Computational Problems in Tensors

Shmuel Friedland
Univ. Illinois at Chicago

Numerical analysis & scientific computing seminar
NUY, Courant Institute, May 14, 2014

Overview

- Uniqueness of best approximation
- Primer on tensors
- Best rank one approximation of tensors
- Number of critical points
- Numerical methods for best rank one approximation
- Compressive sensing of sparse matrices and tensors

The approximation problem

$\nu : \mathbb{R}^n \to [0, \infty)$ a norm on $\mathbb{R}^n$

$C \subset \mathbb{R}^n$ a closed subset,

Problem: approximate a given vector $\mathbf{x} \in \mathbb{R}^n$ by a point $\mathbf{y} \in C$:

$\mathrm{dist}_\nu(\mathbf{x}, C) := \min\{\nu(\mathbf{x} - \mathbf{y}),\ \mathbf{y} \in C\}$

$\mathbf{y}^\star \in C$ is called a best ν-(C)approximation of $\mathbf{x}$

if $\nu(\mathbf{x} - \mathbf{y}^\star) = \mathrm{dist}_\nu(\mathbf{x}, C)$

$\|\cdot\|$ the Euclidean norm on $\mathbb{R}^n$, $\mathrm{dist}(\mathbf{x}, C) = \mathrm{dist}_{\|\cdot\|}(\mathbf{x}, C)$.

We call a best $\|\cdot\|$-approximation briefly a best (C)-approximation

Main Theoretical Result: In most of applicable cases a best approximation is unique outside a corresponding variety

Uniqueness of ν-approxim. in semi-algebraic setting

Thm F-Stawiska:

Let $C \subset \mathbb{R}^n$ semi-algebraic, ν semi-algebraic norm, ν and ν^* are differentiable. Then the set of all points $\mathbf{x} \in \mathbb{R}^n \setminus C$, denoted by $S(C)$, where ν-approximation to $\mathbf{x}$ in C is not unique is a semi-algebraic set which does not contain an open set. In particular $S(C)$ is contained in some hypersurface $H \subset \mathbb{R}^n$.

Def: $S \subset \mathbb{R}^n$ is semi-algebraic if it is a finite union of basic semi-algebraic sets :

$p_i(\mathbf{x}) = 0,\ i \in \{1, ..., \lambda\}, q_j(\mathbf{x}) > 0,\ j \in \{1, ..., \lambda'\}$

$f : \mathbb{R}^n \to \mathbb{R}$ semi-algebraic if $G(f) = \{(\mathbf{x}, f(\mathbf{x})) : x \in \mathbb{R}^n\}$ semi-algebraic

ℓ_p norms are semi-algebraic if $p \geq 1$ is rational

Numerical challenges

Most numerical methods for finding best approximation are local

Usually they will converge to a critical point or at best to a local minimum

In many cases the number of critical points is exponential in n

How far our minimal numerical solution is from a best approximation?

Give a lower bound for best approximation

Give a fast approximation for big scale problems

We will address these problems for tensors

Primer on tensors: I

d-mode tensor $\mathcal{T} = [t_{i_1,\ldots,i_d}] \in \mathbb{F}^{n_1\times\ldots\times n_d}, i_j \in [n_j] := \{1,\ldots,n_j\}, j \in [d]$

$d = 1$ vector: $\mathbf{x}$; $\quad d = 2$ matrix $A = [a_{ij}]$

rank one tensor $\mathcal{T} = [x_{i_1,1}x_{i_2,2}\cdots x_{i_d,d}] = \mathbf{x}_1 \otimes \mathbf{x}_2 \cdots \otimes \mathbf{x}_d = \otimes_{j=1}^d \mathbf{x}_j \neq 0$

rank of tensor $\mathrm{rank}\,\mathcal{T} := \min\{r : \quad \mathcal{T} = \sum_{k=1}^r \otimes_{j=1}^d \mathbf{x}_{j,k}\}$

It is an NP-hard problem to determine $\mathrm{rank}\,\mathcal{T}$ for $d \geq 3$.

border rank brank $\mathcal{T}$ the minimal r s.t. $\mathcal{T}$ is limit of tensors of rank r

brank $\mathcal{T} < \mathrm{rank}\,\mathcal{T}$ for some $d \geq 3$ mode tensors (Nongeneric case)

Unfolding tensor in mode k: $T_k(\mathcal{T}) \in \mathbb{F}^{n_k\times\frac{N}{n_k}}, N = n_1\cdots n_d$

grouping indexes $(i_1,\ldots,i_d)$ into two groups i_k and the rest

$\mathrm{rank}\ T_k(\mathcal{T}) \leq$ brank $\mathcal{T} \leq \mathrm{rank}\,\mathcal{T}$ for each $k \in [d]$

$R(r_1,\ldots,r_d) \subset \mathbb{F}^{n_1\times\ldots\times n_d}$ variety of all tensors $\mathrm{rank}\ T_k(\mathcal{T}) \leq r_k, k \in [d]$

$R(1,\ldots,1) = \otimes_{j=1}^d \mathbb{F}^{n_j}$ - Segre variety (variety of rank one tensors)

Primer on tensors: II

Contraction of tensors $\mathcal{T} = [t_{i_1,\ldots,i_d}], \mathcal{X} = [x_{i_{k_1},\ldots,i_{k_l}}], \{k_1,\ldots,k_l\} \subset [d]$

$\mathcal{T} \times \mathcal{X} := \sum_{i_{k_1}\in[n_{k_1},\ldots,i_{k_l}\in[n_{k_l}]} t_{i_1,\ldots,i_l} x_{i_{k_1},\ldots,i_{k_l}}$

Symmetric d-mode tensor $\mathcal{S} \in \mathrm{S}(\mathbb{F}^n, d)$: $n_1 = \cdots = n_d = n$,

entries $s_{i_1,\ldots,i_d}$ are symmetric in all indexes

rank one symmetric tensor $\otimes^d \mathbf{x} := \mathbf{x} \otimes \cdots \otimes \mathbf{x} \neq 0$

symmetric rank (Waring rank) srank $\mathcal{S} := \min\{r, \quad \mathcal{S} = \sum_{k=1}^r \otimes^d \mathbf{x}_k\}$

Conjecture (P. Comon 2009) srank $\mathcal{S} = \mathrm{rank}\, \mathcal{S}$ for $\mathcal{S} \in \mathrm{S}(\mathbb{C}^n, d)$

Some cases proven by Comon-Golub-Lim-Mourrain 2008

For finite fields $\exists \mathcal{S}$ s.t. srank $\mathcal{S}$ not defined F-Stawiska

Examples of approximation problems

$\mathbb{R}^N := \mathbb{R}^{n_1 \times \ldots \times n_d}$ - and C:

1. Tensors of border rank k-at most, denoted as C_k
2. $C(\mathbf{r}) := R(r_1, \ldots, r_d)$

$\nu(\cdot) = \|\cdot\|$ - Hilbert-Schmidt norm (other norms sometime)

$n_1 = \cdots = n_d = n, \quad r_1 = \cdots = r_d = r$ and $\mathcal{S} \in \mathrm{S}(\mathbb{R}^n, d)$

Problem: Can a best approximation can be chosen symmetric?

