Tensor Completion Via Optimization on the Product of Matrix Manifolds

Josh Girson, Shuchin Aeron
School of Engineering, Tufts University, Medford, MA 02155
joshua.girson@tufts.edu, shuchin@ece.tufts.edu

***Abstract*—We present a method for tensor completion using optimization on low-rank matrix manifolds. Our notion of tensor-rank is based on the recently proposed framework of tensor-Singular Value Decomposition (t-SVD) in [1], [2]. In contrast to convex optimization methods used in [1] that operate in a high-dimensional space, in the manifold setting, one works directly in the reduced dimensionality space and is thus able to significantly reduce the computational costs [3], [4]. In this paper we focus on 3-D data and under the tensor algebraic framework of [1], [2] we show that a 3-D tensor of fixed tubal-rank can be seen as an element of the product manifold of fixed low-rank matrices in the Fourier domain. The tensor completion problem then reduces to finding the best approximation to the sampled data on this product manifold.**

Further, for 3-D data we consider and compare recovery performance under two approaches. In the first approach one samples entire mode-3 fibers of the tensor, which we refer to as *tubal-sampling*. The second approach employs element-wise sampling and we simply refer to this method as sampling. For these two types of sampling approaches, we present simulation results for surveillance video data and show that recovery under random sampling has better performance compared to the random tubal-sampling.

I. Introduction

In this paper we consider manifold based optimization for tensor completion. The theory and methods for tensor completion depend on the particular algebraic framework and the kind of tensor factorization used, see section II for a brief survey. These factorizations yield different notions of tensor rank. Tensor completion methods then seek to find a tensor of lowest tensor-rank subject to data matching constraints. In some cases one can relax this criteria to a computable convex surrogate function, such as the tensor nuclear norm, appropriately defined in many different ways by different authors. This leads to computationally efficient algorithms as well as sufficient conditions for provable guarantee in recovery under given sampling constraints. However, all these methods are iterative and they require computing the rank revealing tensor factorization (possibly involving sparse matrices) at each step in order to regularize the spectra. This makes them computationally very expensive and leads to issues of scalability for many practical problems.

On the other hand if the tensor rank is known a-priori then manifold based optimization methods have shown to be promising [3], [4] in significantly reduce computational costs (per iteration) by exploiting manifold geometry and using tools from numerical linear algebra. In this paper we consider using such methods for tensor completion under the recently proposed algebraic framework for tensor decomposition [1], [2] as summarized in section II-B below.

In addition we consider two types of sampling methods, namely tubal-sampling (which naturally arises in a number of scenarios) where entire tensor fibers are sampled at a time and element-wise sampling where each element is sampled at a time. Then we look at the recovery performance for both of these cases. We show that element-wise sampling has better performance compared to tubal-sampling by direct evaluation on real data sets.

Notation: Tensors are denoted by bold-face calligraphic letters $\boldsymbol{\mathcal{X}}$, matrices by bold-face uppercase letters $\mathbf{X}$ and vectors by bold-face lower case $\mathbf{x}$ and sometimes with an arrow $\vec{\mathbf{x}}$ to denote tensor fibers. The elements of the arrays are denoted using MATLAB convention. For example, $\boldsymbol{\mathcal{X}}(:,:,i)$ denotes the i-th frontal slice, $\boldsymbol{\mathcal{X}}(i,j,:)$ denotes the (i,j)-th fiber going into the board and so on. $\boldsymbol{\mathcal{X}}^{(i)}$ is shorthand to denote i-th frontal face $\boldsymbol{\mathcal{X}}(:,:,i)$. Additional notation is introduced as needed.

II. Background: Tensor algebra and rank

A. Tensors as multilinear operators

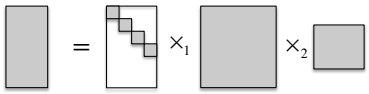

Fig. 1: SVD expressed in terms of mode multiplications.

