Low-rank Promoting Transformations and Tensor Interpolation - Applications to Seismic Data Denoising

Curt Da Silva[1] and Felix J. Herrmann[2]
[1] Dept. of Mathematics [2] Dept. of Earth and Ocean Sciences, University of British Columbia Vancouver, BC, Canada

January 15, 2014

Abstract

In this abstract, we extend our previous work in Hierarchical Tucker (HT) tensor completion, which uses an extremely efficient representation for representing high-dimensional tensors exhibiting low-rank structure, to handle subsampled tensors with noisy entries. We consider a 'low-noise' case, so that the energies of the noise and the signal are nearly indistinguishable, and a 'high-noise' case, in which the noise energy is now scaled to the amplitude of the entire data volume. We examine the effect of the noise in terms of the singular values along different matricizations of the data, i.e. reshaping of the tensor along different modes. By interpreting this effect in the context of tensor completion, we demonstrate the inefficacy of denoising by this method in the source-receiver domain. In light of this observation, we transform the decimated, noisy data in to the midpoint-offset domain, which promotes low-rank behaviour in the signal and high-rank behaviour in the noise. This distinction between signal and noise allows low-rank interpolation to effectively denoise the signal with only a marginal increase in computational cost. We demonstrate the effectiveness of this approach on a 4D frequency slice

Introduction

Acquiring 3D seismic data volumes is a time- and cost-intensive process owing to the *multidimensional* nature of the data. These volumes, which have dimensions (source x, source y, receiver x, receiver y, time), need to be sampled sufficiently in multiple dimensions, which necessarily drives up the associated costs. Moreover, the five-dimensional nature of the data makes it inherently expensive to process, due to the so called *curse of dimensionality*, i.e. exponential dependence of the number of dimensions.

Despite its size, 3D seismic data has an inherent *low-dimensionality* that one can exploit in order to interpolate missing sources and/or receivers. In this abstract, we briefly review the Hierarchical Tucker (HT) format, a structured tensor format that we have previously exploited for interpolating seismic data volumes with missing receivers. Despite its proficiency at recovering seismic data volumes in the noiseless case, the organization of the data in the source-receiver domain has difficulties in interpolating data with noisy receivers as we shall see. We develop an understanding of the nature of the noise in terms of the singular values of different *matricizations* of the data. We interpret these observations in the context of *matrix completion*, the theory for which states that subsampling and noise artifacts should *increase* the singular value decay in a particular domain in order for rank-minimizing recovery to work effectively. Using this understanding, we transform the data in to the *midpoint-offset* domain before completing it, which results in a significant reduction of the noise imprint without large overhead costs.

This abstract extends our previous work on interpolating seismic data volumes in the Hierarchical Tucker format in Da Silva and Herrmann (2013). Other approaches to the problem of seismic tensor completion, in the Tucker tensor format, include Kreimer and Sacchi (2012), which uses a projection on to convex sets (POCS) technique, and Kreimer and Sacchi (2013), which uses a per-dimension soft thresholding. These techniques require per-iteration SVDs of the underlying data, which are prohibitively expensive when the data volumes are large. We will see, in a publication under preparation, that using an ad-hoc windowing approach to reduce computational costs can adversely affect recovery results. Trickett et al. (2013) uses a generalization of Hankel matrices to the tensor case in order to perform completion in the CP tensor format. This approach squares the number of data points, which scales very poorly for large data volumes, and moreover the CP rank itself is not well defined for general tensors, unlike the Tucker or Hierarchical Tucker ranks.

Hierarchical Tucker Format

Definition 1. If we have a d-dimensional tensor X with dimensions $n_1 \times n_2 \times \cdots \times n_d$, if $t = (i_1, i_2, \ldots, i_p)$ is a set of indices with $i_j \in \{1, 2, \ldots, d\}$, the *matricization* of the tensor X with respect to t, denoted $X^{(t)}$, is given by reshaping the tensor X into the matrix $X^{(t)}$ with the dimensions given by t reshaped along the rows, and the other dimensions reshaped into the columns of the matrix.

