Multi-version Tensor Completion for Time-delayed Spatio-temporal Data

Cheng Qian[1*], Nikos Kargas[2*] Cao Xiao[1], Lucas Glass[1], Nicholas D. Sidiropoulos[3], Jimeng Sun[4]

[1] ACOE, IQVIA; [2]Dept of ECE, University of Minnesota Twin Cities; [3]Dept of ECE, University of Virginia; [4]Dept of CS, University of Illinois Urbana-Champaign

Background

Real-world spatio-temporal data is often incomplete or inaccurate due to various data loading delays.

Recovering such missing or noisy (under-reported) elements of the input tensor can be viewed as a generalized **tensor completion** problem.

Existing tensor completion methods usually assume that

i) missing elements are randomly distributed

ii) noise for each tensor element is i.i.d. zero-mean.

Both assumptions can be violated for spatio-temporal tensor data.

We often observe multiple versions of the input tensor with different under-reporting noise levels.

Problem Statement

Given a spatio-temporal tensor of I locations and J features over time, we introduce the following time concepts:

- **Generation date (GD)** is the time when data items are generated.
- **Loading date (LD)** is when the data items are received.
- At loading date t:
  - The observed tensor $\underline{\mathbf{Z}}_t \in \mathbb{R}^{I \times J \times S_t}$
  - The ground-truth tensor $\underline{\tilde{\mathbf{Z}}}_t \in \mathbb{R}^{I \times J \times S_t}$
  - The update tensor $\underline{\mathbf{X}}_t \in \mathbb{R}^{I \times J \times K \times S_t}$

Challenges:

1) The latest frontal slabs of $\underline{\mathbf{Z}}_t$ are under-reported and thus very noisy.
2) The noise distribution is unknown in practice.
3) The dimension corresponding to the GDs in $\underline{\mathbf{Z}}_t$ is gradually growing as t increases and more data are introduced.

The task is to estimate $\underline{\tilde{\mathbf{Z}}}_t$

Related work

- Tensor completion [Almutairi et al., 2017, Lacroix et al., 2018].
- Joint tensor tracking and imputation [Song et al., 2017].
- Nonlinear (neural network based) tensor completion [Liu et al., 2019].

Approach

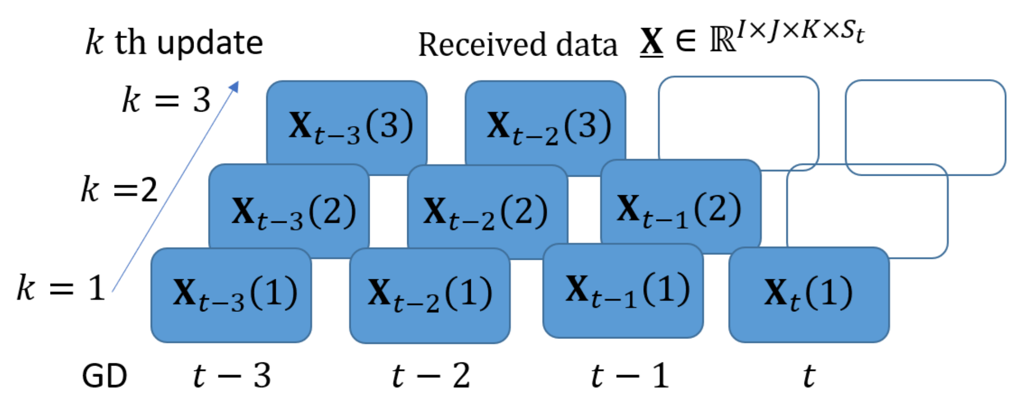

Fig. 1 The updates for the data on GD t.

The key idea is to track the update tensor. We may assume that data corresponding to a given GD is updated at most K times. Then, transform the problem into **an equivalent 4-way tensor completion problem.** Finally, the target tensor can be obtained by **marginalization**!

