Accelerating Online CP Decompositions for Higher Order Tensors

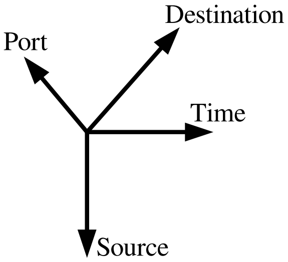
Shuo Zhou[1], Nguyen Xuan Vinh[1], James Bailey[1], Yunzhe Jia[1] and Ian Davidson[2]

[1]The University of Melbourne
[2]University of California, Davis

KDD'2016

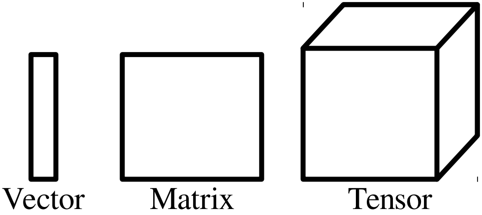
Introduction

Motivation



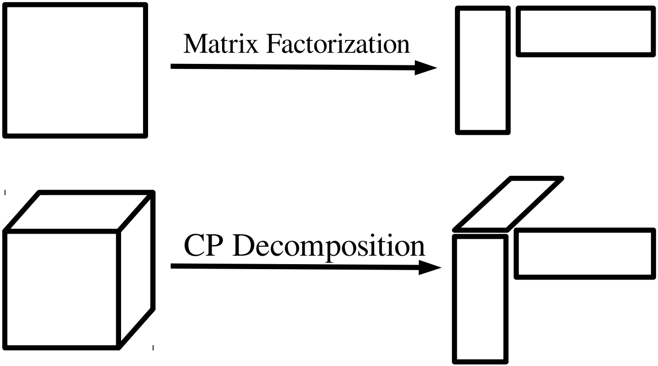
- Multi-way structure
  $\rightarrow$ How to **represent** it?
- Highly complex data
  $\rightarrow$ How to **simplify** it?
- New data keeps arriving
  $\rightarrow$ How to learn **online**?

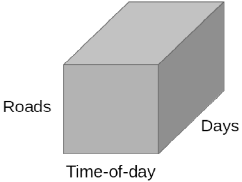
Tensor

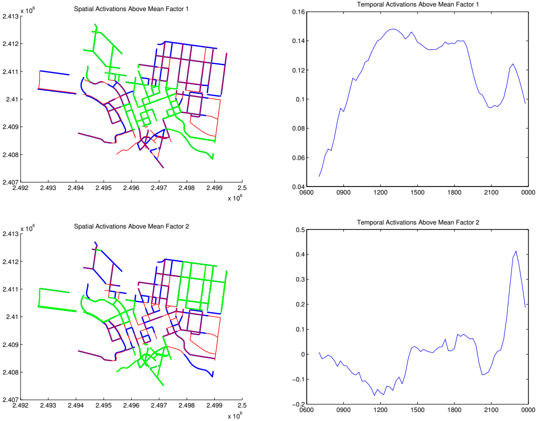


Tensor (multi-way array) is a natural representation for multi-dimensional data, e.g. videos, time-evolving networks

CP Decomposition





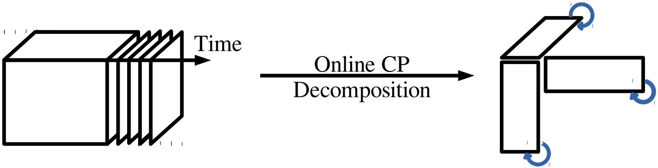


CP decomposition is a method to simplify and summarize tensors

[1] X. N.Vinh, J. Chan, I. Davidson, Simplifying Spatial Temporal Event Objects into Dictionaries

But, online?

Problem Definition



Online CP Decomposition

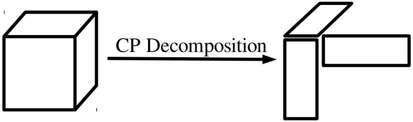
- **Given**:
  - existing data, $\mathcal{X}_{old} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times t_{old}}$
  - its CP decomposition, $[\![\mathbf{A}^{(1)}_{old}, \mathbf{A}^{(2)}_{old}, \ldots, \mathbf{A}^{(N-1)}_{old}, \mathbf{A}^{(N)}_{old}]\!]$
  - new data, $\mathcal{X}_{new} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times t_{new}}$
- **Find**:
  - CP decomposition $[\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N-1)}, \mathbf{A}^{(N)}]\!]$ of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times (t_{old}+t_{new})}$, where $t_{old} \gg t_{new}$.

