Online Tensor Analysis

Sangjun Son
Computer Science and Engineering
Seoul National University
`lucetre@snu.ac.kr`

U Kang
Computer Science and Engineering
Seoul National University
`ukang@snu.ac.kr`

July 2, 2020

Abstract

In this paper, we propose fast and accurate online tensor decomposition method called OnlineTensorAnalysis, a tensor factorization tool for a streaming tensor by drastic data handling. As drastic data income at a certain time step, OnlineTensorAnalysis chooses to split the tensor or update its whole time factor matix. It enhances more accurate and faster decomposition with memory-efficient management of temporal factors.

1 Introduction

Given a temporally growing tensor, how can we analyze it efficiently? Multi-dimensional arrays or tensors have been widely used to model real world data. Tensor decomposition plays a significant role in latent feature discovery and estimation of unobservable entries. Each tensor can be classified as static or dynamic. A tensor whose size and values are temporally changing is dynamic (e.g. sensor data on every point of the room), and the other is static. Most of existing tensor analysis methods such as CP–ALS and HOSVD decompose static tensors with high fitness.

Data stream produces numerous amounts of data every second and it became more important to maintain tensor factorization result. Every dynamic tensor can have an extra temporal mode and previous state of the tensor can be stored along the mode. Growing in the temporal mode, however, applying static tensor factorization methods to dynamic is an inappropriate way in time and space efficiency. It invokes lots of computations to update all the entries of time factor matrix. The contributions of this project are the following:

- OnlineTensorAnalysis performs online tensor decomposition preserving accuracy without time dilation due to short data income intervals.
- OnlineTensorAnalysis is time scalable, being linear on the length of temporal mode.
- OnlineTensorAnalysis automatically detects drastic data changes and creates new starting point of decomposition.

2 Preliminaries
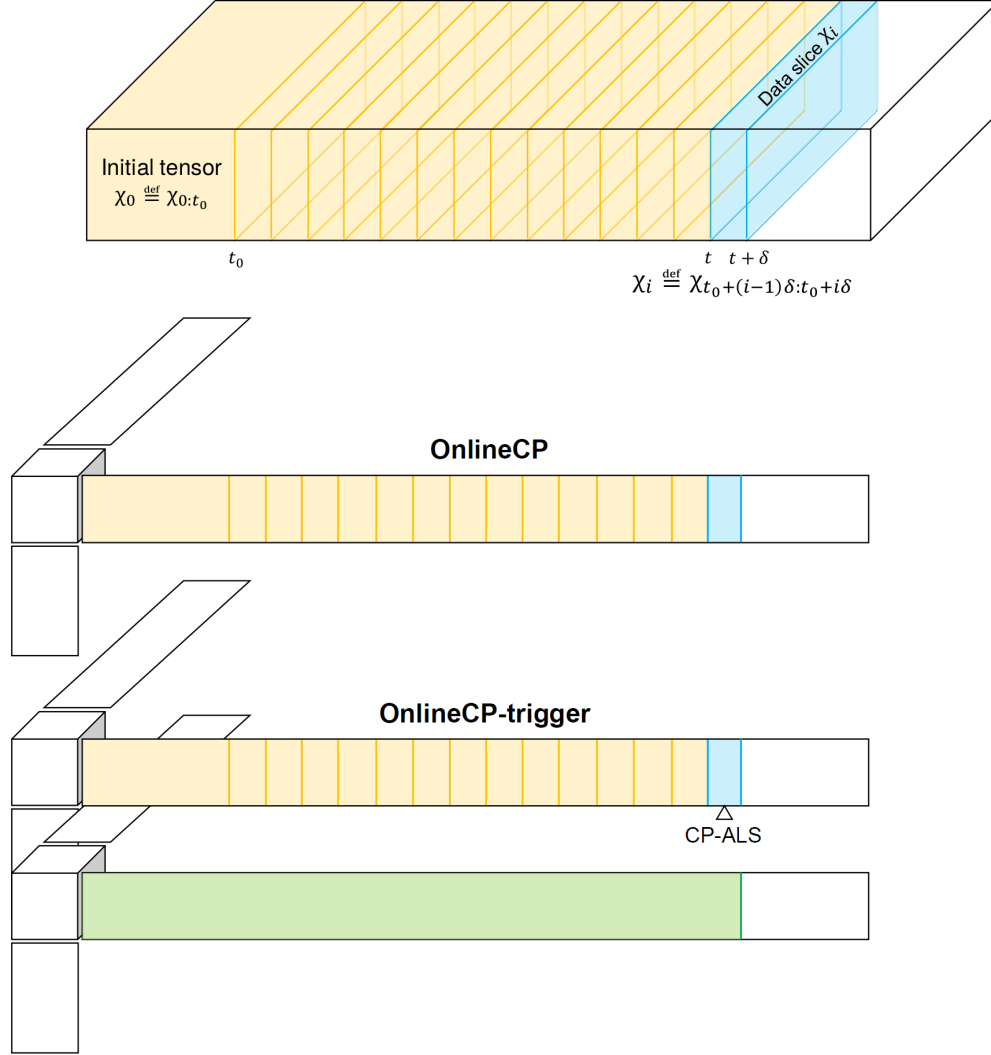
2.1 Preliminaries

...

