7th International Conference on
Combinatorics, Cryptography, Computer Science and Computing
November. 16-17, 2022

A new, better version of high accuracy low rank tensor completion

Rasool Ebrahimi[1]
University of Mazandaran, Babolsar, Iran

Alireza Shojaeifard
Imam Hossein Comprehensive University, Tehran, Iran

Abstract

Tensor completion is one of the efficient methods for restoration of data. Tensor completion problem by minimization of the rank of tensor, leads to an appropriate solution, however, it gives a non-convex objective function that generates an NP-hard problem. In order to be solved this problem, at first, by matricization it is transformed to some matrix completion problems. Next, instead of using the matrix rank, the trace norm is applied. In this paper, to increase the speed of high accuracy low rank tensor completion algorithm, a fast tri-factorization method is presented.

Keywords: Image processing, Low rank tensor completion, Fast tri-factorization method

AMS Mathematical Subject Classification [2010]: 68U10, 15A83, 15A69

1 Introduction

A tensor completion problem is filling the missing or unobserved entries in the given partially observed tensors. Tensors are used for describing complex datasets, due to their multidimensional features. Because of this, tensor completion algorithms have been widely developed for and applied in fields and domains, including data mining, computer vision, signal processing, and neuroscience. In this section, we briefly state some preliminaries for tensor calculus and tensor completion.

We follow [4] to define all notations. We use lowercase letters to denote scalars, e.g., x, boldface capital letters to denote matrices, e.g., $\mathbf{X} \in \Re^{I_1 \times I_2}$, and boldface Euler script letters to denote higher-order tensors, e.g., $\mathbb{X} \in \Re^{I_1 \times I_2 \times \cdots \times I_N}$. The nuclear norm (NN) of a matrix is denoted as $\|\mathbb{X}\|_* = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ is the ith largest singular value of X. The "unfold" operation along the $k - mode$ on a tensor X is defined as $unfold_k(\mathbb{X}) = X_{(k)} \in \Re^{I_k \times (I_1 \ldots I_{k-1} I_{k+1} \ldots I_n)}$.. The opposite operation "fold" is defined as $fold_k(A_{(k)}) = \mathbb{A}$.. Tucker rank or multilinear rank which is introduced by Tucker is defined as $rank_{tc}(\mathbb{X}) = (rank(X_{(1)}), rank(X_{(2)}), \ldots, rank(X_{(n)}))$.

[1]speaker

2 Main results

We start by low rank tensor completion problem

$$\min_{\mathbb{Z}} \quad rank_{tc}(\mathbb{Z}) \quad s.t. \quad P_{\Omega}(\mathbb{Z}-\mathbb{W})=0 \tag{1}$$

where $rank_{tc}(\mathbb{Z})$ represents the tucker rank of tensor $\mathbb{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, Ω is the indices set of known entries of $\mathbb{W}$, and $P_{\Omega}(\mathbb{A}_{ij}) = \begin{cases} \mathbb{A}_{ij} & (i,j) \in \Omega \\ 0 & otherwise. \end{cases}$

In this work, we mainly consider the nuclear (trace) norm minimization model corresponding to low-rank tensor recovery and completion problem. This model can be reformulated as follows [3]:

$$\min_{\mathbb{Z}} \quad \|\mathbb{Z}\|_{*} \quad s.t. \quad P_{\Omega}(\mathbb{Z}-\mathbb{W})=0 \tag{2}$$

where $\|.\|_{*}$ denotes the nuclear norm of a tensor, that is the sum of its singular values of each unfolding matrices. To solving this problem Liu et al. proposed the high accuracy low rank tensor completion(HaLRTC) Algorithm [1, 4].

Though the nuclear norm minimization problem in (2) is convex, while the size of matrices $Z_{(i)}$ are large, the multiple SVDs computation of these matrices are very expensive. In order to reduce SVD computation, the fast tri-factorization method is proposed to approximate nuclear norm minimization problems[2, 6].

