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.
Fast randomized algorithms for covariance matrix computations

Pierre Blanchard 1 - Olivier Coulaud 1 - Eric Darve 2 - Alain Franc 3

ABSTRACT

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 randomized LRA, instead of the standard O(N^3) cost.

Low-rank covariance matrices given as 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.

RANDOM PROJECTION BASED LRA

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

- Form an approximate basis Q ∈ R^{N×N} for the range of C.
- Form a sketch of vector V using Gaussian random projection, i.e., application of C to a N-by-larger random matrix Ω.
- Then, orthogonalize V by QR decomposition
- Thus, we get a low-rank representation of C in the form C ≈ CQ = Q^TQ≈C.

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

- matrix-free method with a O(n × N) memory footprint
- hierarchical methods handle heterogeneous grids more efficiently
- however, the extra error involved by the FMM has to be monitored.
- Ω-structure should apply well to Fast Methods.

EFFICIENT GENERATION OF GRF

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

Correlation kernels: A Gaussian Random Field Y ∼ N(0, C) is a multi-variate Gaussian random variable with mean 0 and covariance C ∈ R^{N×N}. The covariance can be prescribed as a kernel matrix, i.e.,

C = (k(x_i, x_j))_{i,j=1}^{N}

where k is a kernel function. For example, k(x_i, x_j) = exp(-∥x_i - x_j∥^2 / σ^2)

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

Taxonomy via MultiDimensional Scaling (MDS)

Aims: This project aims at developing new strategies for the classification of species that benefits from the massive amount of data provided by Next Generation Sequencing (NGS) techniques.

Metric: MDS aims at reconstructing a cloud of points X in a low-dimensional feature space, e.g., X ∈ R^{N×D}, from a given distance/discrepancy matrix D ∈ R^{N×N} (Smith-Watman scores of local alignment). The algorithm [2] consists in:

- Assembling a covariance/similarity matrix as C_{ij} = σ(x_i, x_j) = 

Computing the SVD of C, i.e., C = UΣ^2

Forming X = C^1/2 × UΣ^{1/2} (LS minimizer)

REFERENCES


FUNDINGS

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