The Hierarchical Tucker format, first introduced in Hackbusch and Kühn (2009), represents a particular class of tensors that exhibit *low-rank* behaviour in particular *matricizations*. We succinctly describe this format via Figure 1. For say, a $4D$ tensor X, e.g. a $3D$ frequency slice, with dimensions $n_1 \times n_2 \times n_3 \times n_4$, we can reshape it in to a matrix $X^{(1,2)}$ with dimensions 1 and 2 along the rows and the other two dimensions along the columns. By performing an SVD-like decomposition on to this matrix, we 'split apart' dimension groups $1,2$ from $3,4$, obtaining 'singular vectors' U_{12} that have dimensions 1 and 2 along its rows. By reshaping this matrix in to a $3D$ cube, we can further split apart dimension 1 from dimension 2 as shown in the bottom of Figure 1. We apply the same splitting for U_{34}.

As a result of this construction, we do not need to store the intermediate matrices U_{12}, U_{34}: the small leaf matrices U_i, $i = 1, \ldots, 4$ and small, intermediate tensors B_t specify the full tensor X completely. The number of parameters needed to represent HT tensors is $\leq dNK + (d-2)K^3 + K^2$, where $N = \max_{i=1,\ldots,d} n_i$, K is the maximum internal rank parameter, and d is the number of dimensions. When $d > 3$ and $K \ll N$, so that the tensor is low-rank, this quantity is much less than N^d, the usual storage requirements needed to represent a $d-$dimensional array, effectively breaking the curse of dimensionality for these tensors.

In the seismic context, we organize X with coordinates $(x_{\text{src}}, x_{\text{rec}}, y_{\text{src}}, y_{\text{rec}})$, which promotes the singular value decay of the seismic data volume and allows it to be well-represented in this format, at least in the low-frequency regime. Owing to source-receiver reciprocity, we assume for concreteness that our data volume is randomly missing receivers and seek to interpolate the data at the missing locations.

In this setting, we let $\phi(x)$ to be the fully-expanded tensor from parameters $x = (U_t, B_t)$. Then, given subsampled data D and a subsampling operator A, we are looking to solve

$$\min_{x=(U_t,B_t)} \|A\phi(x) - b\|_2^2 \quad (1)$$
$$\text{s.t. } U_t^H U_t = I_{k_t}, (B_t^{(k_l,k_r)})^T B_t^{(k_l,k_r)} = I_{k_t}$$

where A is the receiver subsampling operator and b is the input data with missing receivers, i.e. we try to fit the subsampled data subject to orthogonality constraints on the parameters.

The specific algorithmic details of how to solve this problem efficiently have been developed in Da Silva and Herrmann (2013) and will not be elaborated on here. Suffice to say, we can solve these problems efficiently and in a parallelizable manner, without the the need for SVDs of large matrices.

Noisy receivers and singular values

In the noiseless case, the HT format is proficient at interpolating 3D seismic data volumes with missing traces/receivers in the low-frequency regime. However, the $(x_{\text{src}}, x_{\text{rec}}, y_{\text{src}}, y_{\text{rec}})$ organization of the data, is unable to effectively deal with a sparse number of noisy receivers. In order to understand this phenomenon, we randomly remove 50% of the receivers from our data volume D and replace 5% of the remaining receivers by Gaussian noise approximately scaled to the norm of the removed receiver, which we refer to as the 'low-noise' case. Using this energy scaling, the standard noise-removal techniques of trace-by-trace energy comparison cannot mute out the noise effectively. In our 'high-noise' scenario, we scale the noise to the energy of the *undecimated* signal. We display the singular values of the resulting matricizations in Figure 2 with black denoting the original signal, red denoting the subsampled signal, and blue denoting the low noise signal and green denoting the high noise signal.

