Factoring Tensors in the Cloud: A Tutorial on Big Tensor Data Analytics

Nikos Sidiropoulos, UMN, and Evangelos Papalexakis, CMU

ICASSP Tutorial, 5/5/2014
Nikos Sidiropoulos, UMN
- Signal and Tensor Analytics Research (STAR) group
  https://sites.google.com/a/umn.edu/nikosgroup/home
  - Signal processing, communications, optimization
  - Tensor decomposition, factor analysis
  - Big data
  - Spectrum sensing, cognitive radio
  - Sidekicks: preference measurement, conference review & session assignment

Vangelis Papalexakis, Christos Faloutsos, CMU
- http://www.cs.cmu.edu/~epapalex/ and
  http://www.cs.cmu.edu/~christos/
  - Data mining
  - Graph mining
  - Tensor analysis, coupled matrix-tensor analysis
  - Big data, Hadoop

Matrices, rank decomposition

- A matrix (or *two-way array*) is a dataset \( X \) indexed by *two indices*, \((i, j)\)-th entry \( X(i, j) \).
- Simple matrix \( S(i, j) = a(i)b(j), \forall i, j; \) separable, every row (column) proportional to every other row (column). Can write as \( S = ab^T \).
- \( \text{rank}(X) \) := smallest number of ‘simple’ (separable, rank-one) matrices needed to generate \( X \) - a measure of complexity.

\[
X(i, j) = \sum_{f=1}^{F} a_f(i)b_f(j); \text{ or } X = \sum_{f=1}^{F} a_f b_f^T = AB^T.
\]

- Turns out \( \text{rank}(X) \) = maximum number of linearly independent rows (or, columns) in \( X \).
- Rank decomposition for matrices is not unique (except for matrices of rank = 1), as \( \forall \) invertible \( M \):

\[
X = AB^T = (AM)\left(M^{-T}B^T\right) = (AM)\left(BM^{-1}\right)^T = \tilde{A}\tilde{B}^T.
\]
CS ‘slang’ for *three-way array*: dataset $\mathbf{X}$ indexed by *three indices*, $(i, j, k)$-th entry $\mathbf{X}(i, j, k)$.

In plain words: a ‘shoebox’!

**Warning**: different meaning in Physics!

For two vectors $\mathbf{a} (I \times 1)$ and $\mathbf{b} (J \times 1)$, $\mathbf{a} \odot \mathbf{b}$ is an $I \times J$ rank-one matrix with $(i, j)$-th element $a(i)b(j)$; i.e., $\mathbf{a} \odot \mathbf{b} = \mathbf{a} \mathbf{b}^T$.

For three vectors, $\mathbf{a} (I \times 1)$, $\mathbf{b} (J \times 1)$, $\mathbf{c} (K \times 1)$, $\mathbf{a} \odot \mathbf{b} \odot \mathbf{c}$ is an $I \times J \times K$ rank-one three-way array with $(i, j, k)$-th element $a(i)b(j)c(k)$.

The *rank of a three-way array* $\mathbf{X}$ is the smallest number of outer products needed to synthesize $\mathbf{X}$. 
Motivating example: NELL @ CMU / Tom Mitchell

- Crawl web, learn language ‘like children do’: encounter new concepts, learn from context
- NELL triplets of “subject-verb-object” naturally lead to a 3-mode tensor

\[
\mathbf{X} \approx \sum_{f=1}^{F} \mathbf{a}_f \mathbf{b}_f \mathbf{c}_f
\]

Each rank-one factor corresponds to a \textit{concept}, e.g., ‘leaders’ or ‘tools’

E.g., say \( \mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1 \) corresponds to ‘leaders’: subjects/rows with high score on \( \mathbf{a}_1 \) will be “Obama”, “Merkel”, “Steve Jobs”, objects/columns with high score on \( \mathbf{b}_1 \) will be “USA”, “Germany”, “Apple Inc.”, and verbs/fibers with high score on \( \mathbf{c}_1 \) will be ‘verbs’, like “lead”, “is-president-of”, and “is-CEO-of”.

Nikos Sidiropoulos, Evangelos Papalexakis

Factoring Big Tensors in the Cloud

ICASSP Tutorial, 5/5/2014 5 / 55
Kronecker and Khatri-Rao products

- $\otimes$ stands for the Kronecker product:

$$A \otimes B = \begin{bmatrix} BA(1, 1), BA(1, 2), \cdots \\ BA(2, 1), BA(2, 2), \cdots \\ \vdots \end{bmatrix}$$

- $\odot$ stands for the Khatri-Rao (column-wise Kronecker) product: given $A$ ($I \times F$) and $B$ ($J \times F$), $A \odot B$ is the $JI \times F$ matrix

$$A \odot B = \begin{bmatrix} A(:, 1) \otimes B(:, 1) & \cdots & A(:, F) \otimes B(:, F) \end{bmatrix}$$

- vec$(ABC) = (C^T \otimes A)vec(B)$

- If $D = \text{diag}(d)$, then vec$(ADC) = (C^T \odot A)d$
Rank decomposition for tensors

- **Tensor:**
  \[ X = \sum_{f=1}^{F} a_f \circ b_f \circ c_f \]

- **Scalar:**
  \[ X(i, j, k) = \sum_{f=1}^{F} a_{i,f}b_{j,f}c_{k,f}, \quad \forall i \in \{1, \ldots, I\} \]
  \[ \forall j \in \{1, \ldots, J\} \]
  \[ \forall k \in \{1, \ldots, K\} \]

- **Slabs:**
  \[ X_k = AD_k(C)B^T, \quad k = 1, \ldots, K \]

- **Matrix:**
  \[ X^{(KJ \times I)} = (B \circ C)A^T \]

- **Tall vector:**
  \[ x^{(KJI)} := \text{vec} \left( X^{(KJ \times I)} \right) = (A \circ (B \circ C)) 1_{F \times 1} = (A \circ B \circ C) 1_{F \times 1} \]
Do I need to worry about higher-way (or, higher-order) tensors?

