \frac{1}{2} \sum_{(i,j) \in S} (r_{ij} - \sum_{q=1}^{k} u_{iq} v_{jq})^2 \tag{3}$$

$$u_{iq} \Leftarrow u_{iq} + \alpha \sum_{j:(i,j) \in S} (r_{ij} - \sum_{q=1}^{k} u_{iq} v_{jq}) v_{jq} \tag{4}$$

$$v_{jq} \Leftarrow v_{jq} + \alpha \sum_{i:(i,j) \in S} (r_{ij} - \sum_{q=1}^{k} u_{iq} v_{jq}) u_{iq} \tag{5}$$

According to the above method, the missing position of the matrix A_L is filled with the predicted value, and a new matrix $\hat{A}_L \in \mathbb{R}^{N \times M}$ is obtained by the padding stage.

2) Core tensor initialization: The above tensor modeling method and contextual processing method are collectively referred to context-aware tensor decomposition (CATD) [4]. In CATD, the context-aware tensor decomposition is performed, and the initial drawing of a core matrix and three matrices for solving is randomly initialized. In this method, the core tensor S and three matrices on three modes R, U, T obtained by the CATD method is used as the initialization of the model.

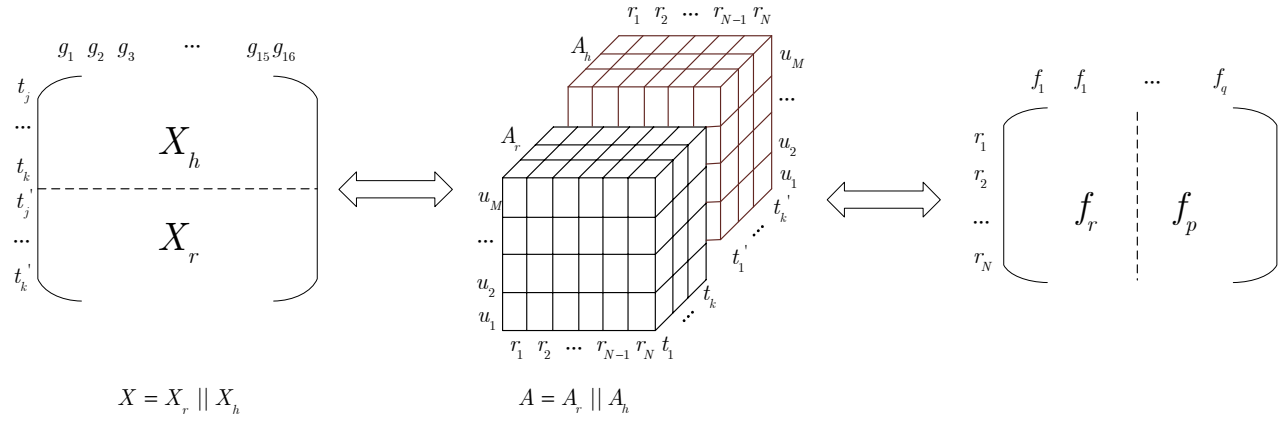

Fig. 3. The model of dealing with dataset sparsity

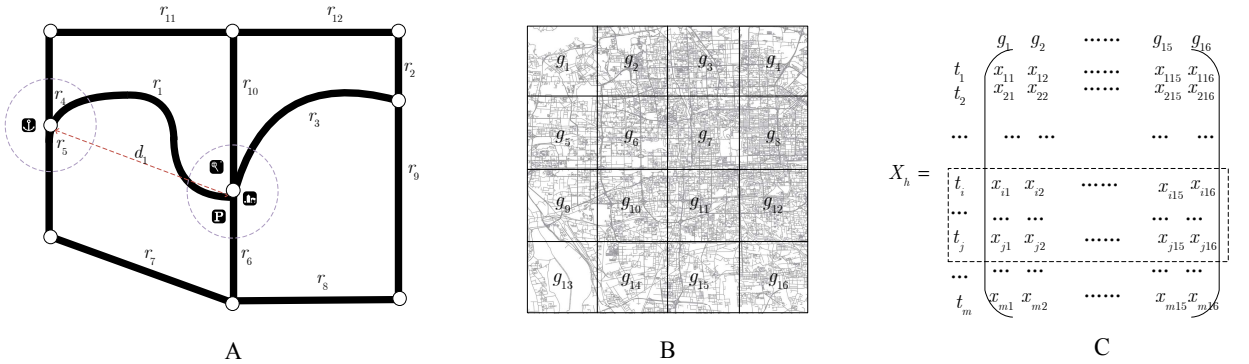

Fig. 4. Constructing context matrices

D. Tensor decomposition

In order to improve the accuracy of tensor decomposition, we combine the already processed $\mathcal{A}_r$ with the historical trajectory tensor $\mathcal{A}_{\mathrm{h}}$, as shown in Fig 2, to form a tensor $\mathcal{A} \in \mathbb{R}^{N\times M\times 2L}$, and use the GPS trajectory dataset to extract the contextual information, the tensor $\mathcal{A}$ performs decomposition. Since this method utilizes context information, the optimized objective function is (6).

