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

Madhav Nimishakavi\textsuperscript{1}  Bamdev Mishra\textsuperscript{2}  Manish Gupta\textsuperscript{2}  Partha Talukdar\textsuperscript{1}

\textsuperscript{1}Indian Institute of Science  \textsuperscript{2}Microsoft
Outline

1. Introduction
2. Preliminaries
3. Side Information infused Incremental Tensor Analysis (SIITA)
4. Results
Introduction

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

- Tensors are used for representing multidimensional data.
In practice, many multidimensional datasets are often incomplete.

Tensor Completion is the task of predicting or imputing missing values in a partially observed tensor.
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.
Most of the existing works make an assumption that the tensor grows only in one mode.

This assumption is restrictive!
Recently Song et al. [4] proposed the more general Multi-aspect streaming tensor completion.

Figure: Multi-aspect streaming tensor sequence
Besides the tensor, additional side information data is also available in the form of matrices in many applications.

- For example, \( \text{movie} \times \text{genre} \) matrix for online movie recommendation etc.

- Incorporating the side information matrices into tensor completion can help achieve better results, particularly in sparse settings.
We propose a framework to handle the following sequences.

(a) Streaming sequence with side information

(b) Multi-aspect streaming sequence with side information
Definition (Multi-aspect streaming Tensor Sequence) [4]: A tensor sequence of $N^{th}$-order tensors $\{\mathbf{X}(t)\}$ is called a multi-aspect streaming tensor sequence if for any $t \in \mathbb{Z}^+$, $\mathbf{X}(t-1) \in \mathbb{R}^{l_1^{t-1} \times l_2^{t-1} \times \ldots \times l_N^{t-1}}$ is the sub-tensor of $\mathbf{X}(t) \in \mathbb{R}^{l_1^{t} \times l_2^{t} \times \ldots \times l_N^{t}}$, i.e.,

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

Here, $t$ increases with time, and $\mathbf{X}(t)$ is the snapshot tensor of this sequence at time $t$. 
Definition (Multi-aspect streaming Tensor Sequence with Side Information): Given a time instance \( t \), let \( A_i^{(t)} \in \mathbb{R}^{l_i \times M_i} \) be a side information (SI) matrix corresponding to the \( i^{th} \) mode of \( \mathcal{X}^{(t)} \), we have,

\[
A_i^{(t)} = \begin{bmatrix} A_i^{(t-1)} & \Delta_i^{(t)} \end{bmatrix}, \text{ where } \Delta_i^{(t)} \in \mathbb{R}^{[l_i^{(t)} - l_i^{(t-1)}] \times M_i}.
\]

let the side information set \( \mathcal{A}^{(t)} = \{ A_1^{(t)}, \ldots, A_N^{(t)} \} \).

Given an \( N^{th} \)-order multi-aspect streaming tensor sequence \( \{ \mathcal{X}^{(t)} \} \), we define a multi-aspect streaming tensor sequence with side information as \( \{ (\mathcal{X}^{(t)}, \mathcal{A}^{(t)}) \} \).
Problem Definition: Given a multi-aspect streaming tensor sequence with side information \(\{(\mathcal{X}(t), \mathcal{A}(t))\}\), the goal is to predict the missing values in \(\mathcal{X}(t)\) by utilizing only entries in the relative complement \(\mathcal{X}(t) \setminus \mathcal{X}(t-1)\) and the available side information \(\mathcal{A}(t)\).
We propose Side Information infused Incremental Tensor Analysis (SIITA).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Streaming</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Multi-Aspect Streaming</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Side Information</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sparse Solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Table: Summary of different tensor streaming algorithms.
SIITA (cont.)

\[
\min_{G \in \mathbb{R}^{r_1 \times \cdots \times r_N}, \forall i \in [1:N]} \quad F(x(t), A(t), G, \{U_i\}_{i=1:N}),
\]

where

\[
F(x(t), A(t), G, \{U_i\}_{i=1:N}) = \left\| \hat{X}(t) - \hat{\mathcal{P}}_{\Omega}(G \times_1 A_1 \times_2 \cdots \times_N A_N U_1 \times_2 \cdots \times N) \right\|_F^2 + \lambda_g \|G\|_F^2 + \sum_{i=1}^{N} \lambda_i \|U_i\|_F^2.
\]
SIITA (cont.)

