

Wine or Brandy?

Block low rank and block separable matrices

Cleve Ashcraft

Livermore Software Technology Corporation

14th Scheduling for Large Scale Systems Workshop
Bordeaux, June 26 - 28, 2019

The modeling of many physical systems generates large dense matrices with tens and hundreds of thousands of rows and columns. A common property is that the dense matrices can be blocked into a semi-regular grid of submatrices where each of the submatrices may have a more efficient representation as a low rank matrix. Storage for and operations with the submatrices is much more efficient than using the large dense matrix.

There are many ways of taking advantage of this property. All use some form of tree to organize the matrix structure. The simplest tree is a star graph, where all children are connected to a single parent. The BLR (block low rank) matrix representation uses this tree.

With BLR, each submatrix is a completely independent entity. Think of a small domain vintage, grown on a small microclimate. BLR is a collection of distinct wines, one for each submatrix.

Many submatrices have something in common, particularly those from the same block rows and/or columns. There is some redundancy, overlap in the rows and columns spaces. Taking advantage of this overlap can reduce the storage and resulting computation by a considerable amount.

Take a number of different vintages (submatrices), blend them together (assemble into a larger submatrix), and distill (compute the low rank representation). If you merge all submatrices on an entire block row or a block column and then distill, we have the BS (block separable) matrix of Gillman and Martinsson.

BLR is wine. Block separable is brandy. One is made from the other. Sometimes one is better than the other. The desired accuracy makes a difference, so does terroir (choice of locality of the submatrices).

We explore these issues on some BEM (boundary element) matrices from electromechanics and underwater acoustics, and some frontal matrices from PDE's, factored using multifrontal. We consider the representation of the matrix, the matrix-vector multiply (used in Krylov methods), and the matrix factor and solve.