- Semantic analysis of Brain fMRI data
- fMRI $\rightarrow$ semantic category scores

- fMRI mode is vectorized ($O(10^5 - 10^6)$)
- Spatial coherence - better to treat as three separate spatial modes $\rightarrow$ 5-way array
- Temporal dynamics - include time as another dimension $\rightarrow$ 6-way array
Rank decomposition for higher-order tensors

• Tensor:
  \[ X = \sum_{f=1}^{F} a_{f}^{(1)} \circ \cdots \circ a_{f}^{(N)} \]

• Scalar:
  \[ X (i_1, \cdots, i_N) = \sum_{f=1}^{F} \prod_{n=1}^{N} a_{in,f}^{(n)} \]

• Matrix:
  \[ X^{(I_1I_2 \cdots I_{N-1} \times I_N)} = \left( A^{(N-1)} \circ A^{(N-2)} \circ \cdots \circ A^{(1)} \right) \left( A^{(N)} \right)^T \]

• Tall vector:
  \[ x^{(I_1 \cdots I_N)} := \text{vec} \left( X^{(I_1I_2 \cdots I_{N-1} \times I_N)} \right) = \left( A^{(N)} \circ A^{(N-1)} \circ A^{(N-2)} \circ \cdots \circ A^{(1)} \right) 1_{F \times 1} \]
For matrices, SVD is instrumental: rank-revealing, Eckart-Young

So is there a tensor equivalent to the matrix SVD?

Yes, ... and no! In fact there is no single tensor SVD.

Two basic decompositions:

- CANonical DECOMPosition (CANDECOMP), also known as PARAllel FACtor (PARAFAC) analysis, or CANDECOMP-PARAFAC (CP) for short: non-orthogonal, unique under certain conditions.
- Tucker3, orthogonal without loss of generality, non-unique except for very special cases.

Both are outer product decompositions, but with very different structural properties.

Rule of thumb: use Tucker3 for subspace estimation and tensor approximation, e.g., compression applications; use PARAFAC for latent parameter estimation - recovering the ‘hidden’ rank-one factors.
I × J × K three-way array X
A : I × L, B : J × M, C : K × N mode loading matrices
G : L × M × N Tucker3 core
Consider an $I \times J \times K$ three-way array $X$ comprising $K$ matrix slabs \( \{X_k\}_{k=1}^{K} \), arranged into matrix $X := [\text{vec}(X_1), \cdots, \text{vec}(X_K)]$.

The Tucker3 model can be written as

$$X \approx (B \otimes A)GC^T,$$

where $G$ is the Tucker3 core tensor $G$ recast in matrix form. The non-zero elements of the core tensor determine the interactions between columns of $A$, $B$, $C$.

The associated model-fitting problem is

$$\min_{A,B,C,G} ||X - (B \otimes A)GC^T||_F^2,$$

which is usually solved using an alternating least squares procedure.

$\text{vec}(X) \approx (C \otimes B \otimes A) \text{vec}(G)$.

Highly non-unique - e.g., rotate $C$, counter-rotate $G$ using unitary matrix.

Subspaces can be recovered; Tucker3 is good for tensor approximation, not latent parameter estimation.
PARAFAC

Low-rank tensor decomposition / approximation

\[ X \approx \sum_{f=1}^{F} a_f \circ b_f \circ c_f, \]

PARAFAC [Harshman ’70-’72], CANDECOMP [Carroll & Chang, ’70], now CP; also cf. [Hitchcock, ’27]

\[ X_k \approx AD_k(C)B^T, \] where \( D_k(C) \) is a diagonal matrix holding the \( k \)-th row of \( C \) in its diagonal.

Combining slabs and using Khatri-Rao product,

\[ X \approx (B \odot A)C^T \iff \text{vec}(X) \approx (C \odot B \odot A)1 \]
Under certain conditions, PARAFAC is essentially unique, i.e., \((A, B, C)\) can be identified from \(X\) up to permutation and scaling of columns - there’s no rotational freedom; cf. [Kruskal ’77, Sidiropoulos et al ’00 - ’07, de Lathauwer ’04-, Stegeman ’06-, Chiantini, Ottaviani ’11-, ...]

\(I \times J \times K\) tensor \(X\) of rank \(F\), vectorized as \(IJK \times 1\) vector
\(x = (A \odot B \odot C) 1\), for some \(A (I \times F), B (J \times F),\) and \(C (K \times F)\) - a PARAFAC model of size \(I \times J \times K\) and order \(F\) parameterized by \((A, B, C)\).

