Hierarchical Tucker Tensor Optimization - Applications to 4D Seismic Data Interpolation

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

February 19, 2013

Abstract

In this work, we develop optimization algorithms on the manifold of Hierarchical Tucker (HT) tensors, an extremely efficient format for representing high-dimensional tensors exhibiting particular low-rank structure. With some minor alterations to existing theoretical developments, we develop an optimization framework based on the geometric understanding of HT tensors as a smooth manifold, a generalization of smooth curves/surfaces. Building on the existing research of solving optimization problems on smooth manifolds, we develop Steepest Descent and Conjugate Gradient methods for HT tensors. The resulting algorithms converge quickly, are immediately parallelizable, and do not require the computation of SVDs. We also extend ideas about favourable sampling conditions for missing-data recovery from the field of Matrix Completion to Tensor Completion and demonstrate how the organization of data can affect the success of recovery. As a result, if one has data with randomly missing source pairs, using these ideas, coupled with an efficient solver, one can interpolate large-scale seismic data volumes with missing sources and/or receivers by exploiting the multidimensional dependencies in the data. We are able to recover data volumes amidst extremely high subsampling ratios (in some cases greater than 75%) using this approach.

Introduction

Three dimensional seismic data acquisition is a time- and cost-intensive process, resulting in an enormous sampled data volume with five dimensions (source x, source y, receiver x, receiver y, and time). Many of the challenges associated with acquiring seismic data are inherently practical in nature: physical limitations and/or budgetary constraints limit the size and scope of data that can be acquired. In spite of these practical challenges, having fully sampled data is of paramount importance in processes such as Full Waveform Inversion and multidimensional convolution for multiple prediction and thus one is interested in interpolating the subsampled data back to the full data volume.

Luckily, even in the case of severe subsampling, multi-dimensional seismic data has a large amount of structure that can be exploited in order to recover the original, fully-sampled data. In this abstract, we use extended notions of *rank* from linear algebra in conjunction with high dimensional tensors, which will enable us to extend ideas from *matrix completion* (that is, recovering a low-rank matrix from only a fraction of its entries) to *tensor completion*. In tensor completion, we only observe a fraction of the entries of the underlying tensor (e.g. due to budgetary constraints that limit the number of seismic sources we can use) yet, by utilizing knowledge of the underlying structure of the fully sampled tensor, we are able to recover the full data volume in a computationally tractable manner. We apply this technique to low temporal-frequency seismic data, which tends to exhibit more low-rank structure than for high-frequencies. This technique is particular suitable for FWI using a frequency continuation method, where there are not too many oscillations at low-frequencies. This will lead us into the *Hierarchical Tucker format*, a relatively novel structured tensor, and our own contributions, which involve solving optimization problems in this format.

Previous work in tensor completion of seismic data includes work in Kreimer and Sacchi (2012), which uses the Tucker format and a projection onto convex sets (POCS) approach, which has to compute SVDs of the underlying tensor at each iteration, and work in Gao et al. (2011), which uses a Toeplitz matrix as a skeleton for their interpolant and exploits the relationship between FFTs and Toeplitz matrices to perform matrix-vector products efficiently, again using a POCS approach. This technique suffers greatly in terms of reconstruction error when the subsampling rate is high. Both of these drawbacks are not present in our current work and our method is much more general than interpolating Toeplitz structure. Another approach, detailed in Kumar et al. (2012), uses a rank-penalization approach to matrix completion coupled with an SPGL1-type solver to quickly interpolate frequency slices.

Methodology

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 columns, and the other dimensions reshaped into the rows of the matrix.

The Hierarchical Tucker (HT) format is a novel, structured tensor format for representing tensors which exhibit certain *low-rank* behaviour in different *matricizations*, first introduced in Hackbusch and Kühn (2009). To avoid a deluge of mathematical notation, we use the example shown in Figure 1 as the representative for the general HT format. As we can see in this figure, the main insight of the HT decomposition - as opposed to a regular, low-rank matrix formulation - is that the 'singular vectors' U_{12} of the matricization $X^{(1,2)}$ contain the dimensions 1 and 2. As such, we can reshape U_{12} into a $3D$ cube with dimensions $n_1 \times n_2 \times k_{12}$ and further split apart this cube along the dimensions 1 and 2, as in the bottom portion of Figure 1. We apply the same splitting for the matrix U_{34}.

