Parallel Algorithms for Tensor Completion in the CP Format

Daniel Kressner
Chair of Numerical Algorithms and HPC
MATHICSE / SMA / SB / EPF Lausanne
daniel.kressner@epfl.ch http://anchp.epfl.ch

Joint work with:
Lars Karlsson (Umeå University)
André Uschmajew (University of Bonn)

SIAM PP 2016

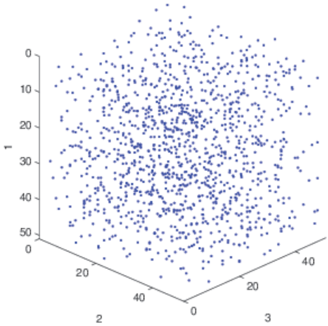
14.4.2016

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

center for advanced modeling science

Tensor completion

Goal: Complete multivariate data.



Applications:

- Completion of corrupted hyperspectral images, CT Scans, . . .
- Learning of multivariate functions
- Non-intrusive methods for stochastic/parametric PDEs
- Context-aware recommender systems
- . . .

Tensor completion

Goal: Complete multivariate data.

Mathematical setting:

- $I_1 \times I_2 \times \cdots \times I_N$ tensor $\boldsymbol{\mathcal{X}}$ with very few entries known.
- $\Omega \subset [1, I_1] \cdots \times \cdots [1, I_N]$ contains (multi)indices of known entries.
- $P_\Omega : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{R}^{|\Omega|}$ orthogonal projection onto known entries.

Tensor completion:

$$\min_{\boldsymbol{\mathcal{X}}} \quad \frac{1}{2} \|\text{known entries} - \mathrm{P}_\Omega \boldsymbol{\mathcal{X}}\|^2$$

- Ill-posed problem.
- Regularize with (multilinear) low-dimensional model for $\boldsymbol{\mathcal{X}}$.

Low-rank tensor completion

Goal: Complete multivariate data.

Mathematical setting:

- $I_1 \times I_2 \times \cdots \times I_N$ tensor $\mathcal{X}$ with very few entries known.
- $\Omega \subset [1, I_1] \cdots \times \cdots [1, I_N]$ contains (multi)indices of known entries.
- $P_\Omega : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{R}^{|\Omega|}$ orthogonal projection onto known entries.

Low-rank tensor completion:

$$\begin{aligned} \min_{\mathcal{X}} \quad & \frac{1}{2} \| \mathrm{P}_\Omega \mathcal{A} - \mathrm{P}_\Omega \mathcal{X} \|^2 \\ \text{subject to} \quad & \mathcal{X} \text{ has tensor rank } L \end{aligned}$$

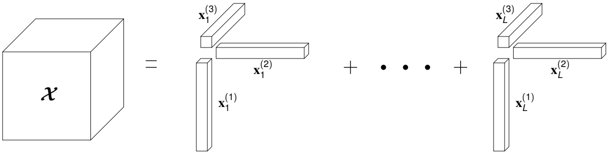
Alternatives to tensor rank (CP decomposition):

- Multilinear rank (Tucker decomposition) and mixtures.
- TT ranks/decomposition, HT ranks/decomposition, general tensor networks.

CANDECOMP/PARAFAC (CP) decomposition

$\mathcal{X}$ has tensor rank L if it admits CP decomposition

$$\mathcal{X} = \sum_{\ell=1}^{L} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)}, \qquad \mathbf{x}_\ell^{(n)} \in \mathbb{R}^{I_n}. \tag{1}$$



Properties of CP:

- Low data complexity: For constant L, linear (instead of exponential) complexity in N.
- Linear wrt *each* factor matrix $\mathbf{X}^{(n)} = \left[\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, \ldots, \mathbf{x}_L^{(n)}\right] \in \mathbb{R}^{I_n \times L}$.
- Tensor rank L *not* upper semi-continuous.