The \textit{Kruskal-rank} of \(A\), denoted \(k_A\), is the maximum \(k\) such that any \(k\) columns of \(A\) are linearly independent \((k_A \leq r_A := \text{rank}(A))\).

Given \(X (\Leftrightarrow x)\), if \(k_A + k_B + k_C \geq 2F + 2\), then \((A, B, C)\) are unique up to a common column permutation and scaling/counter-scaling (e.g., multiply first column of \(A\) by 5, divide first column of \(B\) by 5, outer product stays the same) - cf. [Kruskal, 1977]

\(N\)-way case: \(\sum_{n=1}^{N} k_A^{(n)} \geq 2F + (N - 1)\) [Sidiropoulos & Bro, 2000]
Special case: Two slabs

- Assume square/tall, full column rank $A(I \times F)$ and $B(J \times F)$; and distinct nonzero elements along the diagonal of $D$:

\[
\begin{align*}
X_1 &= AB^T, \\
X_2 &= ADB^T
\end{align*}
\]

- Introduce the compact $F$-component SVD of

\[
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix} =
\begin{bmatrix}
A \\
AD
\end{bmatrix}B^T = U\Sigma V^H
\]

- $\text{rank}(B) = F$ implies that $\text{span}(U) = \text{span}\left(\begin{bmatrix}
A \\
AD
\end{bmatrix}\right)$; hence there exists a nonsingular matrix $P$ such that

\[
U =
\begin{bmatrix}
U_1 \\
U_2
\end{bmatrix} =
\begin{bmatrix}
A \\
AD
\end{bmatrix}P
\]
Special case: Two slabs, continued

Next, construct the auto- and cross-product matrices

\[
\begin{align*}
R_1 &= U_1^H U_1 = P^H A^H A P =: QP, \\
R_2 &= U_1^H U_2 = P^H A^H A D P = QDP.
\end{align*}
\]

All matrices on the right hand side are square and full-rank. Solving the first equation and substituting to the second, yields the eigenvalue problem

\[
\left( R_1^{-1} R_2 \right) P^{-1} = P^{-1} D
\]

\(P^{-1}\) (and \(P\)) recovered up to permutation and scaling

From bottom of previous page, \(A\) likewise recovered, then \(B\)

Starts making sense ...
Alternating Least Squares (ALS)

- Based on matrix view:
  \[ X^{(KJ \times I)} = (B \odot C)A^T \]

- Multilinear LS problem:
  \[
  \min_{A,B,C} \left\| X^{(KJ \times I)} - (B \odot C)A^T \right\|_F^2
  \]

- **NP-hard** - even for a single component, i.e., vector \(a, b, c\). See [Hillar and Lim, “Most tensor problems are NP-hard,” 2013]

- But ... given interim estimates of \(B, C\), can easily solve for conditional LS update of \(A\):
  \[ A_{CLS} = \left( (B \odot C)^\dagger X^{(KJ \times I)} \right)^T \]

- Similarly for the CLS updates of \(B, C\) (symmetry); alternate until cost function converges (monotonically).
Other algorithms?

- Many! - first-order (gradient-based), second-order (Hessian-based) Gauss-Newton, line search, Levenberg-Marquardt, weighted least squares, majorization
- Algebraic initialization (matters)
- See Tomasi and Bro, 2006, for a good overview
- Second-order advantage when close to optimum, but can (and do) diverge
- First-order often prone to local minima, slow to converge
- Stochastic gradient descent (CS community) - simple, parallel, but very slow
- Difficult to incorporate additional constraints like sparsity, non-negativity, unimodality, etc.
ALS

- No parameters to tune!
- Easy to program, uses standard linear LS
- Monotone convergence of cost function
- Does not require any conditions beyond model identifiability
- Easy to incorporate additional constraints, due to multilinearity, e.g., replace linear LS with linear NNLS for NN
- Even non-convex (e.g., FA) constraints can be handled with column-wise updates (optimal scaling lemma)
- Cons: sequential algorithm, convergence can be slow
- Still workhorse after all these years
Outliers, sparse residuals

Instead of LS,

$$\min ||X^{(KJ \times I)} - (B \circ C)A^T||_1$$

Conditional update: LP

Almost as good: coordinate-wise, using \textit{weighted median filtering} (very cheap!) [Vorobyov, Rong, Sidiropoulos, Gershman, 2005]

PARAFAC CRLB: [Liu & Sidiropoulos, 2001] (Gaussian); [Vorobyov, Rong, Sidiropoulos, Gershman, 2005] (Laplacian, etc differ only in pdf-dependent scale factor).

Alternating optimization algorithms approach the CRLB when the problem is well-determined (meaning: not barely identifiable).
Tensors can easily become really big! - size exponential in the number of dimensions (‘ways’, or ‘modes’).

Datasets with millions of items per mode - examples in second part of this tutorial.

Cannot load in main memory; may reside in cloud storage.

Sometimes very sparse - can store and process as (i,j,k,value) list, nonzero column indices for each row, runlength coding, etc.

(Sparse) Tensor Toolbox for Matlab [Kolda et al].

Avoids explicitly computing dense intermediate results.
Tensor partitioning?

- Parallel algorithms for matrix algebra use data partitioning
- Can we reuse some of these ideas?

