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

August 31, 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

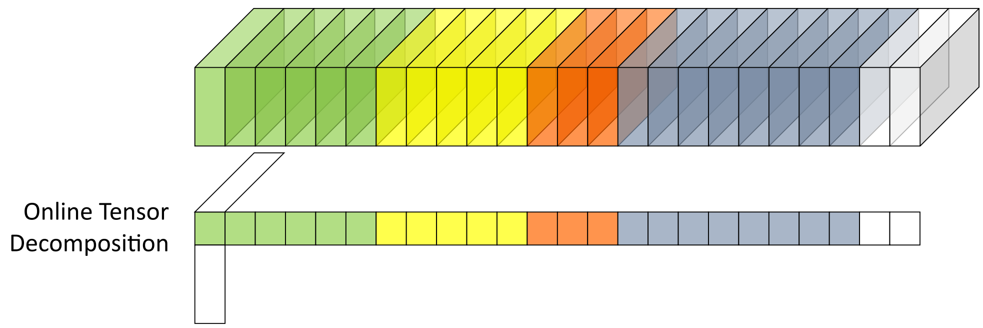
2.1 Preliminaries

...

3 Proposed Method

3.1 Online Tensor Decomposition

Since dynamic tensor decomposition pursues shorter time factor updates, fast decomposition process results low accuracy factorization when real-time data incomes.



3.1.1 *Transformed Online CP*

We've developed TransformedOnlineCP by extending the basic intuition of OnlineCP. To optimize the speed accuracy problem, TransformedOnlineCP resembles CP–ALS by iteratively updating the factor matrix of temporal mode. In spite of time consumption, however it achieves remarkable accuracy improvement.

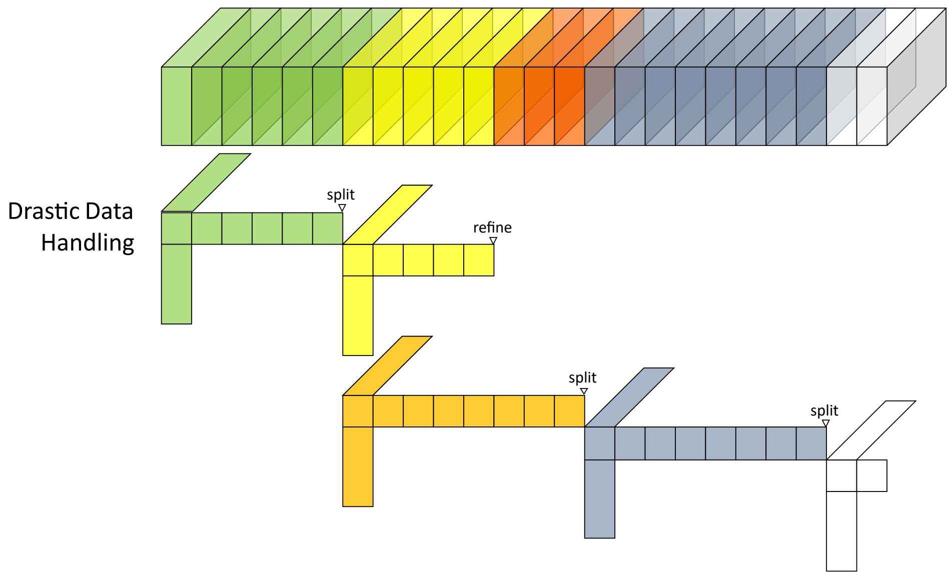
3.1.2 *DTD*

temporally growing dtd mentioned in Multi-aspect tensor completion

3.2 Drastic Data Handling

When we decompose a tensor, we can find error norm comparing real and estimated entries. As the tensor temporally grows, we can calculate local and global error norm by measuring incoming slices and the whole tensor respectively.