In the multilinear algebraic framework, a tensor is treated as an element of a multilinear space, which in turn is constructed as *outer products of vector spaces* [5]. To appreciate this idea, consider two vector spaces U and V of dimensions n_1 and n_2 respectively. Then an element $\mathbf{X}$ of the outer product $U \otimes V$ of vector spaces U, V is an $n_1 \times n_2$ matrix of the form $\mathbf{X} = \mathbf{U}\mathbf{W}\mathbf{V}^\top$ where columns of $\mathbf{U}$ span U and columns of $\mathbf{V}$ span V. In multilinear algebraic notation, this can also be written as $\mathbf{W} \times_1 \mathbf{U} \times_2 \mathbf{V}$, where $\times_1$ denotes matrix multiplication by mode-1 fibers (columns in this case) and $\times_2$ denotes multiplication by mode-2 fibers (rows in this case), see [6]. When $\mathbf{W}$ is restricted to be diagonal and $\mathbf{U}$, $\mathbf{V}$ to be unitary, we obtain the Singular Value Decomposition (SVD) as shown in Figure 1.

For order N-tensors when seen as elements of outer product of N vector spaces, $V_1 \otimes V_2 \otimes \ldots \otimes V_N$, the Canonical Polyadic (CP) decomposition [6] generalizes the SVD (with some caveats, [7], [8]) via, $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{G}_1 \times_2 \ldots \times_N \mathbf{G}_N$, where $\boldsymbol{\mathcal{S}}$ is a super-diagonal tensor and $\times_k$ denotes the multiplication along the mode k fibers. On the other hand Higher Order SVD (HOSVD) or Tucker decompositions [9]

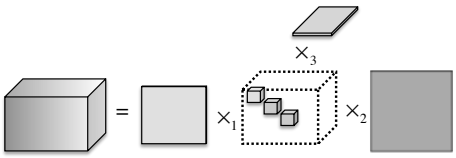

Fig. 2: CP and HOSVD for 3-D tensors. For CP the core tensor is super-diagonal.

flatten or matricize the data along various modes and use SVD of these matrices to obtain the following factorization, $\mathcal{X} = \mathcal{C} \times_1 \mathbf{G}_1 \times_2 \mathbf{G}_2 \times_3 .... \times_N \mathbf{G}_N$, where the core tensor $\mathcal{C}$ is not super-diagonal. The CP and HOSVD are illustrated in Figure 2. These models have shown to be promising for modeling various types of data as low-rank tensors with gains over traditional methods [6].

B. Tensors as linear operators

In contrast to treating tensors as multilinear operators, in this paper we will consider the linear algebraic setting developed in [2] that treats tensors as linear operators. This construction has shown to be useful in a variety of applications [1], [10].

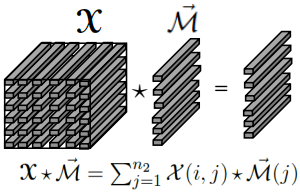

Fig. 3: 3-D tensors as operators on oriented matrices.

In the framework proposed in [2] a 3-D array is defined as a linear operator using the t-product defining the multiplication action. As shown in [1] one views a 3-D tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ as an $n_1 \times n_2$ matrix of tubes (vectors oriented into the board). Similarly one can consider a $n_1 \times 1 \times n_3$ tensor as a vector of tubes. Such tensors are referred to as oriented matrices [2], and are denoted by $\vec{\mathcal{M}}$.

Now in order to define the 3-D tensor as a linear operator on the set of oriented matrices $\vec{\mathcal{M}}$ [2], one defines a multiplication operation between two tubes $\vec{\mathbf{v}} \in \mathbb{R}^{1 \times 1 \times n_3}$ and $\vec{\mathbf{u}} \in \mathbb{R}^{1 \times 1 \times n_3}$ resulting in another tube of the same length. Specifically this multiplication operation is given by circular convolution denoted by $\star$. See appendix for the definition of circular convolution.

