
Asymmetric Multiresolution Matrix Factorization

Pramod Kaushik Mudrakarta
University of Chicago

Shubhendu Trivedi
CSAIL, MIT

Risi Kondor
Flatiron Institute, NY
University of Chicago

Abstract

Multiresolution Matrix Factorization (MMF) was recently introduced as an alternative to the dominant low-rank paradigm in order to capture structure in matrices at multiple different scales. Using ideas from multiresolution analysis (MRA), MMF teased out hierarchical structure in symmetric matrices by constructing a sequence of wavelet bases. While effective for such matrices, there is plenty of data that is more naturally represented as nonsymmetric matrices (e.g. directed graphs), but nevertheless has similar hierarchical structure. In this paper, we explore techniques for extending MMF to any square matrix. We validate our approach on numerous matrix compression tasks, demonstrating its efficacy compared to low-rank methods. Moreover, we also show that a combined low-rank and MMF approach, which amounts to removing a small global-scale component of the matrix and then extracting hierarchical structure from the residual, is even more effective than each of the two complementary methods for matrix compression.

1 Introduction

A wide variety of naturally occurring data, represented as large matrices, often has complex hierarchical structure. For instance, this is the case in data originating from social networks, complex biological interactions, circuit simulations and so on. With the increasing size of such matrices, there is a need to represent and compress them efficiently. Towards this end, a number of

methods have been proposed (Mahoney and Drineas, 2009; Coifman and Maggioni, 2006; Hackbusch, 1999; Halko et al., 2009). All such methods make an implicit or explicit assumption about the nature of such matrices, which serves as a prior to facilitate efficient factorization and compression. Frequently, the assumption in question is that the matrices are of low rank. For example, Principal Component Analysis (PCA) decomposes a symmetric matrix A into the form

$$A = Q^T D Q$$

where Q is a orthogonal matrix and D is diagonal. Notably, such methods are single-level algorithms. Moreover, the columns of Q are usually dense, which does not adequately capture strong locality properties that might be exhibited, for instance, in real world graphs.

Taking a different tack, Multiresolution Matrix Factorization (MMF) (Kondor et al., 2014) framed matrix factorization as a type of multiresolution analysis (MRA) over a discrete metric space whose metric structure is represented by a self-adjoint operator. Thus, in contrast to low-rank approaches, MMF factorizes a symmetric matrix A in the following multi-level manner:

$$A = Q_1^T Q_2^T \dots Q_L^T H Q_L \dots Q_2 Q_1$$

where each of the carefully selected Q_i matrices are orthogonal and extremely sparse, and the matrix H is close to diagonal. MMF is able to discover soft hierarchical arrangements of localized structures in the data represented by the matrix. It is amenable to parallelization and fast implementations (Kondor et al., 2015). The wavelet bases uncovered by MMF also provide a natural sparse approximation for functions defined over the space. For instance, if the matrix A is the normalized graph Laplacian, then the factorization provides us with wavelet bases for approximating functions on graphs.

While MMF has been shown to be useful in various practical settings (Teneva et al., 2016; Ding et al., 2017; Ithapu, 2017), one notable drawback of the method is that it is wholly defined using a self-adjoint

operator such as the graph Laplacian. Therefore, MMF can only handle a multi-level factorization of symmetric matrices. However, many naturally occurring data are represented more concisely as a nonsymmetric matrix; the underlying space might only have a quasi-metric structure. This is the case for instance, when we are working with directed graphs. For data living on such spaces, hierarchical or multiscale structure is as important, albeit more complicated, and thus there is much to be gained by defining appropriate sparse wavelet bases for function approximation.

Defining such wavelet bases, in particular for functions on directed graphs, is an active area of research (Sevi et al., 2018; Furutani et al., 2019; Mhaskar, 2018). However, there is currently no consensus on the correct approach for doing so. Further, existing methods are difficult to implement in practice for downstream applications. In this paper, we do not make an attempt to directly extend multiresolution analysis (MRA) in the spirit of MMF to such spaces, but rather provide practical methods to leverage intuitions used in MMF to make it operable on general square matrices. We show experimentally that our methods can perform very well in various real world matrix compression tasks.

To conclude this section, our contributions in this paper are twofold: 1) we provide two practical ways to adapt MMF to nonsymmetric matrices, 2) we argue that extending MRA to nonsymmetric matrices is not analogous to developing a multi-level SVD; in particular, we show experimentally that it is important to consider interactions between different levels in the hierarchical structure and lastly 3) we also show that a combined low-rank and MMF approach, which amounts to removing a small global-scale component of the matrix and then extracting hierarchical structure from the residual, is even more effective than each of the two complementary methods for matrix compression.

