FMR: Fast randomized algorithms for covariance matrix computations

Pierre Blanchard, Olivier Coulaud, Eric Darve, Alain Franc

To cite this version:

Pierre Blanchard, Olivier Coulaud, Eric Darve, Alain Franc. FMR: Fast randomized algorithms for covariance matrix computations. Platform for Advanced Scientific Computing (PASC), Jun 2016, Lausanne, Switzerland. 2016. hal-01334747

HAL Id: hal-01334747
https://hal.archives-ouvertes.fr/hal-01334747
Submitted on 23 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
We present an open-source library implementing fast algorithms for covariance matrices computations, e.g., randomized low-rank approximations (LRA) and fast multipole matrix multiplication (FMM). The library can be used to approximate square roots of low-rank covariance matrices in O(N^2) operations in SVD form using randomized LRA, instead of the standard O(N^3) cost. Low-rank covariance matrices are often kernels, e.g., Gaussian decay, evaluated on 3D grids can be decomposed in O(N) operations using the FMM. The performance of the library is illustrated on two examples:

- *Generation of Gaussian Random Fields (GRF) on large spatial grids*
- *MultiDimensional Scaling (MDS) for the classification of species.*

Randomized SVD is a random projection-based LRA algorithms made popular by Halko et al. [4], which returns an approximate SVD of a symmetric matrix C \( \in \mathbb{R}^{N \times N} \) given a prescribed numerical rank \( r \) in \( O(N^2 \times r) \) operations:

- Form an approximate basis \( \mathbf{Q} \in \mathbb{R}^{N \times r} \) for the range of \( C \).
- Form a well-conditioned matrix \( C \mathbf{Q} \mathbf{Q}^T \) using randomized tricks.
- Form a low-rank representation of \( C \) in the form \( C \approx C_0 = \mathbf{Q} \mathbf{Q}^T \), with Frobenius/spectral error bounds that hold with high probability.

\[ \text{Factorize } C_0 \text{ in SVD form: } C_0 = \mathbf{U} \sigma \mathbf{V}^T. \]

- We start by assembling the small low-rank matrix \( \mathbf{B} = \mathbf{Q}^T \mathbf{C} \).
- Then, perform a small SVD, i.e., \( \mathbf{B} = \mathbf{Q} \mathbf{\Sigma} \mathbf{V}^T \).
- Form \( \mathbf{U} = \mathbf{Q} \mathbf{\Sigma} \mathbf{V}^T \) and \( \Sigma = \mathbf{\Sigma} \).

If \( C \) is positive semi-definite, then \( C \approx C_0 = A A^T \), where \( A = \mathbf{Q} \mathbf{\Sigma} \mathbf{V}^T \).

The method offers many advantages:

- Easily implemented and parallelized,
- Easily extended to Chebyshev, Interpolative Decomposition ... cost dominated by matrix multiplication, i.e., \( O(N^2 \times N) \).

However, \( C \) should fulfill the following conditions:

- \( C \) should be low-rank (\( r \ll N \)),
- \( C \) should have a fast decreasing spectrum \( (\log(C_{ij}) \sim O(1/r^{1/3}) \) \( \ll N \).

Fast Multipole Acceleration of the matrix multiplications involved in the randomized SVD provides an algorithm for approximating \( A \) in \( O(N^2 \times r) \) operations with many benefits:

- matrix-free method with \( O(1/r^2) \) memory footprint
- hierarchical methods handle heterogeneous grids more efficiently

However, the extra error involved by the FMM has to be monitored.

\[ \mathbf{C} \text{-structure should apply well to } C. \]

Efficient Generation of GRF

**AIM:** This project aims at promoting new highly efficient FMM algorithms to perform resource demanding computations in geostatistics.

**Correlation kernels**

A Gaussian Random Field \( Y \sim \mathcal{N}(0, C) \) is a multivariate Gaussian random variable with mean \( \mathbf{0} \) and covariance \( C \in \mathbb{R}^{N \times N} \). The covariance can be prescribed as a kernel matrix, i.e.,

\[ C = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^{N \times N}, \]

where \( \mathbf{r} = || \mathbf{x}_i - \mathbf{x}_j || \) denotes the distances between points of an arbitrary grid and \( \mathbf{x} \) is a correlation kernel such as

\[ K_{\text{exp}}(\mathbf{r}) = e^{-r/r_\ell}, \quad K_{\text{Gauss}}(\mathbf{r}) = e^{-r^2/(2\ell^2)}. \]

The length scale \( \ell \) characterizes the decreasing speed of the correlation.

**Square-root algorithms**

Covariance matrices are sparsely defined by correlation kernels. Hence, \( C \) admits the following representation:

\[ C = \mathbf{A} \mathbf{A}^T \]

where the matrix factor \( \mathbf{A} \in \mathbb{R}^{N \times N} \) is often called a square root of \( C \). Methods for generating Gaussian Random Fields usually differ by the way \( \mathbf{A} \) is precomputed:

- Standard matrix decompositions (O(N^3))
- Circular embedding (O(N log(N)) for equispaced grids)
- The turning bands method (approximate)

Most of them become computationally prohibitive for large \( N \), i.e., \( N \) over a few thousands.

**RANdomized approach**

Dehbori and Deutsch [3] used the RandSVD in order to precompute \( A \) in low-rank form in \( O(N^2 \times r) \) operations and thus efficiently generate realizations of Gaussian Random Fields at \( O(N \times r) \) cost. This approach still requires \( C \) to be fully assembled.

Fast Multipole Acceleration of the matrix multiplications involved in the randomized SVD provides an algorithm for approximating \( A \) in \( O(N^2 \times r) \) operations with many benefits:

- matrix-free method with \( O(1/r^2) \) memory footprint
- hierarchical methods handle heterogeneous grids more efficiently

However, the extra error involved by the FMM has to be monitored.

**Perspectives**

- Develop automatic procedures for community inventories
- Analyze clustering, concentration of reads, . . .
- Improve visualization tools and methods
- Enhance algorithm and numerical analysis
- Compare with existing approaches based on random column selection
- Improve storage and running time by partitioning data sets and compressing covariance matrices.

**REFERENCES**


**FUNDINGS**

This work was partially supported by the associate team FastLA (Inria, Stanford University & Lawrence Berkeley National Laboratory).