Local Tensor Completion Based on Locality Sensitive Hashing

Kun Xie[1,2,3], Yuxiang Chen[1], Xin Wang[3], Gaogang Xie[4], Jigang Wen[4], Dafang Zhang[1]
[1] College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
[2] CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, China
[3] Department of Electrical and Computer Engineering, State of New York University at Stony Brook
[4] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
xiekun@hnu.edu.cn, csyuxiang1988@gmail.com, x.wang@stonybrook.edu, xie@ict.ac.cn, wenjigang@ict.ac.cn,
dfzhang@hnu.edu.cn

***Abstract*—Tensor completion can be applied to fill in the missing data, which is import for many data applications where the data are incomplete. To infer the missing data, existing tensor-completion algorithms generally assume that the tensor data have global low-rank structure and apply a single model to fit the overall observed data through the global optimization. However, there are different correlation levels among application data, thus the ranks of some sub-tensors can be even lower relative to that of the large tensor. Fitting a single model to all data will compromise the performance of data recovery. To increase the accuracy in missing data recovery, we propose to apply local tensor completion (Local-TC) to recover data from sub-tensors, with each containing data of higher correlations. Although promising, as the tensor data are only organized logically, it is difficult to determine the relationship among data. We propose to exploit locality-sensitive hash (LSH) to quickly find the data correlation and reorganize tensor data, based on which data entries with high correlations are put into the same sub-tensor. The experiment results demonstrate that Local-TC is very effective in increasing the recovery accuracy.**

***Keywords*-Tensor Completion;Locality Sensitive Hashing;**

I. Introduction

Data missing is often observed in many data related applications. Recovering missing data from its partial samples is a fundamental problem, and has attracted continuous attentions in data analysis. Although matrix completion approaches [1]–[4] present good performance under low data missing ratio, the recovery performance suffers when there is a large amount of data missing.

For better data recovery, it helps to represent the data as a higher dimensional array called *tensor*, a higher-order generalization of vector and matrix. Tensor has proven to be a good data structure for dealing with multi-dimensional data in various fields [5], [6]. As tensor completion can well take advantage of the multilinear structures in the data to provide higher information precision thus higher performance in missing data recovery, it continues to draw attention for use in data analyses. For example, our recent study [7], [8] models the end-to-end traffic volume data between network Origin (source) nodes and Destination nodes in the Internet as a 3-way traffic tensor, which exploits the temporal stability, spatial correlation, and traffic periodicity for more accurate Internet traffic data recovery.

Despite that it is more promising to recover missing data through tensor completion, existing tensor completion algorithms generally have the strong assumption that the tensor data have a global low-rank structure, and try to find a single and global model to fit the data of the whole tensor. However, in many practical applications, data entries in a large tensor may have different levels of correlation. Taking the 3-way Internet traffic tensor as an example, its three dimensions are origin destination (OD) pairs, time slots, and days. A subset of OD pairs may have similar end-to-end traffic behaviors in a subset of time slots (i.e., working time) of a subset of days (i.e., working day). Such a subset of OD pairs/time slots/days may construct a sub-tensor with a lower rank.

Although it is well acknowledged that the ranks of sub-tensors would be lower if the data in the sub-tensors have higher correlation, we are not aware of any work that exploit this feature in tensor completion. The sub-tensors with lower ranks are not accurately estimated by conventional tensor completion performed based on a higher global rank. Intuitively, different sub-tensors contain different sets of data. In an extreme case, applying a global tensor completion model to a large tensor consisting of multiple low-rank sub-tensors is akin to fitting one single model to the concatenation of all the data sets. As different data sets have different structure features, a single model can not capture features of all data sets, which results in low missing data recovery accuracy.