For matrices: yes

For $k = 1$: yes - Banach's theorem 1938

For some range of k: yes for some open semi-algebraic set of

$\mathcal{S} \in \mathrm{S}(\mathbb{R}^n, d)$ - F - Stawiska

Best rank one approximation of 3-tensors

$\mathbb{R}^{m\times n\times l}$ IPS: $\langle\mathcal{A},\mathcal{B}\rangle=\sum_{i=j=k}^{m,n,l} a_{i,j,k}b_{i,j,k},\ \|\mathcal{T}\|=\sqrt{\langle\mathcal{T},\mathcal{T}\rangle}$

$\langle\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{z},\mathbf{u}\otimes\mathbf{v}\otimes\mathbf{w}\rangle=(\mathbf{u}^\top\mathbf{x})(\mathbf{v}^\top\mathbf{y})(\mathbf{w}^\top\mathbf{z})$

$\mathbf{X}$ subspace of $\mathbb{R}^{m\times n\times l}$, $\mathcal{X}_1,\ldots,\mathcal{X}_d$ an orthonormal basis of $\mathbf{X}$

$\mathrm{P}_{\mathbf{X}}(\mathcal{T})=\sum_{i=1}^d\langle\mathcal{T},\mathcal{X}_i\rangle\mathcal{X}_i,\quad \|\mathrm{P}_{\mathbf{X}}(\mathcal{T})\|^2=\sum_{i=1}^d\langle\mathcal{T},\mathcal{X}_i\rangle^2$

$\|\mathcal{T}\|^2=\|\mathrm{P}_{\mathbf{X}}(\mathcal{T})\|^2+\|\mathcal{T}-\mathrm{P}_{\mathbf{X}}(\mathcal{T})\|^2$

Best rank one approximation of $\mathcal{T}$:

$\min_{\mathbf{x},\mathbf{y},\mathbf{z}}\|\mathcal{T}-\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{z}\|=\min_{\|\mathbf{x}\|=\|\mathbf{y}\|=\|\mathbf{z}\|=1,a}\|\mathcal{T}-a\,\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{z}\|$

Equivalent: $\|\mathcal{T}\|_\infty:=\max_{\|\mathbf{x}\|=\|\mathbf{y}\|=\|\mathbf{z}\|=1}\sum_{i=j=k}^{m,n,l} t_{i,j,k}x_iy_jz_k$

Hillar-Lim 2013: computation of $\|\mathcal{T}\|_\infty$ NP-hard

Lagrange multipliers: $\mathcal{T}\times\mathbf{y}\otimes\mathbf{z}:=\sum_{j=k=1} t_{i,j,k}y_jz_k=\lambda\mathbf{x}$

$\mathcal{T}\times\mathbf{x}\otimes\mathbf{z}=\lambda\mathbf{y},\ \mathcal{T}\times\mathbf{x}\otimes\mathbf{y}=\lambda\mathbf{z}$

λ singular value, $\mathbf{x},\mathbf{y},\mathbf{z}$ singular vectors

Lim 2005

Number of singular values of 3-tensor: I

$c(m,n,l)$ - $\#$ distinct singular values for a generic $\mathcal{T} \in \mathbb{C}^{m\times n\times l}$

is coefficient of $t_1^{m-1}t_2^{n-1}t_3^{l-1}$ in pol. $\frac{((t_2+t_3)^m-t_1^m)}{(t_2+t_3-t_1)}\frac{((t_1+t_3)^n-t_2^n)}{(t_1+t_3-t_2)}\frac{((t_1+t_2)^l-t_3)}{(t_1+t_2-t_3)}$

Recall $\frac{x^m-y^m}{x-y} = x^{m-1}+x^{m-2}y+\cdots+xy^{m-2}+y^{m-1}$

d_1, d_2, d_3	$c(d_1, d_2, d_3)$	
$2, 2, 2$	6	
$2, 2, n$	8	$n \geq 3$
$2, 3, 3$	15	
$2, 3, n$	18	$n \geq 4$
$2, 4, 4$	28	
$2, 4, n$	32	$n \geq 5$
$2, 5, 5$	45	
$2, 5, n$	50	$n \geq 6$
$2, m, m+1$	$2m^2$	

Table : Values of $c(d_1, d_2, d_3)$

Number of singular values of 3-tensor: II

d_1, d_2, d_3	$c(d_1, d_2, d_3)$	
$3, 3, 3$	37	
$3, 3, 4$	55	
$3, 3, n$	61	$n \geq 5$
$3, 4, 4$	104	
$3, 4, 5$	138	
$3, 4, n$	148	$n \geq 6$
$3, 5, 5$	225	
$3, 5, 6$	280	
$3, 5, n$	295	$n \geq 7$
$3, m, m+2$	$\frac{8}{3}m^3 - 2m^2 + \frac{7}{3}m$	

Table : Values of $c(d_1, d_2, d_3)$

Number of singular values of 3-tensor: III

d_1, d_2, d_3	$c(d_1, d_2, d_3)$	
$4, 4, 4$	240	
$4, 4, 5$	380	
$4, 4, 6$	460	
$4, 4, n$	480	$n \geq 7$
$4, 5, 5$	725	
$4, 5, 6$	1030	
$4, 5, 7$	1185	
$4, 4, 4$	240	
$4, 4, 5$	380	
$4, 4, 6$	460	
$4, 4, n$	480	$n \geq 7$

Table : Values of $c(d_1, d_2, d_3)$

Number of singular values of 3-tensor: IV

d_1, d_2, d_3	$c(d_1, d_2, d_3)$	
$4, 5, 5$	725	
$4, 5, 6$	1030	
$4, 5, 7$	1185	
$4, 4, 4$	240	
$4, 4, 5$	380	
$4, 4, 6$	460	
$4, 4, n$	480	$n \geq 7$
$4, 5, 5$	725	
$4, 5, 6$	1030	
$4, 5, 7$	1185	
$4, 5, 7$	1185	
$4, 5, n$	1220	$n \geq 8$

Table : Values of $c(d_1, d_2, d_3)$

Number of singular values of 3-tensor: V

d_1, d_2, d_3	$c(d_1, d_2, d_3)$	
$5, 5, 5$	1621	
$5, 5, 6$	2671	
$5, 5, 7$	3461	
$5, 5, 8$	3811	
$5, 5, n$	3881	$n \geq 9$

Table : Values of $c(d_1, d_2, d_3)$

Friedland-Ottaviani 2014

Alternating least squares

Denote $\mathrm{S}^{m-1} := \{\mathbf{x} \in \mathbb{R}^m,\ \|\mathbf{x}\| = 1\}$, $\mathrm{S}(m,n,l) : \mathrm{S}^{m-1} \times \mathrm{S}^{n-1} \times \mathrm{S}^{l-1}$

$f(\mathbf{x},\mathbf{y},\mathbf{z}) = \langle \mathcal{T}, \mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z} \rangle : \mathrm{S}(m,n,l) \to \mathbb{R}$