After a description of notation used in this paper, in Section 3, we give a brief introduction to Multiresolution Matrix Factorization.

2 Notation

We denote $[n] = 1, 2, \dots, n$. Inner products between vectors are denoted by the operator $\langle \cdot, \cdot \rangle$. The (i, j) -th entry of a matrix A is denoted by $A_{i,j}$. If $S \subseteq [n]$ is a set of indices, $A_{S,S}$ represents the submatrix of A corresponding to those indices. \bar{S} denotes the complement of a set. When S can be interpreted as an ordered set, for instance, if it is a set of indices, then $S[i]$ represents the i -th index from that set.

3 Multiresolution Matrix Factorization

Given a measurable space X and square integrable functions $L_2(X)$ on X , multiresolution analysis (MRA) constructs a sequence of spaces of functions of increasing smoothness:

$$\begin{array}{ccccccccccc} L_2(X) = V_0 & \rightarrow & V_1 & \rightarrow & V_2 & \rightarrow & \dots & \rightarrow & V_L \\ & & \searrow & & \searrow & & \searrow & & \searrow & & \\ & & & & W_1 & & W_2 & & \dots & & W_L \end{array}$$

where each V_ℓ is split into a smoother space $V_{\ell+1}$ and a rougher space $W_{\ell+1}$, such that $V_\ell = V_{\ell+1} \oplus W_{\ell+1}$. The length-scale over which a function $f \in V_\ell$ varies increases with ℓ and thus makes this decomposition an analysis of $L_2(X)$ multiscale/multiresolution.

Kondor et al. (2014) extends MRA to functions on discrete spaces (e.g. undirected graphs) by interpreting a given symmetric matrix as a discrete version of a continuous self-adjoint operator. Specifically, for $|X| = n$, Kondor et al. (2014) defines multiresolution analysis with respect to a symmetric matrix $A \in \mathbb{R}^{n \times n}$ as a successive filtration

$$V_L \subset \dots \subset V_1 \subset V_0 = L(X) \cong \mathbb{R}^n \quad (1)$$

where each V_ℓ has an orthonormal basis $\Phi_\ell := \{\phi_m^\ell\}$, and its complementary space W_ℓ has an orthonormal basis $\Psi_\ell := \{\psi_m^\ell\}$ that satisfy the following three axioms:

MRA1 The sequence equation (1) is a filtration of \mathbb{R}^n in terms of smoothness. In particular,

$$\eta_\ell = \sup_{v \in V_\ell} \frac{\langle v, Av \rangle}{\langle v, v \rangle}$$

decays with ℓ .

MRA2 The basis vectors of W_ℓ are localized in the sense that

$$\mu_\ell = \max_{m \in \{1, \dots, d_\ell\}} \|\psi_m^\ell\|_0$$

increases no faster than a certain rate.

MRA3 Suppose the matrix U_ℓ represents the matrix expressing $\Psi_\ell \cup \Phi_\ell$ in terms of the previous basis $\Phi_{\ell-1}$. Then, U_ℓ is sparse. (Φ_0 is taken to be the standard basis.)

Once an overcomplete multiscale basis satisfying MRA1-3 for \mathbb{R}^n , i.e., $U_\ell, \ell = 1, \dots, L$, is determined, it can be used to compress the matrix A . Applying

each of the U_ℓ 's sequentially, we obtain versions of A compressed to various scales:

$$A \mapsto \underbrace{U_1 A U_1^T}_{A_1} \mapsto \dots \mapsto \underbrace{U_L \dots U_2 U_1 A U_1^T U_2^T \dots U_L^T}_{A_L=H}$$

Since the choice of the U_ℓ 's aligns with MRA1, Kondor et al. (2014) argues that the matrices A_1, \dots, A_L and H are approximately in core-diagonal form. That is, up to a permutation, each of those matrices has a block-diagonal form, with one block being a diagonal matrix.

Formally, Multiresolution Matrix Factorization is defined as an approximate multilevel factorization of the form

$$A \approx Q_1^T Q_2^T \dots Q_L^T H Q_L \dots Q_2 Q_1, \quad (2)$$

where the matrices Q_1, \dots, Q_L and H obey the following conditions:

1. Each $Q_\ell \in \mathbb{R}^{n \times n}$ is orthogonal and highly sparse. In the simplest case, each Q_ℓ is a Givens rotation, i.e., a matrix which differs from the identity in just the four matrix elements

$$\begin{aligned} [Q_\ell]_{i,i} &= \cos \theta, & [Q_\ell]_{i,j} &= -\sin \theta, \\ [Q_\ell]_{j,i} &= \sin \theta, & [Q_\ell]_{j,j} &= \cos \theta, \end{aligned}$$

for some pair of indices (i, j) and rotation angle θ . Multiplying a vector with such a matrix rotates it counter-clockwise by θ in the (i, j) plane. More generally, Q_ℓ is a so-called k -point rotation, which rotates not just two, but k coordinates.