- Low-hanging fruit?
- First considered in [Phan, Cichocki, Neurocomputing, 2011]
- Later revisited in [Almeida, Kibangou, CAMSAP 2013, ICASSP 2014]
Tensor partitioning

In [Phan & Cichocki, Neurocomputing, 2011] each sub-block is independently decomposed, then low-rank factors of the big tensor are ‘stitched’ together.

- The decomposition of each sub-block must be identifiable → far more stringent ID conditions;
- Each sub-block has different permutation and scaling of the factors - all these must be reconciled before stitching (not discussed in paper);
- Noise affects sub-block decomposition more than full tensor decomposition.

More recently, [Almeida & Kibangou, CAMSAP 2013, ICASSP 2014]:

- Also rely on partitioning, so sub-block identification conditions are required;
- But more robust, based on collaborative consensus-averaging, and the matching of different permutations and scales is explicitly taken into account;
- Requires considerable communication overhead across clusters and nodes; i.e., the parallel threads are not independent.
Tensor compression


\[ X = G + B + E \]

- Implemented in COMFAC
  
  http://www.ece.umn.edu/~nikos/comfac.m

- Lossless if exact mode bases used [CANDELINC]; but Tucker3 fitting is itself cumbersome for big tensors (big matrix SVDs), cannot compress below mode ranks without introducing errors
Consider compressing \( \mathbf{x} = \text{vec}(\mathbf{X}) \) into \( \mathbf{y} = \mathbf{Sx} \), where \( \mathbf{S} \) is \( d \times IJK \), \( d \ll IJK \).

In particular, consider a specially structured compression matrix \( \mathbf{S} = \mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T \).

Corresponds to multiplying (every slab of) \( \mathbf{X} \) from the \( I \)-mode with \( \mathbf{U}^T \), from the \( J \)-mode with \( \mathbf{V}^T \), and from the \( K \)-mode with \( \mathbf{W}^T \), where \( \mathbf{U} \) is \( I \times L \), \( \mathbf{V} \) is \( J \times M \), and \( \mathbf{W} \) is \( K \times N \), with \( L \leq I \), \( M \leq J \), \( N \leq K \) and \( LMN \ll IJK \).
Due to a property of the Kronecker product

\[
\left( U^T \otimes V^T \otimes W^T \right) \left( A \odot B \odot C \right) = \\
\left( (U^T A) \odot (V^T B) \odot (W^T C) \right),
\]

from which it follows that

\[
y = \left( (U^T A) \odot (V^T B) \odot (W^T C) \right) 1 = \left( \tilde{A} \odot \tilde{B} \odot \tilde{C} \right) 1.
\]

i.e., the compressed data follow a PARAFAC model of size $L \times M \times N$ and order $F$ parameterized by $(\tilde{A}, \tilde{B}, \tilde{C})$, with $\tilde{A} := U^T A$, $\tilde{B} := V^T B$, $\tilde{C} := W^T C$. 
Random multi-way compression can be better!

- Assume that the columns of $A$, $B$, $C$ are sparse, and let $n_a$ ($n_b$, $n_c$) be an upper bound on the number of nonzero elements per column of $A$ (respectively $B$, $C$).
- Let the mode-compression matrices $U$ ($I \times L$, $L \leq I$), $V$ ($J \times M$, $M \leq J$), and $W$ ($K \times N$, $N \leq K$) be randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{IL}$, $\mathbb{R}^{JM}$, and $\mathbb{R}^{KN}$, respectively.
- If
  \[
  \min(L, k_A) + \min(M, k_B) + \min(N, k_C) \geq 2F + 2, \quad \text{and} \\
  L \geq 2n_a, \quad M \geq 2n_b, \quad N \geq 2n_c,
  \]
  then the original factor loadings $A$, $B$, $C$ are almost surely identifiable from the compressed data.
Proof rests on two lemmas + Kruskal

- Lemma 1: Consider $\tilde{A} := U^T A$, where $A$ is $I \times F$, and let the $I \times L$ matrix $U$ be randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{IL}$ (e.g., multivariate Gaussian with a non-singular covariance matrix). Then $k_{\tilde{A}} = \min(L, k_A)$ almost surely (with probability 1) [Sidiropoulos & Kyrillidis, 2012]

- Lemma 2: Consider $\tilde{A} := U^T A$, where $\tilde{A}$ and $U$ are given and $A$ is sought. Suppose that every column of $A$ has at most $n_a$ nonzero elements, and that $k_{U^T} \geq 2n_a$. (The latter holds with probability 1 if the $I \times L$ matrix $U$ is randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{IL}$, and $\min(I, L) \geq 2n_a$.) Then $A$ is the unique solution with at most $n_a$ nonzero elements per column [Donoho & Elad, ’03]
First fitting PARAFAC in compressed space and then recovering the sparse $A$, $B$, $C$ from the fitted compressed factors entails complexity $O(LMNF + (I^{3.5} + J^{3.5} + K^{3.5})F)$ in practice.

Using sparsity first and then fitting PARAFAC in raw space entails complexity $O(IJKF + (IJK)^{3.5})$ - the difference is huge.