Preliminary

Notation

Symbol	Meaning
$\mathbf{a}, \mathbf{A}, \mathcal{X}$	vector, matrix, tensor
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$	N^{th}-order tensor
$\mathbf{A}^\top$, $\mathbf{A}^{-1}$, $\mathbf{A}^\dagger$, $\lVert\mathbf{A}\rVert$	transpose, inverse, pseudoinverse and Frobenius norm of $\mathbf{A}$
$\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$	a sequence of N matrices
$\odot$, $\circledast$, $\oslash$	Khatri-Rao product, element-wise multiplication and division
$\mathbf{X}_{(n)}$	mode-n unfolding of tensor $\mathcal{X}$

CP Decomposition



$$\mathbf{X}_{(1)} \approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top$$
$$\mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\top$$
$$\mathbf{X}_{(3)} \approx \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top$$

Algorithm 1 CP-ALS

1: initialize $\mathbf{A}, \mathbf{B}, \mathbf{C}$

2: **repeat**

3: $\quad \arg\min_{\mathbf{A}} \frac{1}{2} \left\| \mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top \right\|^2$

4: $\quad \arg\min_{\mathbf{B}} \frac{1}{2} \left\| \mathbf{X}_{(2)} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\top \right\|^2$

5: $\quad \arg\min_{\mathbf{C}} \frac{1}{2} \left\| \mathbf{X}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top \right\|^2$

6: **until** converged

Online CP Decomposition

- Batch [3]
  - initialize with previous decomposition
  - still computationally expensive
  - poor scalability
- SDT and RLST [6]
  - online tensor decomposition $\rightarrow$ online matrix factorization
  - work on third-order tensors only
- GridTF [7]
  - divide into small grids and decompose them in parallel

Our Approach

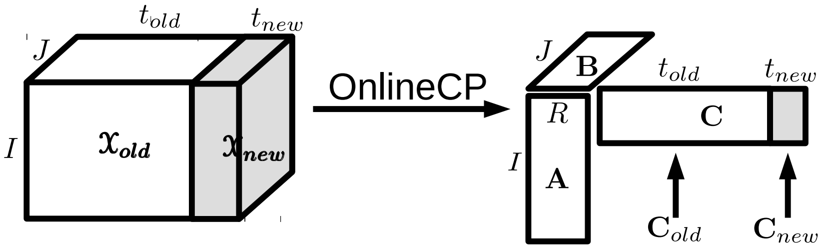
Main Idea

- **ALS** like algorithm
- for each **non-temporal** mode, **complimentary matrices** are used for storing useful information from previous decomposition
- complementary matrices are **incrementally** updated, and loading matrices are estimated based on them

Update Time Mode

fix **A** and **B**, update **C**

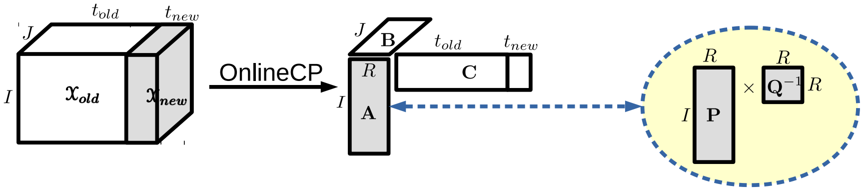
$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{old} \\ \mathbf{C}_{new} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{old} \\ \mathbf{X}_{new\,(3)}((\mathbf{B} \odot \mathbf{A})^\top)^\dagger \end{bmatrix}$$



Update Non-temporal Modes

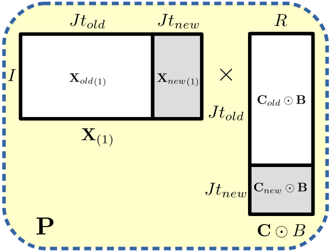
fix $\mathbf{B}$ and $\mathbf{C}$, update $\mathbf{A}$

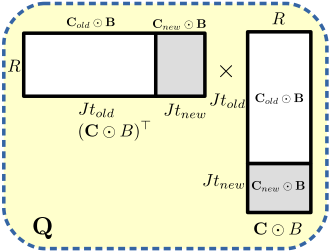
$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}\left\|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\top\right\|^2 \\
\frac{\partial\mathcal{L}}{\partial\mathbf{A}} &= \mathbf{A}(\mathbf{C}\odot\mathbf{B})^\top(\mathbf{C}\odot\mathbf{B}) - \mathbf{X}_{(1)}(\mathbf{C}\odot\mathbf{B}) \\
\mathbf{A} &= (\mathbf{X}_{(1)}(\mathbf{C}\odot\mathbf{B}))((\mathbf{C}\odot\mathbf{B})^\top(\mathbf{C}\odot\mathbf{B}))^{-1} \\
&= \mathbf{PQ}^{-1}
\end{aligned}$$