Best rank one approximation to $\mathcal{T}$ is equivalent to

$\max_{(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathrm{S}(m,n,l)} f(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{x}_\star,\mathbf{y}_\star,\mathbf{z}_\star)$

Alternating least square (ALS) method starts with

$(\mathbf{x}_0,\mathbf{y}_0,\mathbf{z}_0) \in \mathrm{S}(m,n,l), f(\mathbf{x}_0,\mathbf{y}_0,\mathbf{z}_0) \neq 0$:

$\mathbf{x}_i = \frac{\mathcal{T}\times(\mathbf{y}_{i-1}\otimes\mathbf{z}_{i-1})}{\|\mathcal{T}\times(\mathbf{y}_{i-1}\otimes\mathbf{z}_{i-1})\|}, \mathbf{y}_i = \frac{\mathcal{T}\times(\mathbf{x}_{i}\otimes\mathbf{z}_{i-1})}{\|\mathcal{T}\times(\mathbf{x}_{i}\otimes\mathbf{z}_{i-1})\|}, \mathbf{z}_i = \frac{\mathcal{T}\times(\mathbf{x}_{i}\otimes\mathbf{y}_{i})}{\|\mathcal{T}\times(\mathbf{x}_{i}\otimes\mathbf{y}_{i})\|}$, for $i = 1, 2, \ldots,$

$f(\mathbf{x}_{i-1},\mathbf{y}_{i-1},\mathbf{z}_{i-1}) \le f(\mathbf{x}_{i},\mathbf{y}_{i-1},\mathbf{z}_{i-1}) \le f(\mathbf{x}_{i},\mathbf{y}_{i},\mathbf{z}_{i-1}) \le f(\mathbf{x}_{i},\mathbf{y}_{i},\mathbf{z}_{i})$

$(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i)$ converges(?) to 1-semi-maximal critical point $(\mathbf{x}_*,\mathbf{y}_*,\mathbf{z}_*)$

Definition: $(\mathbf{x}_*,\mathbf{y}_*,\mathbf{z}_*)$ - k-semi-maximal critical point if

it is maximal with respect to each set of k vector variables,

while other vector variables are kept fixed

Alternating SVD method: F-Merhmann-Pajarola-Suter

Fix one vector variable in $f(\mathbf{x},\mathbf{y},\mathbf{z}) = \langle \mathcal{T}, \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{z}\rangle$, e.g. $\mathbf{z} \in \mathrm{S}^{l-1}$

$\max\{f(\mathbf{x},\mathbf{y},\mathbf{z}),\ \mathbf{x}\in \mathrm{S}^{m-1}, \mathbf{y}\in \mathrm{S}^{n-1}\}$ achieved at $\mathbf{x}=\mathbf{u}(\mathbf{z}), \mathbf{y}=\mathbf{v}(\mathbf{z})$

singular vectors of bilinear form $f(\mathbf{x},\mathbf{y},\mathbf{z})$ of max. singular value

$(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i) \mapsto (\mathbf{x}_i',\mathbf{y}_i',\mathbf{z}_i) = (\mathbf{u}(\mathbf{z}_i),\mathbf{v}(\mathbf{z}_i),\mathbf{z}_i) \mapsto$

$(\mathbf{x}_{i+1},\mathbf{y}_i',\mathbf{z}_i') = (\mathbf{u}'(\mathbf{y}_i')),\mathbf{y}_i',\mathbf{w}(\mathbf{y}_i')) \mapsto$

$(\mathbf{x}_{i+1},\mathbf{y}_{i+1},\mathbf{z}_{i+1}) = (\mathbf{x}_{i+1},\mathbf{v}'(\mathbf{x}_{i+1}),\mathbf{w}'(\mathbf{x}_{i+1})) \mapsto \ldots$

$(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i)$ converges(?) to 2-semi-maximal critical point $(\mathbf{x}_*,\mathbf{y}_*,\mathbf{z}_*)$

ASVD is more expensive than ALS

Since for finding $\|A\|_2$ one uses (truncated) SVD

ASVD is a reasonable alternative to ALS (see simulations)

Modified ALS and ASVD

Theoretical problem: Let $(\mathbf{x}_*, \mathbf{y}_*, \mathbf{z}_*)$ accumulation point of $\{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\}$

Is it 1-semi-maximal for ALS; 2-semi-maximal for ASVD? (Don't know)

Modified ALS and ASVD: MALS and MASVD

First time 3 maximizations, in other iterations 2 maximizations:

MALS (e.g.) $\max(\max_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1}), \max_{\mathbf{y}} f(\mathbf{x}_{i-1}, \mathbf{y}, \mathbf{z}_{i-1}))$

MSVD (e.g.) $\max(\max_{\mathbf{x},\mathbf{y}} f(\mathbf{x}, \mathbf{y}, \mathbf{z}_{i-1}), \max_{\mathbf{x},\mathbf{z}} f(\mathbf{x}, \mathbf{y}_{i-1}, \mathbf{z}))$

Theorem Any accumulation point of $\{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)\}$ of MALS and MASVD is 1 or 2 semi-maximal respectively

Simulation Setup: I

Implemenation of C++ library supporting the rank one tensor decomposition using vmmlib, LAPACK and BLAS to test the performance of the different best rank one approximation algorithms. The performance was measured via the actual CPU-time (seconds) needed to compute the approximate best rank one decomposition, by the number of optimization calls needed, and whether a stationary point was found. (whether a stationary point or a global maxima is found.)
All performance tests have been carried out on a 2.8 GHz Quad-Core Intel Xeon Macintosh computer with 16GB RAM.

The performance results are discussed for synthetic and real data sets of third-order tensors. In particular, we worked with three different data sets: (1) a real computer tomography (CT) data set (the so-called MELANIX data set of OsiriX), (2) a symmetric random data set, where all indices are symmetric, and (3) a random data set. The CT data set has a 16bit, the random data set an 8bit value range.

Simulation Setup: II

All our third-order tensor data sets are initially of size $512 \times 512 \times 512$, which we gradually reduced by a factor of 2, with the smallest data sets being of size $4 \times 4 \times 4$. The synthetic random data sets were generated for every resolution and in every run; the real data set was averaged (subsampled) for every coarser resolution.

Our simulation results are averaged over different decomposition runs of the various algorithms. In each decomposition run, we changed the initial guess, Additionally, we generated for each decomposition run new random data sets. The presented timings are averages over 10 different runs of the algorithms.

All the best rank one approximation algorithms are alternating algorithms, and based on the same convergence criterion The partial SVD is implemented by applying a symmetric eigenvalue decomposition (LAPACK DSYEVX) to the product AA^T (BLAS DGEMM) as suggested by the ARPACK package.