Also note that the proposed approach does not require computations in the uncompressed data domain, which is important for big data that do not fit in memory for processing.
Further compression - down to $O(\sqrt{F})$ in 2/3 modes

- Assume that the columns of $A$, $B$, $C$ are sparse, and let $n_a$ ($n_b$, $n_c$) be an upper bound on the number of nonzero elements per column of $A$ (respectively $B$, $C$).
- Let the mode-compression matrices $U$ ($I \times L$, $L \leq I$), $V$ ($J \times M$, $M \leq J$), and $W$ ($K \times N$, $N \leq K$) be randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{IL}$, $\mathbb{R}^{JM}$, and $\mathbb{R}^{KN}$, respectively.
- If
  \[ r_A = r_B = r_C = F \]
  \[ L(L - 1)M(M - 1) \geq 2F(F - 1), \quad N \geq F, \quad \text{and} \]
  \[ L \geq 2n_a, \quad M \geq 2n_b, \quad N \geq 2n_c, \]
then the original factor loadings $A$, $B$, $C$ are almost surely identifiable from the compressed data up to a common column permutation and scaling.
Proof: Lemma 3 + results on a.s. ID of PARAFAC

Lemma 3: Consider $\tilde{A} = U^T A$, where $A (I \times F)$ is deterministic, tall/square ($I \geq F$) and full column rank $r_A = F$, and the elements of $U (I \times L)$ are i.i.d. Gaussian zero mean, unit variance random variables. Then the distribution of $\tilde{A}$ is nonsingular multivariate Gaussian.

From [Stegeman, ten Berge, de Lathauwer 2006] (see also [Jiang, Sidiropoulos 2004], we know that PARAFAC is almost surely identifiable if the loading matrices $\tilde{A}, \tilde{B}$ are randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{(L+M)F}$, $\tilde{C}$ is full column rank, and $L(L-1)M(M-1) \geq 2F(F-1)$. 
Theorem 3: Let $x = (A_1 \odot \cdots \odot A_\delta) 1 \in \mathbb{R}^{\prod_{d=1}^{\delta} l_d}$, where $A_d$ is $l_d \times F$, and consider compressing it to $y = (U^T_1 \otimes \cdots \otimes U^T_\delta) x = ((U^T_1 A_1) \odot \cdots \odot (U^T_\delta A_\delta)) 1 = (\tilde{A}_1 \odot \cdots \odot \tilde{A}_\delta) 1 \in \mathbb{R}^{\prod_{d=1}^{\delta} L_d}$, where the mode-compression matrices $U_d$ ($l_d \times L_d$, $L_d \leq l_d$) are randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{l_d L_d}$. Assume that the columns of $A_d$ are sparse, and let $n_d$ be an upper bound on the number of nonzero elements per column of $A_d$, for each $d \in \{1, \cdots, \delta\}$. If

$$\sum_{d=1}^{\delta} \min(L_d, k_{A_d}) \geq 2F + \delta - 1, \quad \text{and} \quad L_d \geq 2n_d, \quad \forall d \in \{1, \cdots, \delta\},$$

then the original factor loadings $\{A_d\}_{d=1}^{\delta}$ are almost surely identifiable from the compressed data $y$ up to a common column permutation and scaling.

Various additional results possible.
Luca Chiantini and Giorgio Ottaviani, On Generic Identifiability of 3-Tensors of Small Rank, SIAM. J. Matrix Anal. & Appl., 33(3), 1018–1037:

Consider an $I \times J \times K$ tensor $X$ of rank $F$, and order the dimensions so that $I \leq J \leq K$

Let $i$ be maximal such that $2^i \leq I$, and likewise $j$ maximal such that $2^i \leq J$

If $F \leq 2^{i+j-2}$, then $X$ has a unique decomposition almost surely

For $I, J$ powers of 2, the condition simplifies to $F \leq \frac{IJ}{4}$

More generally, condition implies:

- if $F \leq \frac{(I+1)(J+1)}{16}$, then $X$ has a unique decomposition almost surely
Even further compression

Assume that the columns of $A$, $B$, $C$ are sparse, and let $n_a$ ($n_b$, $n_c$) be an upper bound on the number of nonzero elements per column of $A$ (respectively $B$, $C$).

Let the mode-compression matrices $U$ $(I \times L, L \leq I)$, $V$ $(J \times M, M \leq J)$, and $W$ $(K \times N, N \leq K)$ be randomly drawn from an absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{R}^{IL}$, $\mathbb{R}^{JM}$, and $\mathbb{R}^{KN}$, respectively.

Assume $L \leq M \leq N$, and $L$, $M$ are powers of 2, for simplicity.

If

$$r_A = r_B = r_C = F$$

$$LM \geq 4F, \ N \geq M \geq L, \ \text{and}$$

$$L \geq 2n_a, \ M \geq 2n_b, \ N \geq 2n_c,$$

then the original factor loadings $A$, $B$, $C$ are almost surely identifiable from the compressed data up to a common column permutation and scaling.

Allows compression down to order of $\sqrt{F}$ in all three modes.
What if \( A, B, C \) are not sparse?