Low-rank tensor completion

Inserting CP into low-rank tensor completion $\rightsquigarrow$

$$\min \left\| P_\Omega \left(\mathcal{A} - \sum_{\ell=1}^{L} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)} \right) \right\|_2^2$$

This will *not* work well because of

- Non-uniqueness:

$$\alpha \mathbf{x}_\ell^{(1)} \circ \cdots \circ \frac{1}{\alpha} \mathbf{x}_\ell^{(N)} = \mathbf{x}_\ell^{(1)} \circ \cdots \circ \mathbf{x}_\ell^{(N)}$$

  for any $\alpha \neq 0$.
- Unboundedness: $\|x_\ell^{(n)}\|$ may become arbitrarily large.

Fix: Penalize factors with large norms.

Low-rank tensor completion

Low-rank tensor completion with regularization:

$$\min \left\| P_\Omega \left(\mathcal{A} - \sum_{\ell=1}^{L} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)} \right) \right\|_2^2 + \lambda \sum_{\ell=1}^{L} \sum_{n=1}^{N} \|\mathbf{x}_\ell^{(n)}\|_2^2$$

- λ regularization parameter (typically small, e.g., $\lambda = 10^{-3}$)
- nonlinear, nonconvex optimization problem
- no closed-form solution in terms of SVD for $N \geq 3$
- quadratic convex problem in *each* individual factor matrix $\mathbf{X}^{(n)} = \left[\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_L^{(n)}\right] \rightsquigarrow$ alternating optimization methods!

Previous parallel approaches to low-rank completion

$N = 2$ (matrix completion):

- ALS (alternating least-squares)
  [Teflioudi/Makari/Gemulla'2012, Zhou/Wilkinson/Schreiber/Pan'2008]
- CCD (cyclic coordinate descent)
  [Pilászy/Zibriczky/Tikk'2010, Yu et al.'2012–2013]
- SGD (stochastic gradient descent)
  [Gemulla et al.'2011, Recht/Ré'2013, Makari et al.'2015]

$N > 2$ (tensor completion):

- Well studied: Complete data = approximation of complete tensor
- Incomplete data usually "fixed" via weight matrices or imputation
  [Hidasi/Tikk'2012], [Acar et al.2011]
- Alternating proximal gradient method under additional nonnegativity constraints
  [Xu/Yin'2013]
- Parallelization of ALS based on local CP models
  [Phan/Cichoki'2011]

ALS

ALS (Alternating Least-Squares)

- ALS alternatingly optimizes factor matrices $\mathbf{X}^{(n)} = \left[\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_L^{(n)}\right]$ for $n = 1, \ldots, N$.
- Optimization of $I_1 \times L$ factor matrix $\mathbf{X}^{(1)}$ (while keeping all other factor matrices fixed):

$$\min \left\| P_\Omega \left(\mathcal{A} - \sum_{\ell=1}^{L} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)} \right) \right\|_2^2 + \lambda \cdots$$

  This decouples wrt rows of $\mathbf{X}^{(1)}$.

- I_1 decoupled optimization problems:

$$\min \sum_{\substack{\mathbf{i} \in \Omega \\ i_1 = \hat{\imath}}} \left(a_{\mathbf{i}} - \sum_{\ell=1}^{L} z_\ell \prod_{n=2}^{N} \left[\mathbf{x}_\ell^{(n)}\right]_{i_n} \right)^2 + \lambda \sum_{\ell=1}^{L} {z_\ell}^2, \qquad \hat{\imath} = 1, \ldots, I_1.$$

$\hat{\imath}$th row of $\mathbf{X}^{(1)}$ determined by $|\Omega_{1,\hat{\imath}}| \times L$ LSQ:

$$\min \|\mathbf{a}_{1,\hat{\imath}} - \mathbf{H}_{1,\hat{\imath}} \mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_2^2.$$

Solution $\mathbf{z}^* = (\mathbf{H}_{1,\hat{\imath}}^T \mathbf{H}_{1,\hat{\imath}} + \lambda I)^{-1} \mathbf{H}_{1,\hat{\imath}}^T \mathbf{a}_{1,\hat{\imath}}$.

