Factor Fitting, Rank Allocation, and Partitioning in Multilevel Low Rank Matrices

Tetiana Parshakova[1] Trevor Hastie[2] Eric Darve[3] Stephen Boyd[4]

[1]ICME, Stanford University [2]Department of Statistics, Stanford University [3]Department of Mechanical Engineering, Stanford University [4]Department of Electrical Engineering, Stanford University

Motivation

Multilevel low rank (MLR) matrices are an extension of low rank matrices and factor models in the squared case. MLR generalize low rank matrices in the sense of giving a substantial reduction in storage and speed up in computing matrix-vector products, compared to a full dense matrix. MLR matrices are closely related to hierarchical matrices, which have been studied since at least the late 1980s [1, 2].

Multilevel low rank matrices

An $m \times n$ contiguous MLR matrix A with L levels has the form

$$A = A^1 + \cdots + A^L,$$

where $A^l = \mathbf{blkdiag}(A_{l,1}, \ldots, A_{l,p_l})$, $l = 1, \ldots, L$, and p_l is the size of the partition at level l, with $p_1 = 1$. We refer to $A_{l,k}$ as the kth block on level l of size $m_{l,k} \times n_{l,k}$, for $l = 1, \ldots, L$, $k = 1, \ldots, p_l$.

We require $\mathbf{rank}\, A_{l,k} \leq r_l$, and for $l = 1, \ldots, L$, $k = 1, \ldots, p_l$ the factored form is

$$A_{l,k} = B_{l,k} C_{l,k}^T, \quad B_{l,k} \in \mathbf{R}^{m_{l,k} \times r_l}, \quad C_{l,k} \in \mathbf{R}^{n_{l,k} \times r_l}.$$

We refer to $r = r_1 + \cdots + r_L$ as the **MLR-rank** of A.

Example. The sparsity patterns of A^2 and A^3 are shown below.

$$A^2 = \begin{bmatrix} * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \end{bmatrix}, \quad A^3 = \begin{bmatrix} * & * & & & & & & \\ * & * & & & & & & \\ & & * & * & & & & \\ & & * & * & & & & \\ & & & & * & * & & \\ & & & & * & * & & \\ & & & & & & * & * \\ & & & & & & * & * \end{bmatrix}.$$

A general $m \times n$ MLR matrix $\tilde{A}$ has the form

$$\tilde{A} = PAQ^T,$$

where A is a contiguous MLR matrix, P and Q are permutation matrices.

Two-matrix form

Factor form. For each level $l = 1, \ldots, L$ define

$$\tilde{B}_l = \mathbf{blkdiag}(B_{l,1}, \ldots, B_{l,p_l}) \in \mathbf{R}^{m \times p_l r_l}, \qquad \tilde{C}_l = \mathbf{blkdiag}(C_{l,1}, \ldots, C_{l,p_l}) \in \mathbf{R}^{n \times p_l r_l}.$$

Define

$$\tilde{B} = \begin{bmatrix} \tilde{B}_1 & \cdots & \tilde{B}_L \end{bmatrix} \in \mathbf{R}^{m \times s}, \qquad \tilde{C} = \begin{bmatrix} \tilde{C}_1 & \cdots & \tilde{C}_L \end{bmatrix} \in \mathbf{R}^{n \times s},$$

with $s = \sum_l p_l r_l$. Then we can write A as

$$A = \tilde{B}\tilde{C}^T,$$

exactly the form of a low rank factorization of A.

Compressed factor form. We also arrange the factors into two arrays or matrices with dimensions $m \times r$ and $n \times r$. We vertically stack the factors at each level

$$B^l = \begin{bmatrix} B_{l,1} \\ \vdots \\ B_{l,p_l} \end{bmatrix} \in \mathbf{R}^{m \times r_l}, \qquad C^l = \begin{bmatrix} C_{l,1} \\ \vdots \\ C_{l,p_l} \end{bmatrix} \in \mathbf{R}^{n \times r_l}, \quad l = 1, \ldots, L.$$

We horizontally stack these matrices to obtain two matrices

$$B = \begin{bmatrix} B^1 & \cdots & B^L \end{bmatrix} \in \mathbf{R}^{m \times r}, \qquad C = \begin{bmatrix} C^1 & \cdots & C^L \end{bmatrix} \in \mathbf{R}^{n \times r}.$$

Our Contribution

1. We introduce a novel definition of multilevel low rank matrix.
2. We present two complementary block coordinate descent algorithms for factor fitting.
3. We present a greedy algorithm for rank allocation that is able to re-allocate the rank assigned to each level in the hierarchy, to improve the fitting.
4. An open-source package that implements these methods, github.com/cvxgrp/mlr_fitting.

General MLR fitting

The most general MLR fitting problem is

$$\begin{array}{ll} \text{minimize} & \|A - \hat{A}\|_F^2 \\ \text{subject to} & \hat{A} \text{ is rank } r \text{ MLR}, \end{array}$$

where $A \in \mathbf{R}^{m \times n}$ is the given matrix to be fit, $\hat{A}$ is the variable. In this general version of the problem the data are A, the matrix to be fit, and r, the rank.