2. Typically, in MMF factorizations $L = O(n)$, and the size of the active part of the Q_ℓ matrices decreases according to a set schedule $n = \delta_0 \geq \delta_1 \geq \dots \geq \delta_L$. More precisely, there is a nested sequence of sets $[n] = S_0 \supseteq S_1 \supseteq \dots \supseteq S_L$ such that the $[Q_\ell]_{\overline{S_{\ell-1}}, \overline{S_{\ell-1}}}$ part of each rotation is the $n - \delta_{\ell-1}$ dimensional identity. S_ℓ is called the **active set** at level ℓ . In the simplest case, $\delta_\ell = n - \ell$.
3. H is an S_L -core-diagonal matrix, which means that it is block diagonal with two blocks: H_{S_L, S_L} , called the core, which is dense, and $H_{\overline{S_L}, \overline{S_L}}$ which is diagonal. In other words, $H_{i,j} = 0$ unless $i, j \in S_L$ or $i = j$.

Given a symmetric matrix, the factorization is carried out by minimizing the Frobenius norm error between the original matrix and its factorized form. Algorithms such as GREEDYJACOBI (Kondor et al., 2014) or pMMF (Teneva et al., 2016) are typically used in

practical applications (Teneva et al., 2016; Ding et al., 2017; Mudrakarta and Kondor, 2017)

Having discussed the core ideas behind MMF, we briefly review some related art before moving forward.

4 Related Work

The work most closely related to some of the ideas in this paper is that on defining wavelets on functions on directed graphs. Two notable recent works on the theme are that of Sevi et al. (2018) and Mhaskar (2018). Sevi et al. (2018) did so by using a random walk operator, while Mhaskar (2018) used an extension of diffusion polynomial frames (Mhaskar and Maggioni, 2008), using a different operator. In fact, the influential work on diffusion wavelets (Coifman and Maggioni, 2006), provides a more general theory in which symmetry of the diffusion operator is not necessary. However, it is unclear how to lift these ideas for defining a principled asymmetric MMF that has also a clear MRA interpretation. On the side of matrix factorization, works that have some resemblance to our approach are Li et al. (2015); Li and Yang (2017), in which the left and right matrices don't have to be related by transposition. However these approaches don't have a clear wavelet based motivation.

5 MMF on nonsymmetric matrices

Based on the MMF approach described in Section 3, in the sequel we provide two approaches that extend MMF to general square matrices.

5.1 MMF via an additive decomposition

We begin by observing that any square matrix $A \in \mathbb{R}^{n \times n}$ can be written as the sum of a symmetric matrix and a skew-symmetric matrix as follows

$$A = \underbrace{\frac{A + A^T}{2}}_{\text{symmetric}} + \underbrace{\frac{A - A^T}{2}}_{\text{skew-symmetric}} \quad (3)$$

We propose performing MRA on each term separately. Since the first term is symmetric, it directly yields to a multiresolution matrix factorization. In particular, the factorization of $\frac{A+A^T}{2}$ can be obtained via a parallel variant of MMF such as pMMF (Kondor et al., 2015). Thus, the problem of computing a multiresolution factorization of an asymmetric square matrix is now reduced to computing a multiresolution factorization for a skew-symmetric matrix.

5.1.1 MMF for skew-symmetric matrices

For skew-symmetric matrices, the MRA axioms, especially MRA1, are no longer applicable. The reason being that under the definition of smoothness in MRA1 $\sup_{v \in V \subseteq \mathbb{R}^n} \frac{\langle v, Av \rangle}{\langle v, v \rangle}$, for a skew-symmetric matrix A , the smoothness is always zero.

In symmetric MMF, the choice of the sparse orthogonal factors and the diagonal part of the core-diagonal matrix H are based on the assumption that the multi-scale basis vectors determined by the MMF algorithm are as close to eigenvectors as possible. In other words, the filtration defined in equation (1) is a sequence of subspaces that is as close to a spectral filtration as possible. Indeed, MMF approximately diagonalizes a symmetric matrix using $O(n)$ rotations, where n is the dimension of the matrix. We consider this intuition as our starting off point to extend MMF to skew-symmetric matrices.

