Inductive Framework for Multi-Aspect Streaming Tensor Completion with Side Information

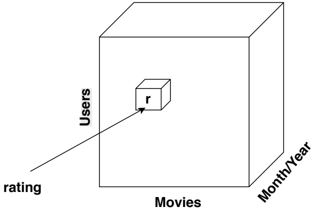
Madhav Nimishakavi[1] Bamdev Mishra[2] Manish Gupta[2]
Partha Talukdar[1]

[1]Indian Institute of Science [2]Microsoft

Outline

Introduction

- A **Tensor** is a multi-way extension of a matrix.



- Tensors are used for representing multidimensional data.

Introduction (cont.)

- In practice, many multidimensional datasets are often incomplete.
- **Tensor Completion** is the task of predicting or imputing missing values in a partially observed tensor.

Introduction (cont.)

- However, in many real world applications the data is dynamic. Some examples include,
  - Online recommendation systems.
  - Social networks.
  - ...
- **Dynamic Tensor Completion** is the task of predicting missing values in a dynamically growing partially observed tensor.

Introduction (cont.)

- Most of the existing works make an assumption that the tensor grows only in one mode.

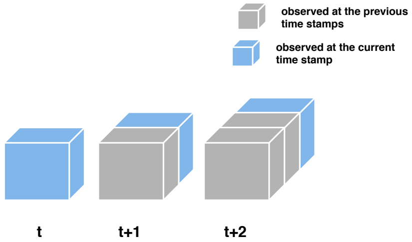


Figure : Streaming tensor sequence

- This assumption is restrictive !

Introduction (cont.)

- Recently Song et al. [4] proposed the more general Multi-aspect streaming tensor completion.

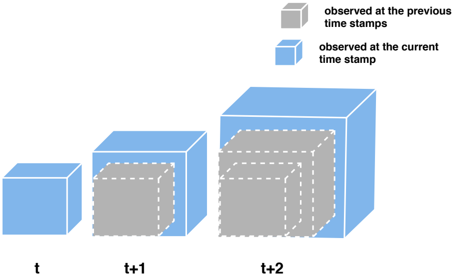


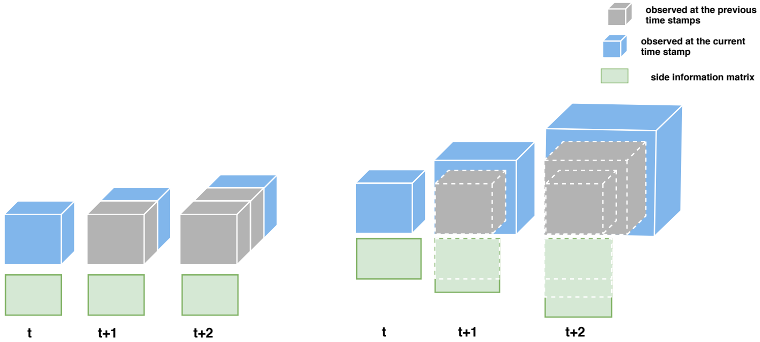
Figure : Multi-aspect streaming tensor sequence

Introduction (cont.)

- Besides the tensor, additional side information data is also available in the form of matrices in many applications.
  - For example, *movie* $\times$ *genre* matrix for online movie recommendation etc.
- Incorporating the side information matrices into tensor completion can help achieve better results, particularly in sparse settings.

Introduction (cont.)

- We propose a framework to handle the following sequences.