$$\mathcal{L}(S,R,U,T,F,G) = \frac{1}{2}\| \mathcal{A}_r - S \times_R R \times_U U \times_T T \|^2 + \frac{\lambda_1}{2}\| X - TG \|^2 + \frac{\lambda_2}{2}\| Y - RF \|^2 + \frac{\lambda_3}{2}(\| S \|^2 + \| R \|^2 + \| U \|^2 + \| T \|^2 + \| F \|^2 + \| G \|^2) \tag{6}$$

Where $\mathcal{A} \in \mathbb{R}^{N\times M\times 2L}$, $X \in \mathbb{R}^{2L\times P}$, $Y \in \mathbb{R}^{N\times Q}$. P is the number of grids divided as shown in Figure 4 B, and Q is the dimension of geographic features. Matrices $T \in \mathbb{R}^{2L\times d_T}$, $G \in \mathbb{R}^{d_T\times P}$, $R \in \mathbb{R}^{N\times d_R}$, $F \in \mathbb{R}^{d_R\times Q}$ are all potential factor matrices, using the formula $\mathcal{A}_{rec} = S \times_R R \times_U U \times_T T$ restores the tensor to obtain the element value of the invisible position, where $S \in \mathbb{R}^{d_R\times d_U\times d_T}$, $U \in \mathbb{R}^{M\times d_U}$.

Since the tensor is very large, the drivers and roads in a city are thousands. Decomposition of such a large tensor is time consuming, especially if the tensor is particularly sparse, with only some locations having values. In the reference material of this paper, the whole city is divided into multiple grid partitions, and the tensor models are respectively established for the GPS trajectories of each partition. It has been confirmed that selecting a suitable partition separately decomposes the small tensor will not affect the accuracy of the original tensor decomposition [4]. That is, by selecting a suitable partition, each partition establishes a relatively small tensor, and each partition extracts its own contextual information for decomposition, which can solve the error and time-consuming problem caused by the decomposition of large and sparse tensors

V. Experiments

A. Datasets

The dataset we use is based on the actual case of the Beijing Urban Road Network. The city's road network covers a spatial range of 40×50km, with a total length of 21,985 kilometers. The GPS trajectory dataset used for the experiment was generated from Sept. 1 to Oct. 31, 2013. The number of GPS nodes reached 673,469,757, and the total length of the trajectories exceeded 26,218,407 kilometers. The average sampling frequency was 96 seconds per point [4], [17].

According to the reference, the morning peak time period is divided into four time periods from 7:00 to 9:00 [4], and each time period is 30 minutes. Because 4 has a good tradeoff between effectiveness and efficiency according to the past research [4]. We removed road segments that were traversed less than 50 times (less than once a day), and the travel time on these roads may be noisy, either because of inappropriate map matching algorithms, or because of these roads cannot be traveled by vehicles.

The experiment of this paper selects the dataset with 7400 taxis and 9992 roads in our model, and extract the context information X and Y from the above dataset. Table Ⅰ shows the statistics of the established tensors and matrices. As shown in Table 1, the tensor is particularly sparse, and only some of the positions are visible.

B. Verification indicator

In this paper we use mean absolute error (MAE) and root mean square error (RMSE) as the validation indicators for our model. The raw value of the processed data set is used as a base fact to test the accuracy of the estimate. The estimation formulas are (7) and (8), where y_i and $\widehat{y}_i$ are the ground truth and its estimation, respectively. As defined in Table Ⅱ.

$$MAE = \frac{\sum_i |y_i - \widehat{y}_i|}{n} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_i (y_i - \widehat{y}_i)^2}{n}} \tag{8}$$

TABLE I. STATISTICS ON THE DATA MODELS

	Size	**Average non-zero entries**
$\mathcal{A}_r$	7499×9992×4	7.0065389e-04
$\mathcal{A}_h$	7499×9992×4	3.1042302e-03
X	8×9	1
Y	7499×18	1

TABLE II. VALIDATION METRICES

	Best	**Worst**
MAE	The smaller, the better.	
RMSE	The smaller, the better.	