- If \( A, B, C \) are sparse with respect to known bases, i.e., \( A = R\tilde{A}, B = S\tilde{B}, \) and \( C = T\tilde{C} \), with \( R, S, T \) the respective sparsifying bases, and \( \tilde{A}, \tilde{B}, \tilde{C} \) sparse.
- Then the previous results carry over under appropriate conditions, e.g., when \( R, S, T \) are non-singular.
- OK, but what if such bases cannot be found?
Assume $\tilde{\mathbf{A}}_p, \tilde{\mathbf{B}}_p, \tilde{\mathbf{C}}_p$ identifiable from $\mathbf{Y}_p$ (up to perm & scaling of cols)

Upon factoring $\mathbf{Y}_p$ into $F$ rank-one components, we obtain

$$
\tilde{\mathbf{A}}_p = \mathbf{U}_p^T \mathbf{A} \Pi_p \Lambda_p.
$$

(1)

Assume first 2 columns of each $\mathbf{U}_p$ are common, let $\bar{\mathbf{U}}$ denote this common part, and $\bar{\mathbf{A}}_p :=$ first two rows of $\tilde{\mathbf{A}}_p$. Then

$$
\bar{\mathbf{A}}_p = \bar{\mathbf{U}}^T \mathbf{A} \Pi_p \Lambda_p.
$$

Dividing each column of $\bar{\mathbf{A}}_p$ by the element of maximum modulus in that column, denoting the resulting $2 \times F$ matrix $\hat{\mathbf{A}}_p$,

$$
\hat{\mathbf{A}}_p = \bar{\mathbf{U}}^T \mathbf{A} \Lambda \Pi_p.
$$

$\Lambda$ does not affect the ratio of elements in each $2 \times 1$ column. If ratios are distinct, then permutations can be matched by sorting the ratios of the two coordinates of each $2 \times 1$ column of $\hat{\mathbf{A}}_p$. 

In practice using a few more ‘anchor’ rows will improve perm-matching.

When $S$ anchor rows are used, the opt permutation matching cast as

$$\min_{\Pi} ||\hat{A}_1 - \hat{A}_p \Pi||_F^2,$$

Optimization over set of permutation matrices - hard?

$$||\hat{A}_1 - \hat{A}_p \Pi||_F^2 = \text{Tr} \left( (\hat{A}_1 - \hat{A}_p \Pi)^T (\hat{A}_1 - \hat{A}_p \Pi) \right) =$$

$$||\hat{A}_1||_F^2 + ||\hat{A}_p \Pi||_F^2 - 2\text{Tr}(\hat{A}_1^T \hat{A}_p \Pi) =$$

$$||\hat{A}_1||_F^2 + ||\hat{A}_p||_F^2 - 2\text{Tr}(\hat{A}_1^T \hat{A}_p \Pi).$$

$$\iff \max_{\Pi} \text{Tr}(\hat{A}_1^T \hat{A}_p \Pi),$$

*Linear Assignment Problem* (LAP), efficient soln via *Hungarian Algorithm.*
After perm-matching, back to (1) and permute columns → $\tilde{A}_p$ satisfying

$$\tilde{A}_p = U_p^T A \Pi \Lambda_p.$$  

Remains to get rid of $\Lambda_p$. For this, we can again resort to the first two common rows, and divide each column of $\tilde{A}_p$ with its top element →

$$\tilde{A}_p = U_p^T A \Pi \Lambda.$$  

For recovery of $A$ up to perm-scaling of cols, we then require that

$$\begin{bmatrix} \tilde{A}_1 \\
\vdots \\
\tilde{A}_p \end{bmatrix} = \begin{bmatrix} U_1^T \\
\vdots \\
U_p^T \end{bmatrix} A \Pi \Lambda$$  

be full column rank.
This implies that

\[ 2 + \sum_{p=1}^{P} (L_p - 2) \geq I. \]

Every sub-matrix contains the two common anchor rows, duplicate rows do not increase rank.

Once dim requirement met, full rank w.p. 1, non-redundant entries drawn from jointly cont. distribution (by design).

Assuming \( L_p = L, \forall p \in \{1, \cdots, P\} \)

\[ P \geq \frac{I - 2}{L - 2}. \]

Common column perm of \( \tilde{A}_p, \tilde{B}_p, \tilde{C}_p \) is common →

\[ P \geq \max \left( \frac{I - 2}{L - 2}, \frac{J}{M}, \frac{K}{N} \right) \]
If compression ratios in different modes are similar, makes sense to use longest mode for anchoring; if this is the last mode, then

\[ P \geq \max \left( \frac{I}{L}, \frac{J}{M}, \frac{K - 2}{N - 2} \right) \]

**Theorem:** Assume that \( F \leq I \leq J \leq K \), and \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) are full column rank (\( F \)). Further assume that \( L_p = L, M_p = M, N_p = N, \forall p \in \{1, \ldots, P\}, \)

\( L \leq M \leq N, (L + 1)(M + 1) \geq 16F, \) random \( \{\mathbf{U}_p\}_{p=1}^{P}, \{\mathbf{V}_p\}_{p=1}^{P}, \) each \( \mathbf{W}_p \) contains two common anchor columns, otherwise random \( \{\mathbf{W}_p\}_{p=1}^{P}. \)

Then \( \left( \tilde{\mathbf{A}}_p, \tilde{\mathbf{B}}_p, \tilde{\mathbf{C}}_p \right) \) unique up to column permutation and scaling.