Note that a matrix $A \in \mathbb{R}^{n \times n}$ is skew-symmetric if $A^T = -A$. Skew-symmetric matrices have complex eigenvalues and are therefore not diagonalizable in real space. Thus, a skew-symmetric matrix can never be perfectly represented using the same form of MMF as for a symmetric matrix over reals. In order to overcome this issue, we consider a well known observation (Murnaghan and Wintner, 1931) that a skew-symmetric matrix, under a unitary transformation, assumes the so-called Murnaghan canonical form,

$$\begin{bmatrix} \Lambda_1 & 0 & \dots & 0 \\ 0 & \Lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \Lambda_L \end{bmatrix} \quad (4)$$

where $\Lambda_i, i = 1, \dots, L$ are 2×2 matrices of the form

$$\Lambda_i = \begin{pmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{pmatrix}$$

where $\lambda_i \in \mathbb{R}$.

This observation affords a multiresolution factorization for skew-symmetric matrices identical to that for symmetric matrices with the difference that the core-diagonal matrix is allowed to have the following form:

$$H = \begin{bmatrix} H_{\text{core}} & 0 & \dots & \dots & 0 \\ 0 & \Lambda_1 & 0 & \dots & 0 \\ 0 & 0 & \Lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \Lambda_L \end{bmatrix} \quad (5)$$

where $\Lambda_i, i = 1, \dots, L$ are 2×2 matrices as defined above.

A couple of observations based on this form are in order. First, notice that the skew-symmetry of the matrix in its MMF is always preserved. Second, this form also allows for interaction between wavelets at adjacent levels. This feature is distinct from the original MMF procedure in which no such interactions occur. To obtain the factorization, the procedure is similar to the GREEDYJACOBI algorithm from Kondor et al. (2014), with the difference that in the last step, instead of zeroing out off-diagonal elements, we sparsify the matrix into the Murnaghan normal form as described above.

Thus, the multiresolution decomposition of a square matrix can be done by factorizing each of the two parts from equation (3) separately and writing A as a sum of two MMFs. In other words, we may write

$$A \approx P_1^T \dots P_L^T H_{\text{sym}} P_L \dots P_1 + Q_1^T \dots Q_L^T H_{\text{skew}} Q_L \dots Q_1$$

where

$$\frac{A + A^T}{2} \approx P_1^T \dots P_L^T H_{\text{sym}} P_L \dots P_1 \quad (6)$$

$$\frac{A - A^T}{2} \approx Q_1^T \dots Q_L^T H_{\text{skew}} Q_L \dots Q_1 \quad (7)$$

where H_{skew} has the form defined in equation (5).

In the next section, we present another approach for multiresolution factorization.

5.2 MMF via direct factorization

In this approach, we aim to capture hierarchical structure by operating on the square matrix directly rather than decomposing it into an additive form. Recall that we would like to identify clusters of coordinates that a matrix A more closely couples than others, and determine an appropriate multiresolution analysis of \mathbb{R}^n . When the matrix is symmetric, the linear transformation that it represents is self-adjoint. In the asymmetric case, depending on whether we are transforming $x \rightarrow x^T A$ or $x \rightarrow Ax$, we may end up with different ways in which A might couple the coordinates.

We consider two different ways of performing multiresolution analysis of \mathbb{R}^n , one that is based on A , and another on A^T . Suppose $P \in \mathbb{R}^{n \times n}$ represents the wavelet transform in the former case, and $Q \in \mathbb{R}^{n \times n}$ of the latter. Thus, we have

$$H = P^T A Q$$

which is the matrix A in the bases P and Q . In the symmetric case, H takes a core-diagonal form, based

on the fact that the eigenvalue decomposition converts a symmetric matrix to diagonal, and MMF uses orthonormal transformations that are designed to be as close to eigenvectors as possible. In the asymmetric case, we relax this assumption and allow some of the off-diagonal elements of H to be nonzero. This is motivated by the fact that the frequencies or smoothness coefficients filtered by the left and right hand sides may not align with each other.

Remark. When A represents the adjacency matrix of a directed graph, we are essentially identifying hierarchical cluster structures considering 1) only the outgoing edges, and 2) only the incoming edges. The matrix H gives a mapping between these two cluster structures.

We define asymmetric MMF to be a multiresolution factorization of the following form

$$A = P_1 P_2 \dots P_L H Q_L^T Q_{L-1}^T \dots Q_1^T \quad (8)$$

, where

- Each $P_i, Q_i, i = 1, \dots, L$ are highly sparse, orthogonal rotation matrices similar to the rotations in a symmetric MMF
- $H \in \mathbb{R}^{n \times n}$ is a core-sparse matrix, that is, under an appropriate permutation of row and columns, H would be a 2×2 block-diagonal matrix with the upper left block being dense, and the other 3 blocks would be highly sparse
- When $P_i = Q_i$ and H is in core-diagonal form with a symmetric core, the above factorization would be a symmetric MMF.