TABLE III. TRAVEL TIME ESTIMATION ACCURACIES FOR DIFFERENT TIME SLOTS

Time slots	**MAE**	**RMSE**
1(7:00-7:30)	32.1718	49.4520
2(7:30-8:00)	31.9325	60.9153
3(8:00-8:30)	34.7732	80.6516
4(8:30-9:00)	26.9320	58.7127
Average	31.4541	62.4329

TABLE IV. PERFORMANCES COMPARISON FOR DIFFERENT MODELS.

Methods	Time slot 1		Time slot 2	
	MAE	**RMSE**	**MAE**	**RMSE**
TD	38.4647	57.6113	39.0498	66.3487
CATD	38.0231	57.1282	38.6528	66.0757
MDCATD	**37.5085**	**56.5621**	**38.1757**	**65.7424**
CABTD	32.4420	49.8761	32.5628	61.4596
MDCABTD	**32.1718**	**49.4520**	**31.9325**	**60.9153**
	Time slot 3		**Time slot 4**	
TD	43.7600	89.9982	31.7715	62.4658
CATD	43.2410	89.6218	31.3388	62.1951
MDCATD	**42.6241**	**89.1436**	**30.8435**	**61.8823**
CABTD	35.3911	81.6086	27.2345	59.0341
MDCABTD	**34.7732**	**80.6516**	**26.9320**	**58.7127**

TABLE V. COMPARISON OF MODEL RESULTS FOR DIFFERENT NUMBERS OF KNOWN VALUES

Randomly delete(%)	Methods	MAE	RMSE
30	TD	33.5659	80.8128
	CATD	32.9161	80.4539
	MDCATD	**32.1565**	**80.0109**
	CABTD	27.4715	74.2934
	MDCABTD	**27.2112**	**73.5072**
60	TD	34.3293	83.1374
	CATD	33.9437	82.9452
	MDCATD	**33.3947**	**82.6674**
	CABTD	28.0527	78.9031
	MDCABTD	**27.1917**	**78.0146**
90	TD	32.0777	55.4096
	CATD	31.6367	55.0700
	MDCATD	**31.0663**	**54.5962**
	CABTD	26.6512	49.5867
	MDCABTD	**25.7923**	**48.2912**

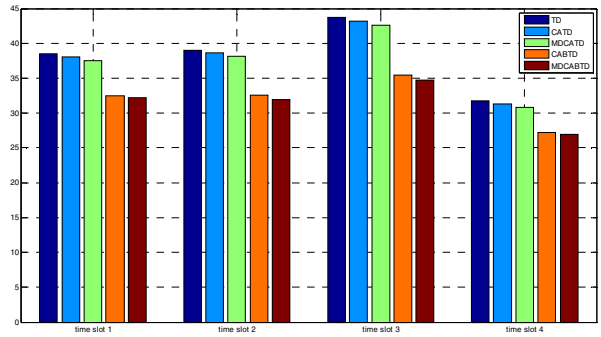

Fig. 5 Mean absolute error of different methods at different time slots.

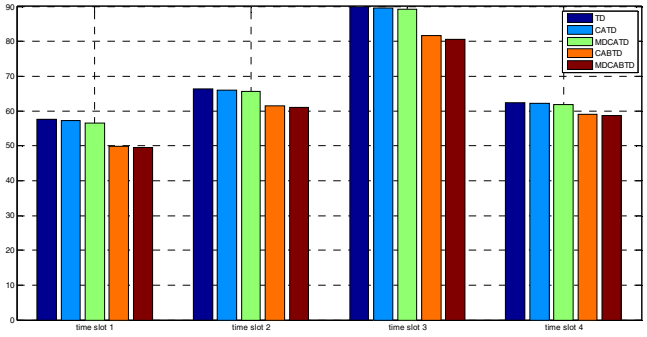

Fig. 6 Root mean square error of different methods at different time slots.

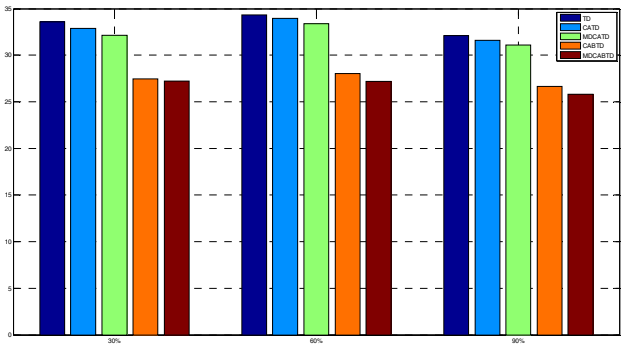

Fig. 7 Mean absolute error of different methods.