ALS

Algorithm 1: ALS

Initialize all vectors $\mathbf{x}_\ell^{(n)}$.
for *each outer iteration* **do**
  for $\hat{n} = 1, 2, \ldots, N$ **do**
    for $\hat{\imath} = 1, 2, \ldots, I_{\hat{n}}$ **do in parallel**
      Set up and solve $\mathbf{z}^* = (\mathbf{H}_{\hat{n},\hat{\imath}}^\mathsf{T} \mathbf{H}_{\hat{n},\hat{\imath}} + \lambda I)^{-1} \mathbf{H}_{\hat{n},\hat{\imath}}^\mathsf{T} \mathbf{a}_{\hat{n},\hat{\imath}}$.
      Update L entries $\left[\mathbf{x}_\ell^{(\hat{n})}\right]_{\hat{\imath}}$, $\ell = 1, 2, \ldots, L$, using $\mathbf{z}^*$.

Tempting parallelization:

- Parallelize wrt variables in inner loop.
- Extension of distributed ALS [Teflioudi/Makari/Gemulla'2012] to tensors: Each node owns corresponding data $\mathbf{a}_{\hat{n},\hat{\imath}}$.
- Requires N replications of $\mathcal{A}$!

Parallelization of ALS

- $\Omega = \Omega^{(1)} \cup \cdots \cup \Omega^{(p)}$ partition of index set Ω over p processors
- Processor q owns $a_{\mathbf{i}}$ for all $\mathbf{i} \in \Omega^{(q)}$.
- Factor matrices $\mathbf{X}^{(n)}$ for $n = 1, \ldots, N$ of CP decomposition of $\boldsymbol{\mathcal{X}}$ replicated on all p processors.

Algorithm 2: Parallel formulation of ALS.

Initialize all vectors $\mathbf{x}_{\ell}^{(n)}$ in parallel;
for *each outer iteration* **do**
  for $\hat{n} = 1, 2, \ldots, N$ **do**
    Construct local contributions to $\mathbf{C}_{\hat{\imath}} := \mathbf{H}_{\hat{n},\hat{\imath}}^{\mathsf{T}} \mathbf{H}_{\hat{n},\hat{\imath}}$ and $\mathbf{d}_{\hat{\imath}} := \mathbf{H}_{\hat{n},\hat{\imath}}^{\mathsf{T}} \mathbf{a}_{\hat{n},\hat{\imath}}$ for $\hat{\imath} = 1, \ldots, I_{\hat{n}}$.
    Reduce local contributions.
    Scatter $I_{\hat{n}}$ coefficients $\mathbf{C}_{\hat{\imath}}$, $\mathbf{d}_{\hat{\imath}}$ across all processors.
    Solve (roughly $I_{\hat{n}}/p$) local normal equations $(\mathbf{C}_{\hat{\imath}} + \lambda \mathbf{I})\mathbf{z}_{\hat{\imath}} = \mathbf{d}_{\hat{\imath}}$.
    Replicate updated factor matrix $\mathbf{X}^{(\hat{n})}$;

Cyclic coordinate descent

Cyclic coordinate descent

Motivation: ALS expensive for larger tensor ranks L.
Computation $\mathcal{O}(L^3)$, Memory/communication bandwidth $\mathcal{O}(L^2)$.