Under this construction, the operation of a tensor $\mathcal{X}$ on $\vec{\mathcal{M}} \in \mathbb{R}^{n_2 \times 1 \times n_3}$ is another oriented matrix of size $n_1 \times 1 \times n_3$ whose i-th tubal element is given by,

$$\mathcal{X} \star \vec{\mathcal{M}} = \sum_{j=1}^{n_2} \mathcal{X}(i, j, :) \star \vec{\mathcal{M}}(j, 1. :)$$

as illustrated in Figure 3. This product between is referred to as the t-product.

t-SVD - Under the above construction viewing a 3-D tensor as a linear operator over the set of oriented matrices, one can compute a tensor-Singular Value Decomposition (t-SVD) as shown in Figure 4. Since $\star$ is given by the circular convolution

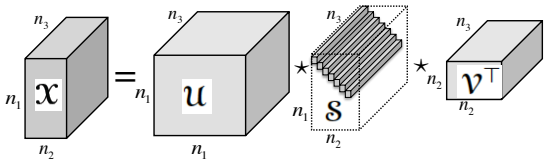

Fig. 4: t-SVD under the t-product

the t-SVD can be computed using the Fast Fourier Transform (`fft`) using Algorithm 1 [2]. In the algorithm $\texttt{fft}(\mathcal{X}, [], 3)$ denotes the `fft` along the 3rd dimension and $\texttt{ifft}(\mathcal{X}, [], 3)$ denotes the inverse `fft` along the third dimension.

The component tensors $\mathcal{U}$ and $\mathcal{V}$ obey the orthogonality conditions $\mathcal{U}^\top \star \mathcal{U} = \mathcal{I}$, $\mathcal{V}^\top \star \mathcal{V} = \mathcal{I}$ with the following definitions for tensor transpose $(\cdot)^\top$ and and identity tensor $\mathcal{I}$ (of appropriate dimensions).

Definition II.1. ***Tensor Transpose***. Let $\mathcal{X}$ be a tensor of size $n_1 \times n_2 \times n_3$, then $\mathcal{X}^\top$ is the $n_2 \times n_1 \times n_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of the transposed frontal slices 2 through n_3.

Definition II.2. ***Identity Tensor***. The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is a tensor whose first frontal slice is the $n \times n$ identity matrix and all other frontal slices are zero.

Note: This construction can be generalized considerably as recently shown in [11], but in this paper we restrict ourselves to using circular convolution to define t-SVD.

Algorithm 1 tSVD

Input: $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$
$\widehat{\mathcal{X}} \leftarrow \texttt{fft}(\mathcal{X}, [\,], 3)$;
for $i = 1$ to n_3 **do**
  $[\hat{\mathbf{U}}, \hat{\mathbf{S}}, \hat{\mathbf{V}}] = \texttt{SVD}(\widehat{\mathcal{X}}(:, :, i))$
  $\widehat{\mathcal{U}}^{(i)} = \hat{\mathbf{U}}; \widehat{\mathcal{S}}^{(i)} = \hat{\mathbf{S}}; \widehat{\mathcal{V}}^{(i)} = \hat{\mathbf{V}}$;
end for
$\mathcal{U} \leftarrow \texttt{ifft}(\widehat{\mathcal{U}}, [\,], 3)$; $\mathcal{S} \leftarrow \texttt{ifft}(\widehat{\mathcal{S}}, [\,], 3)$;
$\mathcal{V} \leftarrow \texttt{ifft}(\widehat{\mathcal{V}}, [\,], 3)$;

III. Manifold structure of fixed rank tensors

It has been shown that tensors of fixed multi-rank under HOSVD, and H-Tucker decompositions form a smooth manifold. Using this several algorithms have been considered in the literature [12], [13], [14] for numerically efficient inferencing using optimization on smooth manifolds. Similarly, under the t-SVD and using the notion of tubal-rank as defined below, we see that the 3-D tensors of fixed tubal-rank also form a smooth manifold. First we introduce the notion of tubal-rank under the t-SVD.