To take advantage of the lower ranks in sub-tensors for more accurate missing data recovery, we propose a local tensor completion scheme, termed as Local-TC. Although it is promising, as a sub-tensor should be built based on similarity and correlation among the tensor data, there is a challenge to efficiently calculate the correlation among the large amount of tensor data. In addition, the tensor data are only organized logically. It is unknown which part of data actually have the close relationship. Thus it is hard to separate a large tensor with partially-observed data into a set of sub-tensors, with closely-related data grouped into the same sub-tensor.

To address the set of challenges above, taking advantage

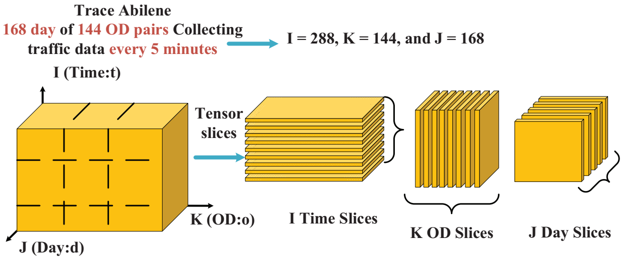
2375-026X/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDE.2018.00113

of the property of locality-sensitive hash (LSH) functions, we propose three novel LSH tables to re-order the three dimensions (corresponding to OD pairs, time slots, and days) in the traffic tensor very quickly, based on which the data in the tensor is re-organized so that adjacent data entries in the tensor have closer correlations. With the re-organized tensor, we propose a novel correlation-based sub-tensor formulation algorithm by putting OD pairs, time slots, and days with higher correlations into the same local sub-tensor. As data similarity reduces the ranks of local sub-tensors, the missing data in sub-tensors can be more accurately inferred compared to the completion of the large tensor directly.

II. Problem

To illustrate the scheme of Local-TC, we present our scheme using the Internet traffic data recovery as an example, where the tensor records the data volume between every source and destination pair and the missing traffic volume data are inferred based on a set of observed entries.

For a network consisting of N nodes, there are $N \times N$ origin and destination (OD) pairs. Based on the analyses of real traffic trace, our recent work on tensor completion [7], [8] reveals that the traffic data have the features of temporal stability, spatial correlation, and periodicity. To fully exploit these traffic features for accurate traffic data recovery, we can model the traffic data as a 3-way traffic tensor $\mathcal{M} \in \mathbb{R}^{I \times J \times K}$ (Fig.1(a)), where K corresponds to the number of the origin and destination (OD) pairs in the network, and there are J days to consider with each day having I time slots. Fig.1(b) shows that the traffic tensor can be divided into slices along the dimensions of time, OD pair, and day, which are exploited to find correlations of data points in the three domains respectively through locality-sensitive hash. For the Abilene trace [9], $I = 288$, $K = 144$, and $J = 168$. Our Local-TC, however, is general and does not depend on how the tensor is modeled.


Trace Abilene
168 day of 144 OD pairs Collecting traffic data every 5 minutes
I = 288, K = 144, and J = 168
I (Time:t)
Tensor slices
K (OD:o)
J (Day:d)
I Time Slices
K OD Slices
J Day Slices

(a) Traffic tensor (b) Traffic tensor slice

Fig. 1. 3-way monitoring tensor.

As mentioned in Section I, to reduce the high network monitoring and communication cost, sample-based network monitoring strategy is often adopted. As a result, the network monitoring data are usually incomplete.

Among various methods to fill in the missing tensor items, we adopt tensor factorization to infer the missing entries of a tensor. Given a traffic monitoring tensor $\mathcal{M}$, the (i, j, k)-th entry, m_{ijk}, represents the data measurement taken from OD pair i at the time slot j of the day k. If there are no monitoring data between a pair of nodes in a given time interval, it leaves the corresponding entry in $\mathcal{M}$ empty. Let Ω be the set of indices of the sample entries in $\mathcal{M}$. The missing data recovery problem is to recover the tensor $\mathcal{M}$ based on its samples through the tensor factorization. Based on CP decomposition, the problem is defined as