(a) Streaming sequence with side information

(b) Multi-aspect streaming sequence with side information

Preliminaries

Definition (Multi-aspect streaming Tensor Sequence) [4]: A tensor sequence of N^{th}-order tensors $\{\boldsymbol{\mathcal{X}}^{(t)}\}$ is called a multi-aspect streaming tensor sequence if for any $t \in \mathbb{Z}^{+}$, $\boldsymbol{\mathcal{X}}^{(t-1)} \in \mathbb{R}^{I_1^{t-1} \times I_2^{t-1} \times \ldots \times I_N^{t-1}}$ is the sub-tensor of $\boldsymbol{\mathcal{X}}^{(t)} \in \mathbb{R}^{I_1^{t} \times I_2^{t} \times \ldots \times I_N^{t}}$, i.e.,

$$\boldsymbol{\mathcal{X}}^{(t-1)} \subseteq \boldsymbol{\mathcal{X}}^{(t)}, \text{ where } I_i^{t-1} \leq I_i^{t}, \ \forall 1 \leq i \leq N.$$

Here, t increases with time, and $\boldsymbol{\mathcal{X}}^{(t)}$ is the snapshot tensor of this sequence at time t.

Preliminaries (cont.)

Definition (Multi-aspect streaming Tensor Sequence with Side Information) : Given a time instance t, let $\mathbf{A}_i^{(t)} \in \mathbb{R}^{I_i^t \times M_i}$ be a side information (SI) matrix corresponding to the i^{th} mode of $\boldsymbol{\mathcal{X}}^{(t)}$, we have,

$$\mathbf{A}_i^{(t)} = \begin{bmatrix} \mathbf{A}_i^{(t-1)} \\ \Delta_i^{(t)} \end{bmatrix}, \text{ where } \Delta_i^{(t)} \in \mathbb{R}^{[I_i^{(t)} - I_i^{(t-1)}] \times M_i}.$$

let the side information set $\mathcal{A}^{(t)} = \{\mathbf{A}_1^{(t)}, \ldots, \mathbf{A}_N^{(t)}\}$.
Given an N^{th}-order multi-aspect streaming tensor sequence $\{\boldsymbol{\mathcal{X}}^{(t)}\}$, we define a multi-aspect streaming tensor sequence with side information as $\{(\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{A}^{(t)})\}$.

Preliminaries (cont.)

Problem Definition: Given a multi-aspect streaming tensor sequence with side information $\{(\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{A}^{(t)})\}$, the goal is to predict the missing values in $\boldsymbol{\mathcal{X}}^{(t)}$ by utilizing only entries in the relative complement $\boldsymbol{\mathcal{X}}^{(t)} \setminus \boldsymbol{\mathcal{X}}^{(t-1)}$ and the available side information $\mathcal{A}^{(t)}$.

SIITA

- We propose Side Information infused Incremental Tensor Analysis (**SIITA**).

Property	TeCPSGD[3]	OLSTEC[2]	MAST[4]	AirCP[1]	SIITA
Streaming	✓	✓	✓		✓
Multi-Aspect Streaming			✓		✓
Side Information				✓	✓
Sparse Solution					✓

Table : Summary of different tensor streaming algorithms.

SIITA (cont.)

$$\min_{\substack{\mathcal{G}\in\mathbb{R}^{r_1\times\cdots\times r_N}\\ \mathbf{U}_i\in\mathbb{R}^{M_i\times r_i}, i=1:N}} F(\mathcal{X}^{(t)}, \mathcal{A}^{(t)}, \mathcal{G}, \{\mathbf{U}_i\}_{i=1:N}), \tag{1}$$

where

$$F(\mathcal{X}^{(t)}, \mathcal{A}^{(t)}, \mathcal{G}, \{\mathbf{U}_n\}_{i=1:N}) = \left\| \overbrace{\mathcal{P}_\Omega(\mathcal{X}^{(t)})}^{\text{observed tensor at t}} - \mathcal{P}_\Omega(\mathcal{G} \times_1 \overbrace{\mathbf{A}_1^{(t)}}^{\text{side info at t}} \mathbf{U}_1 \times_2 \ldots \times_N \overbrace{\mathbf{A}_N^{(t)}}^{\text{side info at t}} \mathbf{U}_N) \right\|_F^2 + \lambda_g \|\mathcal{G}\|_F^2 + \sum_{i=1}^{N} \lambda_i \|\mathbf{U}_i\|_F^2. \tag{2}$$

SIITA (cont.)