Idea:

- Optimize only one term in $\boldsymbol{\mathcal{X}} = \sum_{\ell=1}^{L} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)}$.
- Attempt to optimize $\hat{\ell}$th term by one sweep of rank-one ALS applied to

$$\boldsymbol{\mathcal{A}} - \sum_{\ell \neq \hat{\ell}} \mathbf{x}_\ell^{(1)} \circ \mathbf{x}_\ell^{(2)} \circ \cdots \circ \mathbf{x}_\ell^{(N)}.$$

- $\hat{\imath}$th entry of $\mathbf{x}_{\hat{\ell}}^{(\hat{n})}$ is replaced by

$$z^* = \frac{\sum\limits_{\substack{\mathbf{i}\in\Omega \\ i_{\hat{n}}=\hat{\imath}}} \left(a_{\mathbf{i}} - \sum\limits_{\substack{\ell=1 \\ \ell\neq\hat{\ell}}}^{L} \prod\limits_{n=1}^{N} \left[\mathbf{x}_\ell^{(n)}\right]_{i_n} \right) \prod\limits_{\substack{n=1 \\ n\neq\hat{n}}}^{N} \left[\mathbf{x}_{\hat{\ell}}^{(n)}\right]_{i_n}}{\lambda + \sum\limits_{\substack{\mathbf{i}\in\Omega \\ i_{\hat{n}}=\hat{\imath}}} \left(\prod\limits_{\substack{n=1 \\ n\neq\hat{n}}}^{N} \left[\mathbf{x}_{\hat{\ell}}^{(n)}\right]_{i_n} \right)^2}.$$

Cyclic coordinate descent

Algorithm 3: CCD++

```
Initialize all vectors x_ℓ^(n).
for each outer iteration do
    for ℓ̂ = 1, 2, ..., L do
        for n̂ = 1, 2, ..., N do
            for î = 1, 2, ..., I_n̂ do in parallel
                Update entry [x_ℓ̂^(n̂)]_î .
```

Initialize all vectors $\mathbf{x}_\ell^{(n)}$.
for *each outer iteration* **do**
  for $\hat{\ell} = 1, 2, \ldots, L$ **do**
    for $\hat{n} = 1, 2, \ldots, N$ **do**
      for $\hat{\imath} = 1, 2, \ldots, I_{\hat{n}}$ **do in parallel**
        Update entry $\left[\mathbf{x}_{\hat{\ell}}^{(\hat{n})}\right]_{\hat{\imath}}$.

- Proposed by [Yu/Hsieh/Si/Dhillon'2012] for matrix completion.
- Keeping track of

$$r_{\mathbf{i}} := a_{\mathbf{i}} - \sum_{\substack{\ell=1 \\ \ell \neq \hat{\ell}}}^{L} \prod_{n=1}^{N} \left[\mathbf{x}_\ell^{(n)}\right]_{i_n}$$

  reduces cost by a factor $\approx 1/N$, at the expense of storing sparse tensor $\mathcal{R}$.
- Convergence to critical point follows from [Xu/Yin'2013].

Parallel cyclic coordinate descent

- $\Omega = \Omega^{(1)} \cup \cdots \cup \Omega^{(p)}$ partition of index set Ω over p processors
- Processor q owns $a_{\mathbf{i}}$ *and* $r_{\mathbf{i}}$ for all $\mathbf{i} \in \Omega^{(q)}$.
- Factor matrices $\mathbf{X}^{(n)}$ for $n = 1, \ldots, N$ of CP decomposition of $\boldsymbol{\mathcal{X}}$ replicated on all p processors.