Since \( \{(\mathbf{x}^{(t-1)}, \mathbf{A}^{(t-1)})\} \subseteq \{(\mathbf{x}^{(t)}, \mathbf{A}^{(t)})\} \), we have

\[
F(\mathbf{x}^{(t)}, \mathbf{A}^{(t)}, \mathbf{G}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N}) = \]

\[
F(\mathbf{x}^{(t-1)}, \mathbf{A}^{(t-1)}, \mathbf{G}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N}) + \]

\[
F(\mathbf{x}^{(\Delta t)}, \mathbf{A}^{(\Delta t)}, \mathbf{G}^{(t-1)}, \{\mathbf{U}_i^{(t-1)}\}_{i=1:N})
\]

\[
\text{delta term between } t \text{ and } t-1
\]

(3)
We propose the following incremental update scheme,
\[
\begin{align*}
U_i^{(t)} &= U_i^{(t-1)} - \gamma \frac{\partial F(\Delta t)}{\partial U_i^{(t-1)}}, \quad i = 1 : N \\
G^{(t)} &= G^{(t-1)} - \gamma \frac{\partial F(\Delta t)}{\partial G^{(t-1)}},
\end{align*}
\]
where $\gamma$ is the step size for the gradients. $R(\Delta t)$, needed for computing the gradients of $F(\Delta t)$, is given by
\[
R(\Delta t) = \mathcal{X}(\Delta t) - G^{(t-1)} \times_1 A_1^{(\Delta t)} U_1^{(t-1)} \times_2 \ldots \\
\times_N A_N^{(\Delta t)} U_N^{(t-1)}.
\]
Algorithm 1: Proposed SIITA Algorithm

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

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

for \( t = 1, 2, \ldots \) do

\( \mathbf{U}^{(t)}_i := \mathbf{U}^{(t-1)}_i, i = 1 : N; \)
\( \mathbf{G}^{(t)} := \mathbf{G}^{(t-1)}; \)

for \( k = 1:K \) do

\{Inner iterations\}

Compute \( \mathbf{R}^{(\Delta t)} \) from (4) using \( \mathbf{U}^{(t)}_i, i = 1 : N \) and \( \mathbf{G}^{(t)}_k \);

Compute \( \frac{\partial F(\Delta t)}{\partial \mathbf{U}^{(t)}_i} \) for \( i = 1 : N \);

Update \( \mathbf{U}^{(t)}_i \) using \( \frac{\partial F(\Delta t)}{\partial \mathbf{U}^{(t)}_i} \) and \( \mathbf{U}^{(t-1)}_i \); \{Updating Factor Matrices\}

Compute \( \frac{\partial F(\Delta t)}{\partial \mathbf{G}^{(t)}_k} \);

Update \( \mathbf{G}^{(t)}_k \) using \( \mathbf{G}^{(t-1)}_k \) and \( \frac{\partial F(\Delta t)}{\partial \mathbf{G}^{(t)}_k} \); \{Updating Core Tensor\}

end

\( \mathbf{U}^{(t)}_i := \mathbf{U}^{(t)}_i K; \quad \mathbf{G}^{(t)} := \mathbf{G}^{(t)} K; \)

end

Return: \( \mathbf{U}^{(t)}_i, i = 1 : N, \mathbf{G}^{(t)}. \)
## Results

<table>
<thead>
<tr>
<th></th>
<th>MovieLens 100K</th>
<th>YELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modes</td>
<td>user × movie × week</td>
<td>user × business × year-month</td>
</tr>
<tr>
<td>Tensor Size</td>
<td>943 × 1682 × 31</td>
<td>1000 × 992 × 93</td>
</tr>
<tr>
<td>Starting size</td>
<td>19 × 34 × 2</td>
<td>20 × 20 × 2</td>
</tr>
<tr>
<td>Increment step</td>
<td>19, 34, 1</td>
<td>20, 20, 2</td>
</tr>
<tr>
<td>Sideinfo matrix</td>
<td>1682 (movie) × 19 (genre)</td>
<td>992 (business) × 56 (city)</td>
</tr>
</tbody>
</table>

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

Figure: Runtime comparison between TeCPSGD, OLSTEC and SIITA.

(b) MovieLens 100K (20% Missing)

(a) YELP (20% Missing)
### Static Setting

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Missing%</th>
<th>Rank</th>
<th>AirCP</th>
<th>SIITA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens 100K</td>
<td>20%</td>
<td>3</td>
<td>3.351</td>
<td>1.534</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>3.687</td>
<td>1.678</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>3.797</td>
<td>2.791</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>3</td>
<td>3.303</td>
<td>1.580</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>3.711</td>
<td>1.585</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>3.894</td>
<td>2.449</td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>3</td>
<td>3.883</td>
<td>1.554</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>3.997</td>
<td>1.654</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>3.791</td>
<td>3.979</td>
</tr>
</tbody>
</table>

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

---

Madhav Nimishakavi  
CIKM 2018
## Static Setting (cont.)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Missing%</th>
<th>Rank</th>
<th>AirCP</th>
<th>SIITA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1.094</td>
<td>1.052</td>
</tr>
<tr>
<td>YELP</td>
<td>20%</td>
<td>5</td>
<td>1.086</td>
<td>1.056</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>1.077</td>
<td>1.181</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.096</td>
<td>1.097</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.095</td>
<td>1.059</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.719</td>
<td>1.599</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.219</td>
<td>1.199</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.118</td>
<td>1.156</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.210</td>
<td>2.153</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

where $r_i$ is the number of clusters along mode-$i$. 
Nonnegative Setting (cont.)

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.)

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.

(a) MovieLens 100K

(b) YELP
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


Multi-aspect streaming tensor completion.