The upshot of this construction is that the intermediate matrices in the recursive formulation (e.g. U_{12}, U_{34}) *do not* need to be stored: only the (small) leaf matrices (U_i for $i = 1, \ldots, 4$) and (small) intermediate tensors B_t (also known as *transfer* tensors) need to be stored in order to specify the (full) tensor exactly. As a result, the number of parameters that one has to store to represent HT tensors is

$\leq dNK + (d-2)K^3 + K^2$, where d is the number of dimensions, $N = \max_{i=1,\ldots,d} n_i$ is the maximum dimension size, and $K = \max_{t \in T} k_t$ is the maximum of various internal rank parameters.

Note that when $d > 3$ and $K \ll N$, the number of parameters required to represent the tensor is $\ll N^d$, the usual requirement for storing a d dimensional array of dimension sizes N, and thus the HT format effectively breaks the curse of dimensionality for this class of tensors.

By the analysis done in Uschmajew and Vandereycken (2012), we know that by parametrizing the set of all HT tensors by the matrices/tensors (U_t, B_t), the set of all such tensors forms a *differentiable manifold* in $\mathbb{R}^{n_1 n_2 \ldots n_d}$. Differentiable manifolds are higher-dimensional analogues to curves and surfaces in $\mathbb{R}^2$ and $\mathbb{R}^3$ and there is a large body of existing research on formulating optimization algorithms on differentiable manifolds. For computational purposes, one needs to restrict the parameter matrices (not at the root) to satisfy orthogonality conditions. In our case, if we write $\phi(x)$ 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

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

The Jacobian and Jacobian transpose of $\phi(x)$ can be computed extremely efficiently by exploiting the recursive structure of $\phi(x)$ as well as formulae for matrix derivatives. By modifying the theory developed in Uschmajew and Vandereycken (2012), we can formulate algorithms such as Steepest Descent and Conjugate Gradient on this (Riemannian) manifold, and, as a result, we are able to obtain reasonable interpolants for even high levels of source subsampling. The end result is an algorithm that is SVD-free, is immediately parallelizable in a distributed environment, and is able to recover tensors exhibiting this particular low-rank structure despite very high levels of subsampling.

Higher Dimensional Sampling

We can draw insights about our approach from a large body of literature in the Matrix Completion field, wherein we are trying to recover a low-rank matrix from a partial set of observations of its entries. The low-rank approach to matrix completion tells us that randomly removing entries tends to increase the singular values of the underlying matrix and thus makes recovery more favourable for a rank-minimization optimization scheme. On the other hand, removing entire columns/rows from the matrix zeros-out the smallest singular values of the matrix, and thus rank-minimization will fail to recover the subspace spanned by those columns/rows (e.g. see Candès and Recht (2009)).

In our case of a 4D frequency slice, we have essentially two choices of underlying matricization, both of which are shown in Figure 2. Namely, we can choose between placing the (src x, src y) dimensions in the rows and (rec x, rec y) dimensions in the columns, or placing the (src x, rec x) dimensions in the rows and (src y, rec y) dimensions in the columns. In the case when we are, say, missing receivers, the

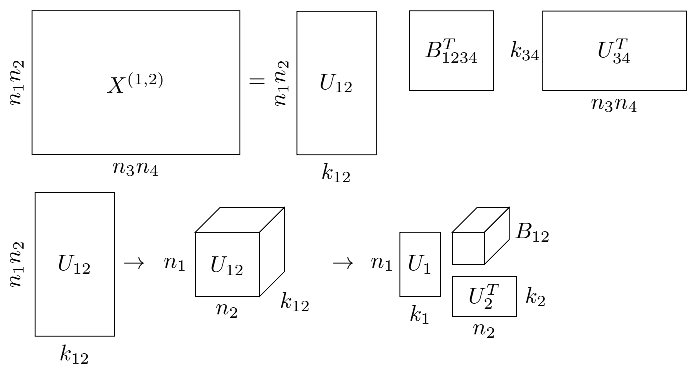


Figure 1: Hierarchical Tucker decomposition for a tensor X of size $n_1 \times n_2 \times n_3 \times n_4$

former organization of data has the effect that subsampling will tend to remove columns of this matrix, and hence the singular values will not increase and in fact *are set to zero* at the low end (the worst-case scenario for the purposes of rank-minimizing recovery). On the other hand, the latter organization of data results in a subsampling that randomly removes blocks from the underlying matrix, which is a much more favourable situation from a low-rank recovery perspective, as we can see from the singular values of the resulting matrix. The same situation holds for matricizations in the singleton dimensions, adding further degrees of regularity to the computed solution compared to standard matrix completion.