Average CPU times for best rank one approximations per algorithm and per data set taken over 10 different initial random guesses medium sizes

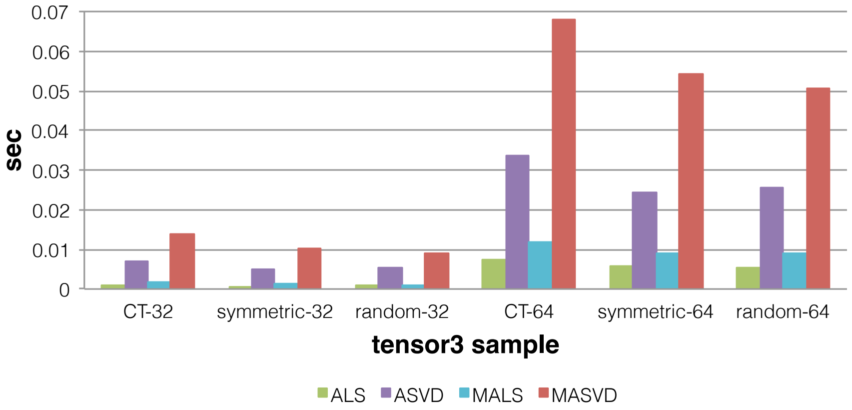


Figure : CPU time (s) for medium sized 3-mode tensor samples

Average CPU times for best rank one approximations per algorithm and per data set taken over 10 different initial random guesses larger sizes

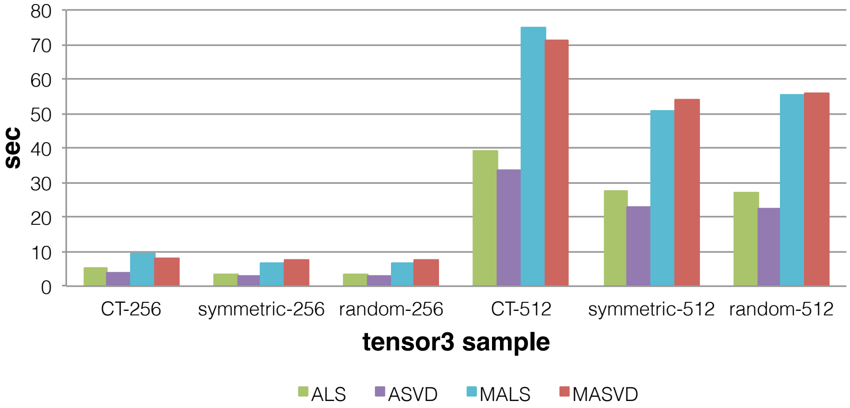


Figure : CPU time (s) for larger sized 3-mode tensor samples

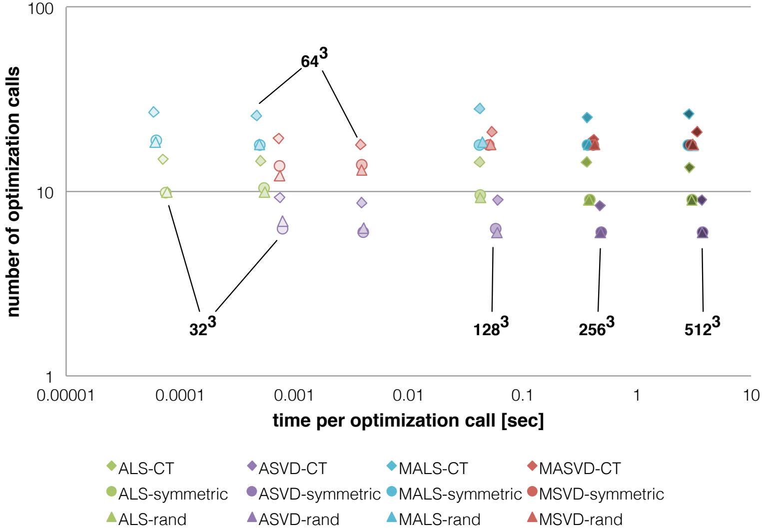
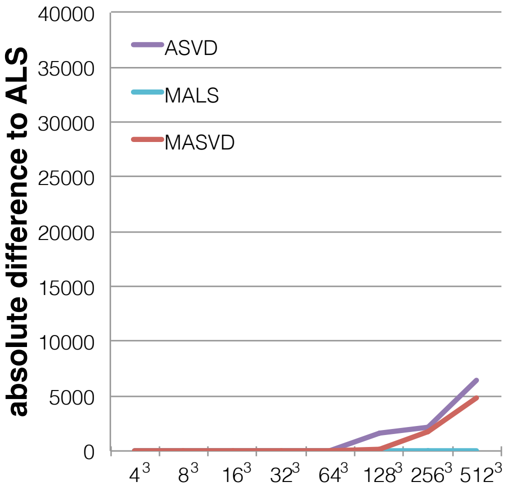
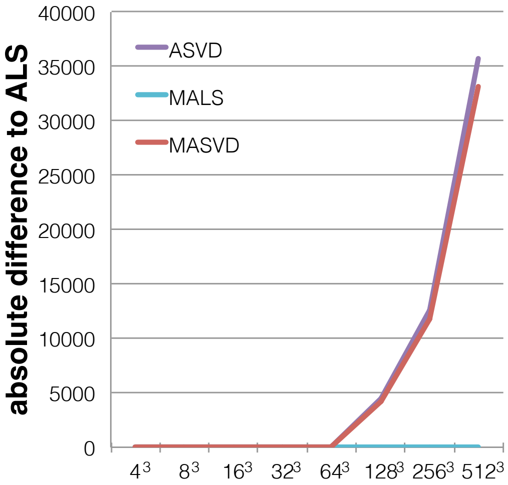


Figure : Average time per optimization call put in relationship to the average number of optimization calls needed per algorithm and per data set taken over 10 different initial random guesses.

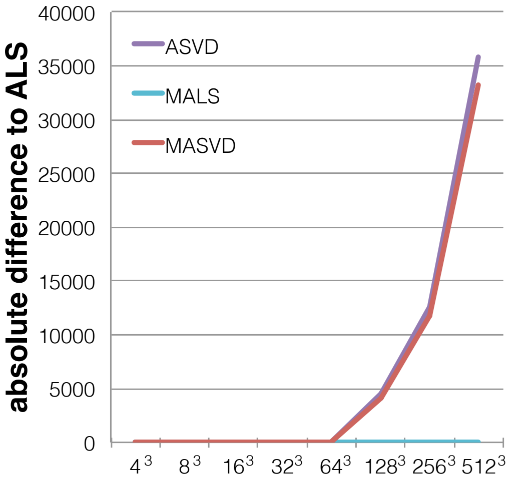
Differences of the achieved Frobenius norms by ALS, ASVD, MALS, and MASVD: CT-data



Differences of the achieved Frobenius norms by ALS, ASVD, MALS, and MASVD: Symmetric



Differences of the achieved Frobenius norms by ALS, ASVD, MALS, and MASVD: Random



Remarks to differences of ALS, ASVD, MALS, and MASVD

The algorithms reach the same stationary point for the smaller and medium data sets. However, for the larger data sets ($\geq 128^3$) the stationary points differ slightly. We suspect that either the same stationary point was not achieved, or the precision requirement of the convergence criterion was too high.

Best rank one approximation for symmetric tensors using ALS, MALS, ASVD and MASVD show that the best rank one approximation is also symmetric, i.e., is of the form $a\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$, where $\mathbf{u} \approx \mathbf{v} \approx \mathbf{w} \in S^{m-1}$

(Banach's theorem.)