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{B},\mathbf{C}} L(\mathbf{A},\mathbf{B},\mathbf{C}) \\ &s.t. L(\mathbf{A},\mathbf{B},\mathbf{C}) = \begin{array}{l} \left\|(\mathcal{M} - [\mathbf{A},\mathbf{B},\mathbf{C}])_\Omega\right\|_F^2 \\ +\alpha\|\mathbf{A}\|_F^2 + \alpha\|\mathbf{B}\|_F^2 + \alpha\|\mathbf{C}\|_F^2 \end{array} \end{aligned} \tag{1}$$

In (1), $\left\|(\mathcal{M} - [\mathbf{A},\mathbf{B},\mathbf{C}])_\Omega\right\|_F^2$ is the recovery loss (error), $\alpha\|\mathbf{A}\|_F^2 + \alpha\|\mathbf{B}\|_F^2 + \alpha\|\mathbf{C}\|_F^2$ is the regularization added to prevent the over-fitting problem. There is an over-fitting of data when the recovery losses (errors) corresponding to the sampled tensor elements are very small while the errors for the data items inferred from existing samples are very large.

Problem in (1) requires the finding of the factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ to approximate the tensor $\mathcal{M}$ with the minimum $L(\mathbf{A},\mathbf{B},\mathbf{C})$, where R is the tensor rank. After obtaining the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, the monitoring tensor can be recovered through

$$\hat{\mathcal{M}} = [\mathbf{A},\mathbf{B},\mathbf{C}], \tag{2}$$

where $\hat{\mathcal{M}}$ denotes the estimated monitoring tensor.

III. Solution

The tensor completion problem in (1) is established based on the assumption that the rank of the overall tensor is low. This global low-rank assumption serves as the base for traditional tensor completion algorithms to fit a single model with the overall tensor data through the global optimization. However, the data correlations in different neighborhood may be different, which causes the difference in the ranks of sub-tensors. A sub-tensor formed with closely related data would have lower rank, while simply applying the global optimization over the whole traffic tensor can not benefit from the higher local correlation among data.

In this work, we propose to design a *local tensor completion algorithm* that can take advantage of the low-rank structure of local tensor for more accurate missing data recovery. Obviously, the sub-tensor should be built based on the similarity and correlation among the tensor data. However, this requires the address of two challenging issues: ***1) how to efficiently determine the correlations among tensor data*, and *2) how to locate low-rank sub-tensors in a tensor.***

Samples taken from similar OD pairs/time slots/days have higher impacts on each other. To well exploit the correlations from the three corresponding domains for more accurate missing data recovery, we propose a novel LSH-facilitated local tensor completion model to form sub-tensors each containing similar OD pairs, time slots, and day times. Fig.2 illustrates the basic procedures of our solution.

Fig.2(a) shows the original traffic tensor to be recovered. In Fig.2(b), to facilitate the forming of correlation-aware sub-tensors, we propose the use of three LSH tables to re-order OD pairs, time slots, and the day time. The good property of the LSH guarantees that similar OD pairs/time slots/days are