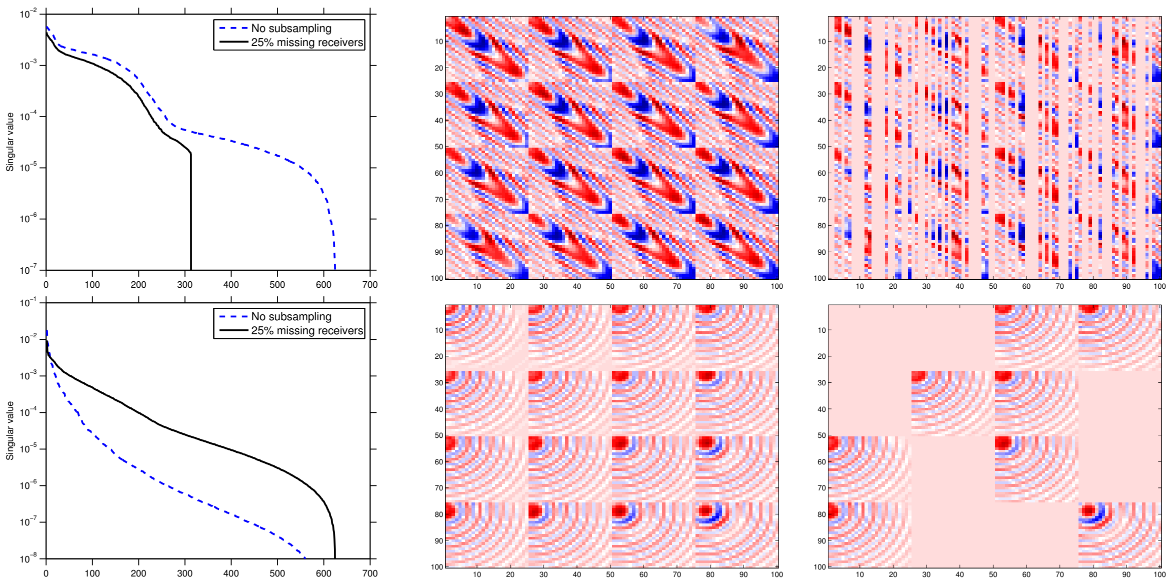
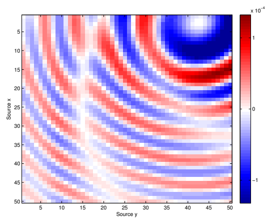


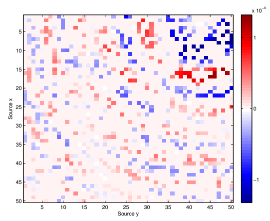
Figure 2: *Top*: (Src x, Src y) Matricization *Bottom*: (Src x, Rec x) Matricization
Left: Singular values *Middle*: Fully sampled data *Right*: Subsampled data

Examples

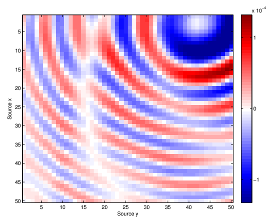
We apply the aforementioned techniques in order to recover a single frequency slice of data generated from a single-reflector model. The data cube D is of size $50 \times 50 \times 50 \times 50$ (src x $\times$ src y $\times$ rec x $\times$ rec y) and we run a non-linear CG algorithm for solving Problem 1 with 75% source subsampling for 200 iterations. Below are merely a tiny sample of the possible common receiver gathers one can extract from the interpolated data, but they are representative of the recovered data volume as a whole. Even amidst many missing sources, as we can see in Figures 3 and 4, the multidimensional redundancies in the data coupled with an efficient CG solver allow one to recover the data to a high accuracy.