Algorithm 4: Parallel CCD++

Initialize in parallel all vectors $\mathbf{x}_{\ell}^{(n)}$ and $\mathcal{R}$.
for *each outer iteration* **do**
  for $\hat{\ell} = 1, 2, \ldots, L$ **do**
    for $\hat{n} = 1, 2, \ldots, N$ **do**
      Construct local contributions to tensor contractions
      $\alpha_{\hat{\imath}} = \sum_{\substack{\mathbf{i} \in \Omega \\ i_{\hat{n}} = \hat{\imath}}} r_{\mathbf{i}} \gamma_{\mathbf{i}}$ and $\beta_{\hat{\imath}} = \sum_{\substack{\mathbf{i} \in \Omega \\ i_{\hat{n}} = \hat{\imath}}} \prod_{\substack{n=1 \\ n \neq \hat{n}}}^{N} \left[\mathbf{x}_{\hat{\ell}}^{(n)}\right]_{i_n}^2$ for $\hat{\imath} = 1, \ldots, I_{\hat{n}}$.
      Reduce local contributions by summation.
      Perform $\left[\mathbf{x}_{\hat{\ell}}^{(\hat{n})}\right]_{\hat{\imath}} \leftarrow \frac{\alpha_{\hat{\imath}}}{\lambda + \beta_{\hat{\imath}}}$ on master processor.
      Replicate updated variables $\mathbf{x}_{\hat{\ell}}^{(\hat{n})}$.
    Update $\mathcal{R}$ in parallel.

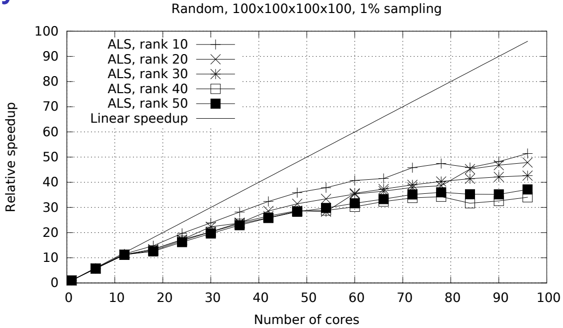
Numerical experiments

Computational environment

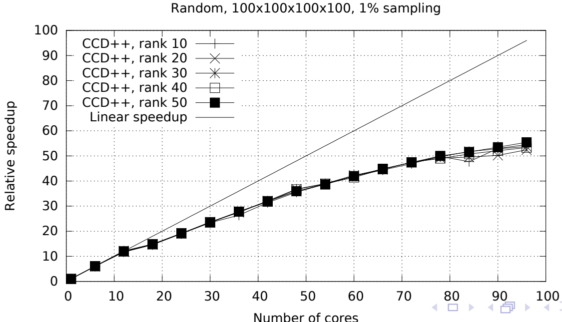
- Abisko distributed memory system at HPC2N in Umeå/Sweden.
- Each node contains four sockets with AMD Opteron 6238 processors.
- Each processor contains 12 cores partitioned into two NUMA domains.
- Nodes are interconnected with 40 Gb/s Mellanox Infiniband.
- PathScale C++ compiler with OpenMPI.

Results: Synthetic data

ALS:



CCD++

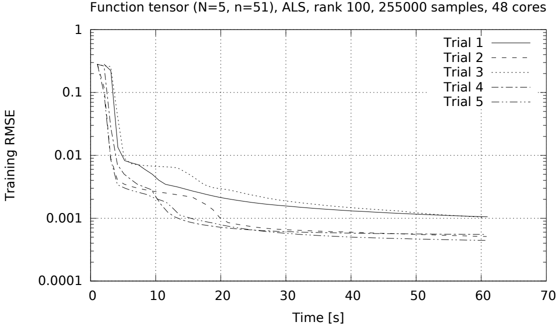


Function-related tensors

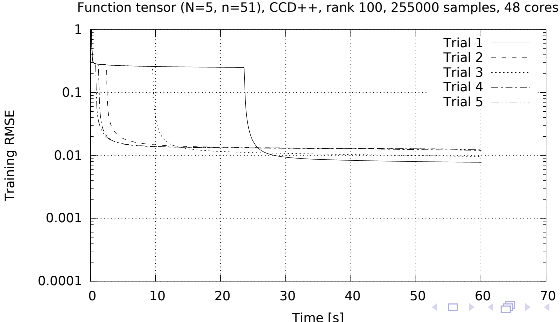
- Discretization of $\mathbf{x} \mapsto \exp(-\|\mathbf{x}\|_2)$, which has a cusp singularity at the origin, on a tensor product grid.
- $a_{i_1,i_2,\ldots,i_N} = \exp(-\sqrt{\xi_{i_1}^2 + \xi_{i_2}^2 + \cdots + \xi_{i_N}^2})$
- Sample $10nNL$ entries of $\mathcal{A}$.
- Application of tensor completion: High-dimensional integration.
- Experiments: $n = 51$, $N = 5$, $L = 100$, 48 cores.