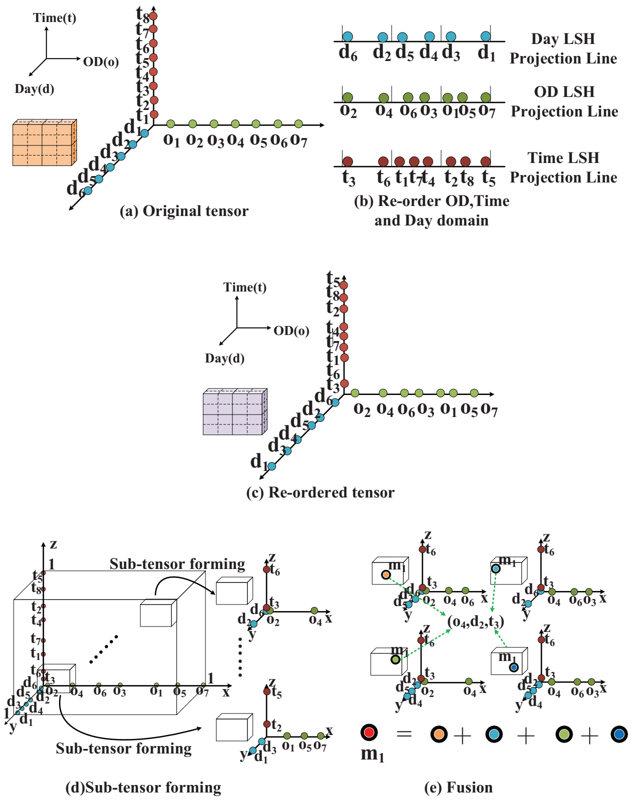


Fig. 2. Overview solution

projected to nearby locations in the LSH hash tables. LSH [10] is based on the simple idea that, if two data points are close together, then after a "projection" operation, they will remain close. To group similar OD pairs together, we apply LSH to the OD pair slice (shown in Fig.1(b)) of the tensor. Given a traffic tensor $\mathcal{M}$ in Fig.1, the slice corresponding to an OD pair i is an $I \times J$ matrix that contains the data measuring the OD pair in J days with each day having I time slots. More specifically, we take the following procedures to map the OD pairs into the LSH table.

1) ***Projecting OD pairs to a line***. Given an OD pair slice $\mu_i \in \mathbb{R}^{I\times J}\,(1 \le i \le K)$, after the vectorization, it can be denoted as a vector $\vec{\mu}_i \in \mathbb{R}^{IJ}\,(1 \le i \le K)$. We define the LSH hash function as

$$h_{\vec{a}}(\vec{\mu}_i) = \vec{a}\cdot\vec{\mu}_i \tag{3}$$

Eq(3) applies a scalar dot operation to project the OD pair $\vec{\mu}_i$ to a point on a straight line, with the point position determined by a random vector $\vec{a}$ whose entries are drawn independently from $\mathcal{N}(0,1)$. In this paper, we call this line the *projection line*.

2) ***Building the LSH table***. We denote the first projected value and the last projected value on the line as p_s and p_e, respectively. We will show our algorithm in partitioning the tensor to form sub-tensors. Each partition includes a group of data from similar OD pairs which also have higher correlation along the dimensions of time slots and days. Given the total number of groups to divide, λ, we partition the projection line between p_s and p_e into λ parts to build the hash table, with the bucket width in the table being $\frac{p_e - p_s}{\lambda}$.

In Fig.2(c), according to the LSH tables, we re-order data in the tensor so that adjacent data entries in the tensor have closer correlations. Based on the re-ordered traffic tensor in Fig.2(c), we propose a sub-tensor formulation algorithm to build low-rank sub-tensors from the re-ordered large tensor in Fig.2(d). By putting OD pairs/time slots/days with higher correlation into the same sub-tensor, the rank of the sub-tensor would be much lower, which further helps to more accurately infer the missing data.

After the tensor re-ordering, to ensure that every sub-tensor has stronger correlation for better missing entry recovery, a straightforward way for sub-tensor formulation is to directly partition the big tensor based on buckets of the LSH tables, with each sub-tensor consisting of one bucket of OD pairs, time slots, and days.

