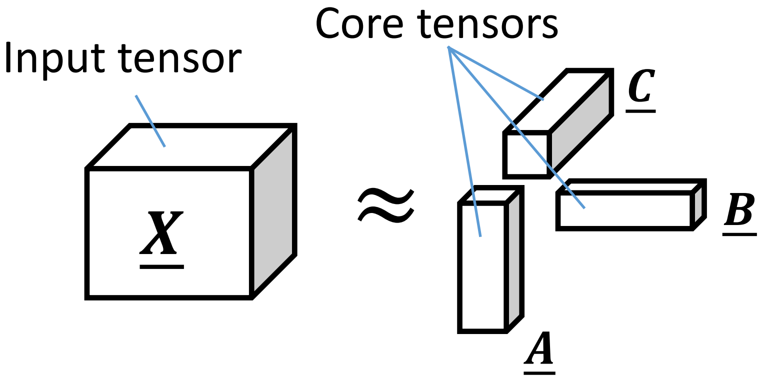
A Sampling-Based Method for Tensor Ring Decomposition

Osman Asif Malik Stephen Becker

Department of Applied Mathematics
University of Colorado Boulder

Tensor ring decomposition via sampling



Elementwise:

R_1, R_2, R_3 are the tensor ring ranks

$$x_{ijk} \approx \sum_{r_1, r_2, r_3 = 1}^{R_1, R_2, R_3} a_{r_3 i r_1} b_{r_1 j r_2} c_{r_2 k r_3} =: \mathrm{TR}(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}})_{ijk}$$

Can formulate the fitting problem as an optimization problem:

$$\underset{\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}}{\mathrm{argmin}} \left\| \mathrm{TR}(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}) - \underline{\boldsymbol{X}} \right\|_{\mathrm{F}}$$

Standard approach is to fit using alternating least squares (ALS):

Algorithm: Tensor Ring ALS [Zhao et al. '16]

`Randomly initialize cores` $\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}$

`for i = 1:max_iter`

$$\underline{\boldsymbol{A}} = \underset{\underline{\boldsymbol{A}}'}{\mathrm{argmin}} \left\| \mathrm{TR}(\underline{\boldsymbol{A}}', \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}) - \underline{\boldsymbol{X}} \right\|_{\mathrm{F}}$$

$$\underline{\boldsymbol{B}} = \underset{\underline{\boldsymbol{B}}'}{\mathrm{argmin}} \left\| \mathrm{TR}(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}', \underline{\boldsymbol{C}}) - \underline{\boldsymbol{X}} \right\|_{\mathrm{F}}$$

$$\underline{\boldsymbol{C}} = \underset{\underline{\boldsymbol{C}}'}{\mathrm{argmin}} \left\| \mathrm{TR}(\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}') - \underline{\boldsymbol{X}} \right\|_{\mathrm{F}}$$

`Check convergence criteria`

`end`

Least squares problems are sampled efficiently

Consider a subproblem:

$$\underline{\boldsymbol{A}} = \underset{\underline{\boldsymbol{A}}'}{\operatorname{argmin}} \left\| \operatorname{TR}\left(\underline{\boldsymbol{A}}', \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}\right) - \underline{\boldsymbol{X}} \right\|_{\mathrm{F}}$$

This can be reshaped into a matrix linear least squares problem of the form

$$A^* \coloneqq \underset{A}{\operatorname{argmin}} \|GA - X\|_F$$

where

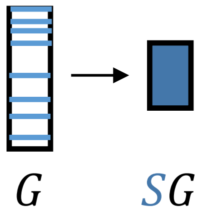
- G comes from merging and reshaping $\underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}$
- A is a reshaped version of $\underline{\boldsymbol{A}}'$
- X comes from reshaping $\underline{\boldsymbol{X}}$

For N-way $I \times \cdots \times I$ input tensor $\underline{\boldsymbol{X}}$:

- forming G costs $NI^{N-1}R^3$
- solving least squares problem costs $I^N R^2$

R is the target rank, assumed to be the same for all dimensions

Can reduce these costs via sampling:



$$\tilde{A} \coloneqq \underset{A}{\operatorname{argmin}} \|SGA - SX\|_F$$

When J rows are sampled:

- forming SG costs JN^2R^3
- solving least squares problem costs $JNIR^2$

Key challenge: Computing distribution of S efficiently given structure of G

Theorem [M., Becker '21]

Can construct sampling distribution in IR^4 time such that the solution $\tilde{A}$ satisfies

$$\|G\,\tilde{A} - X\|_F \leq (1+\varepsilon)\|G\,A^* - X\|_F$$

with probability at least $1 - \delta$ provided at least $J > 4R^{2N}/(\varepsilon\delta)$ rows are sampled*.