Since $\{(\boldsymbol{\mathcal{X}}^{(t-1)}, \mathcal{A}^{(t-1)})\} \subseteq \{(\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{A}^{(t)})\}$, we have

$$F(\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{A}^{(t)}, \boldsymbol{\mathcal{G}}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N}) = \overbrace{F(\boldsymbol{\mathcal{X}}^{(t-1)}, \mathcal{A}^{(t-1)}, \boldsymbol{\mathcal{G}}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N})}^{\text{term at time } t-1} + \underbrace{F(\boldsymbol{\mathcal{X}}^{(\Delta t)}, \mathcal{A}^{(\Delta t)}, \boldsymbol{\mathcal{G}}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N})}_{\text{delta term between } t \text{ and } t-1} \quad (3)$$

SIITA (cont.)

We propose the following incremental update scheme,

$$\left.\begin{cases} \mathbf{U}_i^{(t)} = \mathbf{U}_i^{(t-1)} - \gamma \frac{\partial F^{(\Delta t)}}{\partial \mathbf{U}_i^{(t-1)}}, i = 1 : N \\ \boldsymbol{\mathcal{G}}^{(t)} = \boldsymbol{\mathcal{G}}^{(t-1)} - \gamma \frac{\partial F^{(\Delta t)}}{\partial \boldsymbol{\mathcal{G}}^{(t-1)}}, \end{cases}\right\} \text{SGD style updates}$$

where γ is the step size for the gradients. $\boldsymbol{\mathcal{R}}^{(\Delta t)}$, needed for computing the gradients of $F^{(\Delta t)}$, is given by

$$\boldsymbol{\mathcal{R}}^{(\Delta t)} = \boldsymbol{\mathcal{X}}^{(\Delta t)} - \boldsymbol{\mathcal{G}}^{(t-1)} \times_1 \mathbf{A}_1^{(\Delta t)} \mathbf{U}_1^{(t-1)} \times_2 \cdots \times_N \mathbf{A}_N^{(\Delta t)} \mathbf{U}_N^{(t-1)}. \tag{4}$$

SIITA (cont.)

Algorithm 1: Proposed SIITA Algorithm

Input : $\{\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{A}^{(t)}\}, \lambda_i, i = 1 : N, (r_1, \ldots, r_N)$

Randomly initialize $\mathbf{U}_i^{(0)} \in \mathbb{R}^{M_i \times r_i}, i = 1 : N$ and $\boldsymbol{\mathcal{G}}^{(0)} \in \mathbb{R}^{r_i \times \cdots \times r_N}$;

for $t = 1, 2, \ldots$ **do**

- $\mathbf{U}_i^{(t)_0} := \mathbf{U}_i^{(t-1)}, i = 1 : N$;
- $\boldsymbol{\mathcal{G}}^{(t)_0} := \boldsymbol{\mathcal{G}}^{(t-1)}$;
- **for** $k = 1{:}K$ **do**
  - {Inner iterations}
  - Compute $\boldsymbol{\mathcal{R}}^{(\Delta t)}$ from (4) using $\mathbf{U}_i^{(t)_{k-1}}, i = 1 : N$ and $\boldsymbol{\mathcal{G}}^{(t)_{k-1}}$;
  - Compute $\frac{\partial F^{(\Delta t)}}{\partial \mathbf{U}_i^{(t)_{k-1}}}$ for $i = 1 : N$;
  - Update $\mathbf{U}_i^{(t)_k}$ using $\frac{\partial F^{(\Delta t)}}{\partial \mathbf{U}_i^{(t)_{k-1}}}$ and $\mathbf{U}_i^{(t)_{k-1}}$; {Updating Factor Matrices}
  - Compute $\frac{\partial F^{(\Delta t)}}{\partial \boldsymbol{\mathcal{G}}^{(t)_{k-1}}}$;
  - Update $\boldsymbol{\mathcal{G}}^{(t)_k}$ using $\boldsymbol{\mathcal{G}}^{(t)_{k-1}}$ and $\frac{\partial F^{(\Delta t)}}{\partial \boldsymbol{\mathcal{G}}^{(t)_{k-1}}}$; {Updating Core Tensor}
- **end**
- $\mathbf{U}_i^{(t)} := \mathbf{U}_i^{(t)_K}$; $\boldsymbol{\mathcal{G}}^{(t)} := \boldsymbol{\mathcal{G}}^{(t)_K}$;