The computation of the asymmetric MMF is obtained using an approach similar to the symmetric MMF. The algorithm we propose is designed to minimize the Frobenius norm error

$$\|A - P_1 P_2 \dots P_L H Q_L^T Q_{L-1}^T \dots Q_1^T\|_F$$

We determine the left and right rotations in a greedy manner. That is, we compute a sequence of rotated version of A

$$A \rightarrow P_1^T A Q_1 \rightarrow \dots \rightarrow P_L^T \dots P_1^T A Q_1 \dots Q_L$$

Each of the rotations is computed by identifying sets of similar rows (for the left rotations) and columns (for the right rotations), and determining an appropriate k -point rotation. This requires us to compute Gram matrices on both the rows as well as the columns. Finally, we sparsify the final rotated version of A to determine the core-sparse matrix H . We explore three ways of sparsification

1. COREDIAGONAL: retain only the diagonal elements of H
2. TOPN: retain the top n elements in terms of magnitude
3. GREEDYTOPN: select n elements by magnitude greedily such that no two of the selected elements fall in the same row/column

The full procedure is summarized in Algorithm 1.

The running time of this algorithm is $O(n^3)$, similar to GREEDYJACOBI routine used in the original symmetric MMF. Nevertheless, computation can be made faster by similarly extending the parallel MMF algorithm to maintain row and column Gram matrices and computing separate left and right rotations. It must be noted that applying the left rotations to A does not affect the column Gram matrix, and applying the right rotations does not affect the row Gram matrix.

Algorithm 1 Asymmetric MMF via direct factorization

Input: core size d , matrix $A_0 = A \in \mathbb{R}^{n \times n}$

Set of active rows $S_0^r \leftarrow [n]$

Set of active columns $S_0^c \leftarrow [n]$

for $\ell = 1$ **to** $n - d$ **do**

 Select a random row i from the set S_ℓ^r

$G \leftarrow A_{S_\ell^r, S_\ell^c} A_{i, S_\ell^c}^T$

$j \leftarrow S_\ell^c[\arg \max(G)]$

$P_\ell \leftarrow$ Jacobi rotation on rows i, j

$A \leftarrow P_\ell^T A$

 Remove $\arg \min_{t \in \{i, j\}} \|A_{t, S_\ell^c}\|_2$ from S_ℓ^r

 Select a random column i' from the set S_ℓ^c

$G' \leftarrow A_{S_\ell^r, S_\ell^c}^T A_{S_\ell^r, i'}$

$j' \leftarrow S_\ell^c[\arg \max(G')]$

$Q_\ell \leftarrow$ Jacobi rotation on rows i', j'

$A \leftarrow Q_\ell$

 Remove $\arg \min_{t \in \{i', j'\}} \|A_{S_\ell^r, t}\|_2$ from S_ℓ^c

end for

$H =$ COREDIAGONAL(A) or TOPN(A) or GREEDYTOPN(A)

return $P_1, \dots, P_{n-d}, H, Q_1, \dots, Q_{n-d}$

5.3 Numerical experiments

Similar to the evaluation of MMF in Teneva et al. (2016), we compare our asymmetric MMF formulations with the low-rank-based CUR decomposition (Mahoney and Drineas, 2009) for the task of compressing square matrices. The CUR decomposition is a widely used alternative to the singular value decomposition to compress large matrices. To reduce a matrix $A \in \mathbb{R}^{n \times n}$ to rank- k , the method proceeds by carefully

selecting k rows ($R \in \mathbb{R}^{n \times n}$) and columns ($C \in \mathbb{R}^{k \times n}$) and determines a matrix $U \in \mathbb{R}^{k \times k}$ such that the product CUR is as close to the best rank- k approximation of A as possible.

To set up a comparison, we sample 70 square matrices from the UFlorida sparse matrix repository (Davis and Hu, 2011). The matrices in this repository were sourced from various fields of science and engineering and contain data arising from circuit simulations, social networks, chemical processes, 2D/3D problems, etc. All the sampled matrices were chosen such that the numerical symmetry was less than 25%, and had a maximum dimensionality of 100,000.

5.3.1 Results on MMF via an additive decomposition

We first present results comparing the asymmetric MMF obtained by the additive decomposition to CUR. Table 1 shows the percentage of times the asymmetric MMF had a better compression error over CUR over 65 matrices. Since both the CUR and MMF algorithms are randomized, we report results averaged over 3 trials. CUR performs better than MMF in the majority of the cases.