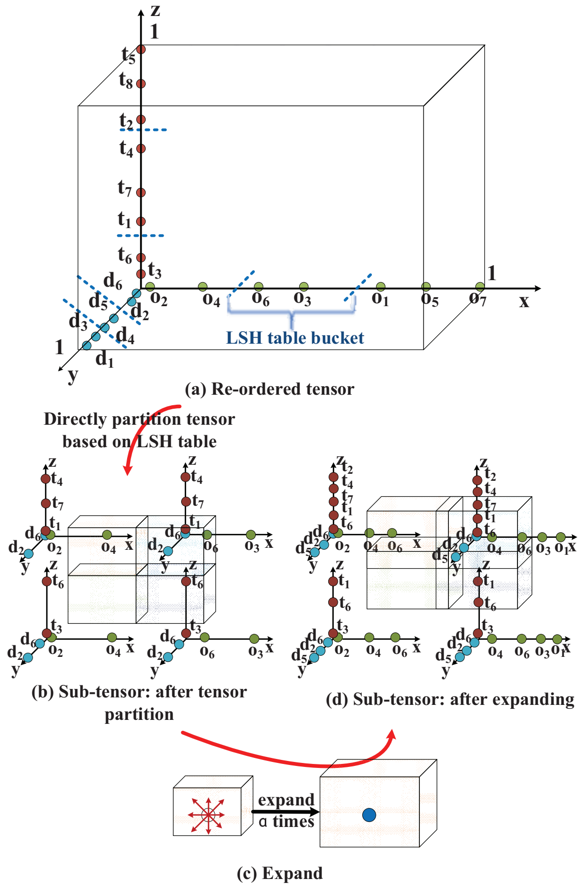

LSH table bucket
(a) Re-ordered tensor
Directly partition tensor based on LSH table
(b) Sub-tensor: after tensor partition
(d) Sub-tensor: after expanding
expand α times
(c) Expand

Fig. 3. Sub-tensor formulation.

Although this straightforward way allows that the large tensor is fully covered by the sub-tensors partitioned, the direct partition may lead to some undesirable results. For example, in Fig.3(b), o_4 in the first bucket of OD LSH table is close to o_6 in the second bucket, but its impact on recovering the missing data on the column o_6 is not considered. As an alternative way of partition, o_4 and o_6 are put into two adjacent sub-tensors, so they can exploit overlapping entries for a possibly better recovery performance.

To address the above issue, we propose an expanding principle. As shown in Fig.3(c), the coverage of each sub-tensor under the direct partition is expanded α times with $\alpha > 1$ to make adjacent OD pairs, time slots, and days included in all the adjacent sub-tensors. As a result, the sub-tensors have overlaps, and the missing data on the overlapping area will be recovered by multiple sub-tensors.

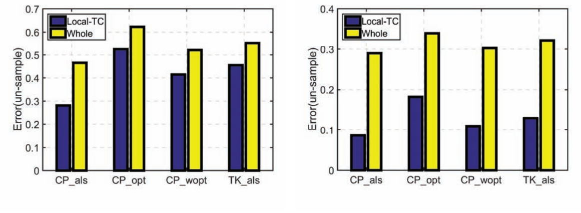
In Fig.2(e), after inferring missing items in sub-tensors, we fuse their overlapped entries to obtain the final data.

IV. Experiment

We use the public traffic traces Abilene [9] and GÈANT [11] to evaluate the performance of Local-TC.

In the experiment, we first randomly select sub set of the trace data as the measurement samples, and then apply the proposed Local-TC to recover the full traffic monitoring data from partial measurement samples. Then, using the raw traffic trace data as a reference, we evaluate the performance by comparing the recovered data with the original data trace. In all the experiments, we set the default sample ratio to be 50%.

To evaluate the accuracy of different tensor completion algorithms, we use the relative error metric defined as follows: **Error(un-sample)** $= \frac{\sqrt{\sum_{(i,j,k)\in\overline{\Omega}} (m_{i,j,k} - \hat{m}_{i,j,k})^2}}{\sqrt{\sum_{(i,j,k)\in\overline{\Omega}} (m_{i,j,k})^2}}$ where $1 \leq i \leq I$, $1 \leq j \leq J$, $1 \leq k \leq K$. $m_{i,j,k}$ and $\hat{m}_{i,j,k}$ denote the (i,j,k)-th element of the raw data tensor $\mathcal{M}$ and the tensor $\hat{\mathcal{M}}$ recovered through the tensor completion.