end

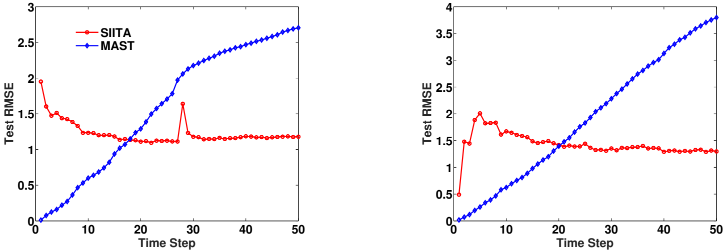
Return: $\mathbf{U}_i^{(t)}, i = 1 : N, \boldsymbol{\mathcal{G}}^{(t)}$.

Results

	MovieLens 100K	**YELP**
Modes	user × movie × week	user × business × year-month
Tensor Size	943×1682×31	1000×992×93
Starting size	19×34×2	20×20×2
Increment step	19, 34, 1	20, 20, 2
Sideinfo matrix	1682 (movie) × 19 (genre)	992 (business) × 56 (city)

Table : Summary of datasets used in the paper.

Multi-Aspect Streaming Setting

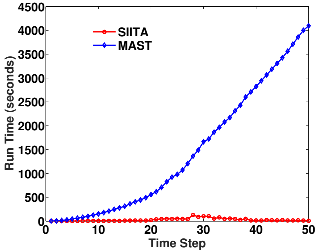


(a) MovieLens 100K (20% Missing)

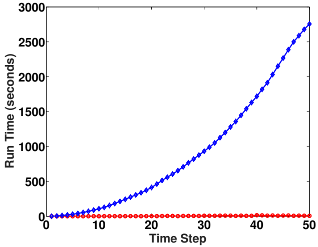
(b) YELP (20% Missing)

Figure : Evolution of Test RMSE (lower is better) of MAST and SIITA with each time step.

Multi-Aspect Streaming Setting (cont.)



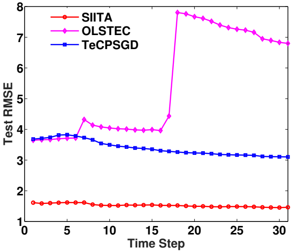
(a) MovieLens 100K
(20% Missing)



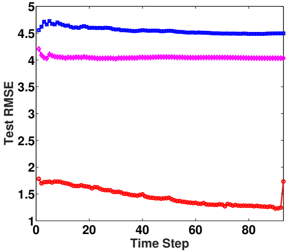
(b) YELP
(20% Missing)

Figure : Runtime comparison between MAST and SIITA at every time step.

Streaming Setting



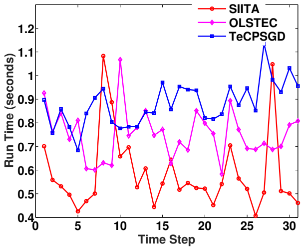
(b) MovieLens 100K
(20% Missing)



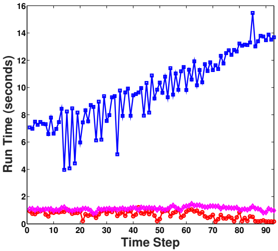
(a) YELP
(20% Missing)

Figure : Evolution of Test RMSE (lower is better) of TeCPSGD, OLSTEC and SIITA with each time step.

Streaming Setting (cont.)



(b) MovieLens 100K
(20% Missing)



(a) YELP
(20% Missing)

Figure : Runtime comparison between TeCPSGD, OLSTEC and SIITA.

