A Rank-One Tensor Updating Algorithm for Tensor Completion

Yuning Yang, Yunlong Feng, and Johan A. K. Suykens, *Fellow, IEEE*

***Abstract*—In this letter, we propose a rank-one tensor updating algorithm for solving tensor completion problems. Unlike the existing methods which penalize the tensor by using the sum of nuclear norms of unfolding matrices, our optimization model directly employs the tensor nuclear norm which is studied recently. Under the framework of the conditional gradient method, we show that at each iteration, solving the proposed model amounts to computing the tensor spectral norm and the related rank-one tensor. Because the problem of finding the related rank-one tensor is NP-hard, we propose a subroutine to solve it approximately, which is of low computational complexity. Experimental results on real datasets show that our algorithm is efficient and effective.**

***Index Terms*—Frank–Wolfe (conditional gradient) method, rank-one tensor, tensor completion, tensor nuclear/spectral norm.**

I. Introduction

TENSORS are higher order generalizations of matrices. During recent years, tensor completion has drawn much attention [1]–[5]. The task of tensor completion is to infer a tensor (possibly low rank) with missing entries from partial observations. Typically, tensor completion has been modeled as matrix nuclear norm minimization or regularization problems [1]–[3], [6]. More specifically, tensor completion has been formulated as an optimization problem involving sum of the nuclear norms of unfolding matrices of the unknown tensor. However, due to the sum of matrix nuclear norms, algorithms have to solve several subproblems involving full singular value decompositions at each iteration. This can be problematic when the size of the tensor goes large. Another limitation is that as shown in [7], using sum of nuclear norms may fail to exploit the tensor structure and lead to a suboptimal procedure.

Manuscript received December 23, 2014; revised February 10, 2015; accepted March 26, 2015. Date of publication April 06, 2015; date of current version April 10, 2015. This work was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC AdG A-DATADRIVE-B (290923). This work reflects only the authors' views, the Union is not liable for any use that may be made of the contained information; Research Council KULGOA/10/09 MaNet; CoEPFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO Projects G.0377.12 (Structured systems) and G.088114N (Tensor based data similarity); PhD/Postdoc grants; IWT Project SBO POM (100031); PhD/Postdoc grants; iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office (IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Qian Du.

The authors are with ESAT-STADIUS, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium (e-mail: yuning.yang@esat.kuleuven.be; yunlong.feng@esat.kuleuven.be; johan.suykens@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2420592

The contribution of this letter is to propose an efficient algorithm for tensor completion. Our optimization model adopts the tensor nuclear norm introduced in [8] and our work is inspired by [7]. Under the framework of the conditional gradient method [9], we show that at each iteration, solving the proposed model amounts to computing the tensor spectral norm and the related rank-one tensor. Because finding the related rank-one tensor is NP-hard, we present an efficient subroutine to solve it approximately. We also establish a lower bound result for the subroutine. Experimental results on real datasets show the efficiency and effectiveness of our algorithm. After finishing our work, we notice that [10] proposed to solve tensor learning problems based on generalized conditional gradient methods, where computing the tensor spectral norm is also involved as a subproblem. However, our subroutine for computing the tensor spectral norm has some advantages over that used in [10] (See Section III-A).

In the next section, we recall the necessary preliminaries. We present our algorithm in Section III. Experimental results are reported in Section IV. Some conclusions are drawn in Section V.

II. Preliminaries on Tensors

Vectors are written as boldface letters $(\mathbf{a}, \mathbf{b}, \ldots)$, matrices correspond to italic capitals $(A, B, \ldots)$, and tensors are written as calligraphic capitals $(\mathcal{A}, \mathcal{B}, \cdots)$. We use $\mathbb{T} := \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$ to denote an N-th order tensor space.

For two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{T}$, their inner product is given by $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{n_1} \cdots \sum_{i_N=1}^{n_N} \mathcal{A}_{i_1 \cdots i_N} \mathcal{B}_{i_1 \cdots i_N}$. The Frobenius norm of $\mathcal{A}$ is defined by $\|\mathcal{A}\|_F = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}$. The outer product of vectors $\mathbf{x}_i \in \mathbb{R}^{n_i}$, $i = 1, \ldots, N$ is denoted as $\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_N$ and is a rank-one tensor in $\mathbb{T}$ defined by $(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_N)_{i_1 \cdots i_N} = \prod_{j=1}^{N} \mathbf{x}_{j,i_j}$. The mode-$d$ tensor-matrix multiplication of a tensor $\mathcal{X} \in \mathbb{T}$ with a matrix $U \in \mathbb{R}^{J_d \times n_d}$ is a tensor of size $n_1 \times \cdots \times n_{d-1} \times J_d \times n_{d+1} \times \cdots \times n_N$.