3 Proposed Method

We've developed our methods by extending the basic intuition of OnlineCP. Since dynamic tensor decomposition pursues shorter time factor updates, this results low accuracy factorization when real-time data incomes. To optimize the speed accuracy problem, we'd like to trigger static decomposition method, CP–ALS while dynamic method like OnlineCP is being done.
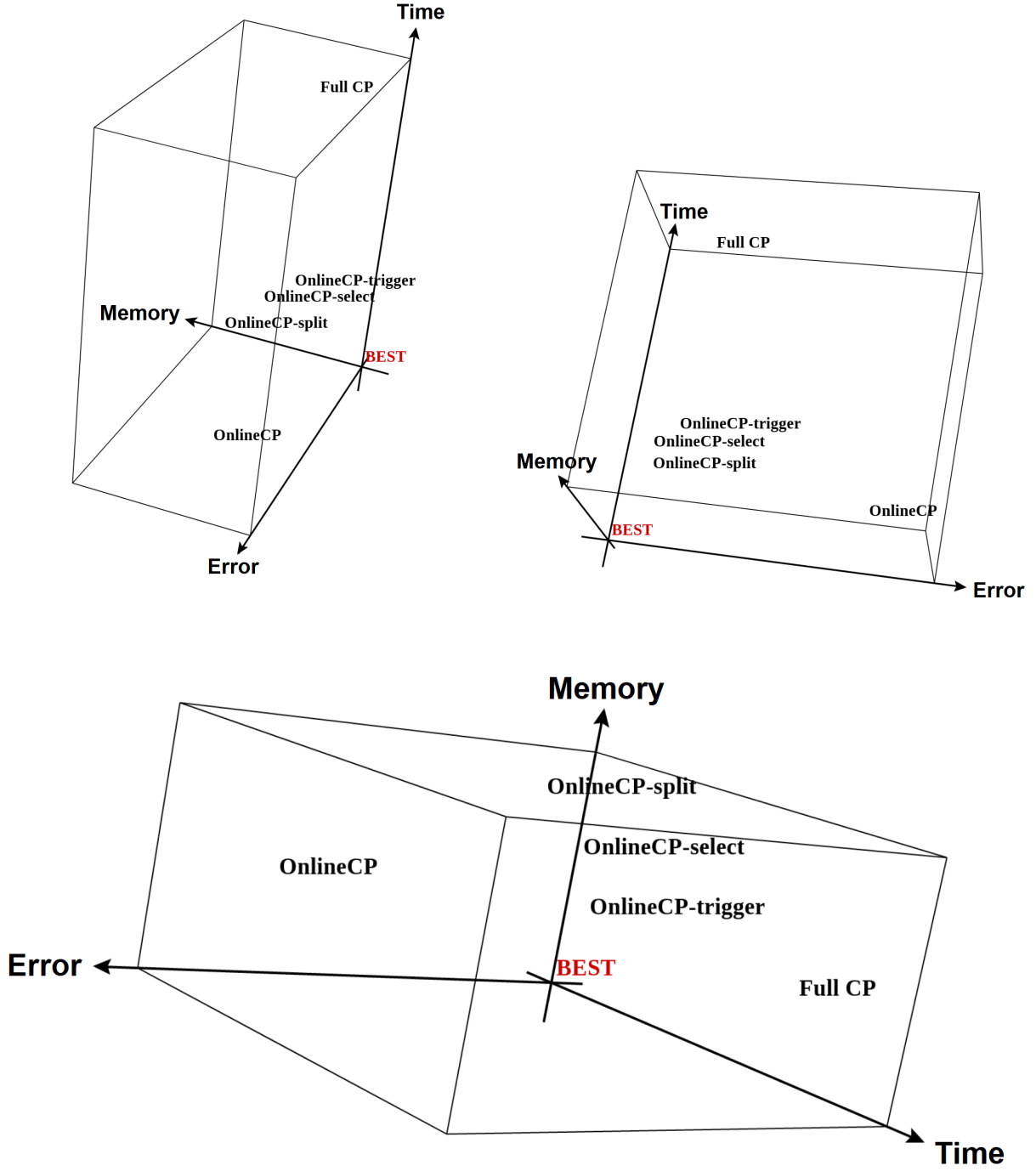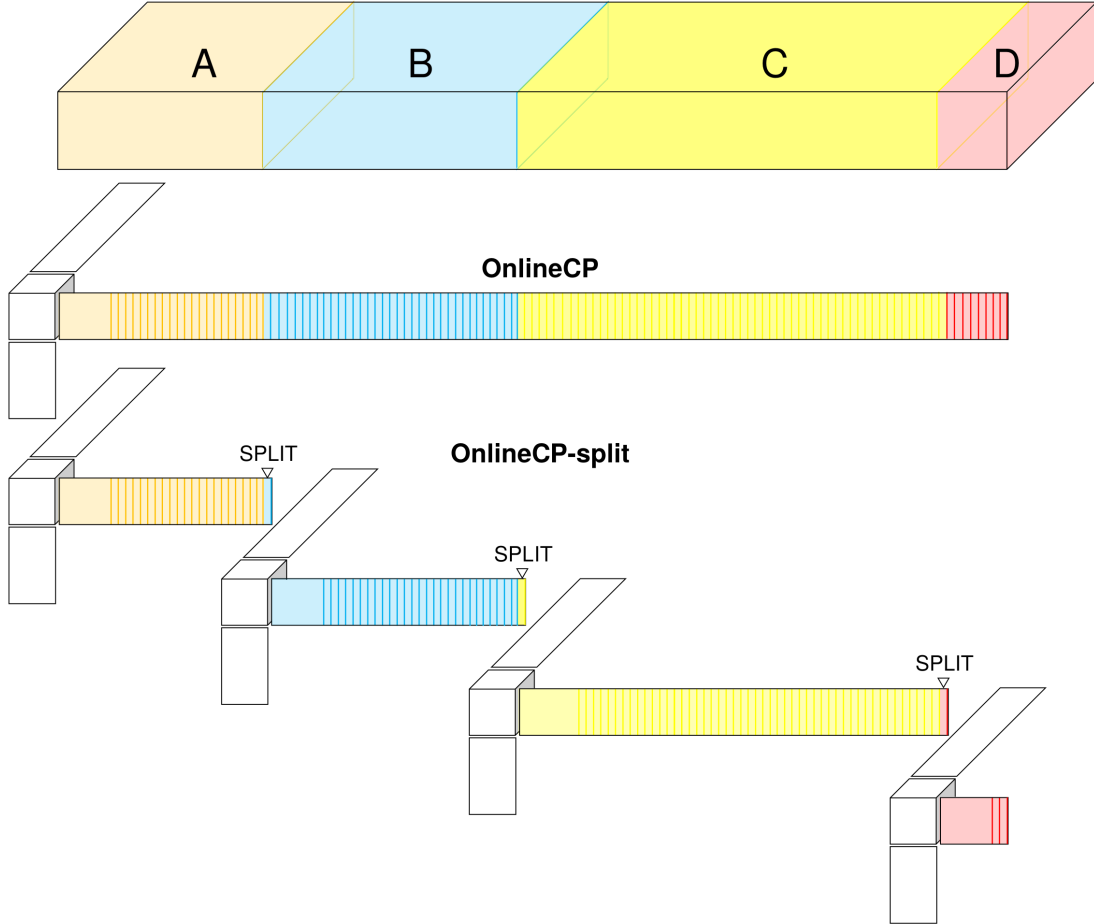
3.1 *OnlineCP-trigger*