5.3.2 Results on MMF via direct factorization

Next, we report results on the direct factorization variant of the asymmetric MMF. In particular, we consider two different versions of the method based on what elements are retained in the core-sparse matrix H . In the first case, we retain the compressed $k \times k$ dense core, and $n - k$ elements chosen greedily according to magnitude and made sure that no two of these $n - k$ elements fall in the same row or column.

Table 2 shows the percentage times this version of the asymmetric MMF wins over CUR in compression error. We observe that this version of MMF is significantly better than the additive version, and is able to capture structure in both problems with underlying known geometry and without underlying geometry.

Relaxing the structure of the matrix H further, we report the results for simply storing the top $n - k$ elements in the core-sparse matrix in Table 3. Here, we see that MMF is able to capture structure and compress nearly all kinds of matrices at all compression levels.

From the numerical results, it seems that allowing interaction between wavelet levels is critical to achieving good approximation errors. This indicates that hierarchical structure in nonsymmetric matrices is more complex than that uncovered by simply extending the

symmetric MMF analogous to an SVD.

For smaller compression ratios, CUR seems to be preferable for some types of problems such as directed graphs. It is possible that some matrices may have global-scale behavior that is superimposed over the hierarchical structure. This is possible, for instance, when the data is noisy. In the next part, we explore ways of combining MMF and low-rank approaches to achieve even better compression errors.

5.3.3 Matrix compression via a combination of CUR and MMF

In order to better motivate this section, we first consider Figure 1, which shows the error of MMF approximation of a matrix A as a function of the decay rate of its singular values.

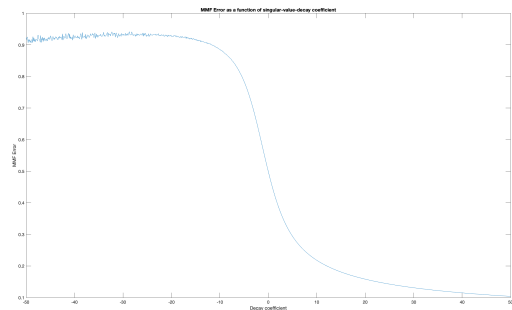


Figure 1: Approximation error of symmetric MMF with different rates of decay in the eigenvalues. The 200×200 random matrices were generated by computing QDQ^T , where $Q \in \mathbb{R}^{200 \times 200}$ is a fixed, randomly generated orthogonal matrix. D is a diagonal matrix whose entries are given by $\frac{1 - e^{t(x-1)}}{1 - e^t}$, where t is the decay coefficient and x takes 200 uniformly spaced values between 0 and 1. The X-axis corresponds to t .

We observe that when the singular values decay at a super-linear rate, MMF does not approximate the matrix well. However, in many real-world matrices, the decay of singular values may not be uniform and may fluctuate to very high and low values. An example of such a spectrum is shown in Figure 2. In this case we see that although there are segments where the decay rate is sub-linear, the large decay rate around singular value 1500 may significantly impact the effectiveness of MMF on this matrix. Thus, it might be well motivated to first compress the matrix using low-rank methods and then use MMF on the residual. We describe this technique in Algorithm 2.

For various values of k , we report the errors for compressing various matrices to 5% of their original size in Figure 3.

Matrix kind	num. matrices	1%	10%	25%	50%	75%
<i>problems with underlying 2D/3D geometry</i>						
structural problem	4	50.0	50.0	50.0	0.0	0.0
computational fluid dynamics problem	3	0.0	0.0	0.0	0.0	0.0
2D/3D problem	5	80.0	80.0	60.0	60.0	60.0
materials problem	2	0.0	0.0	0.0	0.0	0.0
<i>problems with no underlying geometry</i>						
economic problem	11	18.2	9.1	9.1	9.1	9.1
directed graph	19	5.3	5.3	5.3	0.0	0.0
power network problem	3	0.0	0.0	0.0	0.0	0.0
circuit simulation problem	2	0.0	0.0	0.0	0.0	0.0
chemical process simulation problem	23	0.0	0.0	8.7	0.0	0.0
statistical/mathematical problem	1	100.0	0.0	0.0	0.0	0.0
counter-example problem	1	0.0	0.0	0.0	0.0	0.0
total wins:		13.5	10.8	12.2	5.4	5.4

Table 1: Matrix compression using asymmetric MMF (additive). Each entry of the rightmost 5 columns is the percentage of matrices on which MMF performed better than CUR, as measured by Frobenius norm compression error. Each column corresponds to the percentage size (w.r.t original size) to which a matrix is compressed.