Tensor tubal-rank - Under the t-SVD, a useful notion of rank is tubal-rank defined to be the number of non-zero singular tubes in $\mathcal{S}$.

Product manifold structure - In this paper we want to restrict ourselves to real manifolds.

But since the t-product uses Fourier domain computation, we first lift a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ to an *even* tensor $\mathcal{X}_e \in$

$\mathbb{R}^{n_1 \times n_2 \times 2n_3-1}$ by adding *dummy* frontal slices so that the tensor $\mathcal{X}_e$ has tubes that are *even*, i.e. $\mathcal{X}_e(i,j,n_3+(k-1)) = \mathcal{X}(i,j,n_3-(k-2))$, $\forall (i,j)$, $2 \leq k \leq n_3$.

In this case, by *construction* the set of even 3-D tensors $\mathcal{X}_e$ of fixed *tubal*-rank, form a smooth embedded manifold $\mathbb{M}$ of $\mathbb{R}^n$ for some n. In particular, the smooth manifold can be realized as a *real product manifold* of rank r matrices in the Fourier domain,

$$\widehat{\mathcal{X}_e} \in \mathbb{M} = \underbrace{\mathbb{M}_r \times \cdots \times \mathbb{M}_r}_{2n_3-1}.$$

This is because the `fft` of a real and even signal is real and even. Note that we are not enforcing the symmetry (even) condition in this embedding.

This allows us to rapidly adapt optimization methods on real matrix manifolds with computationally efficient Reimannian gradient and retraction computations under various geometries [15]. In this paper we employ MANOPT MATLAB toolbox [16] with its inbuilt manifold factory for fixed rank tensors that uses the Reimannian geometry for fixed rank matrices described in [3]. Note that using the product structure the gradient and retraction can be done independently across the manifolds and can even be parallelized.

IV. Tensor completion from missing entries

We consider the problem of tensor completion from missing entries assuming that the tensor has low tensor tubal-rank. Specifically given a sampling set Ω the problem is to find a tensor $\mathcal{X}$ of tubal-rank r that closely approximates the observations. We consider two cases.

A. Tubal-sampling

The first sampling method that we consider is the tubal-sampling. For example, such scenarios typically arise in seismics where at a given location one typically records a temporal signal of certain length [14], [10]. In this case one is given observations $\mathcal{X}(i,j,:)$ at indices $(i,j) \in \Omega, i \in \{1,2,...,n_1\}, j \in \{1,2,...,n_2\}$, where Ω denotes the sampling set. Let $\mathcal{P}_\Omega$ denote the corresponding (linear) sampling operator.

First note that the problem can be equivalently converted to recovering the even tensor $\mathcal{X}_e$. Now note that for tubal-sampling, the optimization problem can be solved in the Fourier domain where it separates out slice by slice. That is, the slices $\widehat{\mathcal{X}_e}(:,:,i)$ can be recovered independently by solving for,

$$\min_{\widehat{\mathbf{x}_e^{(i)}} \in \mathbb{M}_r} \|\hat{\mathcal{P}}^{(i)}_{e,\Omega}(\hat{\mathcal{X}}^{(i)}_e - \hat{\mathcal{Y}}^{(i)}_e)\|_F^2 \,. \tag{1}$$

Note that all the $\widehat{\mathcal{P}}^{(i)}_{e,\Omega}$ are the same and hence the problem reduces to individual matrix completion problems in the Fourier domain.

B. Element-wise Sampling

In element-wise sampling, indices $(i,j,k) \in \Omega, i \in \{1,...,n_1\}, j \in \{1,...,n_2\}, k \in \{1,...,n_3\}$ are given as observations. Let $\mathcal{P}_\Omega$ denote the corresponding (linear) sampling operator. Now note that unlike the tubal-sampling, in the Fourier domain one cannot separate the optimization problem slice-wise. Here we explicitly need to use the product manifold structure together with explicit characterization of the linear operator $\widehat{\mathcal{P}}_\Omega$ as given by Equation (3) in the Appendix.