If, in addition, \( P \geq \max \left( \frac{I}{L}, \frac{J}{M}, \frac{K - 2}{N - 2} \right) \), then \( \left( \mathbf{A}, \mathbf{B}, \mathbf{C} \right) \) are almost surely identifiable from \( \left\{ \left( \tilde{\mathbf{A}}_p, \tilde{\mathbf{B}}_p, \tilde{\mathbf{C}}_p \right) \right\}_{p=1}^{P} \) up to a common column permutation and scaling.
PARACOMP - Significance

- Indicative of a family of results that can be derived.
- Theorem shows that fully parallel computation of the big tensor decomposition is possible – first result that guarantees ID of the big tensor decomposition from the small tensor decompositions, without stringent additional constraints.

**Corollary:** If \( \frac{K-2}{N-2} = \max\left(\frac{I}{L}, \frac{J}{M}, \frac{K-2}{N-2}\right) \), then the memory / storage and computational complexity savings afforded by PARACOMP relative to brute-force computation are of order \( \frac{IJ}{F} \).

- Note on complexity of solving master join equation: after removing redundant rows, system matrix in (2) will have approximately orthogonal columns for large \( I \rightarrow \) left pseudo-inverse \( \approx \) its transpose, complexity \( I^2F \).
Cases where $F > \min(I, J, K)$ can be treated using Kruskal’s condition; total storage and complexity gains still of order $\frac{IJ}{F^2}$.

Latent sparsity: Assume that every column of $A$ ($B$, $C$) has at most $n_a$ (resp. $n_b$, $n_c$) nonzero elements. A column of $A$ can be uniquely recovered from only $2n_a$ incoherent linear equations.

Therefore, we may replace the condition

$$P \geq \max \left( \frac{I}{L}, \frac{J}{M}, \frac{K - 2}{N - 2} \right),$$

with

$$P \geq \max \left( \frac{2n_a}{L}, \frac{2n_b}{M}, \frac{2n_c - 2}{N - 2} \right).$$
Color of compressed noise

- \( \mathbf{Y} = \mathbf{X} + \mathbf{Z} \), where \( \mathbf{Z} \): zero-mean additive white noise.
- \( \mathbf{y} = \mathbf{x} + \mathbf{z} \), with \( \mathbf{y} := \text{vec} (\mathbf{Y}) \), \( \mathbf{x} := \text{vec} (\mathbf{X}) \), \( \mathbf{z} := \text{vec} (\mathbf{Z}) \).
- Multi-way compression \( \rightarrow \mathbf{Y}_c \)

\[
\mathbf{y}_c := \text{vec} (\mathbf{Y}_c) = \left( \mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T \right) \mathbf{y} = \\
\left( \mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T \right) \mathbf{x} + \left( \mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T \right) \mathbf{z}.
\]

- Let \( \mathbf{z}_c := (\mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T) \mathbf{z} \). Clearly, \( E[\mathbf{z}_c] = 0 \); it can be shown that

\[
E \left[ \mathbf{z}_c \mathbf{z}_c^T \right] = \sigma^2 \left( \left( \mathbf{U}^T \mathbf{U} \right) \otimes \left( \mathbf{V}^T \mathbf{V} \right) \otimes \left( \mathbf{W}^T \mathbf{W} \right) \right).
\]

- \( \Rightarrow \) If \( \mathbf{U}, \mathbf{V}, \mathbf{W} \) are orthonormal, then noise in the compressed domain is white.
\( E[z_c] = 0 \), and

\[
E \left[ z_c z_c^T \right] = \sigma^2 \left( \left( U^T U \right) \otimes \left( V^T V \right) \otimes \left( W^T W \right) \right).
\]

- For large \( I \) and \( U \) drawn from a zero-mean unit-variance uncorrelated distribution, \( U^T U \approx I \) by the law of large numbers.
- Furthermore, even if \( z \) is not Gaussian, \( z_c \) will be approximately Gaussian for large \( IJK \), by the Central Limit Theorem.
- Follows that least-squares fitting is approximately optimal in the compressed domain, even if it is not so in the uncompressed domain. Compression thus makes least-squares fitting ‘universal’!
Component energy \(\approx\) preserved after compression

- Consider randomly compressing a rank-one tensor \(\mathbf{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\), written in vectorized form as \(\mathbf{x} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}\).

- The compressed tensor is \(\tilde{\mathbf{X}}\), in vectorized form

\[
\tilde{\mathbf{x}} = \left( \mathbf{U}^T \otimes \mathbf{V}^T \otimes \mathbf{W}^T \right) (\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}) = (\mathbf{U}^T \mathbf{a}) \otimes (\mathbf{V}^T \mathbf{b}) \otimes (\mathbf{W}^T \mathbf{c}).
\]

- Can be shown that, for moderate \(L, M, N\) and beyond, Frobenious norm of compressed rank-one tensor approximately proportional to Frobenious norm of the uncompressed rank-one tensor component of original tensor.

- In other words: compression approximately preserves component energy \(\Rightarrow\) order.

- \(\Rightarrow\) Low-rank least-squares approximation of the compressed tensor \(\Leftrightarrow\) low-rank least-squares approximation of the big tensor, approximately.