Incremental Update

$$\mathbf{P} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \qquad \mathbf{Q} = (\mathbf{C} \odot \mathbf{B})^\top(\mathbf{C} \odot \mathbf{B})$$





$$\mathbf{P} \leftarrow \mathbf{P} + \mathbf{X}_{new(1)}(\mathbf{C}_{new} \odot \mathbf{B})$$
$$\mathbf{Q} \leftarrow \mathbf{Q} + (\mathbf{C}_{new} \odot \mathbf{B})^\top(\mathbf{C}_{new} \odot \mathbf{B})$$
$$\mathbf{A} \leftarrow \mathbf{P}\mathbf{Q}^{-1}$$

Empirical Analysis

Datasets

Datasets	Size	Slice Size $S = \prod_{i=1}^{N-1} I_i$	Source
COIL-3D	$128 \times 128 \times 240$	16,384	[5]
COIL-HD	$64 \times 64 \times 25 \times 240$	102,400	
DSA-3D	$8 \times 45 \times 750$	360	[1]
DSA-HD	$19 \times 8 \times 45 \times 750$	6,840	
FACE-3D	$112 \times 92 \times 400$	10,304	[8]
FACE-HD	$28 \times 23 \times 16 \times 400$	10,304	
FOG	$10 \times 9 \times 1000$	90	[2]
GAS-3D	$30 \times 8 \times 2970$	240	[4]
GAS-HD	$30 \times 6 \times 8 \times 2970$	1,440	
HAD-3D	$14 \times 6 \times 500$	64	[10]
HAD-HD	$14 \times 12 \times 5 \times 6 \times 500$	3,840	
ROAD	$4666 \times 96 \times 1826$	447,936	[9]

Baselines

- **Batch Cold**: ALS in Tensor Toolbox [3].
- **Batch Hot**: ALS + previous decomposition.
- **SDT**
- **RLST**
- **GridTF**

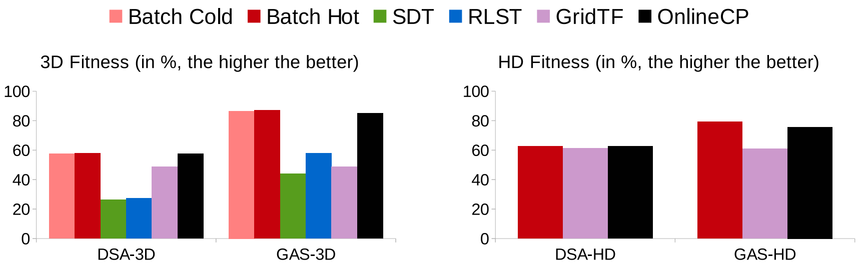
Setup

- Procedure
  - 20% initialization, rest added by slice
  - averaging over 10 runs
- Evaluation metrics
  - effectiveness:

$$\mathit{fitness} \triangleq \left(1 - \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|}{\|\mathcal{X}\|}\right) \times 100\%$$

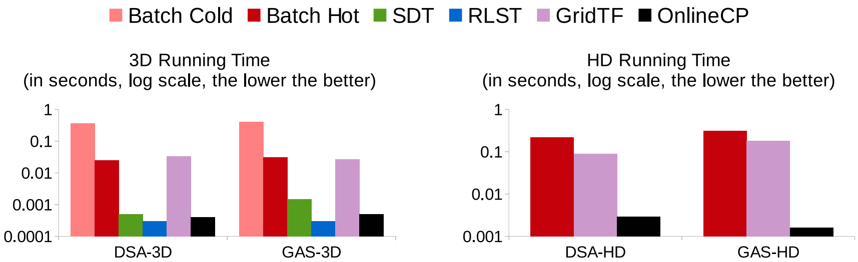
  - efficiency: running time in *seconds*

Effectiveness



- Batch methods always obtain the best fitness overall
- OnlineCP shows comparable results as batch algorithms
- Performance of other online algorithms are much worse

Efficiency



- Batch methods are extremely time-consuming
- Improvement of GridTF is not significant
- Both SDT and RLST show very good performance
- OnlineCP is also quite efficient and significantly faster than batch methods

Effectiveness & Efficiency

Compared to Batch Hot

Relative fitness:

- **OnlineCP (97%)**
- RLST (76%)
- GridTF (68%)
- SDT (67%)

Speedup:

- **OnlineCP (555.59)**
- RLST (113.75)
- SDT (93.56)
- GridTF (1.75)

