Algorithms for Generating Bidiagonal Test Matrices

Masami Takata* Kinji Kimura⋆ Masashi Iwasaki◇ Yoshimasa Nakamura†‡
takata@ics.nara-wu.ac.jp
* Graduate School of Humanities and Sciences, Nara Women's University
⋆ Graduate School of Science & Technology, Niigata University
◇ Faculty of Human Environment, Kyoto Prefectural University
† Graduate School of Informatics, Kyoto University ‡ SORST, JST

Abstract

In this paper, we propose two algorithms to generate matrices with desired singular values and singular vectors. The first algorithm is based on the Golub-Kahan-Lanczos method. This requires $O(m^2)$ computational cost, where m is dimension. The second algorithm uses the Jacobi rotation and the bulge-chasing. If matrices with desired singular values are needed, the computational cost is $O(m^2)$. If matrices with not only desired singular values but also desired singular vectors are needed, the computational cost becomes $O(m^3)$.

上2重対角のテスト行列を作成するためのアルゴリズム

高田 雅美* 木村 欣司⋆ 岩崎 雅史◇ 中村 佳正†‡
* 奈良女子大学大学院人間文化研究科 ⋆ 新潟大学大学院自然科学研究科
◇ 京都府立大学人間環境学部 † 京都大学大学院情報学研究科 ‡ JST, SORST

概要

本論文では，与えられた特異値と特異ベクトルからテスト行列を構成するためのアルゴリズムを2つ提案する．1つ目のアルゴリズムは，Golub-Kahan-Lanczos 法を基にしており，計算量は $O(m^2)$ である．ここで，m は行列サイズを表す．2つ目のアルゴリズムは，Jacobi 法を用いる．与えられた特異値のみから構成する場合，計算量は $O(m^2)$ である．また，特異ベクトルも与える場合，$O(m^3)$ となる．

1 Introduction

When accuracy of Singular Value Decomposition (SVD) schemes is evaluated, accuracy of singular vectors has been discussed with respect to the orthogonality in general. However, the orthogonality does not indicate errors between a computed singular vectors and the exact singular vectors. Thereupon, test matrices with desired singular values and singular vectors should be used. Though, algorithms for generating test matrices are important, such suitable algorithm have not been found. In this paper, we discuss new algorithms for the above purpose.

In §2, we discuss three existence algorithms to evaluate accuracy of numerical scheme computing SVD. In §3 and 4, we propose two algorithms generating accurate test matrices having desired singular values and singular vectors. In §5, we compare performances of two our algorithms.

2 Existing Algorithms for Generating Test Matrices

To evaluate numerical routines, both execution time and accuracy are needed. It is easy to obtain execution time. On the other hand, accuracy should be measured by using matrices with desired singular values and singular vectors. There are three existing algorithm for generating bidiagonal test matrices.

In the first algorithm,*double-precision* type code is transformed into *multiple-precision arithmetic* one, to obtain more accurate singular values and singular vectors of a test matrix. And, desired singular values and singular vectors are not used.

The second algorithm is designed by application of the Cholesky decomposition to a known benchmark symmetric positive definite tridiagonal matrices for eigen decomposition. Bench mark matrices are few.

As the third algorithm, in few matrices, exact singular values and singular vectors are expressed as trigonometric [2]. By using this algorithm, matrices, whose singular values and singular vectors are know with high accuracy, can be generated in any matrix dimension. However, such matrices have special properties, for example, it has uniformly distributed singular values. Hence, accuracy in schemes can not be generally discussed through experimental results.

3 Algorithm designed from the Golub-Kahan-Lanczos method

Let A be an $m\times m$ full matrix whose singular values are nonzero and simple (not multiplicate). Let P and Q denote orthogonal matrices such that $P^\top AQ = B$, where $P = (\mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_m})$, $Q = (\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_m})$, and B is an upper bidiagonal matrix. It is known that the P, Q and B are computed by using the Golub-Kahan-Lanczos method [1]. We obtain $A^\top AQ = A^\top PB = QB^\top B$, which implies that Q transforms $A^\top A$ to $B^\top B$, where the singular values of B are congruent with those of A.

Let us consider the case where A is a just diagonal matrix $\Sigma = \Sigma^\top$ where $A = \Sigma = \Sigma^\top$. Then Σ consists of singular values of B because of $P^\top \Sigma Q = B$. Let us set a vector $\mathbf{q_1}$ such that $|\mathbf{q_1}| = 1$ and an $m \times m$ diagonal matrix Σ. Then Q, P and B are given as $\Sigma Q = PB, \Sigma^\top P = QB^\top$.