Lemma 2.1. *Let L, R and M be given matrices of compatible dimensions, and suppose both L and R have orthogonal columns and orthogonal rows, respectively, i.e.,* $L^T L = I$ *and* $RR^T = I$*, then we have* $\|M\|_{*} = \|LMR\|_{*}$.

According to the Lemma 2.1, it is clear that $\|Z\|_{*} = \|LMR\|_{*} = \|M\|_{*}$. Therefore the nuclear norm minimization problem for the matrix Z is reduced a much smaller matrix M. Substituting $Z = LMR$ into (2) and simplifying, we arrive at the following model for this nuclear norm minimization problem [5]:

$$\min_{Z,L,M,R} \quad \|M\|_{*} \quad s.t. \quad P_{\Omega}(Z-W)=0, \quad Z=LMR, \quad L^T L = I, RR^T = I.$$

And we can get the following partial augmented Lagrange function:

$$\mathbf{L}(Z,L,M,R,Y,\mu) = \|M\|_{*} + \langle Y, Z-LMR \rangle + \frac{\mu}{2}\|Z-LMR\|_F^2. \tag{3}$$

The optimal solution of the problem (3) with respect to the variables M, Z, L, R and Y is respectively derive by Algorithm 1:

3 Numerical result

Now we apply the proposed fast tri-factorization simple low rank tensor completion method for a picture case: The first column images of Figure (1) shows the original image. The second column images shows the corrupted image with 90% lost data. The third column images shows the recovered image by HaLRTC algorithm and the fourth to last columns images shows the recovered image by FTF-HaLRTC algorithm with $r = 6, 9, 12, 15, 18$ and 21:

The numerical results of CPU time, PSNR and RMSE of Algorithms HaLRTC and FTF-HaLRTC with $r = 6, 9, 12, 15, 18$ and 21 are presented in Tables 1, 2 and 3, respectively.

Input : An incomplete data tensor $\mathbb{X}^0 \in \Re^{I_1 \times I_2 \times \cdots \times I_n}$, the observation index set Ω, the size r of matrix $M \in \Re^{r \times r}$ and thresholding value μ

Output: Completed data tensor $\mathbb{X}^{k+1}$

while *not convergence* **do**

$M_{(i)}^{k+1} = D_{\mu^{-1}}\Big((L_{(i)}^{K})^T(unfold_i(\mathbb{X}^k + \frac{1}{\rho}\mathbb{Y}_i^k) + V_{(i)}^k/\mu)(R_{(i)}^k)^T\Big)$

$Z_{(i)}^{k+1} = L_{(i)}^k M_{(i)}^{k+1} R_{(i)}^k - V_{(i)}^k/\mu + P_\Omega(T_{(i)} - L_{(i)}^k M_{(i)}^k R_{(i)}^k + V_{(i)}^k/\mu)$

$L_{(i)}^{k+1} = QR\Big((Z_{(i)}^K + V_{(i)}^k/\mu)(R_{(i)}^k)^T\Big)$

$(R_{(i)}^{k+1})^T = QR\Big((Z_{(i)}^K + V_{(i)}^k/\mu)^T L_{(i)}^{K+1}\Big)$

$V_{(i)}^{k+1} = P_\Omega\Big(V_{(i)}^k + \mu(Z_{(i)}^{k+1} - L_{(i)}^{k+1} M_{(i)}^{k+1} R_{(i)}^{k+1})\Big)$

$\mathbb{X}^{k+1} = \frac{1}{n}\sum_{i=1}^{n} fold_i(Z_{(i)}^{k+1} - \frac{1}{\rho}\mathbb{Y}_i^k)$

$\mathbb{X}_\Omega^{k+1} = \mathbb{X}_\Omega^0$

$\mathbb{Y}_i^{k+1} = \mathbb{Y}_i^k - \rho(\mathbb{W}_i^{k+1} - \mathbb{X}^{k+1})$

end

Algorithm 1: FTF-HaLRTC Algorithm

Table 1: CPU time of Algorithms HaLRTC and FTF-HaLRTC with r=6,9,12,15,18 and 21 on the images of Figure 1