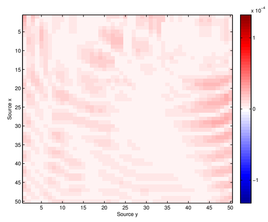
(a) Original data slice



(b) Subsampled data



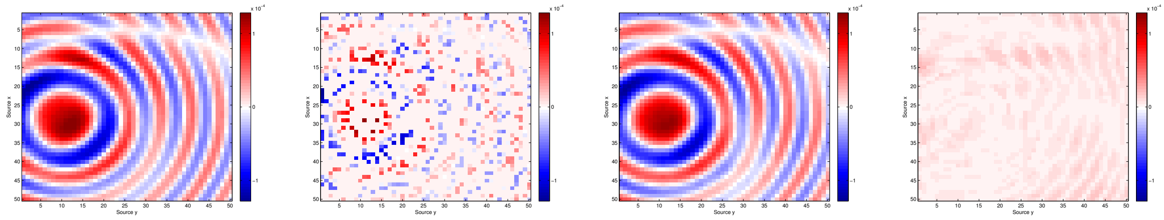
(c) Recovered data



(d) Difference

Figure 3: 75% Source Subsampling - (Rec x, Rec y) = (5, 45) - SNR 15.4 dB

We also include results from interpolating a synthetic data set provided to us by BG. The data set consists of only 200 randomly placed shots (out of a possible 4624 shots), each having 401 x 401 receivers and is generated from an unknown model. We extract a single frequency slice from this data and attempt to recover the full slice using the aforementioned method. Owing to the low-frequency content of the slice, the data is subsampled to 101 x 101 receivers and the data volume is recovered to a 68 x 68 x 101 x 101 tensor. Each figure is then produced by Fourier interpolating each common shot gather to 401 x 401 receivers. We can see in Figure 5 that the interpolated data matches the known data to a sufficiently accurate degree. More importantly, the interpolated results in Figure 5 look very reasonable, given the structure of the other shot gathers in this frequency slice. Unfortunately, we do not have a reference solution for these shot gathers in order to perform quantitative comparisons.



(a) Original data slice (b) Subsampled data (c) Recovered data (d) Difference

Figure 4: 75% Source Subsampling - (Rec x, Rec y) = (30,12) - SNR 17.4 dB

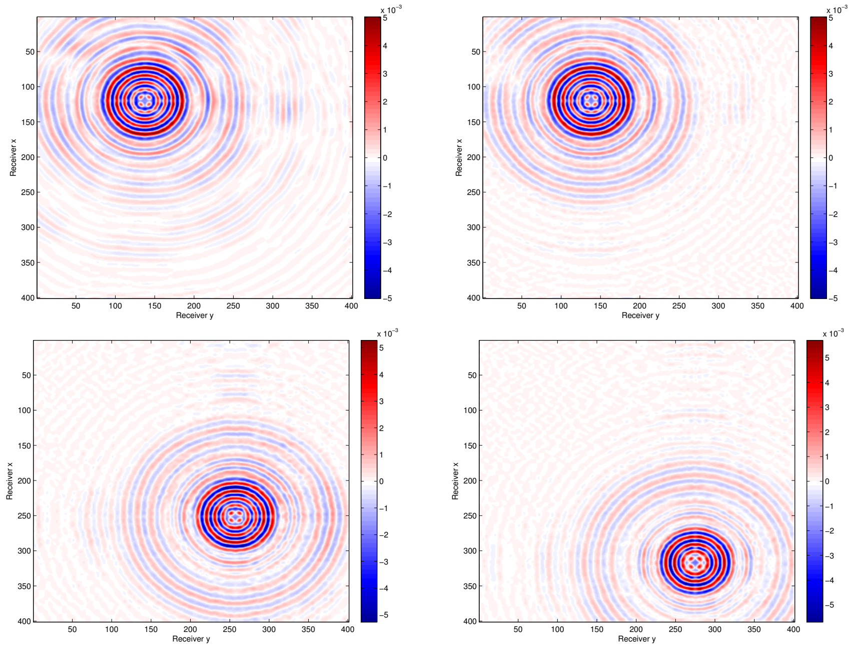


Figure 5: *Top Left*: Original Data *Top Right*: Interpolated Data - SNR 12.5dB *Bottom*: Interpolated Data

Conclusion

Owing to the novelty of this format, as well as the mathematical and optimization details that have to be addressed in order to perform optimization, and hence interpolation, we have only detailed some of the specifics here. We have managed to extend many of the largely theoretical ideas in Uschmajew and Vandereycken (2012) into a computationally practical approach to interpolate 4D seismic data volumes based on extensions to previous work in optimization on differentiable manifolds.

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

Candès, E. and Recht, B. [2009] Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, **9**(6), 717–772.

Gao, J., Sacchi, M. and Chen, X. [2011] A fast rank reduction method for the reconstruction of 5d seismic volumes. *2011 SEG Annual Meeting*.

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.

Kumar, R., Aravkin, A.Y. and Herrmann, F.J. [2012] Fast methods for rank minimization with applications to seismic-data interpolation. Tech. Rep. TR-2012-04, University of British Columbia, Vancouver.

Uschmajew, A. and Vandereycken, B. [2012] The geometry of algorithms using hierarchical tensors. *preprint, http://sma.epfl.ch/~vanderey/papers/geom_htucker.pdf*.