b_{2i-1} $(1 \le i \le m)$ and $\mathbf{p_i}$ are determined by $\Sigma\mathbf{q_i} - b_{2(i-1)}\mathbf{p_{i-1}} = b_{2i-1}\mathbf{p_i}$ with $|\mathbf{p_i}| = 1$, and b_{2i} and $\mathbf{q_{i+1}}$ are decided by $\Sigma^\top\mathbf{p_i} - b_{2i-1}\mathbf{q_i} = b_{2i}\mathbf{p_{i+1}}$ with $|\mathbf{q_i}| = 1$. Let $P = U$ and $Q = V$, then

$$
\begin{aligned}
& b_1 = \|\Sigma\mathbf{v_1}\|, \quad \mathbf{u_1} = \Sigma\mathbf{v_1}/b_1, \\
& \begin{cases} b_{2h} = \|\alpha_h\|, & \mathbf{v_{h+1}} = \alpha_h/b_{2h}, \\ b_{2h+1} = \|\beta_h\|, & \mathbf{u_{h+1}} = \beta_h/b_{2h+1}, \end{cases} \\
& \alpha_h = \Sigma^\top\mathbf{u_h} - b_{2h-1}\mathbf{v_h}, \\
& \beta_h = \Sigma\mathbf{v_{h+1}} - b_{2h}\mathbf{u_h}, \\
& \qquad\qquad (h : 1 \le h \le m-1).
\end{aligned} \tag{1}
$$

Hence, in the case where a vector $\mathbf{v_1}$ such that $|\mathbf{v_1}| = 1$ and an $m \times m$ diagonal matrix Σ are given, V, U, and B can be obtained. This algorithm in double-precision is not always suitable because this algorithm is based on Krylov subspace method. Therefore, a multiple-precision arithmetic library is needed. For the number of necessary precision, we will discuss in §5.

In this algorithm, to obtain a bidiagonal matrix with desired singular values, the computational cost is $O(m^2)$. To generate a bidiagonal matrix with desired singular values and singular vectors, it takes also $O(m^2)$.

4 Algorithm designed from the Jacobi rotations and the bulge-chasing

In this section, we propose an algorithm which transforms a diagonal matrix into bidiagonal matrix with the help of the Jacobi rotations and the bulge-chasing.

Let Σ_0 be an $m\times m$ diagonal matrix whose elements are singular values σ_i.

The Jacobi rotation

$$
J_0 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \cos\theta_0 & \sin\theta_0 \\ & & & -\sin\theta_0 & \cos\theta_0 \end{pmatrix} \tag{2}
$$

is adopted as $\tilde{\Sigma}_0 = \Sigma_0 J_0$, where θ_0 is randomly given. Let L_0 be set to

$$
L_0 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \cos\phi_0 & \sin\phi_0 \\ & & & -\sin\phi_0 & \cos\phi_0 \end{pmatrix} \tag{3}
$$

with ϕ_0 such that the $(m, m-1)^{th}$ element of $L_0\tilde{\Sigma}_0$ becomes 0. Concretely, we have$\Sigma_1 = L_0\tilde{\Sigma}_0$.

Next, another Jacobi rotation J_1

$$
J_1 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & \cos\theta_1 & \sin\theta_1 & \\ & & -\sin\theta_1 & \cos\theta_1 & \\ & & & & 1 \end{pmatrix}, \tag{4}
$$

is adopted as $\tilde{\Sigma}_1 = \Sigma_1 J_1$, where θ_1 is randomly generated. Then, let L_1 be set to

$$L_1 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & \cos\phi_1 & \sin\phi_1 & \\ & & -\sin\phi_1 & \cos\phi_1 & \\ & & & & 1 \end{pmatrix}, \quad (5)$$

where ϕ_1 is a suitable value to change the $(m-1, m-2)^{th}$ element to 0 in $L_1\tilde{\Sigma}_1$. Concretely, the left Given rotation L_1 is adopted as $\hat{\Sigma}_1 = L_1\tilde{\Sigma}_1$. Let R_1 be set to

$$R_1 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & \cos\psi_1 & & \sin\psi_1 \\ & & & 1 & \\ & & -\sin\psi_1 & & \cos\psi_1 \end{pmatrix}, \quad (6)$$

where ψ_1 is a suitable value to change the $(m-2, m)^{th}$ element to 0 in $\hat{\Sigma}_1 R_1$. Then, the right Given rotation R_1 is adopted as $\bar{\Sigma}_1 = \hat{\Sigma}_1 R_1$. $\tilde{L}_0$ is given as follows:

$$\tilde{L}_0 = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \cos\tilde{\phi}_0 & \sin\tilde{\phi}_0 \\ & & & -\sin\tilde{\phi}_0 & \cos\tilde{\phi}_0 \end{pmatrix}, \quad (7)$$

where $\tilde{\phi}_0$ is defined to change the $(m, m-1)^{th}$ element to 0 in $\tilde{L}_0\bar{\Sigma}_1$. Then,

$$\begin{aligned} \breve{\Sigma}_1 &= \tilde{L}_0\bar{\Sigma}_1 = \tilde{L}_0 L_1 L_0 \Sigma_0 J_0 J_1 R_1 \\ &\equiv \begin{pmatrix} d_1 & & & & \\ & \ddots & & & \\ & & d_{m-2} & e_{m-2} & \\ & & & d_{m-1} & e_{m-1} \\ & & & & d_m \end{pmatrix}. \quad (8) \end{aligned}$$

And, the left orthogonal matrix U and the right one V in $\breve{\Sigma}_1$ are given as $U = \tilde{L}_0 L_1 L_0$ and $V = (J_0 J_1 R_1)^\top$.