Image	size	HaLRTC	FTF-HaLRTC					
			r=6	r=9	r=12	r=15	r=18	r=21
Clock tower	$194 \times 259 \times 3$	205	**32**	59	145	272	381	300
Peppers(1)	$194 \times 259 \times 3$	173	50	**37**	117	206	682	609
Dog	$213 \times 238 \times 3$	389	81	**34**	158	222	351	447
Facade	$256 \times 256 \times 3$	343	**21**	53	66	261	219	377
House	$256 \times 256 \times 3$	484	**30**	130	156	227	278	419
Parrot	$256 \times 384 \times 3$	887	**24**	134	126	260	508	1439
F16	$512 \times 512 \times 3$	4388	**143**	480	299	1789	1210	2671
Peppers(2)	$512 \times 512 \times 3$	3485	**356**	634	587	1013	1026	3752

4 conclusion

In this paper, we presented an algorithm called FTF-HaLRTC to reduce the CPU time of high accuracy low rank tensor completion Algorithm. By comparison of the low rank tensor completion algorithm and our proposed algorithm, it is shown that for image tensor in $\Re^{m \times n \times 3}$ the CPU time of the tensor completion method can be reduced and accuracy is increased by FTF-HaLRTC method.

References

[1] A. Dhondea, R. A. Cohen, I. V. Bajić. CALTeC: Content-adaptive linear tensor completion for collaborative intelligence. *IEEE International Conference on Image Processing (ICIP)*, IEEE, (2021) 2179-2183.

[2] J. Fan and T. W. Chow. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, 71 (2017), 290-305.

[3] M. Fazel, H. Hindi and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. *In Proceedings of the 2001 American Control Conference.*IEEE,(2001), 4734-4739.

