Computers and Mathematics with Applications 71 (2016) 1933–1943

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An improved divide-and-conquer algorithm for the banded matrices with narrow bandwidths

Xiangke Liao [a,b], Shengguo Li [a,*], Lizhi Cheng [c,d], Ming Gu [e]

[a] College of Computer, National University of Defense Technology (NUDT), Changsha 410073, China
[b] Science and Technology on Parallel and Distributed Processing Lab, NUDT, China
[c] State Key Lab of High Performance Computing, NUDT, Changsha 410073, China
[d] College of Science, NUDT, Changsha 410073, China
[e] Department of Mathematics, University of California, Berkeley, CA 47920, USA

ARTICLE INFO

Article history:
Received 18 August 2015
Received in revised form 3 March 2016
Accepted 6 March 2016
Available online 11 April 2016

Keywords:
Divide and conquer algorithm
Banded matrices
SVD
HSS matrices
Eigenvalue problem

ABSTRACT

In this paper we propose a novel divide-and-conquer (DC) algorithm to compute the SVD of banded matrices, and further accelerate it by using rank-structured matrix techniques, especially the hierarchically semiseparable (HSS) matrix. The DC algorithm for the symmetric banded eigenvalue problem can also be accelerated similarly. For matrices with few deflations, the banded DC algorithms require more flops than the classical DC algorithm, and thus they are suitable for narrowly banded matrices. While, if there exist many deflations, the banded DC algorithms can be faster than the classical ones for matrices with relatively large bandwidths. Numerous experiments have been done to test the proposed algorithms. Some of the tested matrices are from construction and some are from real applications. Comparing with the DC algorithm in Intel MKL, our proposed algorithms can be hundreds times faster for matrices with narrow bandwidths or many deflations.

1. Introduction

The eigenvalue and SVD problems are involved in various computational science and engineering areas, such as information retrieval [1], quantum physics [2], chemistry [3], and image restoration. Depending on specific applications, the matrices may be sparse, dense or banded. In this work we focus on the banded case. Many matrices generated from electronic structure calculations can be approximated accurately by banded matrices even though they are dense [2,4]. The banded matrices also appear in the block Lanczos algorithm [5] and in the dense eigenvalue or SVD problems. A dense matrix is first reduced to a banded form and finally to the tridiagonal or bidiagonal form [6,7]. The main objective of this paper is to introduce an efficient *divide-and-conquer algorithm* for computing the SVD of a banded matrix. Unlike classic methods, this new method computes the SVD of a banded matrix directly without reducing it to a bidiagonal form. A similar method was originally proposed in [8] without showing any numerical results. In this paper we revisit it and propose some new techniques to accelerate it, and further show some numerical comparisons with the classical DC algorithms. These techniques can be similarly applied to the symmetric banded eigenvalue problem.

State-of-the-art SVD solvers for banded matrices are based on the bidiagonal reduction strategy, consisting of the following three stages. First, a banded matrix is reduced to an upper bidiagonal form by a sequence of two-sided orthogonal

* Corresponding author.
E-mail addresses: xkliao@nudt.edu.cn (X. Liao), nudtlsg@nudt.edu.cn (S. Li).

http://dx.doi.org/10.1016/j.camwa.2016.03.008
0898-1221/

transformations [9], and this step is called *bidiagonal reduction*. Second, the bidiagonal SVD problem is solved by any standard method such as DC [8], QR [10] or MRRR [11]. Finally, the singular vectors are computed by accumulating the orthogonal transformations from the bidiagonal reduction, and this process is usually called *back transformation*.

The banded DC (BDC) for the SVD problem introduced in this paper is quite similar to that for the symmetric banded eigenvalue problem [12–14] which avoids the tridiagonalization stage, see the works by Arbenz and coauthors [12,15], Gansterer and coauthors [16,13,17,4], Haidar and coauthors [14], etc. For the eigenvalue problem, it boils down to computing the eigendecomposition of a diagonal matrix with a rank-b modification, where b is the bandwidth. Arbenz [12] proposed two methods to solve this problem. One approach turns the rank-b modification into a $b \times b$ eigenproblem, and the eigenvalues can be computed by bisection-type algorithm, and it is suggested to compute the eigenvectors via inverse iteration. Another approach computes the rank-b modification as a sequence of rank-one modifications. The first approach requires fewer flops than the second one. Unfortunately, there exists no numerically stable implementation for the first approach. All the works [16,13,17,4,14] use the second approach. For the SVD case, it boils down to solving a rank-b updating SVD problem, see (10). Similarly, we compute it as a sequence of rank-one updating SVD problems.