In the source-receiver organization (top row), the singular values of the source dimension are barely changed by the subsampling and the addition of noise in the receiver direction. From a matrix/tensor recovery point of view, this is a worst case scenario for removing noise, because we need the noise to *increase* the singular values in order for rank-minimizing optimization to distinguish between signal and noise. On the other hand, in the midpoint-offset domain (bottom row), the noise increases the resulting singular values in *both* the midpoint and offset coordinates. As a result, using this domain to complete and denoise our data is much more favourable from a tensor completion perspective.

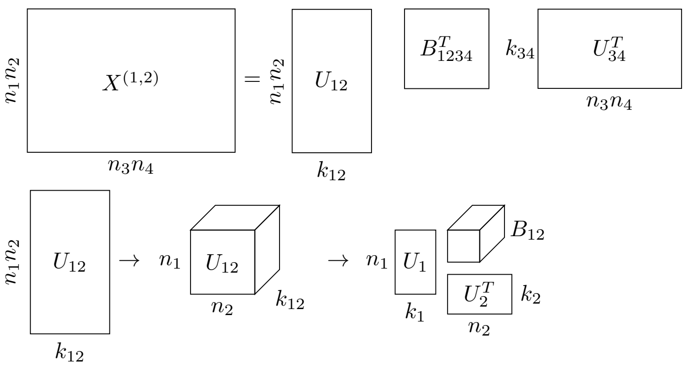


Figure 1: Hierarchical Tucker decomposition for a tensor X of size $n_1 \times n_2 \times n_3 \times n_4$

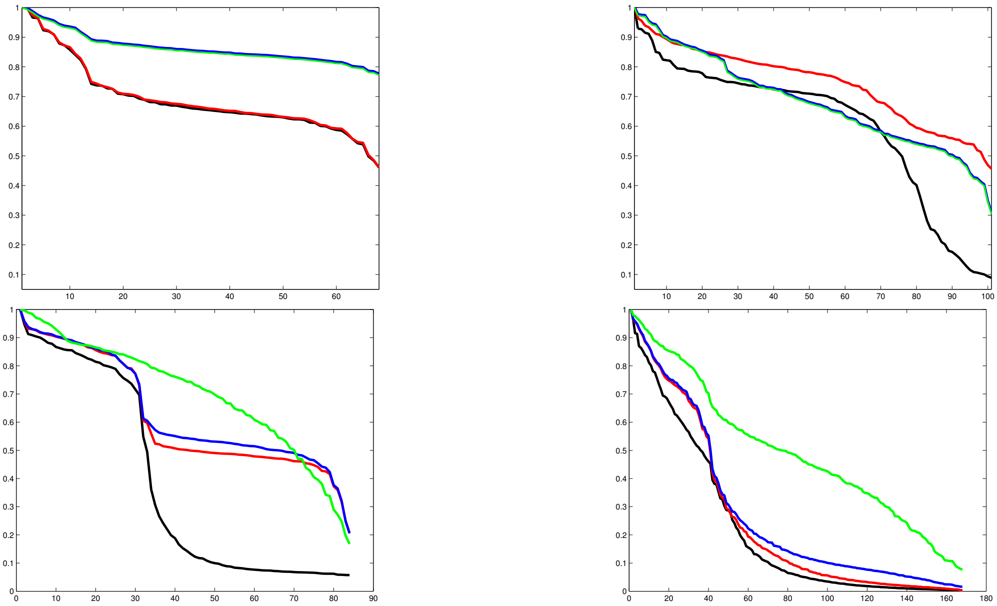


Figure 2: Normalized singular values of various matricizations - low noise.
Black: original, *Red*: subsampled, *Blue*: low noise, *Green*: high noise
Top left: x_{src}, *Top right* x_{rec}, *Bottom left*: x_{midpt}, *Bottom right*: x_{offset}

Numerical Experiments

Our data set is a single frequency slice at $7.34Hz$ from a dataset provided to us by BG Group, with 68×68 sources and 401×401 receivers downsampled to 101×101 receivers. We randomly remove 50% of the receivers and consider the two cases discussed previously: a 'low-noise' scenario, wherein 5% of the remaining receivers are replaced with Gaussian noise approximately scaled to the energy of the removed receiver and a 'high-noise' scenario, wherein the noise is scaled to the energy of the *undecimated* data. Using the techniques discussed previously, we interpolate the data both in the source-receiver (S-R) domain and the midpoint-offset (M-H) domain as shown in Figures 3 and 4. As we can see, performing our tensor completion in the appropriate domain has a *significant* effect ($\sim 4-6$ dB) on the quality of the recovered result without significantly increasing the computational overhead.