In particular, using the lifting to first estimate the even tensor $\mathcal{X}_e$, we solve for the following optimization problem,

$$\min_{\widehat{\mathcal{X}_e} \in \mathbb{M}} \|\hat{\mathcal{P}}_{e,\Omega}(\hat{\mathcal{X}}_e - \hat{\mathcal{Y}}_e)\|_F^2 \quad \text{where } \mathbb{M} \in \underbrace{\mathbb{M}_r \times \cdots \times \mathbb{M}_r}_{2n_3-1}. \tag{2}$$

We now compare the performance of these two types of sampling on some real data sets.

V. Numerical Experiments

We will employ random sampling for both the tubal and element-wise sampling. The test data that we use here is footage from a still camera view of a traffic intersection. This data is prime for testing because the consistency of the background throughout the scope allows for similarities between frames. This property also holds in panning videos, as there is still consistency from frame to frame although the content is still changing. This property does not hold with zooming videos, as the zooming property removes the standard background data that is constant from frame to frame.

We compared the results of recovery at different sampling ratios for both of the methods. The results are shown in Figure 5. We note that the for this data tubal sampling is worse compared to random sampling but has reasonable performance as the sampling rate increases. While it may be intuitively obvious that random sampling should be better compared to tubal-sampling, the result of tensor completion on *panning video* can give comparable performance for the same sampling rates, see [17] for such an example (we omit the manifold results for panning video here due to lack of space).

VI. Conclusions

While both methods seemed to do relatively well, the completion using element-wise sampling performed better than the tubal-sampling. It seems as though the individual frames are overall clearer and that there is more interpolation happening with the completely random sampling. There are spots in the tubal sampling where it seems as though there is no inference whatsoever and some of the points that were originally not sampled remain with values of all zeros in the output tensor. One possible reason for this could be that since the algorithm never had any knowledge of the data at that point, there was nothing that it could base any inference off of. The completely random sampling, on the other hand, would have had data for every pixel location for at least one point in time and could then infer the previous or future time's values based on this information.

In the future we would like to continue by testing this on larger, more diverse datasets. The manifold based optimization can also be made online in that one can update as the data is received in real time. Finally, it would be interesting to see if there is a way to apply this type of algorithm to tensors of higher order. One approach is to use a recursive construction of *product manifold of product manifolds*.

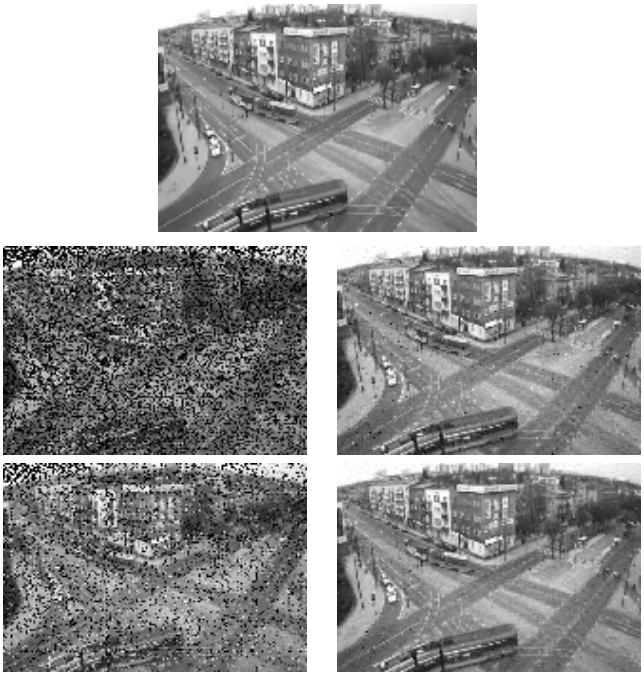

Fig. 5: Original frame and examples of recovery from a video sequence of size $117 \times 160 \times 10$ with tubal-rank fixed at 60 for 50% and 80% sampling rates for random tubal-sampling (Left figures) and random element-wise sampling (Right figures).