Matrix kind	num. matrices	1%	10%	25%	50%	75%
<i>problems with underlying 2D/3D geometry</i>						
structural problem	4	50.0	50.0	25.0	0.0	0.0
materials problem	1	0.0	0.0	0.0	0.0	0.0
2D/3D problem	4	100.0	75.0	75.0	75.0	0.0
computational fluid dynamics problem	3	100.0	100.0	0.0	0.0	0.0
<i>problems with no underlying geometry</i>						
statistical/mathematical problem	1	100.0	100.0	100.0	0.0	0.0
economic problem	9	77.8	22.2	11.1	33.3	11.1
power network problem	2	100.0	0.0	50.0	0.0	0.0
directed graph	15	80	20	6.7	0.0	0.0
chemical process simulation problem	21	80.9	23.8	14.3	0.0	0.0
circuit simulation problem	2	100.0	0.0	0.0	0.0	0.0
counter-example problem	1	0.0	0.0	0.0	0.0	0.0
total wins:		79.4	30.2	17.5	9.5	1.6

Table 2: Matrix compression using asymmetric MMF (TOPN). See Table 1 for a description of the table entries.

Matrix kind	num. matrices	1%	10%	25%	50%	75%
<i>problems with underlying 2D/3D geometry</i>						
structural problem	4	100.0	100.0	100.0	50.0	50.0
materials problem	1	100.0	100.0	100.0	100.0	100.0
2D/3D problem	5	100.0	100.0	100.0	80.0	60.0
computational fluid dynamics problem	3	100.0	100.0	100.0	0.0	0.0
<i>problems with no underlying geometry</i>						
directed graph	18	100.0	50.0	22.3	16.7	0.0
chemical process simulation problem	21	100.0	90.5	67.7	57.1	38.1
circuit simulation problem	2	100.0	100.0	100.0	50.0	0.0
power network problem	3	100.0	100.0	100.0	66.7	33.3
counter-example problem	1	100.0	100.0	100.0	100.0	100.0
economic problem	10	100.0	100.0	100.0	70.0	50.0
statistical/mathematical problem	1	100.0	100.0	100.0	100.0	0.0
total wins:		100.0	84.1	69.6	49.3	29.0

Table 3: Matrix compression using asymmetric MMF (GREEDYTOPN). See Table 1 for a description of the table entries.

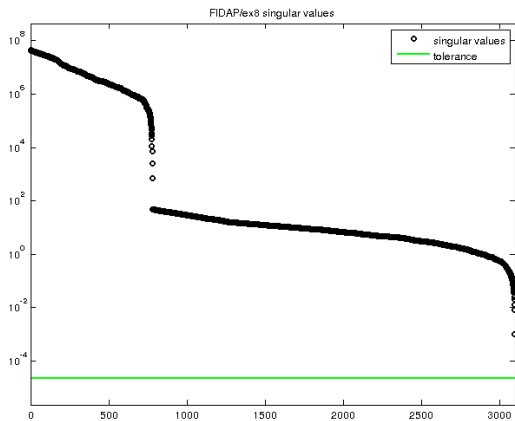


Figure 2: Plot showing the singular values of the ‘ex8’ matrix from the UFlorida sparse matrix repository in descending order. Although the matrix is full-rank, the sharp drop in the singular values around index 800 imparts a low-rank nature to the matrix. (Note that the y-axis is on log-scale). Image sourced from UFlorida Sparse Matrix Repository (Davis and Hu, 2011).

Algorithm 2 Matrix compression using a combination of low-rank and multiresolution methods

Input: Square matrix A , compression size k , and r

$[C, U, R] = \text{CUR}(A, r)$

$[P, H, Q] = \text{MMF}(\text{CUR}, k)$

return $\|PHQ^T\|_F / \|A\|_F$

In most of the matrices, we observe that there exists at least one rank to which the matrix can be first compressed using CUR before compressing with MMF, that results in lower compression errors than each of CUR and MMF methods individually. In practice, there is no method to determine what this rank would be for a given matrix, however, it is an interesting and worthwhile problem to be able to remove a small global-scale component of the matrix, after which the hierarchical structure may be easily extracted.

In all the experiments, the running time of MMF is equal or slightly slower than that of low-rank-based methods. However, we note that our algorithms can be made significantly faster by straightforwardly adapting parallel versions of MMF (Kondor et al., 2015) to the asymmetric case. We expect speedups similar to those achieved in the symmetric case (Teneva et al., 2016).