Conclusions

In this abstract, we considered the practical implications of choosing an appropriate domain for performing tensor completion and denoising. By drawing on insights from matrix and tensor completion, we have extended previously developed work on Hierarchical Tucker tensor completion to handle the noisy receiver case using an appropriate transformation e.g. Midpoint-offset. These results can be understood within the theory of matrix/tensor completion and can potentially be extended to other seismic transforms such as midpoint-offset-azimuth as well.

Acknowledgements: We would like to thank all of the sponsors of our SINBAD project for their continued support and to BG for providing the synthetic data we used in our example.

References

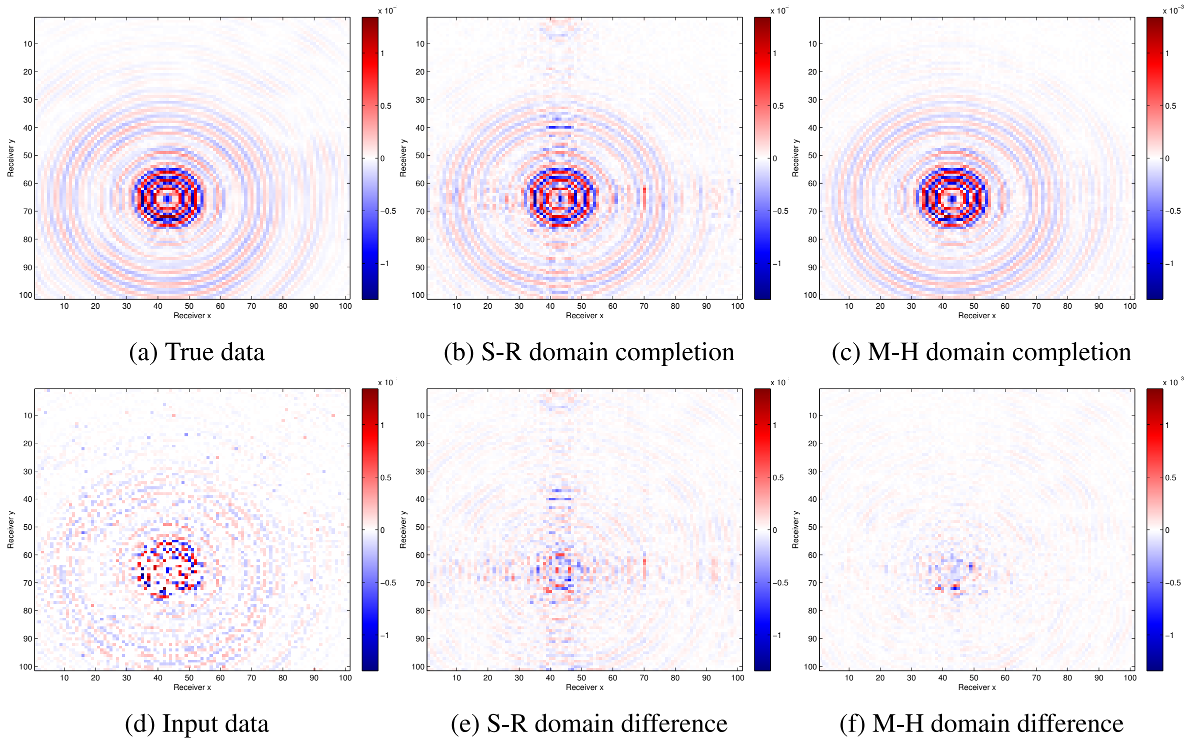
Da Silva, C. and Herrmann, F.J. [2013] Optimization on the hierarchical tucker manifold-applications to tensor completion.