(a) Abilene

(b) GÈANT

Fig. 4. Performance under different tensor completion algorithms.

To evaluate the effectiveness of Local-TC, besides CP_{als}, we implement other three tensor completion algorithms including CP_{opt}, CP_{wopt}, and TK_{als} under our Local-TC scheme. The first three (CP_{opt}, CP_{wopt}, and CP_{als}) are designed based on CP model, the last TK_{als} is designed based on the Tucker model. These algorithms are applied to complete the sub-tensors formed from the large tensor based on Local-TC. We denote these implementations Local-TC in the figure. Then, for performance comparison, we also implement the tensor completion algorithms directly using the whole large tensor without tensor partition, denoted Whole in the figure.

Compared with the tensor completion algorithms executed using the whole data, our Local-TC scheme can bring closer the correlations among entries in the sub-tensors. Thus tensor completion algorithms under Local-TC can more accurately recover the missing data. All the performance results (in Fig.IV) demonstrate that our LSH-facilitated sub-tensor forming scheme is very effective in improving the recovery accuracy, and our Local-TC scheme is general without depending on the underlying tensor completion algorithms.

V. Conclusion

In this paper, we propose a local tensor completion scheme (Local-TC) to infer missing data entries from sub-tensors formulated with more closely related data. In order to identify data with higher correlation and form sub-tensors with lower ranks, we propose several novel techniques with the facilitation of LSH functions: LSH-based tensor re-organization and correlation-aware sub-tensor formulation. To the best of our knowledge, Local-TC is the first study that takes advantage of the local low-rank structure hidden in the large tensor for accurate missing data recovery.

Acknowledgment

The work is supported by the National Natural Science Foundation of China under Grant Nos.61572184, 61725206, 61472130, 61472131, and 61772191, Hunan Provincial Natural Science Foundation of China under Grant No.2017JJ1010, Beijing Natural Science Foundation under Grant No.4162057, Science and Technology Key Projects of Hunan Province under Grant No.2015TP1004, U.S. ONR N00014-17-1-2730, NSF ECCS 1408247, CNS 1526843, and ECCS 1731238, and the open project funding (CASNDST201704) of CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences.

References

[1] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, “Low cost and high accuracy data gathering in wsns with matrix completion,” *IEEE Transactions on Mobile Computing*, DOI:10.1109/TMC.2017.2775230.

[2] K. Xie, X. Ning, X. Wang, D. Xie, J. Cao, G. Xie, and J. Wen, “Recover corrupted data in sensor networks: A matrix completion solution,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1434–1448, 2017.

[3] K. Xie, L. Wang, X. Wang, G. Xie, G. Zhang, D. Xie, and J. Wen, “Sequential and adaptive sampling for matrix completion in network monitoring systems,” in *IEEE INFOCOM*, 2015.

[4] H. Li, K. Li, A. Jiyao, and K. Li, “Msgd: A novel matrix factorization approach for large-scale collaborative filtering recommender systems on gpus,” *IEEE Transactions on Parallel and Distributed Systems*, DOI: 10.1109/TPDS.2017.2718515.

[5] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *Signal Processing Magazine, IEEE*, vol. 32, no. 2, pp. 145–163, 2015.

[6] Y. Han and F. Moutarde, “Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization,” *International Journal of Intelligent Transportation Systems Research*, vol. 14, no. 1, pp. 36–49, 2016.

[7] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, and G. Zhang, “Accurate recovery of internet traffic data: A tensor completion approach,” in *IEEE INFOCOM*, 2016.

[8] X. Kun, P. Can, W. Xin, X. Gaogang, and W. Jigang, “Accurate recovery of internet traffic data under dynamic measurements,” in *IEEE INFOCOM*, 2017.

[9] “The abilene observatory data collections. http://abilene. internet2.edu /observatory/data-collections.html.”

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *ACM SCG*, 2004.

[11] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, “Providing public intradomain traffic matrices to the research community,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.