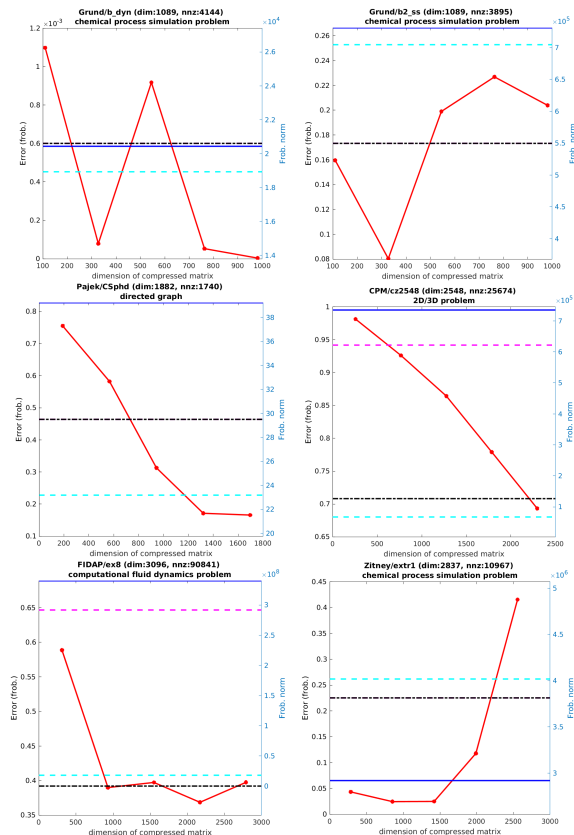


Figure 3: Frobenius norm error as a function of CUR compression rank for compressing the matrix to 5% of its original size using the combined low-rank + multiresolution method. Solid blue line indicates error of compression with only CUR to 5% matrix size, while the dotted blue line indicates the same, but with MMF.

6 Conclusion

In this paper, we have presented approaches to extend Multiresolution Matrix Factorization to nonsymmetric matrices. In general, efficiently estimating hierarchical structure in matrices is an increasingly important problem for which there currently exist very few practically useful methods. For future work we aim to use ideas from wavelet theory on more general spaces to define MMFs for general square matrices in which the MRA interpretation is clear.

References

- Coifman, R. R. and Maggioni, M. (2006). Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94.
- Davis, T. A. and Hu, Y. (2011). The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1.

- Ding, Y., Kondor, R., and Eskreis-Winkler, J. (2017). Multiresolution kernel approximation for gaussian process regression. In Advances in Neural Information Processing Systems, pages 3740–3748.
- Furutani, S., Shibahara, T., Akiyama, M., Hato, K., and Aida, M. (2019). Graph signal processing for directed graphs based on the hermitian laplacian. In Proceedings of ECML-PKDD.
- Hackbusch, W. (1999). A sparse matrix arithmetic based on \mathcal{H} -matrices. part I: introduction to \mathcal{H} -matrices. Computing, 62(2):89–108.
- Halko, N., Martinsson, P.-G., and Tropp, J. A. (2009). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. Computing, 53(2):1–74.
- Ithapu, V. K. (2017). Decoding the deep: Exploring class hierarchies of deep representations using multiresolution matrix factorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 45–54.
- Kondor, R., Teneva, N., and Garg, V. (2014). Multiresolution matrix factorization. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1620–1628.
- Kondor, R., Teneva, N., and Mudrakarta, P. K. (2015). Parallel MMF: a multiresolution approach to matrix computation. arXiv preprint arXiv:1507.04396.
- Li, Y. and Yang, H. (2017). Interpolative butterfly factorization. SIAM Journal on Scientific Computing, 39(2):A503–A531.
- Li, Y., Yang, H., Martin, E. R., Ho, K. L., and Ying, L. (2015). Butterfly factorization. Multiscale Modeling and Simulation, 13(2):714–732.
- Mahoney, M. W. and Drineas, P. (2009). Cur matrix decompositions for improved data analysis. Proceedings of the National Academy of Sciences, 106(3):697–702.
- Mhaskar, H. N. (2018). A unified framework for harmonic analysis of functions on directed graphs and changing data. Applied and Computational Harmonic Analysis, 44(3):611–644.
- Mhaskar, H. N. and Maggioni, M. (2008). Diffusion polynomial frames on metric measure spaces. Applied and Computational Harmonic Analysis, 24(3):329–353.
- Mudrakarta, P. K. and Kondor, R. (2017). A generic multiresolution preconditioner for sparse symmetric systems. arXiv preprint arXiv:1707.02054.
- Murnaghan, F. and Wintner, A. (1931). A canonical form for real matrices under orthogonal transformations. Proceedings of the National Academy of Sciences of the United States of America, 17(7):417.
- Sevi, H., Rilling, G., and Borgnat, P. (2018). Harmonic analysis on directed graphs and applications: from fourier analysis to wavelets. arXiv preprint arXiv:1811.11636.
- Teneva, N., Mudrakarta, P. K., and Kondor, R. (2016). Multiresolution matrix compression. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, volume 51 of Proceedings of Machine Learning Research, pages 1441–1449. PMLR.