Tensor-matrix mode–d unfolding and balanced-unfolding. The mode-d unfolding of tensor $\mathcal{A}$ is denoted as $A_{(d)}$ by arranging the mode-d fibers to be the columns of the resulting matrix. For an N-th order tensor, the balanced-unfolding is to choose $\lfloor \frac{N}{2} \rfloor$ modes and merge them into the first mode (row) of the unfolding matrix, and the remaining $N - \lfloor \frac{N}{2} \rfloor$ modes are merged into the second mode (column). We explain it by an example. For a 4-th order tensor $\mathcal{A}$, we can unfold it to $A_{[1,2;3,4]}, A_{[1,2;4,3]}, \ldots, A_{[4,3;2,1]}$, where a semicolon indicates a new mode. Here $A_{[1,2;3,4]}$ is defined by $(A_{[1,2;3,4]})_{(i_1-1)n_2+i_2,(i_3-1)n_4+i_4} = \mathcal{A}_{i_1 i_2 i_3 i_4}$.

Tensor rank. There are mainly two types of tensor rank, namely the CP-rank and the Tucker-rank [11]. The CP-rank is defined as the minimum integer R such that for a tensor $\mathcal{X}$, it can be factorized as a sum of R rank-one tensors. The Tucker-rank

1070-9908 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

of an N-th order tensor $\mathcal{X}$ is an N tuple, whose i-th entry is the rank of the unfolding matrix $X_{(i)}$.

Tensor nuclear norm and spectral norm. The tensor nuclear norm is defined as in [8]

$$\|\mathcal{X}\|_* := \inf\left\{\sum_{p=1}^{\infty}\sigma_p | \mathcal{X} = \sum_{p=1}^{\infty}\sigma_p\mathcal{X}_p,\ \sigma_p \geq 0\right\},$$

where $\mathcal{X}_p$ are rank-one tensors with $\|\mathcal{X}_p\|_F = 1$. The concept of the tensor nuclear norm dates back to 1950s [12]. The tensor spectral norm is defined as

$$\|\mathcal{X}\|_2 := \max_{\|\mathbf{y}_i\|=1, 1\leq i\leq N} \langle \mathcal{X}, \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \cdots \otimes \mathbf{y}_N\rangle.$$

This is also known as the singular value problem [13]. The tensor nuclear/spectral norm are dual to each other [8], i.e.,

$$\|\mathcal{X}\|_2 = \max_{\|\mathcal{Y}\|_*\leq 1} \langle \mathcal{X}, \mathcal{Y}\rangle. \tag{1}$$

III. A Rank-One Tensor Updating Algorithm

In the literature, tensor completion involving sum of matrix nuclear norms can be modeled as

$$\min_{\mathcal{X}\in\mathbb{T}} \sum_{j=1}^{N}\|X_{(j)}\|_* \quad \text{s.t.} \mathcal{X}_\Omega = \mathcal{B}_\Omega$$

or its corresponding regularized version [1]–[3], [6]. Here $\mathcal{B}_\Omega$ denotes the observed entries of the tensors. The idea behind the above approaches is to restrict the Tucker-rank of the tensors. If we instead control the CP-rank of the tensors, the tensor completion can be modeled as

$$\min_{\mathcal{X}\in\mathbb{T}} \|\mathcal{X}\|_* \quad \text{s.t.} \mathcal{X}_\Omega = \mathcal{B}_\Omega. \tag{2}$$

(2) has been proposed and analyzed theoretically in [7]. Concerning the algorithmic aspect, however, it is unlikely to design an algorithm for solving (2). Because of the structure of the tensor nuclear norm, computing its proximity operator is difficult or even impossible.

Alternatively, we notice that the tensor spectral norm is relatively easier to deal with. In fact, many methods are available for approximately computing the tensor spectral norm, as will be mentioned later. Realizing that the spectral norm is the dual norm of the tensor nuclear norm motivates us to apply the Frank-Wolfe (conditional gradient (CG)) method [9]. The CG method iteratively solves the constrained problem $\min_{\mathbf{x}\in C} f(\mathbf{x})$ via the following two steps

1) Compute $\mathbf{s}^{(k)} := \arg\min_{\mathbf{x}\in C}\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{x}\rangle$.
2) Choose a stepsize $\alpha^{(k)} \in (0, 1]$ via a certain strategy and update $\mathbf{x}^{(k+1)} := (1-\alpha^k)\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}$.

The CG method regains attention in recent years [14]. Under the CG framework, we first transform (2) into

$$\min_{\mathcal{X}\in\mathbb{T}} f(\mathcal{X}) := 2^{-1}\|\mathcal{X}_\Omega - \mathcal{B}_\Omega\|_F^2 \quad \text{s.t.} \|\mathcal{X}\|_* \leq \beta, \tag{3}$$

where $\beta > 0$ is a constraint parameter, and denote $C := \{\mathcal{X} \in \mathbb{T} | \|\mathcal{X}\|_* \leq \beta\}$, which is compact and convex. In the CG algorithm, one notices that a key step is to solve the subproblem in step 1), which in our setting amounts to solving

$$\begin{aligned}\mathcal{S}^{(k)} :&= \arg\min_{\|\mathcal{X}\|_*\leq\beta}\langle \nabla f(\mathcal{X}^{(k)}), \mathcal{X}\rangle \\ &= -\beta \arg\max_{\|\mathcal{X}\|_*\leq 1}\langle \nabla f(\mathcal{X}^{(k)}), \mathcal{X}\rangle \\ &= -\beta \arg\max_{\|\mathbf{x}_i\|=1, 1\leq i\leq N.}\langle \nabla f(\mathcal{X}^{(k)}), \mathbf{x}_1\otimes\cdots\otimes\mathbf{x}_N\rangle,\end{aligned}$$

where the last equality follows from (1).