The results of ASVD and MASVD give a better symmetric rank one approximation, i.e., $\mathbf{u} - \mathbf{v}, \mathbf{u} - \mathbf{w}$ in ASVD and MASVD are smaller than in ALS and MALS.

Compressive sensing for sparse matrices and tensors

joint works with Qun Li, Dan Schonfeld and Edgar A. Bernal

Conventional Compressive sensing (CS) theory relies on data representation in the form of vectors.

Many data types in various applications such as color imaging, video sequences, and multi-sensor networks, are intrinsically represented by higher-order tensors.

We propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order spare tensors.

GTCS offers an efficient means for representation of multidimensional data by providing simultaneous acquisition and compression from all tensor modes. Its draw back is an inferior compression ratio.

Compressive sensing of vectors: Noiseless

$\Sigma_{s,N}$ is the set of all $\mathbf{x} \in \mathbb{R}^N$ with at most s nonzero coordinates

Sparse version of CS: Given $\mathbf{x} \in \Sigma_{s,N}$ compress it to a short vector

$\mathbf{y} = (y_1, \ldots, y_M)^\top, M << N$ and send it to receiver

receiver gets $\mathbf{y}$, possible with noise, decodes to $\mathbf{x}$

Compressible version: coordinates of $\mathbf{x}$ have fast power law decay

Solution: $\mathbf{y} = A\mathbf{x}$, $A \in \mathbb{R}^{M\times N}$ a specially chosen matrix, e.g. s-n. p.

Sparse noiseless recovery: $\mathbf{x} = \arg\min\{\|\mathbf{z}\|_1, A\mathbf{z} = \mathbf{y}\}$

A has s-null property if for each $A\mathbf{w} = \mathbf{0}, \mathbf{w} \neq \mathbf{0}$, $\|\mathbf{w}\|_1 > 2\|\mathbf{w}_S\|_1$

$S \subset [N] := \{1, \ldots, N\}, |S| = s$,

$\mathbf{w}_S$ has zero coordinates outside S and coincides with $\mathbf{w}$ on S

Recovery condition $M \geq cs\log(N/s)$, noiseless reconstruction $O(N^3)$

Compressive sensing of matrices I - noiseless

$X = [x_{ij}] = [\mathbf{x}_1 \ldots \mathbf{x}_{N_1}]^\top \in \mathbb{R}^{N_1 \times N_2}$ is s-sparse.

$Y = U_1 X U_2^\top = [\mathbf{y}_1, \ldots, \mathbf{y}_{M_2}] \in \mathbb{R}^{M_1 \times M_2}$, $U_1 \in \mathbb{R}^{M_1 \times N_1}$, $U_2 = \mathbb{R}^{M_2 \times N_2}$

$M_i \geq cs \log(N_i/s), M = M_1 M_2 \geq (cs)^2 \log(N_1/s) \log(N_2/s)$

U_i has s-null property for $i = 1, 2$

Thm M: X is determined from noiseless Y.

Algo 1: $Z = [\mathbf{z}_1 \; \ldots \mathbf{z}_{M_2}] = X U_2^\top \in \mathbb{R}^{N_1 \times M_2}$

each $\mathbf{z}_i$ a linear combination of columns of X hence s-sparse

$Y = U_1 Z = [U_1 \mathbf{z}_1, \ldots, U_1 \mathbf{z}_{M_2}]$ so $\mathbf{y}_i = U_1 \mathbf{z}_i$ for $i \in [M_2]$

Recover each $\mathbf{z}_i$ to obtain Z

Cost: $M_2 O(N_1^3) = O((\log N_2) N_1^3)$

$Z^\top = U_2 X^\top = [U_2 \mathbf{x}_1 \ldots U_2 \mathbf{x}_{N_1}]$

Recover each $\mathbf{x}_i$ from $i - th$ column of $Z^\top$

Cost: $N_1 O(N_2^3) = O(N_1 N_2^3)$, Total cost: $O(N_1 N_2^3 + (\log N_2) N_1^3)$

Compressive sensing of matrices II - noiseless

Algo 2: Decompose $Y = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^\top$,

$\mathbf{u}_1, \ldots, \mathbf{u}_r$, $\mathbf{v}_1^\top, \ldots, \mathbf{v}_r^\top$ span column and row spaces of Y respectively

for example a rank decomposition of Y: $r = \mathrm{rank}\ Y$

Claim $\mathbf{u}_i = U_1 \mathbf{a}_i, \mathbf{v}_j = U_2 \mathbf{b}_j$, $\mathbf{a}_i, \mathbf{b}_j$ are s-sparse, $i, j \in [r]$.

Find $\mathbf{a}_i, \mathbf{b}_j$. Then $X = \sum_{i=1}^{r} \mathbf{a}_i \mathbf{b}_i^\top$

Explanation: Each vector in column and row spaces of X is s-sparse:

$Range(Y) = U_1 Range(X)$, $Range(Y^\top) = U_2 Range(X^\top)$

Cost: Rank decomposition: $O(rM_1 M_2)$ using Gauss elimination or SVD

Note: $\mathrm{rank}\ Y \le \mathrm{rank}\ X \le s$

Reconstructions of $\mathbf{a}_i, \mathbf{b}_j$: $O(r(N_1^3 + N_2^3))$

Reconstruction of X: $O(rs^2)$

Maximal cost: $O(s \max(N_1, N_2)^3)$

Why algorithm 2 works

Claim 1: Every vector in Range X and Range $X^\top$ is s-sparse.

Claim 2: Let $X_1 = \sum_{i=1}^r \mathbf{a}_i \mathbf{b}_i^\top$. Then $X = X_1$.

Prf: Assume $0 \neq X - X_1 = \sum_{j=1}^k \mathbf{c}_j \mathbf{d}_j^\top$, $\mathbf{c}_1, \ldots, \mathbf{c}_k$ & $\mathbf{d}_1, \ldots, \mathbf{d}_k$ lin. ind.

as Range $X_1 \subset$Range X, Range $X_1^\top \subset$Range $X^\top$

$\mathbf{c}_1, \ldots, \mathbf{c}_k \in$Range X, $\mathbf{d}_1, \ldots, \mathbf{d}_k \in$Range $X^\top$

Claim: $U_1\mathbf{c}_1, \ldots U_1\mathbf{c}_k$ lin.ind..

Suppose $\mathbf{0} = \sum_{j=1}^k t_i U_1 \mathbf{c}_j = U_1 \sum_{j=1}^k t_j \mathbf{c}_j$.

As $\mathbf{c} := \sum_{j=1}^k t_j \mathbf{c}_j \in$ Range X, $\mathbf{c}$ is s-sparse.

As U_1 has null s-property $\mathbf{c} = \mathbf{0} \Rightarrow t_1 = \ldots = t_k = 0$.