Proposed Methods

1) Multi-version Tensor Completion (MTC)

In this work, we approximate $\underline{\mathbf{X}}_t$ using a low-rank CPD model

$$\underline{\mathbf{X}} = \sum_{f=1}^{F} \mathbf{A}(:,f) \circ \mathbf{B}(:,f) \circ \mathbf{C}(:,f) \circ \mathbf{D}(:,f),$$

where **A**, **B**, **C** and **D** are the factor matrices for location, feature, LD and GD, respectively. We propose to solve

$$\min_{\boldsymbol{\theta}, \underline{\mathbf{Y}}} \mathcal{F}(\boldsymbol{\theta}, \underline{\mathbf{Y}}) + \mathcal{R}(\boldsymbol{\theta})$$
$$\text{s. t. } \boldsymbol{\theta} \geq 0, \mathcal{P}_{\Omega_{\mathbf{s}}}(\underline{\mathbf{Y}}(:,:,:,s)) = \mathcal{P}_{\Omega_{\mathbf{s}}}(\underline{\mathbf{X}}(:,:,:,s)),$$
$$\forall \mathbf{s} = S-K+2, \ldots, S,$$

where θ stands for the unknown parameters, $\mathcal{R}(\cdot)$ is the regularization, and

$$\mathcal{F}(\boldsymbol{\theta}, \underline{\mathbf{Y}}) = \alpha \mathcal{F}_1(\boldsymbol{\theta}, \underline{\mathbf{Y}}) + (1-\alpha)\mathcal{F}_2(\boldsymbol{\theta}, \underline{\mathbf{Y}}),$$
$$\mathcal{F}_1(\boldsymbol{\theta}, \underline{\mathbf{Y}}) = \sum_{s=1}^{S-K+1} \|\underline{\mathbf{Y}}(:,:,:,s) - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{d}_s]\!]\|_F^2,$$
$$\mathcal{F}_2(\boldsymbol{\theta}, \underline{\mathbf{Y}}) = \sum_{s=S-K+2}^{S} \|\underline{\mathbf{Y}}(:,:,:,s) - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{d}_s]\!]\|_F^2$$

We solve this optimization problem using BSUM with closed-form expression for updating each variable.

$$\hat{\underline{\mathbf{Z}}} = \textstyle\sum_{k=1}^{K} \hat{\underline{\mathbf{Y}}}(:,:,k,:).$$

2) MTC-online

At *t+1*, a new update $\check{\underline{\mathbf{X}}}_{t+1}$ will be appended to $\underline{\mathbf{X}}_t$, we have

$$\mathrm{vec}(\check{\underline{\mathbf{X}}}_{t+1}) \approx (\mathbf{A}_t \odot \mathbf{B}_t \odot \mathbf{C}_t)(\mathbf{D}_{t+1}(S_{t+1},:))^T$$

We can solve a non-negative least squares problem to find the last row of $\mathbf{D}_{t+1}$, denoted by $\mathbf{d}_{t+1}$.

We initialize MTC using the latest estimate, i.e., $\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t, \mathbf{D}_t, \mathbf{d}_{t+1}$

implement MTC with one iteration to update all the factor matrices.

Results

Datasets:

- Semi-synthetic data
  - Covid-19, 77×32×442×10
  - Chicago-Crime, 51 × 3 × 200 ×8
- Real Spatio-temporal medical claims data, 3027 × 22 × 52 ×12

Baselines:

- Naïve, i.e., $\underline{\mathbf{Z}}_t$
- Structured Data Fusion (Tensorlab),
- COSTCO [Liu et al., 2019]
- ARIMA
- LSTM

Table 1 Performance comparison in static case

	Patient-Claims			Covid-19			Chicago-Crime		
Method	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2
MTC	**220.4**	**29.7**	**0.997**	**74.2**	**26.0**	**0.986**	**1.42**	0.57	**0.983**
Naive	1113.5	107.2	0.896	290.1	97.1	0.559	4.98	1.24	0.594
SDF (3-way)	1149.1	146.2	0.905	291.9	105.0	0.551	4.70	1.24	0.648
SDF (4-way)	278.7	31.5	0.995	101.7	31.8	0.974	1.46	**0.55**	0.981
COSTCO	633.5	96.4	0.972	203.1	99.3	0.877	2.43	0.66	0.908
ARIMA	524.2	66.5	0.981	283.2	99.8	0.780	3.63	1.55	0.915
LSTM	400.6	58.5	0.989	343.9	111.8	0.736	3.65	1.41	0.876