When $N = 2$, the above subproblem can be efficiently solved by the power method or the Arnoldi iteration [15]. However, when $N \geq 3$, finding the spectral norm of a given tensor $\mathcal{A}$ is to compute

$$\|\mathcal{A}\|_2 = \max_{\|\mathbf{x}_i\|=1, i=1,\ldots,N} \langle \mathcal{A}, \mathbf{x}_1\otimes\cdots\otimes\mathbf{x}_N\rangle, \tag{4}$$

which is NP-hard in general [16]. Thus in the next subsection we will present an approximation method to solve (4).

We remark that [10] proposed to solve tensor learning problems using generalized CG method, where solving (4) is also involved as a subproblem. One of the main differences between our method and that of [10] is that, our subroutine for solving (4) has some differences from that used in [10] and has some advantages, as shown in the following subsection.

A. Solving the Subproblem (4)

We review some approximation methods for solving (4). The power method has been generalized to higher order tensors [17], and some variations were proposed in [18], [19]. A Newton method was proposed in [20]. However, these methods are not very efficient in our setting. Some polynomial-time approaches with provable lower bounds have been proposed in [21]–[25]. Here we provide an alternative way to solve (4). Empirically it is more efficient and effective. We first present our method for tensors of order-4:

Subroutine 1 $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \mathbf{ApproxSpectral4}(\mathcal{A})$

1) Unfold $\mathcal{A}$ to a matrix $A_{[1,2;3,4]} \in \mathbb{R}^{n_1n_2\times n_3n_4}$.
2) Compute the singular vector pair $(\mathbf{x}_{[1,2]}, \mathbf{x}_{[3,4]}) \in \mathbb{R}^{n_1n_2}\times\mathbb{R}^{n_3n_4}$ corresponding to the leading singular value of $A_{[1,2;3,4]}$.
3) Fold the vector $\mathbf{x}_{[1,2]}$ to matrix $X_{[1;2]} \in \mathbb{R}^{n_1\times n_2}$, and compute the singular vector pair $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{n_1}\times\mathbb{R}^{n_2}$ corresponding to the leading singular value of $X_{[1;2]}$.
4) Compute the singular vector pair $(\mathbf{x}_3, \mathbf{x}_4) \in \mathbb{R}^{n_3}\times\mathbb{R}^{n_4}$ corresponding to the leading singular value of the matrix $\mathcal{A}\times_1\mathbf{x}_1^\top\times_2\mathbf{x}_2^\top \in \mathbb{R}^{n_3\times n_4}$.
5) Return $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, x_4)$.

Proposition 1: Let $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ be a solution generated by Subroutine 1. Then there holds $\langle \mathcal{A}, \mathbf{x}_1\otimes\mathbf{x}_2\otimes\mathbf{x}_3\otimes\mathbf{x}_4\rangle \geq \|\mathcal{A}\|_2/\sqrt{\min\{n_1, n_2\}\min\{n_3, n_4\}}$.

Proof: Let us denote $v = \langle \mathcal{A}, \mathbf{x}_1\otimes\mathbf{x}_2\otimes\mathbf{x}_3\otimes\mathbf{x}_4\rangle$, $M_1 = \mathcal{A}\times_1\mathbf{x}_1^\top\times_2\mathbf{x}_2^\top$ and $M_2 = \langle \mathcal{A}, \cdot\otimes\cdot\otimes X_{[3;4]}\rangle$. Here $X_{[3;4]} \in \mathbb{R}^{n_3\times n_4}$ is the matrix folded by $\mathbf{x}_{[3,4]}$ that is generated in Step 2), and $M_2 \in \mathbb{R}^{n_1\times n_2}$ is a matrix whose (i, j)-th entry is given by the inner product of $\mathcal{A}(i, j, :, :)$ and $X_{[3;4]}$. We also denote the leading singular value of $A_{[1,2;3,4]}$ as $\sigma_{\max}$. It is clear to see that $\sigma_{\max} \geq \|\mathcal{A}\|_2$. From the subroutine we get