Hackbusch, W. and Kühn, S. [2009] A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, **15**(5), 706–722.

Kreimer, N. and Sacchi, M. [2012] A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation. *Geophysics*, **77**(3), V113–V122.

Kreimer, N., S.A. and Sacchi, M. [2013] Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction. *GEOPHYSICS*, **78**(6), V273–V284.

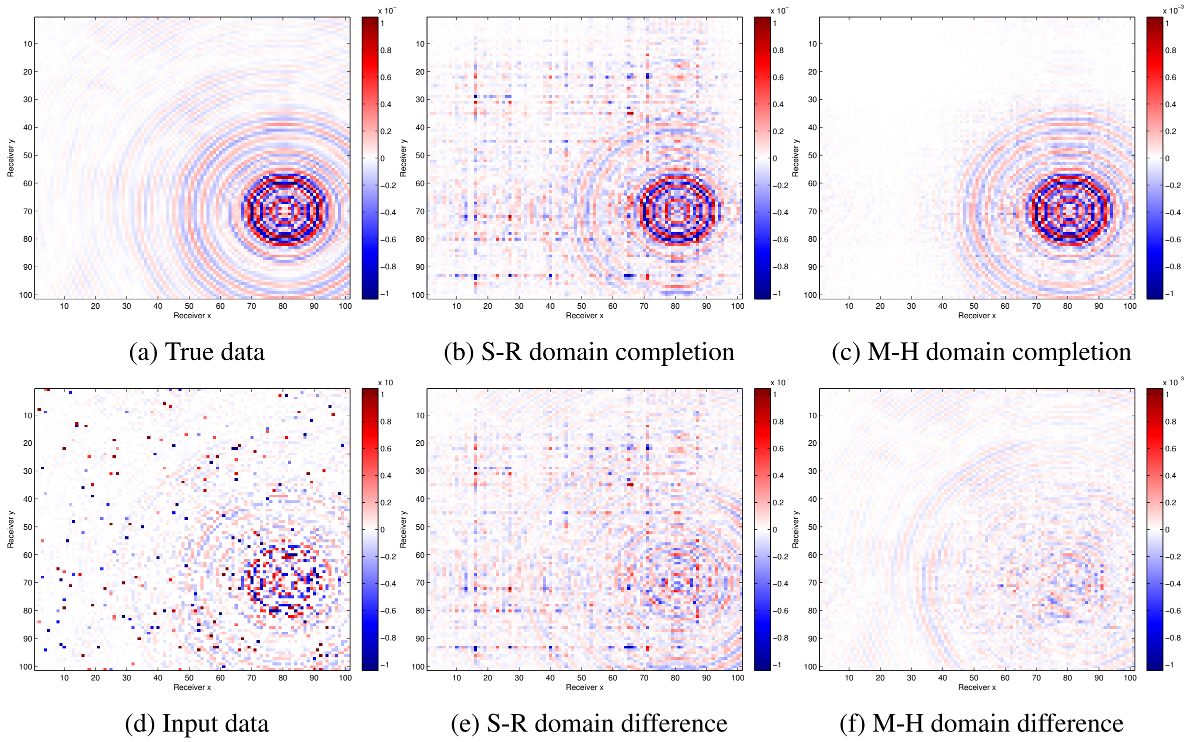
Trickett, S., Burroughs, L. and Milton, A. [2013] Interpolation using hankel tensor completion. *Ratio*, **1**(16), 16.



(a) True data (b) S-R domain completion (c) M-H domain completion

(d) Input data (e) S-R domain difference (f) M-H domain difference

Figure 3: Common source gather in the low noise scenario. S-R SNR - 7.87 dB, M-H SNR - 12.6 dB



(a) True data (b) S-R domain completion (c) M-H domain completion

(d) Input data (e) S-R domain difference (f) M-H domain difference

Figure 4: Common source gather in the high noise scenario. S-R - SNR 3.05 dB, M-H - SNR 8.95 dB