- Match component permutations across replicas by sorting component energies.
Nominal setup:

- \( I = J = K = 500; \ F = 5; \ \mathbf{A}, \ \mathbf{B}, \ \mathbf{C} \sim \text{randn}(500, 5); \)
- \( L = M = N = 50 \) (each replica = 0.1% of big tensor);
- \( P = 12 \) replicas (overall cloud storage = 1.2% of big tensor).
- \( S = 3 \) (vs. \( S_{\text{min}} = 2 \)) anchor rows.
- ↑ Satisfy identifiability without much ‘slack’.
- + WGN std \( \sigma = 0.01 \).
- COMFAC [www.ece.umn.edu/~nikos](http://www.ece.umn.edu/~nikos) used for all factorizations, big and small.
**PARACOMP: MSE as a function of** $L = M = N$

- Fix $P = 12$, vary $L = M = N$.

![Graph showing the MSE as a function of $L = M = N$]

- $I=J=K=500$; $F=5$; $\sigma=0.01$; $P=12$; $S=3$

- **PARACOMP**
  - Direct, no compression
  - 32% of full data
  - 1.2% of full data

(P=12 processors, each w/ 0.1% of full data)

Nikos Sidiropoulos, Evangelos Papalexakis

Factoring Big Tensors in the Cloud

ICASSP Tutorial, 5/5/2014
PARACOMP: MSE as a function of $P$

- Fix $L = M = N = 50$, vary $P$.

\[ l = j = k = 500; \quad l = m = n = 50; \quad F = 5; \quad \text{sigma} = 0.01; \quad S = 3 \]

| $P$ | $||A - A_{hat}||_F^2$ |
|-----|-----------------------|
| 20  | $10^{-3}$              |
| 30  | $10^{-4}$              |
| 40  | $10^{-5}$              |
| 50  | $10^{-6}$              |
| 60  | $10^{-7}$              |
| 70  | $10^{-8}$              |
| 80  | $10^{-9}$              |
| 90  | $10^{-10}$             |
| 100 | $10^{-11}$             |
| 110 | $10^{-12}$             |

12% of the full data in $P=120$ processors with 0.1% each.

1.2% of the full data in $P=12$ processors with 0.1% each.
PARACOMP: MSE vs AWGN variance $\sigma^2$

- Fix $L = M = N = 50$, $P = 12$, vary $\sigma^2$.

![Graph showing the relationship between $\sigma^2$ and $||A - A_{\text{hat}}||_F^2$]
Missing elements

- Recommender systems, NELL, many other datasets: over 90% of the values are missing!
- PARACOMP to the rescue: fortuitous fringe benefit of ‘compression’ (rather: taking linear combinations)!
- Let $\mathcal{T}$ denote the set of all elements, and $\Psi$ the set of available elements.
- Consider one element of the compressed tensor, as it would have been computed had all elements been available; and as it can be computed from the available elements (notice normalization - important!):

$$Y_\nu(l, m, n) = \frac{1}{|\mathcal{T}|} \sum_{(i,j,k) \in \mathcal{T}} u_l(i)v_m(j)w_n(k)X(i, j, k)$$

$$\tilde{Y}_\nu(l, m, n) = \frac{1}{E[|\Psi|]} \sum_{(i,j,k) \in \Psi} u_l(i)v_m(j)w_n(k)X(i, j, k)$$
**Theorem:** [Marcos & Sidiropoulos, IEEE ISCCSP 2014] Assume a Bernoulli i.i.d. miss model, with parameter $\rho = \text{Prob}[(i, j, k) \in \Psi]$, and let $X(i, j, k) = \sum_{f=1}^{F} a_f(i) b_f(j) c_f(k)$, where the elements of $a_f, b_f$ and $c_f$ are all i.i.d. random variables drawn from $a_f(i) \sim \mathcal{P}_a(\mu_a, \sigma_a)$, $b_f(j) \sim \mathcal{P}_b(\mu_b, \sigma_b)$, and $c_f(k) \sim \mathcal{P}_c(\mu_c, \sigma_c)$, with $p_a := \mu_a^2 + \sigma_a^2$, $p_b := \mu_b^2 + \sigma_b^2$, $p_c := \mu_c^2 + \sigma_c^2$, and $F' := (F - 1)$. Then, for $\mu_a, \mu_b, \mu_c$ all $\neq 0$,

$$\frac{E[\|\mathcal{E}_\nu\|_F^2]}{E[\|Y_\nu\|_F^2]} \leq \frac{(1 - \rho)}{\rho |T|} \left(1 + \frac{\sigma_U^2}{\mu_u^2}\right) \left(1 + \frac{\sigma_V^2}{\mu_v^2}\right) \left(1 + \frac{\sigma_W^2}{\mu_w^2}\right) \frac{F'}{F} + \frac{p_ap_bp_c}{F\mu_a^2\mu_b^2\mu_c^2}$$

- Additional results in paper.
Figure: SNR of compressed tensor for different sizes of rank-one $X$
SNR of loadings $A$, for $\text{rank}(\mathbf{X}) = 1$, $(\mu_X, \sigma_X) = (1, 1)$

Figure: SNR of recovered loadings for different sizes of rank-one $\mathbf{X}$
Three-way (emission, excitation, sample) fluorescence spectroscopy

Figure: Measured and imputed data; recovered latent spectra

Works even with systematically missing data!