$0 = Y - Y = U_1(X - X_1)U_2^\top = \sum_{j=1}^k (U_1\mathbf{c}_j)(\mathbf{d}_j^\top U_2^\top) \Rightarrow$

$U_2\mathbf{d}_1 = \ldots = U_2\mathbf{d}_k = \mathbf{0} \Rightarrow \mathbf{d}_1 = \ldots \mathbf{d}_k = \mathbf{0}$ as each $\mathbf{d}_i$ is s-sparse

So $X - X_1 = 0$ contradiction

Sum.-Noiseless CS of matrices & vectors as matrices

1. Both algorithms are highly parallelizable
2. Algorithm 2 is faster by factor $s\min(N_1, N_2)$ at least
3. In many instances but not all algorithm 1 performs better.
4. Caveat: the compression is : $M_1 M_2 \geq C^2(\log N_1)(\log N_2)$.
5. Converting vector of length N to a matrix

Assuming $N_1 = N^\alpha, N_2 = N^{1-\alpha}$

the cost of vector compressing is $O(N^3)$

the cost of algorithm 1 is $O((\log N)N^{\frac{9}{5}}), \alpha = \frac{3}{5}$

the cost of algorithm 2 is $O(sN^{\frac{3}{2}}), \alpha = \frac{1}{2}, s = O(\log N)(?)$

Remark 1: The cost of computing Y from s-sparse X: $2sM_1M_2$

(Decompose X as sum of s standard rank one matrices)

Numerical simulations

We experimentally demonstrate the performance of GTCS methods on sparse and compressible images and video sequences.

Our benchmark algorithm is Duarte-Baraniuk 2010

named Kronecker compressive sensing (KCS)

Another method is multi-way compressed sensing

of Sidoropoulus-Kyrillidis (MWCS) 2012

Our experiments use the ℓ_1-minimization solvers of Candes-Romberg.

We set the same threshold to determine the termination of ℓ_1-minimization in all subsequent experiments.

All simulations are executed on a desktop with

2.4 GHz Intel Core i5 CPU and 8GB RAM.

We set $M_i = K$

UIC logo

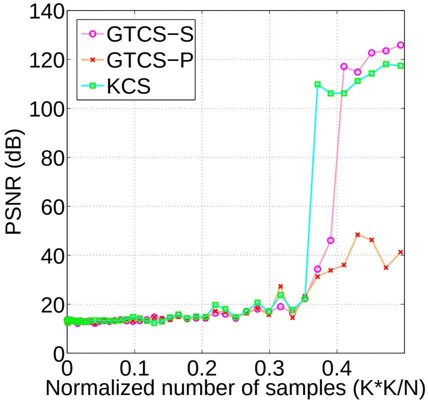
(a) The original sparse image

(b) GTCS-S recovered image

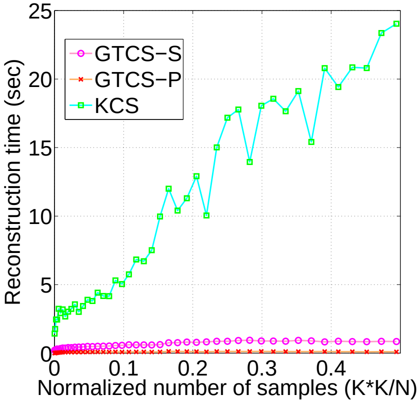
(c) GTCS-P recov-

(d) KCS recovered

PSNR and reconstruction times for UIC logo



(a) PSNR comparison



(b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on sparse image.

Explanation of UIC logo representation I

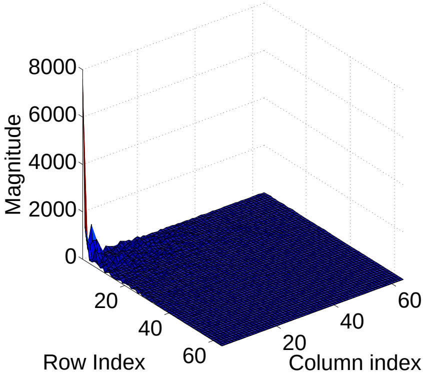
The original UIC black and white image is of size 64×64 ($N = 4096$ pixels). Its columns are 14-sparse and rows are 18-sparse. The image itself is 178-sparse. For each mode, the randomly constructed Gaussian matrix U is of size $K \times 64$. So KCS measurement matrix $U \otimes U$ is of size $K^2 \times 4096$. The total number of samples is K^2. The normalized number of samples is $\frac{K^2}{N}$. In the matrix case, GTCS-P coincides with MWCS and we simply conduct SVD on the compressed image in the decomposition stage of GTCS-P. We comprehensively examine the performance of all the above methods by varying K from 1 to 45.

Explanation of UIC logo representation II

Figure 5(a) and 5(b) compare the peak signal to noise ratio (PSNR) and the recovery time respectively. Both KCS and GTCS methods achieve PSNR over 30dB when $K = 39$. As K increases, GTCS-S tends to outperform KCS in terms of both accuracy and efficiency. Although PSNR of GTCS-P is the lowest among the three methods, it is most time efficient. Moreover, with parallelization of GTCS-P, the recovery procedure can be further accelerated considerably. The reconstructed images when $K = 38$, that is, using 0.35 normalized number of samples, are shown in Figure 4(b)4(c)4(d). Though GTCS-P usually recovers much noisier image, it is good at recovering the non-zero structure of the original image.

Cameraman simulations I

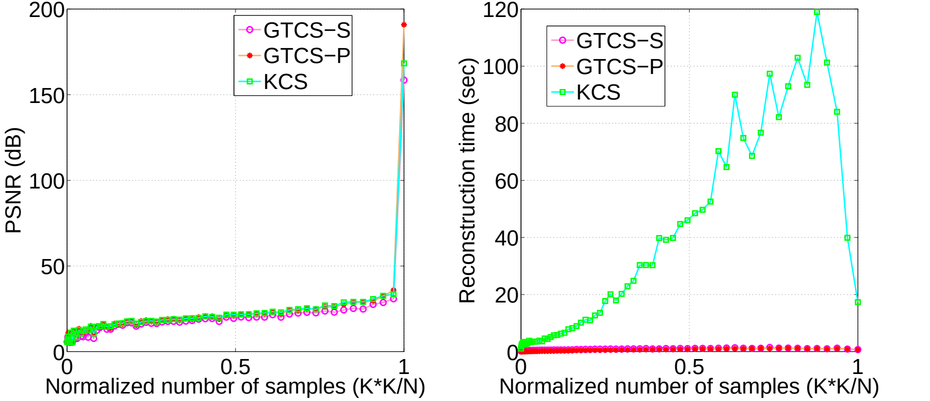
(a) Cameraman in space domain



(b) Cameraman in DCT domain

Figure : The original cameraman image (resized to 64 $\times$ 64 pixels) in space domain and DCT domain.

Cameraman simulations II

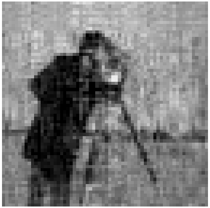


(a) PSNR comparison (b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on compressible image.