$$\begin{aligned}v = \mathbf{x}_3^\top M_1\mathbf{x}_4 &= \|M_1\|_2 \geq \|M_1\|_F/\sqrt{\min\{n_3, n_4\}} \\ &= \max_{\|X\|_F=1}\frac{\langle M_1, X\rangle}{\sqrt{\min\{n_3, n_4\}}} \geq \frac{\langle M_1, X_{[3;4]}\rangle}{\sqrt{\min\{n_3, n_4\}}} \\ &= \frac{\langle \mathcal{A}\times_1\mathbf{x}_1^\top\times_2\mathbf{x}_2^\top, X_{[3;4]}\rangle}{\sqrt{\min\{n_3, n_4\}}} = \frac{\mathbf{x}_1^\top M_2\mathbf{x}_2}{\sqrt{\min\{n_3, n_4\}}},\end{aligned} \tag{5}$$

where the first inequality follows from the relationship between the matrix spectral norm and Frobenius norm. It follows from $A_{[1,2;3,4]}\mathbf{x}_{[3,4]} = \sigma_{\max}\mathbf{x}_{[1,2]}$ and the definition of M_2 that

$$\mathbf{x}_1^\top M_2\mathbf{x}_2 = \sigma_{\max}\mathbf{x}_1^\top X_{[1;2]}\mathbf{x}_2 \geq \frac{\sigma_{\max}\|X_{[1;2]}\|_F}{\sqrt{\min\{n_1,n_2\}}} \geq \frac{\sigma_{\max}}{\sqrt{\min\{n_1,n_2\}}}, \quad (6)$$

where the first inequality again follows from the relationship between the matrix spectral norm and the Frobenius norm. (5) together with (6) implies the desired results. ■

The dominant cost in Subroutine 1 is Step 2). We find that it is not necessary to solve the singular value problem in Step 2) precisely. Empirically, we observe that running only a few power iterations is acceptable. After Subroutine 1, we can perform another step to get a better solution: update $(\mathbf{x}_1,\mathbf{x}_2)$ by denoting them as the singular vector pair corresponding to the leading singular value of the matrix $\mathcal{A}\times_3\mathbf{x}_3^\top\times_4\mathbf{x}_4^\top$.

Concerning the computational complexity, if $n_1 = \cdots = n_4 = n$, then the complexity of the above method is $O(n^4) + O(n^2)$, while that of [22] is $O(n^4)+O(n^3)+O(n^2)$. Comparing on 100 randomly generated tensors of size $50\times 50\times 50$, the solution quality of our method has a 13% improvement, and the CPU time is 16% less than that of [22]. For tensors of size $100\times 100\times 100\times 100$, the solution quality has a 16% improvement, while the CPU time is 122% less than [22]. Note that the method of [22] was used in [10] as also an subroutine for computing the tensor spectral norm. Therefore our subroutine has some advantages over that of [10].

The subroutine for 2^d-th ($d > 2$) order tensors is defined recursively in Subroutine 2. For a tensor of order N which is not the power of two, it can be treated as of order $2^{\lceil\log_2(N)\rceil}$, e.g., a tensor $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ can be seen as in the space $\mathbb{R}^{n_1\times n_2\times 1\times n_3}$, and a 5-th order tensor can be treated as a 8-th order tensor analogously. The complexity is $\sum_{k=2}^{d} O(n^{2^k})$, against $\sum_{k=2}^{2^d} O(n^k)$ of [22].

Subroutine 2 $(\mathbf{x}_1,\ldots,\mathbf{x}_{2^d}) = \mathbf{ApproxSpectral}(\mathcal{A})$.

1) If the order of $\mathcal{A}$ is 4, return $(\mathbf{x}_1,\ldots,\mathbf{x}_4) = \mathbf{ApproxSpectral}4(\mathcal{A})$.
2) Compute the singular vector pair $(\mathbf{x}_{[1,\ldots,2^{d-1}]},\mathbf{x}_{[2^{d-1}+1,2^d]})$ corresponding to the leading singular value of matrix $A_{[1,\ldots,2^{d-1};2^{d-1}+1,2^d]}$.
3) Fold the vector $\mathbf{x}_{[1,\ldots,2^{d-1}]}$ to a 2^{d-1}-th order tensor $\mathcal{X}$; compute $(\mathbf{x}_1,\ldots,\mathbf{x}_{2^{d-1}}) = \mathbf{ApproxSpectral}(\mathcal{X})$.
4) Denote the 2^{d-1}-th order tensor $\mathcal{Y} := \mathcal{A}\times_1\mathbf{x}_1^\top\times_2\cdots\times_{2^{d-1}}\mathbf{x}_{2^{d-1}}^\top$; compute $(\mathbf{x}_{2^{d-1}+1},\ldots,\mathbf{x}_{2^d}) = \mathbf{ApproxSpectral}(\mathcal{Y})$.
5) Return $(\mathbf{x}_1,\ldots,\mathbf{x}_{2^d})$.