Drastic data can be detected by measuring local error norm and distribution of previous norms. Using Welford's algorithm, we can track mean and deviation and find out anomalies in current local error norm by z-score calculation. Differentiating the upper and lower limit of z-score, we can trigger one of accuracy optimizing processes. Trigger function now decides whether to split or to concatenate behind after one temporal factor update. It allows to store the tensor efficiently by grouping tensors with similar themes and splitting them otherwise.

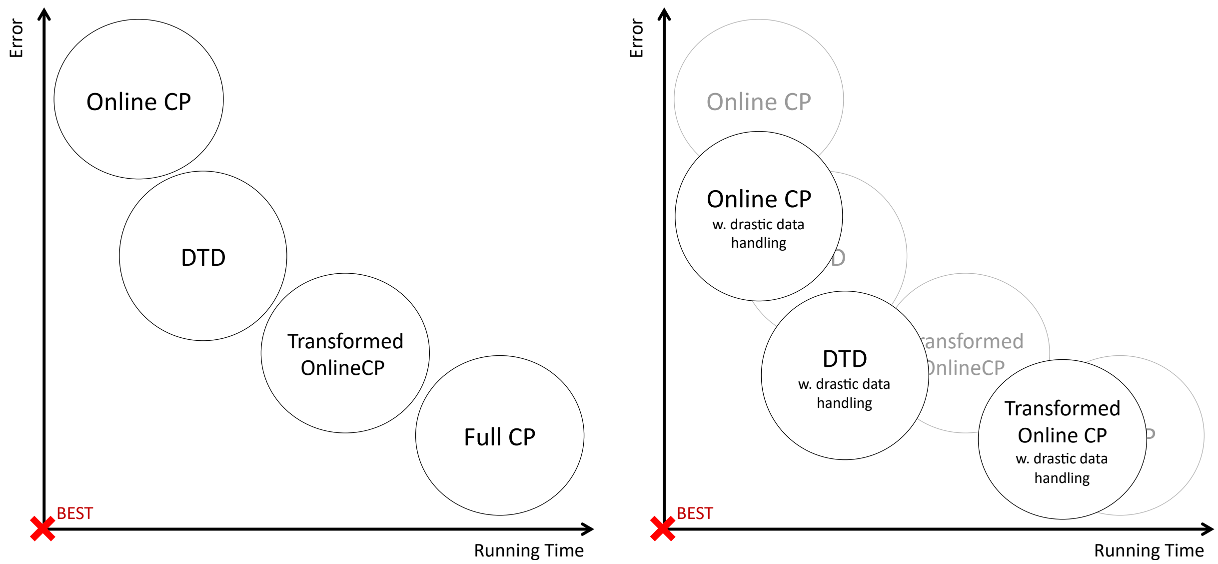


3.2.1 *Split Process*

Anomaly detection in image error norm 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 process, we'd like to apply upper limit ul to trigger splitting the tensor into serial tensors of different themes.

3.2.2 *Refinement Process*

Refinement process is to concatenate incoming tensor slices whose theme is similar to the previous tensor. Exceedance of lower limit ll trigger forgetting preious decomposition result and decompose the tensor once again.


Error
Online CP
DTD
Transformed
OnlineCP
Full CP
BEST
Running Time
Error
Online CP
Online CP
w. drastic data
handling
DTD
w. drastic data
handling
Transformed
Online CP
w. drastic data
handling
BEST
Running Time

4 Experiments

4.1 Datasets

For the demonstration, we will use a synthetic tensor and several real-world tensors. We've made stream of a tensor from starting point and split into parts with batch sizes. Here's our datasets and metadata to construct tensor streams.

Dataset	**Mode**				**Start to Stream**	**Batch Sizes**
Synthetic Data	1 K	10	20	30	5	5 * 199
Sample Video	205	240	320	3	5	5 * 40
Stock	3 K	200	5		10	10 * 299
Sever Room CFD	3 K	3	3	34	10	10 * 299

4.1.1 *Synthetic Data*

We've constructed synthetic data to manually make temporally changing points. This tensor has its size of $10 * 20 * 30 * 1000$ having the last mode as a temporal mode. The tensor was conducted by concatenation of theme tensor T which is addition of three tensors T_{main}, T_{theme} and T_{noise}. Each consisting tensor is 100x, 10x and 1x normal distributed randomized tensor respectively and those are used for making different and similar theme tensors.