* Simplified bound assumes $\varepsilon\delta$ is sufficiently small.

Results in algorithm with complexity *sublinear* in number of entries in input tensor

Experiments

Decomposition

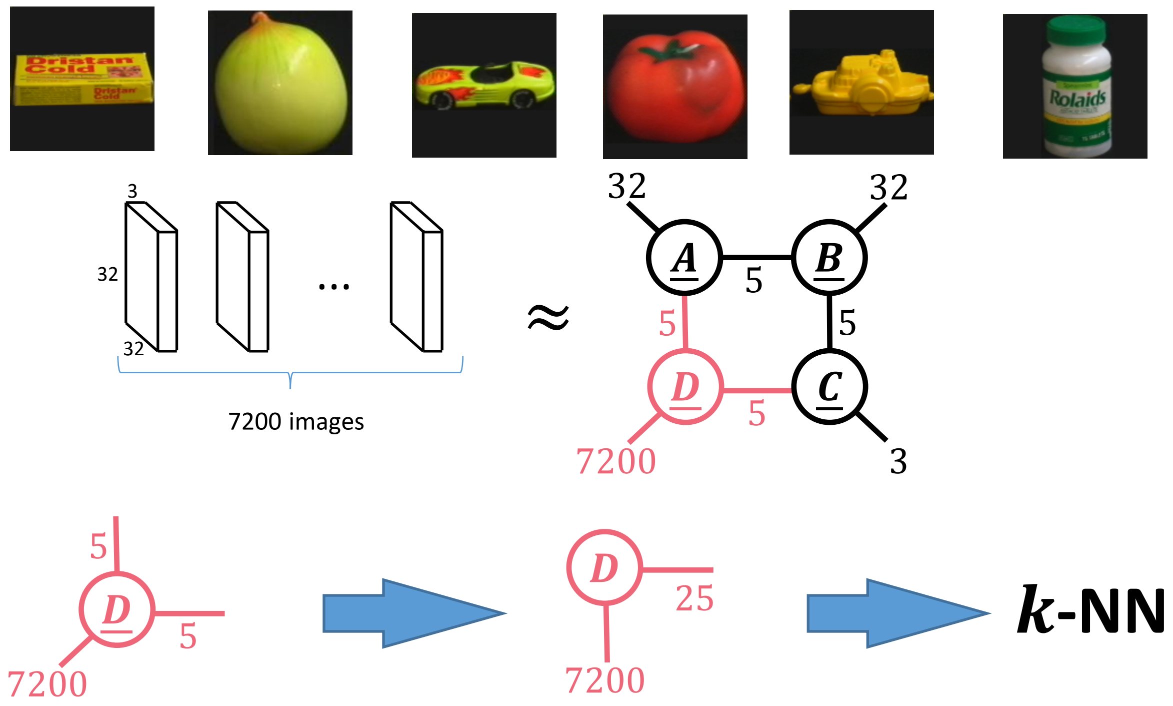
Datasets:

Dataset	Size	Type
Pavia Uni.	$610 \times 340 \times 103$	Hyperspectral
DC Mall	$1280 \times 307 \times 191$	Hyperspectral
Park Bench	$1080 \times 1920 \times 364$	Video
Tabby Cat	$720 \times 1280 \times 286$	Video
Red Truck	$128 \times 128 \times 3 \times 72$	Images

- Substantial speedup over standard algorithm
- Only minor loss in accuracy

Feature extraction

COIL-100 dataset [Nene et al. ’96]:



>99 % accuracy

Thank you for your attention!