Now we go back to the CG method. At the $(k+1)$-th iteration, we let $\mathcal{S}^{(k)}$ be the rank-one tensor generated by the outer product of $\mathbf{ApproxSpectral}(\nabla f(\mathcal{X}^{(k)}))$ multiplied by a factor $-\beta$, and denote $\mathcal{D}^{(k)} = \mathcal{S}^{(k)} - \mathcal{X}^{(k)}$. Since the solution of Subroutine 2 is suboptimal, $\mathcal{D}^{(k)}$ may not be a descent direction. Nevertheless, empirically we find that for most cases $\mathcal{D}^{(k)}$ is a descent direction. The stepsize $\alpha^{(k)}$ is chosen as

$$\alpha^{(k)} := \min\{1, |\langle\nabla f(\mathcal{X}^{(k)}),\mathcal{D}^{(k)}\rangle|/\|\mathcal{D}_\Omega^{(k)}\|_F^2\}. \quad (7)$$

Using the results of [14], [26], we deduce the following results.

Theorem 1: Let $\{\mathcal{X}^{(k)}\}$ be a sequence generated by the CG method for solving (2). If at each iteration $\mathcal{D}^{(k)}$ is a descent direction, and the stepsize is chosen by (7), then we have $f(\mathcal{X}^{(k+1)}) \leq f(\mathcal{X}^{(k)}) - 2^{-1}|\langle\nabla f(\mathcal{X}^{(k)}),\mathcal{D}^{(k)}\rangle|\alpha^{(k)}$.

Our rank-one updating algorithm is summarized as follows.

Algorithm 1 A **R**ank- **O**ne **T**ensor **U**pdating Algorithm (RoTu)

Input: Parameter $\beta > 0$, a rank-one tensor $\mathcal{X}^{(0)}$

While certain stopping criterion is not satisfied **do**
1) Compute a rank-one tensor $\mathcal{S}^{(k)} = -\beta\mathbf{ApproxSpectral}(\nabla f(\mathcal{X}^{(k)}))$.
2) Denote $\mathcal{D}^{(k)} := \mathcal{S}^{(k)} - \mathcal{X}^{(k)}$, choose $\alpha^{(k)}$ via (7) and update $\mathcal{X}^{(k+1)} := \mathcal{X}^{(k)} + \alpha^{(k)}\mathcal{D}^{(k)}$.

End While

Remark 1: Whether or not $\mathcal{D}^{(k)}$ is a descent direction, we can always update the algorithm as follows: Denote $\mathcal{W}^{(k)} := -\beta^{-1}S^{(k)}$; update $\mathcal{X}^{(k+1)} := \alpha_1^{(k)}\mathcal{X}^{(k)} + \alpha_2^{(k)}\mathcal{W}^{(k)}$, where $(\alpha_1^{(k)},\alpha_2^{(k)}) = \arg\min_{\alpha_1,\alpha_2\in\mathbb{R}}\|\alpha_1\mathcal{X}_\Omega^{(k)} + \alpha_2\mathcal{W}_\Omega^{(k)} - \mathcal{B}_\Omega\|_F$. Then one can verify the convergence of the algorithm, which is somewhat similar to that of [27] and can be found in the supplemental material[1]. For a different choice of the stepsize and convergence analysis, one can refer to [10] for details.

IV. Numerical Experiments

We evaluate our method on several real datasets: Spectral Images (size $205\times 246\times 96$), Brain MRI (size $181\times 217\times 181$), INCISIX (size $256\times 256\times 100$) and KNIX[2]. From KNIX, we choose 3 datasets, the sizes of which are $256\times 256\times 3\times 24$, $256\times 256\times 3\times 22$ and $256\times 256\times 3\times 22$, respectively. For each dataset, some entries are set to be missed randomly.

We compare RoTu1 (Alg. 1) and RoTu2 (the method in Remark 1) with GCG [10] HaLRTC [1], TMAC [28] and a matrix completion approach (MC). The matrix completion approach is implemented as follows: we use the CG method to solve the following matrix completion problems

$$\min_{\mathcal{X}\in\mathbb{T}} 2^{-1}\|\mathcal{X}_\Omega - \mathcal{B}_\Omega\|_F^2 \quad \text{s.t.}\|X_{(i)}\|_* \leq \beta_i \quad (8)$$

from $i = 1$ to N, and choose the best solution among the N solutions as the final recovery result.

As suggested in [14], the value $|\langle\nabla f(\mathcal{X}^{(k)}),\mathcal{D}^{(k)}\rangle|$ can be used as a stopping criterion. In our setting, we use the value in (7) as a stopping criterion, as it is proportional to

[1]ftp://ftp.esat.kuleuven.be/pub/SISTA//yyang/rank-one_tensor_update.pdf

[2]The first two datasets can be downloaded from https://code.google.com/p/tensor-related-code/source/browse/trunk/Model/Tensor+Completion/LRTC_Package_Ji/?r=6, and the last two can be downloaded from http://www.osirix-viewer.com/datasets/.