Cameraman simulations III

(a) GTCS-S, K = 46, PSNR = 20.21 dB



(b) GTCS-P/MWCS, K = 46, PSNR = 21.84 dB

(c) KCS, K = 46, PSNR = 21.79 dB

(d) GTCS-S,K = 63, PSNR = 30.88 dB

(e) GTCS-P/MWCS, K = 63, PSNR = 35.95 dB

(f) KCS, K = 63, PSNR = 33.46 dB

Cameraman explanations

As shown in Figure 6(a), the cameraman image is resized to 64×64 ($N = 4096$ pixels). The image itself is non-sparse. However, in some transformed domain, such as discrete cosine transformation (DCT) domain in this case, the magnitudes of the coefficients decay by power law in both directions (see Figure 6(b)), thus are compressible. We let the number of measurements evenly split among the two modes. Again, in matrix data case, MWCS concurs with GTCS-P. We exhaustively vary K from 1 to 64.

Figure 7(a) and 7(b) compare the PSNR and the recovery time respectively. Unlike the sparse image case, GTCS-P shows outstanding performance in comparison with all other methods, in terms of both accuracy and speed, followed by KCS and then GTCS-S. The reconstructed images when $K = 46$, using 0.51 normalized number of samples and when $K = 63$, using 0.96 normalized number of samples are shown in Figure 8.

Compressive sensing of tensors

$\mathbf{M} = (M_1, \ldots, M_d), \mathbf{N} = (N_1, \ldots, N_d) \in \mathbb{N}^d$, $J = \{j_1, \ldots, j_k\} \subset [d]$

Tensors: $\otimes_{i=1}^d \mathbb{R}^{N_i} = \mathbb{R}^{N_1 \times \ldots \times N_d} = \mathbb{R}^{\mathbf{N}}$

Contraction of $\mathcal{A} = [a_{i_{j_1}, \ldots, i_{j_k}}] \in \otimes_{j_p \in J} \mathbb{R}^{N_{j_p}}$ with $\mathcal{T} = [t_{i_1, \ldots, i_d}] \in \mathbb{R}^{\mathbf{N}}$:

$\mathcal{A} \times \mathcal{T} = \sum_{i_{j_p} \in [N_{j_p}], j_p \in J} a_{i_{j_1}, \ldots, i_{j_k}} t_{i_1, \ldots, i_d} \in \otimes_{l \in [d] \setminus J} \mathbb{R}^{N_l}$

$\mathcal{X} = [x_{i_1, \ldots, i_d}] \in \mathbb{R}^{\mathbf{N}}, \mathcal{U} = U_1 \otimes U_2 \otimes \ldots \otimes U_d \in \mathbb{R}^{(M_1, N_1, M_2, N_2, \ldots, M_d, N_d)}$

$U_p = [u^{(p)}_{i_p j_p}] \in \mathbb{R}^{M_p \times N_p}$, $p \in [d], \mathcal{U}$ Kronecker product of $U_1, \ldots, U_d$.

$\mathcal{Y} = [y_{i_1, \ldots, i_d}] = \mathcal{X} \times \mathcal{U} := \mathcal{X} \times_1 U_1 \times_2 U_2 \times \ldots \times_d U_d \in \mathbb{R}^{\mathbf{M}}$

$y_{i_1, \ldots, i_p} = \sum_{j_q \in [N_q], q \in [d]} x_{j_1, \ldots, j_d} \prod_{q \in [d]} u_{i_q, j_q}$

Thm $\mathcal{X}$ is s-sparse, each U_i has s-null property

then $\mathcal{X}$ uniquely recovered from $\mathcal{Y}$.

Algo 1: GTCS-S

Algo 2: GTCS-P

Algo 1- GTCS-S

Unfold $\mathcal{Y}$ in mode 1: $Y_{(1)} = U_1 \mathcal{W}_1 \in \mathbb{R}^{M_1 \times (M_2 \cdot \ldots \cdot M_d)}$,

$\mathcal{W}_1 := X_{(1)}[\otimes_{k=d}^{2} U_k]^\top \in \mathbb{R}^{N_1 \times (M_2 \cdot \ldots \cdot M_d)}$

As for matrices recover the $\tilde{M}_2 := M_2 \cdots M_d$ columns of $\mathcal{W}_1$ using U_1

Complexity: $O(\tilde{M}_2 N_1^3)$.

Now we need to recover

$\mathcal{Y}_1 := \mathcal{X} \times_1 I_1 \times_2 U_2 \times \ldots \times_d U_d \in \mathbb{R} N_1 \times M_2 \ldots \times M_d$

Equivalently, recover N_1, $d-1$ mode tensors in $\mathbb{R}^{N_2 \times \ldots \times N_d}$ from $\mathbb{R}^{M_2 \times \ldots \times M_d}$ using $d-1$ matrices $U_2, \ldots, U_d$.

Complexity $\sum_{i=1}^{d} \tilde{N}_{i-1} \tilde{M}_{i+1} N_i^3$

$\tilde{N}_0 = \tilde{M}_{d+1} = 1, \quad \tilde{N}_i = N_1 \ldots N_i, \quad \tilde{M}_i = M_i \ldots M_d$

$d = 3$: $M_2 M_3 N_1^3 + N_1 M_3 N_2^3 + N_1 N_2 N_3^3$

Algo 2- GTCS-P

Unfold $\mathcal{X}$ in mode k: $X_{(k)} \in \mathbb{R}^{N_k \times \frac{N}{N_k}}$, $N = \prod_{i=1}^{d} N_i$.

As $\mathcal{X}$ is s-sparse $\mathrm{rank}\ _k\mathcal{X} := \mathrm{rank}\ X_{(k)} \le s$.

$Y_{(k)} = U_k X_{(k)} [\otimes_{i \ne k} U_i]^\top \Rightarrow \text{Range } Y_{(k)} \subset U_k \text{Range } X_{(k)}, \mathrm{rank}\ Y_{(k)} \le s$.

$X_{(1)} = \sum_{j=1}^{R_1} \mathbf{u}_i \mathbf{v}_i^\top$, $\mathbf{u}_1, \ldots, \mathbf{u}_{R_1}$ spans range of $X_{(1)}$ so $R_1 \le s$

Each $\mathbf{v}_i$ corresponds to $\mathcal{U}_i \in \mathbb{R}^{N_2 \times \ldots N_d}$ which is s-sparse

So (1) $\mathcal{X} = \sum_{j=1}^{R} \mathbf{u}_{1,j} \otimes \ldots \otimes \mathbf{u}_{d,j}$, $R \le s^{d-1}$

$\mathbf{u}_{k,1}, \ldots, \mathbf{u}_{k,R} \in \mathbb{R}^{N_k}$ span Range $X_{(k)}$ and each is s-sparse

Compute decomposition $\mathcal{Y} = \sum_{j=1}^{R} \mathbf{w}_{1,j} \otimes \ldots \otimes \mathbf{w}_{d,j}$, $R \le s^{d-1}$,

$\mathbf{w}_{k,1}, \ldots, \mathbf{w}_{k,R} \in \mathbb{R}^{M_k}$ span Range $Y_{(k)}$, Compl: $O(s^{d-1} \prod_{i=1}^{d} M_i)$

Find $\mathbf{u}_{k,j}$ from $\mathbf{w}_{k,j} = U_k \mathbf{u}_{k,j}$ and reconstruct $\mathcal{X}$ from (1)

Complexity $O(ds^{d-1} \max(N_1, \ldots, N_d)^3)$, $s = O(\log(\max(N_1, \ldots, N_d)))$

Summary of complexity converting linear data

$N_i = N^{\alpha_i}, M_i = O(\log N)$, $\alpha_i > 0$, $\sum_{i=1}^{d} \alpha_i = 1$, $s = \log N$

$d = 3$

GTCS-S: $O((\log N)^2 N^{\frac{27}{19}})$

GTCS-P: $O((\log N)^2 N)$

GTCS-P: $O((\log N)^{d-1} N^{\frac{3}{d}})$ for any d.