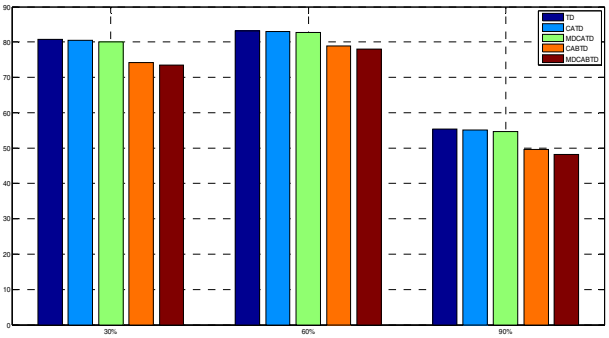

Fig. 8 Root mean square error of different methods.

C. Results

Compared the proposed method with the following three methods to prove its advantage in travel time estimation:

1) Tucker decomposition: A simple tensor Tucker decomposition estimates an unknown entry based only on its own non-zero entry, regardless of any contextual features [3].

2) Context-aware tensor decomposition (CATD): Based on the tensor Tucker decomposition, the contextual information *X* and *Y* are extracted from the GPS trajectory dataset to decompose the tensor. It is a decomposition method based on contextual information considering the geographical features between road segments and the space-time correlation between time periods [4].

3) Block term decomposition and context-aware Tucker decomposition (BTD and CATD combination): BTD unifies the Tucker and CANDECOMP/PARAFAC decompositions [2], and approximates by a sum of low multi-linear rank terms [19]. Unknown entries are not only based on their own non-zero entries estimation, but also consider the contextual information of the GPS trajectories [21].

In order to test the accuracy of our model, we randomly delete 30% of the known values from different time periods of the real-time tensor $\mathcal{A}_r$, and then predict the values through our model, using their original values as a ground truth to calculate both mean absolute error (MAE) and root mean square error (RMSE),where the root mean square error (RMSE) is widely used to measure the error of tensor decomposition. The estimation accuracies of travel times in different time slots are given in different rows in Table Ⅲ. It is evident that travel time in all time slots can be estimated by the proposed model with a high accuracy in terms of all validation metrics.

The performance comparison of different models are reported in Table Ⅳ. The estimation accuracies in terms of MAE, RMSE in the four time slots are all given, and the proposed model is referred to MDCATD and MDCABTD. The table shows that Tucker decomposition is worse than the improved method context-aware tensor decomposition. The method BTD extending CATD by factorizing the travel time tensor into a sum of low multi-linear rank terms, travel time estimations of BTD is more accurate than CATD. However, benefiting of the original tensor pretreatment by matrix decomposition and iteration initialization preprocessing, our model is more accurate than the above methods. Fig 5 and 6 show the MAE and RMSE of each method at different time slots more intuitively. Two histograms show that the proposed method has higher accuracy.

As for the table Ⅴ, except for randomly deleting 30% of the known values, ,we also randomly removed 60%, 90% of known items from the last period of the real-time, respectively, and tested the results of using these methods separately. As shown in Fig 7 and Fig 8, two histograms show that the proposed method, using the core tensor and three matrices iterated by CATD as the initial iteration value and filling the unknown position of the tensor in advance, is still better.

VI. Conclusion

This paper provides a method of using matrix decomposition to fill the invisible position of the tensor in advance, and estimating the personalized travel time of users based on contextual perception. According to the known dataset, the dataset is processed into a third-order tensor representing the driver, the road and the period. The tensor slicing operation is used to take out the matrix of the invisible position, and the matrix decomposition is used to fill the missing elements. Through the context information extracted from the GPS trajectories, the Tucker decomposition method of the gradient descent is used to cooperatively decompose the new tensor. And from the results, the method proposed in this paper can predict more accurate travel time for users. The method is applied to the road network of Beijing. Several trajectories and travel time of several drivers collected in the city for several months are used to provide a large and sparse dataset. [4]

For the prediction of travel time, there are still many improvements that can be made, such as the impact of weather on travel time, and the correlation between taxi drivers. For example, during peak hours, drivers will tell each other that a certain road is too crowded. It is necessary

to consider the travel time of other roads, etc. These can be future research trends for potential problems in travel time prediction.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093, 61402310, 61672364 and 61672365, by the Soochow Scholar Project of Soochow University, by the Six Talent Peak Project of Jiangsu Province of China and by the Graduate Innovation and Practice Program of colleges and universities in Jiangsu Province.

REFERENCES

[1] F. Z. Li, L. Zhang, and Z. Zhang, “Lie Group Machine Learning,” Walterde Gruyter GmbH & Co KG, 2018.