TABLE I
RECOVERY RESULTS OF DIFFERENT METHODS ON REAL DATASETS

		RoTu1 (Alg. 1)		RoTu2 (Remark 1)		GCG [10]		HaLRTC [1]		TMAC [28]		MC (8)	
Datasets	MR (%)	Relerr	Time (S.)	Relerr	Time (S.)	Relerr	Time (S.)	Relerr	Time (S.)	Relerr	Time (S.)	Relerr	Time (S.)
Spectral Image ($205\times246\times96$)	60	1.24E-02	51.53	**1.03E-02**	46.44	3.25E-02	82.89	2.18E-02	43.48	1.15E-02	260.15	1.08E-02	154.01
	70	1.40E-02	41.69	**1.21E-02**	36.12	3.53E-02	66.35	3.00E-02	42.76	1.81E-02	257.93	1.40E-02	155.51
	80	1.63E-02	32.14	**1.49E-02**	27.73	3.79E-02	50.12	4.23E-02	44.65	2.65E-02	239.94	2.03E-02	159.38
	90	**2.18E-02**	21.61	**2.18E-02**	16.57	4.04E-02	30.43	6.47E-02	83.70	6.08E-02	205.45	4.06E-02	153.89
	95	**3.33E-02**	16.09	3.73E-02	10.91	4.41E-02	18.62	1.05E-01	161.33	1.70E-01	194.56	1.26E-01	151.13
	99	8.56E-02	10.95	8.95E-02	5.42	**7.63E-02**	10.30	8.50E-01	159.26	6.94E-01	179.47	6.57E-01	151.65
Brain MRI ($181\times217\times181$)	60	4.27E-02	63.74	1.27E-02	61.52	5.56E-02	111.75	**1.20E-06**	167.92	1.47E-02	304.78	4.50E-02	210.56
	70	4.69E-02	50.64	1.66E-02	48.47	6.03E-02	90.32	**2.99E-05**	170.87	3.55E-02	287.71	6.22E-02	214.39
	80	5.11E-02	37.96	2.37E-02	35.72	6.50E-02	67.19	**2.94E-03**	184.99	7.70E-02	267.30	1.07E-01	214.24
	90	5.85E-02	22.99	**4.64E-02**	20.74	7.08E-02	38.58	9.03E-02	190.18	1.79E-01	245.64	2.34E-01	213.94
	95	**6.98E-02**	14.46	1.04E-01	12.02	8.09E-02	20.96	3.45E-01	215.29	3.73E-01	233.96	4.04E-01	213.39
	99	**2.72E-01**	8.74	3.51E-01	5.00	3.32E-01	9.37	9.95E-01	1.23	9.09E-01	225.17	7.96E-01	214.38
Incisix ($256\times256\times100$)	60	2.07E-01	62.51	2.03E-01	57.09	2.54E-01	106.02	2.19E-01	115.67	**1.36E-01**	206.02	2.00E-01	199.94
	70	2.30E-01	50.87	2.28E-01	45.22	2.76E-01	86.14	2.72E-01	116.63	**1.64E-01**	241.25	2.50E-01	200.08
	80	2.62E-01	39.37	2.63E-01	33.88	2.98E-01	64.90	3.41E-01	131.70	**2.27E-01**	317.32	3.15E-01	199.24
	90	3.31E-01	26.39	3.44E-01	20.96	**3.30E-01**	40.54	4.43E-01	153.40	3.72E-01	297.26	4.50E-01	199.46
	95	4.25E-01	18.77	4.48E-01	13.21	**3.72E-01**	24.71	9.75E-01	1.29	5.61E-01	287.37	5.95E-01	198.52
	99	5.27E-01	12.54	5.16E-01	6.57	**5.15E-01**	14.00	9.95E-01	1.42	8.95E-01	263.67	8.68E-01	198.28
Knix1 ($256\times256\times3\times24$)	60	1.28E-01	50.03	**1.25E-01**	44.41	1.71E-01	83.06	1.56E-01	95.51	**1.25E-01**	117.06	3.04E-01	184.41
	70	1.45E-01	41.61	**1.42E-01**	35.78	1.86E-01	70.18	1.98E-01	99.09	1.48E-01	123.50	3.65E-01	187.73
	80	**1.67E-01**	32.49	1.70E-01	27.01	2.05E-01	58.06	2.58E-01	104.60	1.86E-01	201.37	4.48E-01	188.97
	90	**2.10E-01**	22.80	2.36E-01	17.37	2.48E-01	43.95	3.58E-01	121.62	2.62E-01	288.43	6.13E-01	186.84
	95	**2.75E-01**	17.03	3.15E-01	11.85	3.06E-01	35.30	4.76E-01	175.71	4.14E-01	277.59	7.66E-01	189.04
	99	6.01E-01	1.74	**4.11E-01**	7.03	4.59E-01	29.34	9.95E-01	1.38	1.01E+00	275.31	9.19E-01	194.66
Knix2 ($256\times256\times3\times22$)	60	1.15E-01	48.27	**1.12E-01**	42.23	1.56E-01	79.87	1.30E-01	73.95	1.31E-01	94.96	3.11E-01	167.94
	70	1.30E-01	40.98	**1.27E-01**	35.55	1.72E-01	69.39	1.65E-01	76.52	1.54E-01	88.72	3.67E-01	167.66
	80	**1.50E-01**	32.06	1.51E-01	26.52	1.90E-01	57.72	2.16E-01	78.67	1.90E-01	136.38	4.51E-01	167.73
	90	**1.95E-01**	22.20	2.10E-01	16.85	2.28E-01	43.06	3.07E-01	93.79	2.61E-01	238.33	6.15E-01	168.10
	95	**2.64E-01**	13.71	2.85E-01	11.96	2.79E-01	35.55	4.12E-01	137.50	3.44E-01	228.67	7.73E-01	168.55
	99	6.44E-01	1.18	**3.80E-01**	6.91	3.92E-01	29.56	9.95E-01	1.25	1.01E+00	226.52	9.21E-01	169.85
Knix3 ($256\times256\times3\times22$)	60	1.12E-01	46.46	**1.06E-01**	40.62	1.69E-01	77.55	1.20E-01	80.06	1.60E-01	106.44	3.88E-01	169.23
	70	1.26E-01	40.23	**1.21E-01**	34.10	1.85E-01	68.92	1.59E-01	82.31	1.90E-01	128.90	4.48E-01	171.90
	80	1.46E-01	32.38	**1.45E-01**	26.39	2.04E-01	58.07	2.29E-01	86.99	2.37E-01	153.64	5.33E-01	170.67
	90	**1.96E-01**	23.93	2.08E-01	17.61	2.48E-01	45.49	3.88E-01	102.66	3.20E-01	246.99	7.05E-01	169.61
	95	**2.87E-01**	18.89	3.07E-01	12.60	3.23E-01	38.00	5.50E-01	134.06	4.07E-01	236.82	8.37E-01	169.30
	99	5.15E-01	14.57	**4.93E-01**	7.78	5.29E-01	32.13	9.95E-01	1.37	9.44E-01	222.24	9.53E-01	167.25