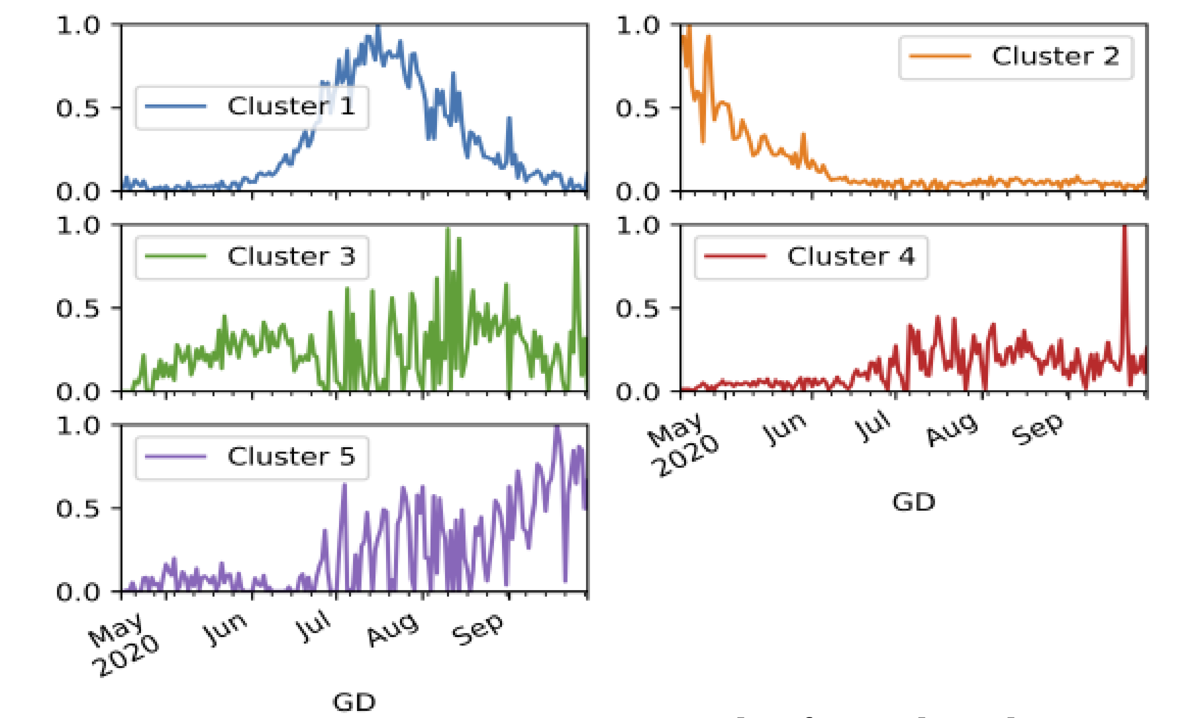

Fig. 2 : Latent components in GD mode of Covid-19 dataset.

Table 2 Performance comparison in dynamic case

Method	RMSE	MAE	R^2
Patient-Claims dataset			
MTC	$\mathbf{237.8 \pm 40.3}$	32.5 ± 3.0	$\mathbf{0.996 \pm 0.001}$
MTC-online	238.5 ± 37.5	$\mathbf{32.2 \pm 2.8}$	$\mathbf{0.996 \pm 0.001}$
SDF (4-way)	253.4 ± 59.5	34.6 ± 4.7	0.996 ± 0.002
Naive	$1,017.5 \pm 69.3$	99.8 ± 6.3	0.912 ± 0.012
SDF (3-way)	$1,052.8 \pm 89.1$	131.0 ± 15.3	0.904 ± 0.015
COSTCO	580.5 ± 37.2	87.9 ± 5.3	0.977 ± 0.003
ARIMA	553.1 ± 31.5	74.6 ± 5.2	0.979 ± 0.002
LSTM	692.1 ± 232.2	98.7 ± 31.8	0.952 ± 0.039

Conclusion

- This paper studies the problem of time-delayed spatio-temporal tensor data estimation.
- We formulated the problem as a multi-version tensor completion (MTC) problem by introducing an extra mode to capture the data updates.
- We proposed static and online version of MTC algorithms to tackle this problem.
- The experimental results on several real datasets have demonstrated the advantages of the proposed methods.