Streaming the Bidiagonal Factorization

Johannes J. Brust and Michael A. Saunders

Abstract

Frequently in online or data-driven applications, new information becomes available in the form of a *stream* (Syamantak et al. [KS24]). Because processing the data for analysis or inference often involves matrix factorizations like the SVD or an eigendecomposition, we develop new updating methods. As repeatedly refactoring a large matrix is expensive, we propose low-rank updates to a previous factorization. Thus we consider the model

$$\bar{A} = A + CW^T,$$

where the previous data A is $m \times n$, and the updates C and W are $m \times t$ and $n \times t$. Simply computing \overline{A} costs mnt multiplications, which we therefore regard as an optimal complexity. For some well known factorizations, efficient updating methods are known. For instance, the methods of Gill et al. [GGMS74] update the Cholesky or LDL^T factorization in $\mathcal{O}(mnt)$ flops, while Golub and Van Loan [GV13, Sec 12.5] describe a method for updating the QR factorization. Also, Bunch et al. [BNS78] describe an algorithm for updating the eigendecomposition, while Brand [Bra06] and Moonen et al. [MVDV92] develop methods for the SVD. The complexity of the latter methods also scales as $\mathcal{O}(mnt)$, but updating an eigendecomposition or SVD typically involves iterative nonlinear equation solves. In SVD computation, the first step is to reduce the matrix to (upper) bidiagonal form before computing the singular values of the bidiagonal (e.g., implemented in LAPACK's bdsqr and gebrd [ABB+99]). Since the bidiagonalization and SVD are closely related, it is not surprising that attempts have been made to replace the SVD with the bidiagonalization for low-rank matrix approximations (Simon and Zha [SZ00]). Even though the SVD guarantees the best low-rank approximations, the bidiagonalization can be computed with a predetermined number of orthogonal transformations, making it computationally attractive.

The most stable method for computing a bidiagonal factorization uses sequences of orthogonal Householder reflectors. Because this requires and overwrites the matrix elements in memory, it is best suited for dense systems. Its complexity scales as $\mathcal{O}(mn^2)$ flops and mn memory and it is therefore limited to small or medium problems. A second approach accesses the data only via matrix-vector products within a short two-vector recursion. This Golub-Kahan bidiagonalization (GKB) produces a partial bidiagonalization after k iterations. For a general A the GKB complexity is $\mathcal{O}(kmn)$. When the data is sparse or otherwise structured, GKB can exploit the structure with potentially much fewer flops (but without further modifications suffers from rapid loss of orthogonality). Our algorithms reuse an existing bidiagonal factorization $A = QBP^T$ to compute the next factorization

$$\bar{Q}\bar{B}\bar{P}^T = QBP^T + CW^T$$

at reduced cost. To exploit previous information fully, we develop sparsity-exploiting bidiagonalization algorithms. One method is gk-bidiag, which we compare to LAPACK's bidiagonalization functions (Table 1). We also propose methods such as a compact representation of products of Householder matrices combined with the GKB iteration.

Problem	m	n	gk-bidiag		LAPACK	
			error	secs	error	secs
GL7d12	8899	1019	0.96	0.031	0.97	26
ch6-6-b2	2400	450	1.6	0.0051	0.94	1.2
ch7-6-b2	4200	630	1.1	0.012	0.95	3.9
ch7-7-b2	7350	882	1.3	0.023	0.97	16
cis-n4c6-b2	1330	210	2.7	0.0017	0.91	0.3
mk11-b2	6930	990	1.2	0.02	0.97	16
n4c6-b2	1330	210	2.9	0.0017	0.91	0.3
rel6	2340	157	0.7	0.0028	0.71	0.56
relat6	2340	157	0.74	0.0029	0.74	0.71

Table 1: Updating a rank r = 50 truncated bidiagonal factorization $\bar{Q}_{1:r}\bar{B}_{1:r}\bar{P}_{1:r}^T$ when a rank-one update is added to a previous factorization. LAPACK subroutines and the sparsity-preserving solver gk-bidiag are applied to 9 SuiteSparse matrices [DH11]. The error is $\|\bar{A} - \bar{Q}_{1:r}\bar{B}_{1:r}\bar{P}_{1:r}^T\|_F/\|A\|_F$.

References

[ABB+99]	Edward Anderson, et al. LAPCK users' guide. SIAM, 1999
[BNS78]	James R Bunch, Christopher P Nielsen, and Danny C Sorensen. Rank-one modification of the symmetric eigenproblem. <i>Numerische Mathematik</i> , 31(1):31–48, 1978.
[Bra06]	Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. Linear Algebra and its Applications, $415(1):20-30$, 2006.
[DH11]	Timothy A Davis and Yifan Hu. The University of Florida sparse matrix collection. ACM Trans. Math. Softw. (TOMS), 38(1):1–25, 2011.
[GGMS74]	Philip E Gill, Gene H Golub, Walter Murray, and Michael A Saunders. Methods for modifying matrix factorizations. <i>Mathematics of Computation</i> , 28(126):505–535, 1974.
[GV13]	G. H. Golub and C. F. Van Loan. <i>Matrix Computations</i> . Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, fourth edition, 2013.
[KS24]	Syamantak Kumar and Purnamrita Sarkar. Streaming PCA for Markovian data. Advances in Neural Information Processing Systems (NeurIIPS), 36, 2024.
[MVDV92]	Marc Moonen, Paul Van Dooren, and Joos Vandewalle. A singular value decomposition updating algorithm for subspace tracking. <i>SIAM Journal on Matrix Analysis and Applications</i> , 13(4):1015–1038, 1992.
[SZ00]	Horst D Simon and Hongyuan Zha. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. <i>SIAM Journal on Scientific Computing</i> , 21(6):2257–2274, 2000.