VII. Appendix

A. Circular convolution and Fourier transforms

Circular Convolution: For two vectors $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathbb{R}^n$, the circular convolution denoted by $\star$ is defined as $\vec{\mathbf{x}} \star \vec{\mathbf{y}} = \vec{\mathbf{z}} \in \mathbb{R}^n$ where $\vec{\mathbf{z}}[j] = \sum_{k=1}^{n} \vec{\mathbf{x}}[k] \star \vec{\mathbf{y}}[(j-k) \bmod n]$.

Property 1: $\vec{\mathbf{x}} \star \vec{\mathbf{y}} = \texttt{ifft}(\texttt{fft}(\vec{\mathbf{x}}) \circ \texttt{fft}(\vec{\mathbf{y}}))$ where $\circ$ denotes the Hadamard product or the element-wise multiplication.

Observation: We can also write $\vec{\mathbf{z}} = \vec{\mathbf{x}} \star \vec{\mathbf{y}}$ in the matrix vector product form:

$$\begin{bmatrix} \vec{\mathbf{z}}[1] \\ \vec{\mathbf{z}}[2] \\ \vdots \\ \vec{\mathbf{z}}[n] \end{bmatrix} = \underbrace{\begin{bmatrix} \vec{\mathbf{x}}[1] & \vec{\mathbf{x}}[n] & \dots & \vec{\mathbf{x}}[2] \\ \vec{\mathbf{x}}[2] & \vec{\mathbf{x}}[1] & \dots & \vec{\mathbf{x}}[3] \\ \vdots & \vdots & \ddots & \vdots \\ \vec{\mathbf{x}}[n] & \vec{\mathbf{x}}[n-1] & \dots & \vec{\mathbf{x}}[1] \end{bmatrix}}_{\texttt{bcirc}(\vec{\mathbf{x}})} \begin{bmatrix} \vec{\mathbf{y}}[1] \\ \vec{\mathbf{y}}[2] \\ \vdots \\ \vec{\mathbf{y}}[n] \end{bmatrix}$$

where the central matrix dealing with the $\vec{\mathbf{x}}$ is the block circulant matrix, $\texttt{bcirc}(\vec{\mathbf{x}})$, formed from $\vec{\mathbf{x}}$.

Property 2: Multiplication in the time domain is equivalent to convolution in the Fourier domain. Thus we see, as before, that $\texttt{fft}(\vec{\mathbf{x}} \circ \vec{\mathbf{y}}) = \texttt{fft}(\vec{\mathbf{x}}) \star \texttt{fft}(\vec{\mathbf{y}})$. Let us denote by $\hat{\vec{\mathbf{x}}}$ the $\texttt{fft}$ of $\vec{\mathbf{x}}$, then we have in simplified terms, $\texttt{fft}(\vec{\mathbf{x}} \circ \vec{\mathbf{y}}) = \hat{\vec{\mathbf{x}}} \star \hat{\vec{\mathbf{y}}}$

Now if we sample in the time domain in a totally random manner (i.e. not a tubal manner but element wise randomly), then each tube of the tensor is $.*$ by a random $(0,1)$ vector. To explain further, let $\boldsymbol{\mathcal{X}}_{ij} \equiv \boldsymbol{\mathcal{X}}(i,j,:)$ denote the i,j^{th} tube of $\boldsymbol{\mathcal{X}}$. Then the sampled data $\boldsymbol{\mathcal{Y}}_{ij}$ is given by $\boldsymbol{\mathcal{Y}}_{ij} = \boldsymbol{\mathcal{X}}_{ij} \circ \boldsymbol{\mathcal{P}}_{ij}$ where $\boldsymbol{\mathcal{P}}$ is the sampling matrix. Thus we see that $\boldsymbol{\mathcal{P}}_{ij}$ is tube of zeros and ones. Now using Property 1 above, we see that this becomes $\hat{\boldsymbol{\mathcal{Y}}}_{ij} = \texttt{fft}(\boldsymbol{\mathcal{X}}_{ij}) \star \texttt{fft}(\boldsymbol{\mathcal{P}}_{ij})$. Now we can rewrite this using our observation:

$$\hat{\boldsymbol{\mathcal{Y}}}_{ij} = \texttt{bcirc}(\hat{\boldsymbol{\mathcal{P}}}_{ij})\, \hat{\boldsymbol{\mathcal{X}}}_{ij}\,.$$

Therefore, over all $i = 1, 2, \dots, n_2$ and $j = 1, 2, \dots, n_3$:

$$\hat{\boldsymbol{\mathcal{Y}}}(:) = \begin{bmatrix} \texttt{bcirc}(\hat{\boldsymbol{\mathcal{P}}}_{11}) & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \texttt{bcirc}(\hat{\boldsymbol{\mathcal{P}}}_{n_2 n_3}) \end{bmatrix} \hat{\boldsymbol{\mathcal{X}}}(:) \quad (3)$$

References

[1] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3842–3849.

[2] M. Kilmer, K. Braman, N. Hao, and R. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013/06/20 2013. [Online]. Available: http://dx.doi.org/10.1137/110837711

[3] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM Journal on Optimization*, 2013. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/110845768

[4] P.-A. Absil and J. Malick, "Projection-like Retractions on Matrix Manifolds," *SIAM Journal on Optimization*, vol. 22, no. 1, pp. 135–158, Jan. 2012. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/100802529

[5] W. Hackbusch, *Tensor spaces and numerical tensor calculus*. Springer, 2012.

[6] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2013/06/30 2009. [Online]. Available: http://dx.doi.org/10.1137/07070111X

[7] A. Stegeman and N. D. Sidiropoulos, "On kruskal's uniqueness condition for the Candecomp/Parafac decomposition," *Linear Algebra and its Applications*, vol. 420, no. 23, pp. 540 – 552, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379506003855

[8] V. D. Silva and L.-H. Kim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, 2008.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2013/06/30 2000. [Online]. Available: http://dx.doi.org/10.1137/S0895479896305696

[10] G. Ely, S. Aeron, N. Hao, and M. E. Kilmer, "5d and 4d pre-stack seismic data completion using tensor nuclear norm (tnn)," *Society of Exploration Geophysicists (SEG) workshop*, 2013.

[11] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor-tensor products with invertible linear transforms," Tufts University, Tech. Rep., 2014, submitted to Linear Algebra and its Applications. [Online]. Available: http://www.ece.tufts.edu/~shuchin/els_cosine_716.pdf

[12] H. Kasai and B. Mishra, "Riemannian preconditioning for tensor completion," *arXiv.org*, Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.02159

[13] D. Kressner, M. Steinlechner, and B. Vandereycken, "Low-rank tensor completion by Riemannian optimization," *Bit Numerical Mathematics*, vol. 54, no. 2, pp. 447–468, Jun. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10543-013-0455-z

[14] C. Da Silva and F. J. Herrmann, "Optimization on the Hierarchical Tucker manifold - applications to tensor completion," *arXiv.org*, May 2014. [Online]. Available: http://arxiv.org/abs/1405.2096v1

[15] B. Mishra, K. A. Apuroop, and R. Sepulchre, "A Riemannian geometry for low-rank matrix completion," *arXiv.org*, Nov. 2012. [Online]. Available: http://arxiv.org/abs/1211.1550

[16] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014. [Online]. Available: http://www.manopt.org

[17] Z. Zhang and S. Aeron, "Exact tensor completion using tsvd," *submitted to IEEE Transactions on Signal Processing*, 2015, http://arxiv.org/abs/1502.04689.