One advantage of banded DC is that the *bidiagonal reduction* and *back transformation* steps are avoided, which are done with memory bandwidth limited BLAS2 routines. While, its disadvantage is that it requires much more floating point operations when the singular vectors are required, the complexity increases from $O(N^3)$ to $O(N^3 b)$, see [12,8]. In this work we always assume the singular vectors are desired, and otherwise the classical approach is usually a better choice. Similar to the bidiagonal DC algorithm, the most expensive part of BDC lies in computing the singular vectors via matrix–matrix multiplications (MMM) in $O(N^3)$. BDC needs b times more MMM and solves b times more secular equations than the bidiagonal DC. The good news is that MMM can be performed efficiently by calling highly optimized BLAS libraries and that solving secular equation is relatively cheap, only costs $O(N^2)$ flops. Therefore, when b is small, the banded DC algorithm can be much faster than the classical approach. When b is relatively large, we can first reduce the matrix to a banded matrix with narrower bandwidth via *bulge*chasing and then use BDC to compute its SVD or eigendecomposition. Numerical results in Section 5 show that this strategy can be faster than the classical approach.

Recently, rank-structured matrices are playing a very important role in the design of fast or superfast algorithms. For example, they are involved in designing fast sparse direct solvers [18], solving matrix equations [19], integral equations [20,21] and eigenvalue problems [22,23]. In [23], the authors show that the computations of singular vectors can be accelerated by using HSS matrices. The fact is that the singular vector matrices of a broken arrow matrix can be *off-diagonally* low rank, which can be approximated accurately by HSS or other rank-structured matrices. As is well-known, the HSS matrix multiplication algorithm only costs $O(N^2 r)$ instead of $O(N^3)$ flops, where N is the dimension and r is a relatively small number. In this paper, we similarly use this technique to accelerate the banded SVD problem. The symmetric banded DC algorithm is quite related, and we also show how to use the HSS technique to speedup its computations. The HSS matrix techniques make the banded DC algorithms much more efficient than without using them, which are verified by the numerical results in Section 5.

In summary, the contributions of this paper are the following:

- A banded DC algorithm is proposed for the SVD problem, which is similar to that in [8]. Some implementation details are included, and the numerical results show that it is efficient and numerically stable.
- The complexity of banded DC algorithm is analyzed in detail, and it turns out that it requires more flops than the standard DC algorithm for matrices with few deflations. Numerical results also show it is suitable for matrices with narrow bandwidths. For a matrix with relatively large bandwidth, we suggest reducing its bandwidth first via bulge chasing.
- The HSS matrix techniques are used to accelerate the banded DC algorithms, which make them even more efficient. The results for the banded eigenvalue problems are also included.

The paper is organized as follows. In Section 2, we briefly introduce some concepts of rank-structured matrices especially the HSS matrix. In Section 3, we present the banded DC algorithm for the SVD problem and analyze its complexity in detail. Section 4 briefly introduces the banded DC algorithm for the symmetric eigenvalue problem. Some numerical results are reported in Section 5 including results for constructed matrices and matrices from real applications [24].

2. Notation and HSS matrix

The rank-structured matrix computations have been the intensive focus of recent research. A matrix is called *rank-structured* if the ranks of all off-diagonal blocks are relatively small compared to the size of matrix. The HSS matrix [25,26] is an important kind of rank-structured matrices. The other kinds include $\mathcal{H}$ and $\mathcal{H}^2$ matrices [27,28], sequentially semiseparable (SSS) matrices [29], quasi-separable and semiseparable matrices [30,31], etc.

We follow the notation in [32] which is a little different from those used in [26,33]. The HSS matrix is represented by using a binary *tree*. Let $\mathcal{I} = \{1, 2, \dots, N\}$ and $\mathcal{T}$ be a binary tree in postordering where node i is associated with a contiguous subset t_i of $\mathcal{I}$, which satisfies the following conditions:

- $t_i \cup t_{\mathrm{sib}(i)} = t_{\mathrm{par}(i)}$, where sib($i$) and par($i$) are the sibling and parent of a node i respectively;
- $t_{\mathrm{root}(\mathcal{T})} = \mathcal{I}$, where root($\mathcal{T}$) denotes the root of $\mathcal{T}$.
- $i_1 < i_2 < i$, where i_1 and i_2 are respectively the left and right child of parent node i.