[2] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness," SIAM Journal on Matrix Analysis and Applications, vol. 30, no. 3, pp.1033-1066, 2008.

[3] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, and Q. Zhao, et al. "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," IEEE Signal Processing Magazine, vol. 32, no. 2, pp.145-163, 2015.

[4] Y. L. Wang, Z. Yu, and X. X. Ye, "Travel time estimation of a path using sparse trajectories," Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp.25-34, 2014.

[5] Z. Yu, "Trajectory data mining: an overview," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 6, no. 3, pp.29, 2015.

[6] X. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, "Urban link travel time estimation using large-scale taxi data with partial information," Transportation Research Part C: Emerging Technologies, vol. 33, pp.37-49, 2013.

[7] C. C. Aggarwal, Recommender Systems, Cham: Springer International Publishing, 2016.

[8] I. Sanaullah, M. Quddus, and M. Enoch, "Developing travel time estimation methods using sparse GPS data," Journal of Intelligent Transportation Systems, vol. 20, no. 6, pp.532-544, 2016.

[9] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial–temporal correlation," Physica A: Statistical Mechanics and its Applications, vol. 446, pp.54-63, 2016.

[10] S. K. S. Fan, C. J. Su, H. T. Nien, P. F. Tsai, and C. Y. Cheng, "Using machine learning and big data approaches to predict travel time based on historical and real-time data from Taiwan electronic toll collection," Soft Computing, vol. 22, no. 17, pp.5707-5718, 2018.

[11] T. G. Kolda, and B. W. Bader, "Tensor decompositions and applications," SIAM Review, vol. 51, no. 3, pp.455-500, 2009.

[12] W. H. Lee, S. S. Tseng, and S. H. Tsai, "A knowledge based real-time travel time prediction system for urban network," Expert Systems with Applications, vol. 36, no. 3, pp.4239-4247, 2009.

[13] K. Tang, S. Chen, and A. J. Khattak, "Personalized travel time estimation for urban road networks: A tensor-based context-aware approach," Expert Systems with Applications, vol. 103, pp.118-132, 2018.

[14] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," Transportation Research Part B: Methodological, vol. 46, no. 9, pp.1097-1122, 2012.

[15] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," IEEE Transactions on Intelligent Transportation Systems, vol.13, no. 4, pp.1679-1693, 2012.

[16] E. Jenelius, and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," Transportation Research Part B: Methodological, vol. 53, pp.64-81, 2013.

[17] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. ACM, pp.713-724, 2013.

[18] F. Zheng, and H. Van Zuylen, "Urban link travel time estimation based on sparse probe vehicle data," Transportation Research Part C: Emerging Technologies, vol. 31, pp.145-157, 2013.

[19] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," IEEE Journal of Selected Topics in Signal Processing, vol. 9, no. 4, pp.586-600, 2015.

[20] Y. Kui, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, pp.431-443, 2012.

[21] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," Proceedings of the Fourth ACM Conference on Recommender Systems. ACM, pp.79-86, 2010

[22] Z. Ma, H. N. Koutsopoulos, L. Ferreira, and M. Mesbah, "Estimation of trip travel time distribution using a generalized Markov chain approach," Transportation Research Part C: Emerging Technologies, vol. 74, pp.1-21, 2017.

[23] D. M. Miranda, and S. V. Conceição, "The vehicle routing problem with hard time windows and stochastic travel and service time," Expert Systems with Applications, vol. 64, pp.104-116, 2016.

[24] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Non-parametric estimation of route travel time distributions from low-frequency floating car data," Transportation Research Part C: Emerging Technologies, vol.58, pp.343-362, 2015.

[25] Y. Jing, Z. Yu, C. Zhang, X. Xie, and G. Z. Sun, "An interactive-voting based map matching algorithm," Proceedings of the 2010 Eleventh International Conference on Mobile Data Management. IEEE Computer Society, pp.43-52, 2010.

[26] D. Woodard, G. Nogin, P. Koch, D. Racz, M. Goldszmidt, and E. Horvitz, "Predicting travel time reliability using mobile phone GPS data," Transportation Research Part C: Emerging Technologies, vol. 75, pp.30-44, 2017.

[27] B. Yang, C. Guo, Y. Ma, and C. S. Jensen, "Toward personalized, context-aware routing," The VLDB Journal—The International Journal on Very Large Data Bases, vol. 24, no. 2, pp.297-318, 2015.

[28] M. Rahmani, H. N. Koutsopoulos, and E. Jenelius, "Travel time estimation from sparse floating car data with consistent path inference: A fixed point approach," Transportation Research Part C: Emerging Technologies, vol. 85, pp.628-643, 2017.