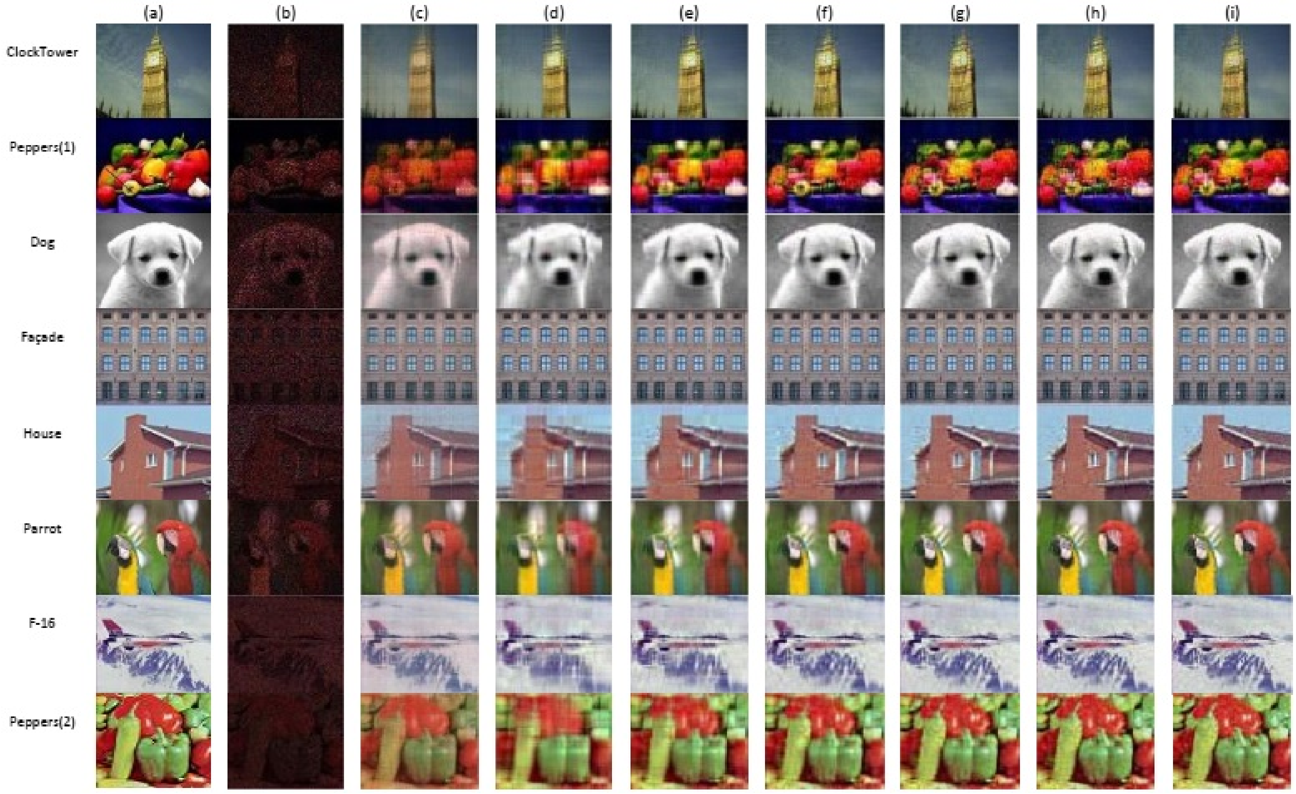

Figure 1: A dataset of images. The columns show (a) original image, (b) corrupted images, (c) reconstructed by HaLRTC algorithm and reconstructed by FTF-HaLRTC algorithm by (d) r=6, (e) r=9, (f) r=12, (g) r=15, (h) r=18 and (i) r=21.

[4] J. Liu, P. Musialski, P. Wonka and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence* 35 (2012), 208-220.

[5] Y. Liu, L. C. Jiao and F. Shang. A fast tri-factorization method for low-rank matrix recovery and completion. *Pattern Recognition* 46 (2013), 163-173.

[6] P. Zhou, C Lu, Z Lin and C. Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing* 27 (2017), 1152-1163.

e-mail: ras.ebrahimi@gmail.com
e-mail: ashojaeifard@ihu.ac.ir

Table 2: PSNR of Algorithms HaLRTC and FTF-HaLRTC with $r = 6, 9, \cdots, 21$ on the images of Figure 1

Image	HaLRTC	FTF-HaLRTC					
		$r = 6$	$r = 9$	$r = 12$	$r = 15$	$r = 18$	$r = 21$
Clock tower	24.5799	25.4203	**25.5697**	25.2483	24.7271	23.9213	24.0201
Peppers(1)	32.7101	**35.2172**	30.3968	28.0007	27.4035	29.6224	31.5904
Dog	20.9779	23.2542	25.5108	26.4452	**26.3876**	26.2137	25.8782
Facade	25.1516	26.1403	26.7955	27.3085	27.1646	**27.3136**	27.2733
House	20.4801	21.4744	22.5607	23.2675	23.8116	**24.5072**	24.1978
Parrot	21.7867	21.1534	22.5848	23.6520	**24.0974**	23.9581	23.6118
F16	21.6511	20.6374	21.5722	22.3328	22.8319	23.2582	**23.6137**
Peppers(2)	17.4122	17.8384	18.9717	19.9032	20.4540	21.0689	**21.2885**

Table 3: RMSE of Algorithms HaLRTC and FTF-HaLRTC with $r = 6, 9, \cdots, 21$ on the images of Figure 1

Image	HaLRTC	FTF-HaLRTC					
		$r = 6$	$r = 9$	$r = 12$	$r = 15$	$r = 18$	$r = 21$
Clock tower	15.0504	13.6623	**13.4293**	13.9357	14.7974	16.2359	16.0522
Peppers(1)	19.5202	**17.1957**	18.4744	19.1874	19.3747	18.6984	18.1397
Dog	22.7848	17.5320	13.5207	**12.1417**	12.2224	12.4697	12.9607
Facade	14.0916	12.5755	11.6617	10.9930	11.1766	**10.9865**	11.0377
House	24.1289	21.5190	18.9892	17.5051	16.4422	**15.1769**	15.7272
Parrot	20.7589	22.3290	18.9366	16.7472	**15.9099**	16.1672	16.8247
F16	21.0855	23.6958	21.2779	19.4940	18.4054	17.5238	**16.8210**
Peppers(2)	34.3503	32.7054	28.7048	25.7856	24.2015	22.5474	**21.9845**