$|\langle\nabla f(\mathcal{X}^{(k)}), \mathcal{D}^{(k)})\rangle|$, and is well scaled and also observed to converge to zero. For all the methods except HaLRTC, the threshold of the stopping criterion is $\epsilon = 10^{-5}$; for HaLRTC, we find that 10^{-5} is too loose and we set 10^{-7} for it. The max iteration for all the methods is 500. The initial guess for all the methods is the zero tensor.

The recovery results of the above methods are averaged over 10 instances and reported in Table I, where "MR" is the missing ratio and "Relerr" refers to the relative error $\|\mathcal{X}^* - \mathcal{B}\|_F / \|\mathcal{B}\|_F$ where $\mathcal{X}^*$ denotes the tensor recovered. From the table we observe that our methods RoTu1 and RoTu2 perform the best, followed by GCG. The reason may be that methods based on tensor nuclear norm can explore the low rank structure of the tensors more intrinsically. Concerning the CPU time, our methods and GCG are much faster than other methods when the MR value is very high. Comparing with MC, which is also solved by using the CG method, tensor-based methods are more accurate. This shows that when the data is tensor-structured, tensor-based methods can reveal better ground truth than that by treating it as unfolding matrices.

To graphically illustrate the effectiveness of our method, in Fig. 1 we show one slide of the recovery results of the spectral image dataset from different methods. The missing ratio is 0.95. We observe that RoTu1 almost recovers all the details, while HaLRTC loses almost all the details. Although TMAC can recover some details, we observe that in other recovery slides, the results are poor. That is why the relative error of TMAC is worse than that of HaLRTC.

We also show the influence of the parameter β of RoTu1 in Fig. 2, where $\beta \in [10^3, 10^4]$. The datasets and the MR value is chosen the same as the above setting. The plots show that RoTu1 is not very sensitive to β in a wide range. RoTu2 does not need any parameter to control the complexity.

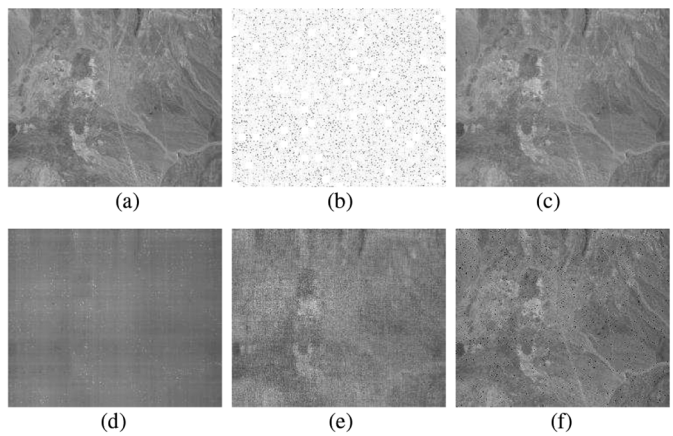


Fig. 1. The figures illustrate the recovery results of different methods on the dataset spectral image. (a) One slide of the dataset. (b) 0.95 missing entries. (c) ~ (f): Recovery results of different methods. (a) One slide. (b) MR = 0.95. (c) RoTu1 (Alg. 1). (d) HaLRTC [1]. (e) TMAC [28] (f) MC (8).