Time Index	1∼100	101∼200	201∼250	251∼500	501∼600	601∼700	701∼750	751∼800	801∼950	951∼1000
Theme	A	A'	B	B'	B''	C	D	E	E'	E''

4.1.2 *Sample Video*

4.1.3 *Stock*

KOSPI200

4.1.4 *Server Room CFD*

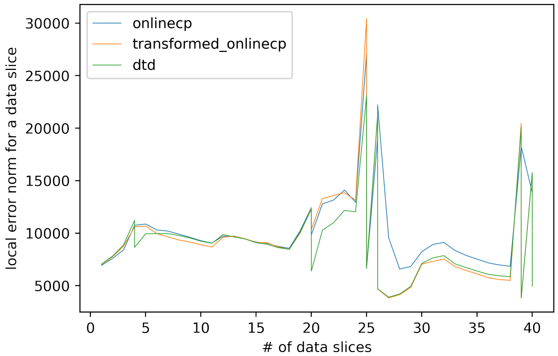
Temperature was measured in a server room equipped with two heterogeneously occupied racks and one roof-mounted cooling device by 34 probes. 3 servers power usage scenarios and also 3 air conditioning temperature scenarios were performed.

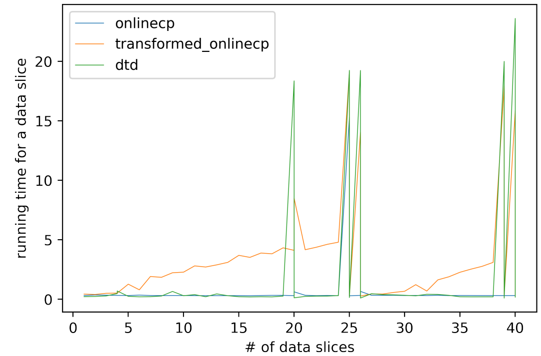
4.2 Experimental Results

For each dataset, decomposition rank and number of iterations may affect performance of decomposition. Triggering condition for split and refinement process depends on upper and lower limit of z-score, ul and ll.

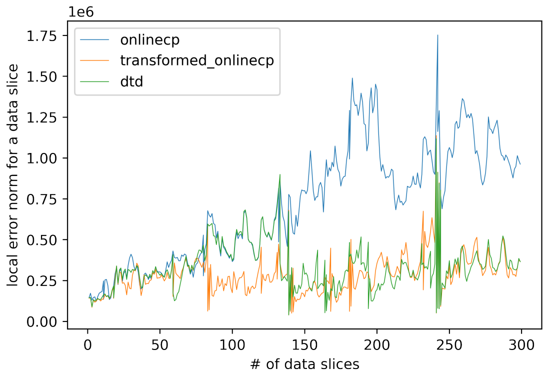
Dataset	Rank	# of Iterations	ul	ll
Synthetic Data	10	1	1	0.5
Sample Video	20	1	5	3
Stock	5	5	5	3
Sever Room CFD	5	5	3	2

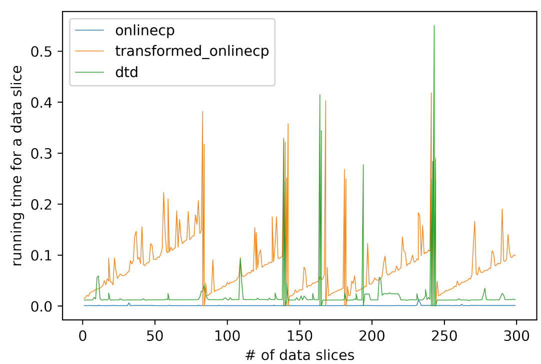
4.2.1 *Sample Video*





4.2.2 *Stock*





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.