Static Setting

Dataset	Missing%	Rank	AirCP	SIITA
MovieLens 100K	20%	3	3.351	**1.534**
		5	3.687	1.678
		10	3.797	2.791
	50%	3	3.303	**1.580**
		5	3.711	1.585
		10	3.894	2.449
	80%	3	3.883	**1.554**
		5	3.997	1.654
		10	3.791	3.979

Table : Mean Test RMSE (lower is better) across multiple train-test splits in the Batch setting.

Static Setting (cont.)

Dataset	Missing%	Rank	AirCP	SIITA
YELP	20%	3	1.094	**1.052**
		5	1.086	1.056
		10	1.077	1.181
	50%	3	1.096	1.097
		5	1.095	**1.059**
		10	1.719	1.599
	80%	3	1.219	1.199
		5	**1.118**	1.156
		10	2.210	2.153

Table : Mean Test RMSE (lower is better) across multiple train-test splits in the Batch setting.

Nonnegative Setting

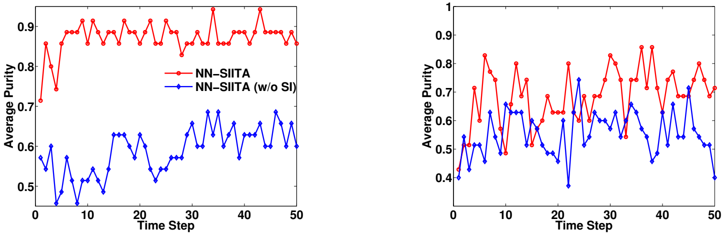
- Incorporating Nonnegative constraints into SIITA (**NN-SIITA**) is useful for unsupervised setting.
- Metrics for evaluating the clusters mined by NN-SIITA
  - Let w_p items of top w items in a cluster belong to the same category, then

$$\text{For a cluster } p, \textbf{Purity}(p) = w_p / w,$$

$$\textbf{average-Purity} = \frac{1}{r_i} \sum_{p=1}^{r_i} \text{Purity}(p),$$

where r_i is the number of clusters along mode-i.

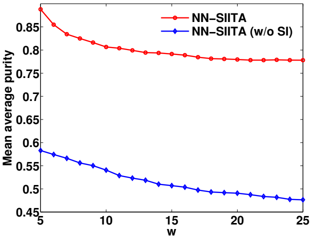
Nonnegative Setting (cont.)



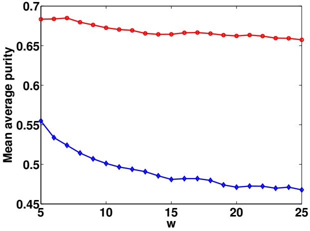
(a) MovieLens 100K (b) YELP

Figure : Average Purity (higher is better) of clusters learned by NN-SIITA and NN-SIITA (w/o SI) at every time step in the unsupervised setting.

Nonnegative Setting (cont.)



(a) MovieLens 100K



(b) YELP

Figure : Evolution of mean average purity (higher is better) with w for NN-SIITA and NN-SIITA (w/o SI) for both MovieLens 100K and YELP datasets.

Takeaways

- SIITA is the first ever algorithm that incorporates side information into dynamic tensor completion.
- SIITA can handle the more general Multi-aspect streaming setting.
- NN-SIITA is the first ever algorithm that incorporates Nonnegative constraints into dynamic tensor analysis.

Codes available at https://madhavcsa.github.io

Thank You!

Bibliography

[1] H. Ge, J. Caverlee, N. Zhang, and A. Squicciarini.
Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information.
CIKM, 2016.

[2] H. Kasai.
Online low-rank tensor subspace tracking from incomplete data by cp decomposition using recursive least squares.
In *ICASSP*, 2016.

[3] M. Mardani, G. Mateos, and G. B. Giannakis.
Subspace learning and imputation for streaming big data matrices and tensors.
IEEE Transactions on Signal Processing, 2015.

Bibliography (cont.)

[4] Q. Song, X. Huang, H. Ge, J. Caverlee, and X. Hu.
Multi-aspect streaming tensor completion.
In *KDD*, 2017.