We seek permutations P and Q, the number of levels L, the block dimensions $m_{l,k}$ and $n_{l,k}$, $l = 1, \ldots, L$, $k = 1, \ldots, p_l$, the two matrices B and C, and the rank allocation, *i.e.*, r_i for which $r_1 + \cdots + r_L = r$.

Factor fitting

We fix the combinatorial aspects of $\hat{A}$, and only optimize over the (real) coefficients in the factors, *i.e.*, the matrices B and C. We propose to use 1) block coordinate descent and 2)alternating least squares for factor fitting.

Rank allocation

We fix the hierarchical partition but not the ranks, *i.e.*, we optimize over B and C, and also the ranks $r_1, \ldots, r_L$, subject to $r_1 + \cdots + r_L = r$. The rank allocation problem is a natural one when the hierarchical row and column partitions are given or already known.

Define the residual matrix

$$R = \tilde{A} - \sum_{j \neq l} \mathbf{blkdiag}(B_{j,1} C_{j,1}^T, \ldots, B_{j,p_j} C_{j,p_j}^T)$$

If we increase/decrease the rank allocated to (each block of) level l by 1, the decrease/increase in Frobenius norm squared error is

$$\delta_l^+ = \sum_{k=1}^{p_l} \sigma_{r_l+1}^2(R_{l,k}), \qquad \delta_l^- = \sum_{k=1}^{p_l} \sigma_{r_l}^2(R_{l,k}).$$

These numbers predict the decrease or increase when only the level l factors are changed. In each iteration of the rank allocation algorithm, we find the level i and j, with $i \neq j$, for which the predicted net decrease is maximized,

$$i, j = \underset{i \neq j}{\operatorname{argmax}} \left(\delta_i^+ - \delta_j^- \right).$$

Numerical Experiments

The discrete Gauss transform (DGT) matrix is given by

$$A_{ij} = e^{-\|t_i - s_j\|_2^2 / h^2},$$

where $s_j \in \mathbf{R}^d$ for $j = 1, \ldots, n$ and $t_i \in \mathbf{R}^d$ for $i = 1, \ldots, m$ are source and target locations respectively and $h > 0$ is the bandwidth. We set sources and targets to be uniformly distributed in a unit cube $[0, 1]^d$ with $d = 3$, $m = 5000$, $n = 7000$, and $h = 0.2$. We fix the rank as $r = 28$.

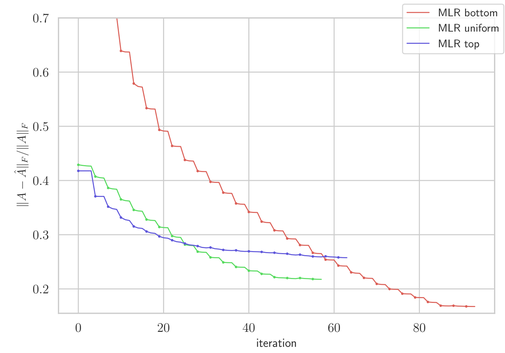

Figure 1. Fitting error during rank allocation, starting from three different initial allocations of rank.

Method	Error (%)	Storage ($\times 10^5$)
LR	41.779	3.360
HODLR	72.549	3.385
MLR bottom	**16.753**	3.360
MLR uniform	21.766	3.360
MLR top	25.759	3.360

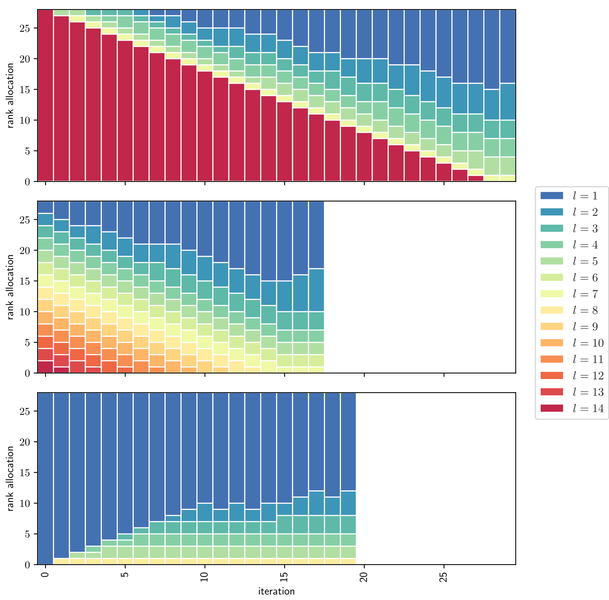

Figure 2. Rank $r = 28$ partitioning across $L = 14$ levels during fitting of DGT matrix, starting from the bottom level $l = 14$, uniform and the top level $l = 1$ initial allocation.

References

[1] Leslie Greengard and Vladimir Rokhlin.
A fast algorithm for particle simulations.
Journal of Computational Physics, 73(2):325–348, 1987.

[2] Wolfgang Hackbusch.
Hierarchical matrices: algorithms and analysis, volume 49.
Springer, 2015.