Detection Approach: CP–ALS activates whole temporal factor updates and enables high accuracy decomposition. Drastic data can be detected by calculating image error norm and its ratio between neighboring time steps. The moment for triggering CP–ALS is when temporal ratio of error norm for incoming data exceeds threshold.

$$threshold < \frac{\|\tilde{\chi}_{i+1} - \chi_{i+1}\|}{\|\tilde{\chi}_i - \chi_i\|}$$

Data slice χ_i

Initial tensor
$\chi_0 \overset{\text{def}}{=} \chi_{0:t_0}$

t_0 t $t+\delta$

$\chi_i \overset{\text{def}}{=} \chi_{t_0+(i-1)\delta:t_0+i\delta}$

OnlineCP

OnlineCP-trigger

CP-ALS

3.2 *OnlineCP-split*

Split Approach: trigger function in detection approach tells us sudden change in data. What if the incoming data may have a new theme unseen before? It implies that new decomposition starting point with tensor split is needed. In this approach, we'd like to apply the trigger function for splitting the tensor into serial tensors of different themes. (e.g. A, B, C, D)

3.3   *OnlineCP-select*

Selection Approach: Similarly to split approach, trigger function now decides whether to split or to concatenate behind after one temporal factor update. It allows to store the tensor efficiently; group tensors with similar themes and split them otherwise. (e.g. A, B, B', C)
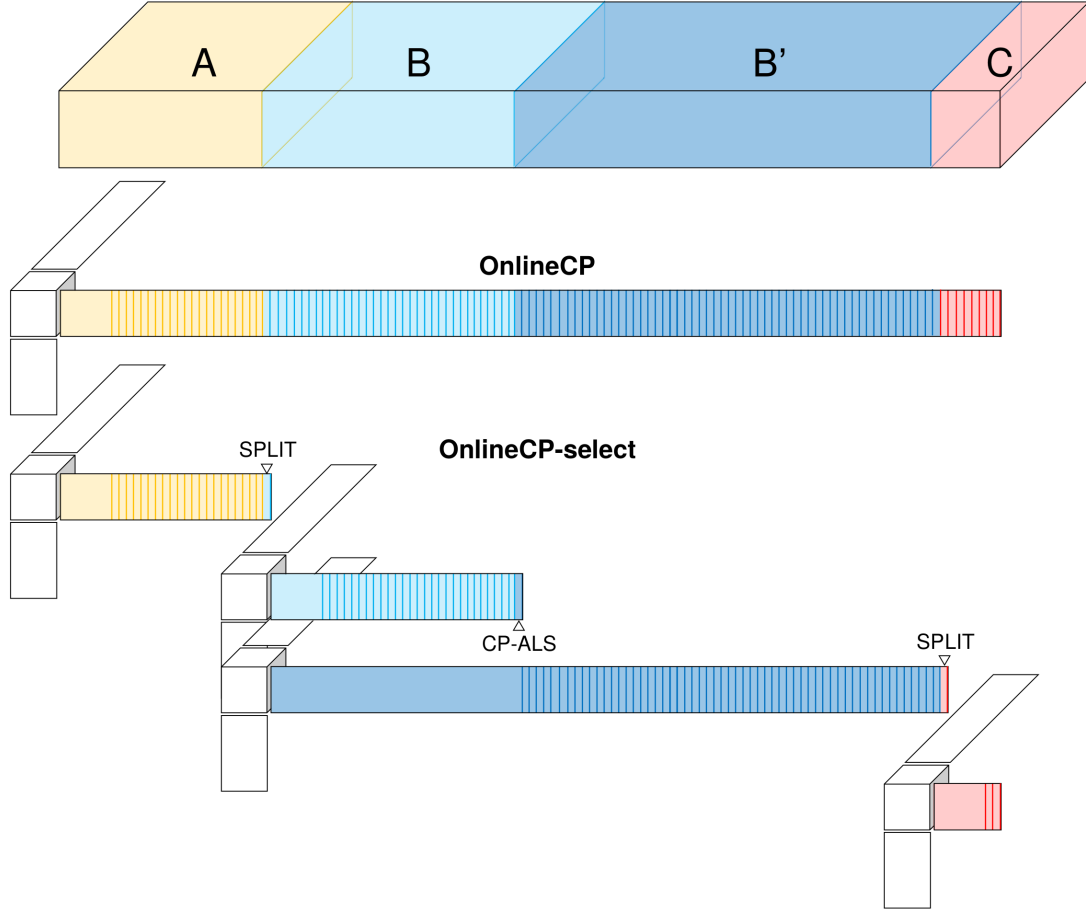
4   Experiments

We implemented our method and compared it with the older ones. The results are very promising.

Figure 1 shows our results: Figure 1(a) gives a scatter-plot of the N sound-clips, where the axis are the two main features we propose to use ... Figure 1(b) shows the wall-clock time of our method, versus the size of the database N.
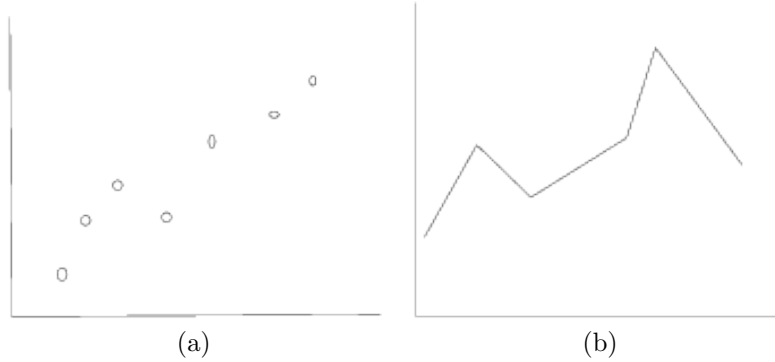

<div align="center">(a)   (b)</div>

Figure 1: A fictitious dataset (a) and its performance plot (b)

5   Related Works

Put related works here.

6   Conclusions

The proposed method *someMETHOD* has the following advantages:

- it gives better classification accuracy than all 10 competitors we tried
- its accuracy is very close to the very best competitor in the *UCR Insect Classification Contest.*
- it is scalable

A Appendix

A.1 Additional Stuff 1

Put contents here.

A.2 Additional Stuff 2

Put contents here.