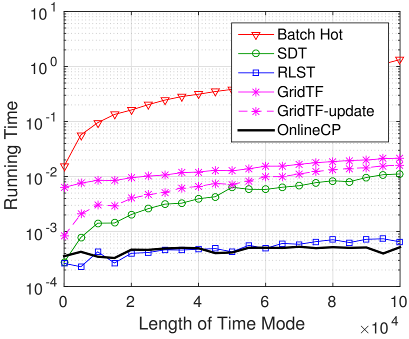
Sensitivity to Initialization

- $\mathcal{X} \in \mathbb{R}^{20\times20\times100}$
- best ALS fitness: 90.14%
- 200 trails of ALS initialization with $\varepsilon \in [9e-1, 1e-4]$
- initial fitness: 65.78 ± 15.3704

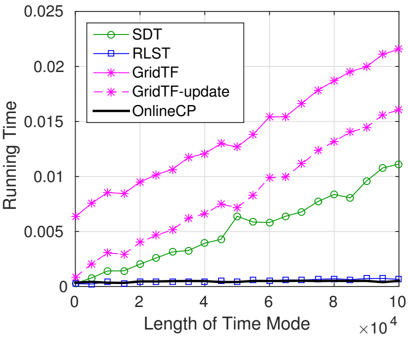
	Final Fitness
Batch Hot	82.57±10.1474
SDT	7.68±55.5205
RLST	33.15±36.9270
GridTF	57.43±15.2148
OnlineCP	67.67±12.9846

Scalability – Time

$$\mathfrak{X} \in \mathbb{R}^{20\times 20\times 10^5}$$



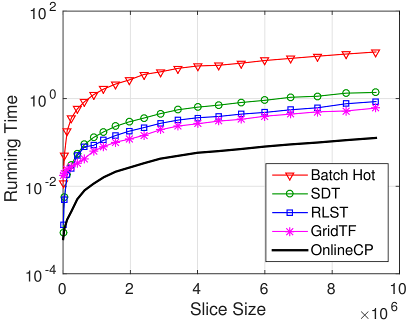
(a) Log scale



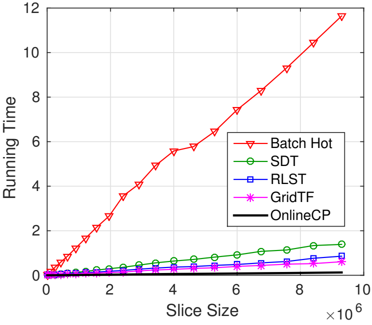
(b) Normal scale

Scalability – Slice Size

100 timestamps, slice size $[100, 9 \times 10^6]$



(a) Log scale



(b) Normal scale

Conclusion & Future Work

- In conclusion, OnlineCP:
  1. is applicable to both **3$^{\text{rd}}$** and **higher** order online tensors
  2. shows very good **quality** of decompositions and significant improvement in **efficiency**
  3. outperforms existing online approaches in terms of **stability** and **scalability**
- Future work:
  1. applications
  2. dynamic tensors
  3. constrained online tensor decomposition

PDF & Code: http://shuo-zhou.info

References I

K. Altun, B. Barshan, and O. Tunçel.
Comparative study on classifying human activities with miniature inertial and magnetic sensors.
Pattern Recognition, 43(10):3605–3620, 2010.

M. Bächlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Tröster.
Wearable assistant for parkinson's disease patients with the freezing of gait symptom.
Information Technology in Biomedicine, IEEE Transactions on, 14(2):436–446, 2010.

B. W. Bader, T. G. Kolda, et al.
Matlab tensor toolbox version 2.6.
Available online, February 2015.

J. Fonollosa, S. Sheik, R. Huerta, and S. Marco.
Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring.
Sensors and Actuators B: Chemical, 215:618–629, 2015.

S. A. Nene, S. K. Nayar, H. Murase, et al.
Columbia object image library (coil-20).
Technical report, technical report CUCS-005-96, 1996.

D. Nion and N. D. Sidiropoulos.
Adaptive algorithms to track the parafac decomposition of a third-order tensor.
Signal Processing, IEEE Transactions on, 57(6):2299–2310, 2009.

References II

A. H. Phan and A. Cichocki.
Parafac algorithms for large-scale problems.
Neurocomputing, 74(11):1970–1984, 2011.

F. S. Samaria and A. C. Harter.
Parameterisation of a stochastic model for human face identification.
In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

F. Schimbinschi, X. V. Nguyen, J. Bailey, C. Leckie, H. Vu, and R. Kotagiri.
Traffic forecasting in complex urban networks: Leveraging big data and machine learning.
In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 1019–1024. IEEE, 2015.

M. Zhang and A. A. Sawchuk.
Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors.
In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 1036–1043. ACM, 2012.

Q & A