Warning: the roundoff error in computing parfac decomposition of $\mathcal{Y}$ and then of $\mathcal{X}$ increases significantly with d.

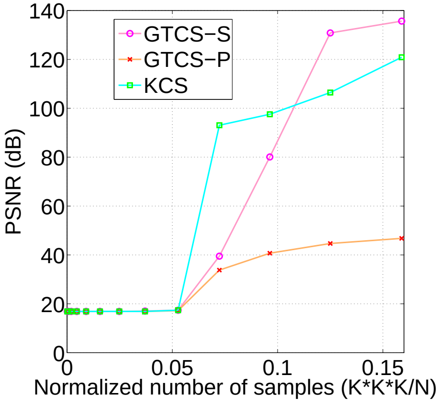
Sparse video representation

We compare the performance of GTCS and KCS on video data. Each frame of the video sequence is preprocessed to have size 24×24 and we choose the first 24 frames. The video data together is represented by a $24 \times 24 \times 24$ tensor and has $N = 13824$ voxels in total. To obtain a sparse tensor, we manually keep only $6 \times 6 \times 6$ nonzero entries in the center of the video tensor data and the rest are set to zero.

The video tensor is 216-sparse and its mode-i fibers are all 6-sparse $i = 1, 2, 3$. The randomly constructed Gaussian measurement matrix for each mode is now of size $K \times 24$ and the total number of samples is K^3. The normalized number of samples is $\frac{K^3}{N}$.

We vary K from 1 to 13.

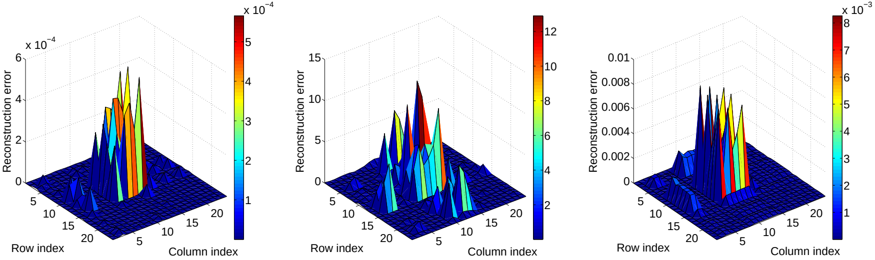
PSNR and reconstruction time of sparse video



(a) PSNR comparison (b) Recovery time comparison

Figure : PSNR and reconstruction time comparison on sparse video.

Reconstruction errors of sparse video



(a) Reconstruction error of GTCS-S

(b) Reconstruction error of GTCS-P

(c) Reconstruction error of KCS

Figure : Visualization of the reconstruction error in the recovered video frame 9 by GTCS-S (PSNR = 130.83 dB), GTCS-P (PSNR = 44.69 dB) and KCS (PSNR = 106.43 dB) when $K = 12$, using 0.125 normalized number of samples.

Conclusion

Real-world signals as color imaging, video sequences and multi-sensor networks, are generated by the interaction of multiple factors or multimedia and can be represented by higher-order tensors. We propose Generalized Tensor Compressive Sensing (GTCS)-a unified framework for compressive sensing of sparse higher-order tensors. We give two reconstruction procedures, a serial method (GTCS-S) and a parallelizable method (GTCS-P). We compare the performance of GTCS with KCS and MWCS experimentally on various types of data including sparse image, compressible image, sparse video and compressible video. Experimental results show that GTCS outperforms KCS and MWCS in terms of both accuracy and efficiency. Compared to KCS, our recovery problems are in terms of each tensor mode, which is much smaller comparing with the vectorization of all tensor modes. Unlike MWCS, GTCS manages to get rid of tensor rank estimation, which considerably reduces the computational complexity and at the same time improves the reconstruction accuracy.

References 1

- S. Banach, Über homogene polynome in (L^2), *Studia Math.* 7 (1938), 36–44.
- B. Chen, S. He, Z. Li, and S, Zhang, Maximum block improvement and polynomial optimization, *SIAM J. Optimization*, 22 (2012), 87–107
- P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, Symmetric tensors and symmetric tensor rank, *SIAM Journal on Matrix Analysis and Applications*, 30 (2008), 1254-1279.
- S. Friedland, Best rank one approximation of real symmetric tensors can be chosen symmetric, *Front. Math. China*, 8 (1) (2013), 19–40.
- S. Friedland, V. Mehrmann, R. Pajarola, S.K. Suter, On best rank one approximation of tensors, *Numerical Linear Algebra with Applications*, 20 (2013), 942–955.

References 2

S. Friedland and G. Ottaviani, The number of singular vector tuples and uniqueness of best rank one approximation of tensors, *Found. Comput. Math.* 2014, arXiv:1210.8316 .

C.J. Hillar and L.-H. Lim, Most tensor problems are NP-hard, *Journal of the ACM*, 60 (2013), no. 6, Art. 45, 39 pp.

L.-H. Lim. Singular values and eigenvalues of tensors: a variational approach. *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing* (CAMSAP '05), 1 (2005), 129-132.

References 3

- C. Caiafa and A. Cichocki, Multidimensional compressed sensing and their applications, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(6), 355-380, (2013).
- E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete information, *Information Theory, IEEE Transactions on* 52 (2006), 489–509.
- E. Candes and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies, *Information Theory, IEEE Transactions on* 52 (2006), 5406–5425.
- D. Donoho, Compressed sensing, *Information Theory, IEEE Transactions on* 52 (2006), 1289–1306.
- M. Duarte and R. Baraniuk, Kronecker compressive sensing, *Image Processing*, IEEE Transactions on, 2 (2012), 494–504.

References 4

- Q. Li, D. Schonfeld and S. Friedland, Generalized tensor compressive sensing, *Multimedia and Expo (ICME)*, 2013 IEEE International Conference on, 15-19 July 2013, ISSN: 1945-7871, 6 pages.
- S. Friedland, Q. Li and D. Schonfeld, Compressive Sensing of Sparse Tensors, arXiv:1305.5777.
- S. Friedland, Q. Li, D. Schonfeld and Edgar A. Bernal, Two algorithms for compressed sensing of sparse tensors, arXiv:1404.1506
- N. Sidiropoulus and A. Kyrillidis, Multi-way compressive sensing for sparse low-rank tensors, *Signal Processing Letters*, IEEE, 19 (2012), 757–760.