Results: Function-related tensors

ALS:

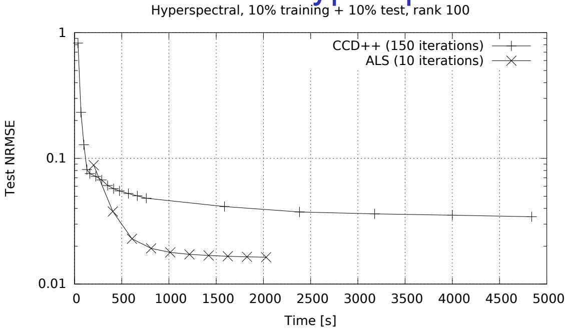


CCD++

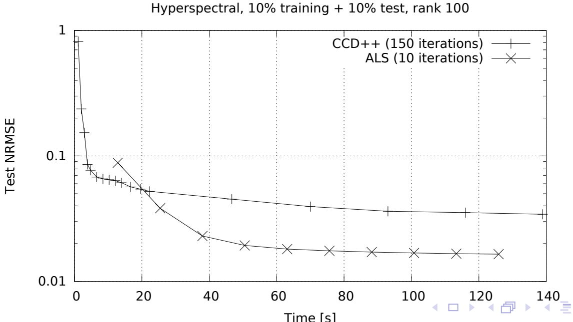


Results: 1017 × 1340 × 33 hyperspectral image

1 core:

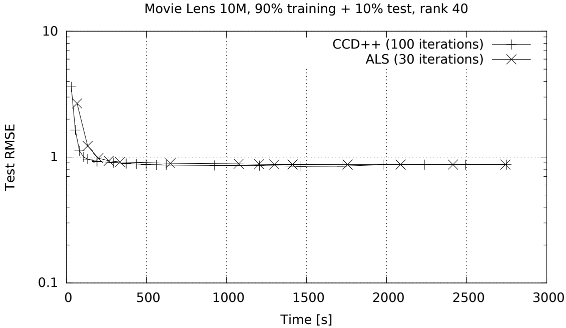


48 cores:

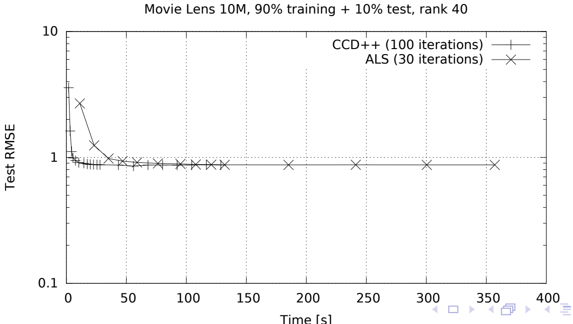


Results: $71\,567 \times 10\,677 \times 731$ MovieLens tensor

1 core:



48 cores:



Conclusions

Conclusions and ongoing work

- Parallel CCD++ method of choice for data-related applications.
- Parallel ALS sometimes better when high accuracy is required.
- Both algorithms are weakly scalable.

More details in:

- L. Karlsson, D. Kressner, and A. Uschmajew. Parallel algorithms for tensor completion in the CP format. Parallel Computing, 2015.

Outlook:

- (MPI+)Cuda: Ongoing joint work with Efthalia Karydi.
- Adjust distribution of Ω to increase data locality.
- Application to UQ. (Talk by Grasedyck on Tuesday)