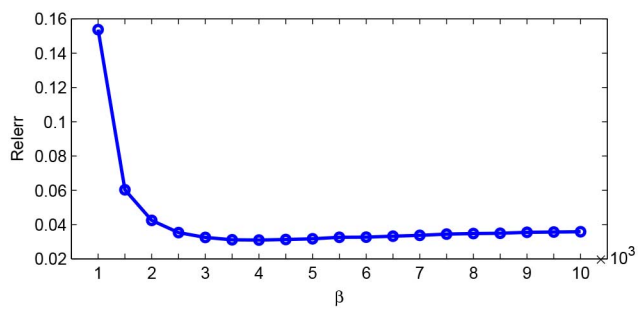


Fig. 2. Plots of relative errors versus β.

V. CONCLUSION

In this letter, we proposed a new algorithm for tensor completion. Using the tensor nuclear/spectral norm and under the framework of the conditional gradient method, we showed that our methods solve the problem via rank-one tensor updating. We presented an efficient subroutine to compute the tensor spectral norm. Numerical experiments on real datasets showed that our methods are efficient and effective.

REFERENCES

[1] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, 2013.

[2] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inv. Probl.*, vol. 27, no. 2, p. 025010, 2011.

[3] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, pp. 303–351, 2014.

[4] N. Li and B. Li, "Tensor completion for on-board compression of hyperspectral images," in *17th IEEE Int. Conf. Image Processing (ICIP), 2010*, 2010, pp. 517–520, IEEE.

[5] M. Signoretto, R. Van De Plas, B. De Moor, and J. A. K. Suykens, "Tensor versus matrix completion: A comparison with application to spectral data," *IEEE Signal Process. .Lett.*, vol. 18, no. 7, pp. 403–406, 2011.

[6] R. Tomioka, K. Hayashi, and H. Kashima, "Estimation of low-rank tensors via convex optimization," *arXiv preprint 1010.0789*, 2010.

[7] M. Yuan and C.-H. Zhang, "On tensor completion via nuclear norm minimization," *arXiv preprint 1405.1773*, 2014.

[8] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Trans. Information Theory*, vol. 60, no. 2, pp. 1260–1280, Feb. 2014.

[9] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, no. 1–2, pp. 95–110, 1956.

[10] Y. Yu, H. Cheng, and X. Zhang, "Approximate low-rank tensor learning," in *7th NIPS Workshop on Optimization for Machine Learning*. : , 2014.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, pp. 455–500, 2009.

[12] A. Grothendieck, "Produits tensoriels topologiques et espaces nucleaires," *Mem. Amer. Math. Soc.*, vol. 16, p. 140, 1955.

[13] L.-H. Lim, "Singular values and eigenvalues of tensors: A variational approach," in *1st IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005*, 2005, vol. 1, pp. 129–132.

[14] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proc. 30th Int. Conf. Machine Learning (ICML-13)*, 2013, pp. 427–435.

[15] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. Washington, DC, USA: U.S. Govt. Press Office, 1950.

[16] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, pp. 45:1–45:39, 2013.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1253–1278, 2000.

[18] B. Chen, S. He, Z. Li, and S. Zhang, "Maximum block improvement and polynomial optimization," *SIAM J. Optim.*, vol. 22, pp. 87–107, 2012.

[19] X. Zhang, C. Ling, and L. Qi, "The best rank-1 approximation of a symmetric tensor and related spherical optimization problems," *SIAM J. Matrix Anal. Applicat.*, vol. 33, no. 3, pp. 806–821, 2012.

[20] T. Zhang and G. H. Golub, "Rank-one approximation to high order tensors," *SIAM J. Matrix Anal. Applicat.*, vol. 23, no. 2, pp. 534–550, 2001.

[21] X. Zhang, L. Qi, and Y. Ye, "The cubic spherical optimization problems," *Math. Comput.*, vol. 81, no. 279, pp. 1513–1525, 2012.

[22] S. He, Z. Li, and S. Zhang, "Approximation algorithms for homogeneous polynomial optimization with quadratic constraints," *Math. Program.*, vol. 125, pp. 353–383, 2010.

[23] A. M.-C. So, "Deterministic approximation algorithms for sphere constrained homogeneous polynomial optimization problems," *Math. Program.*, vol. 129, no. 2, pp. 357–382, 2011.

[24] S. He, B. Jiang, Z. Li, and S. Zhang, "Probability bounds for polynomial functions in random variables," *Math. Oper. Res.*, 2014.

[25] Y. Yang, Y. Feng, X. Huang, and J. A. K. Suykens, Rank-one tensor properties with applications to a class of tensor optimization problems KU Leuven, Leuven, Belgium, Internal Rep. 13-245, 2013, ESAT-SISTA.

[26] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[27] Z. Wang, M.-J. Lai, Z. Lu, and J. Ye, "Orthogonal rank-one matrix pursuit for low rank matrix completion," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. A488–A514, 2014.

[28] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *arXiv preprint 1312.1254*, 2013.