By repeating this process, a bidiagonal matrix B and exact singular vectors can be generated.

The computational cost is only in $O(m^2)$ for generating a bidiagonal matrix B if U and V are not necessary. On the other hand, the computation cost is in $O(m^3)$ for generating a bidiagonal matrix B, singular vectors U and V.

表 1: The max necessary bit number for generating bidiagonal matrix with only desired singular values. [bit]

matrix dimension	$Algorithm_G$	$Algorithm_J$
100	512	128
200,...,400	1024	128
500,...,900	2048	128
1000	4096	128

5 Evaluations

We discuss a necessary bit number and execution time. Here, the necessary bit number means the minimal bit number for generating the same matrices by using larger bit number. For some experiments, we use a computer with 2 CPUs: AMD OPTERON 285 (Dual Core) and GCC 4.1.1. $Algorithm_G$ and $Algorithm_J$ are program codes based on our algorithms proposed in §3 and 4, respectively.

5.1 Bidiagonal matrices

To validate necessary bit numbers, 100 test matrices are generated at each dimension. In our experiments, necessary bit numbers increase twice and twice. Tab.1 shows the max necessary bit number for a generation with only exact singular values. By using $Algorithm_J$, the necessary bit number is always 128 bits, regardless of matrix dimension. On the other hand, in $Algorithm_G$, the necessary bit number increases as matrix dimension be longer.

From the transition of average execution time for generating bidiagonal matrices with only desired singular values by using $Algorithm_J$, we confirm that the execution time is $O(m^2)$. Fig.1 shows transition of average execution time in $Algorithm_J$ and $Algorithm_G$. The execution time in $Algorithm_G$ is longer than that in $Algorithm_J$. Since the necessary bit number is changed, execution time increases above $O(m^2)$ in $m = 100, 200, 500$ and 1000.

It is concluded that the necessary bit number in $Algorithm_J$ is smaller than that in $Algorithm_G$ and the execution time of $Algorithm_J$ is $O(m^2)$. $Algorithm_J$ should be employed in the case where we evaluate accuracy of singular values and orthogo-

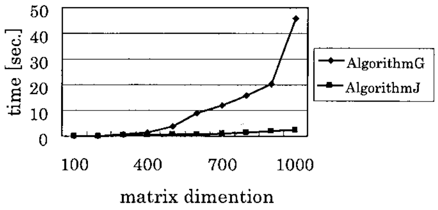

図 1: Transition of average execution time for generating bidiagonal matrices with desired singular values.

表 2: The max number of necessary bit for generating of a bidiagonal matrix with desired singular values and vectors. [bit]

matrix dimension	$Algorithm_G$	$Algorithm_J$
100	2048	512
200	2048	1024
300	4096	1024
400, 500	4096	2048
600,...,1000	8192	2048

nality of singular vectors computed by SVD routines.

5.2 Exact singular vectors

To investigate necessary bit numbers, 3 test matrices are generated at each dimension. Tab.2 shows the max number of necessary bit for generating with not only desired singular values and but also desired singular vectors. In $Algorithm_G$ and $Algorithm_J$, the necessary bit number increases as matrix dimension becomes large. The necessary bit number in $Algorithm_G$ is longer than that in $Algorithm_J$.

Tab.2 shows the transition of computational cost for generating a bidiagonal matrix with desired singular values and vectors. When the number of bit increases at twice, the computational cost becomes three times. The transition in $Algorithm_J$ is more furious than that in $Algorithm_G$, since the computational cost in $Algorithm_G$ and $Algorithm_J$ are $O(m^2)$ and $O(m^3)$, respectively.

Consequently, to generate a bidiagonal test matrix with desired singular values and vectors, $Algorithm_G$ is better if memory size is sufficient level.

表 3: The computational cost for generating with desired singular values and vectors. [sec.]

matrix dimension	$Algorithm_G$ Max.	$Algorithm_G$ Min.	$Algorithm_J$ Max.	$Algorithm_J$ Min
100	0.26	0.04	1.28	0.54
200	0.98	0.35	29.99	9.98
300	6.78	2.21	100.91	99.80
400	12.08	3.93	715.84	236.33
500	18.81	18.89	1400.07	461.18
600	87.76	27.07	2440.43	798.41
700	119.56	37.00	3892.49	1272.62
800	156.10	48.47	5808.66	5726.86
900	197.35	61.53	8184.76	8180.16
1000	243.20	76.63	11346.89	11233.88

6 Conclusion

In this paper, we propose new algorithms for generating bidiagonal test matrices, which is useful for validating SVD routines. By using existing algorithms in §2, we can not obtain a number of bidiagonal matrices whose singular values and singular vectors are known. On the other hand, our new algorithms in §3 and 4 can generate bidiagonal matrices with desired singular values and vectors in a random.

When a bidiagonal matrix with only desired singular values are generated, the computational costs in $Algorithm_G$ and $Algorithm_J$ are both $O(m^2)$. Also, the necessary bit number in $Algorithm_J$ is shorter than that in $Algorithm_G$. From viewpoint of necessary bit number, $Algorithm_J$ is better. To generate a bidiagonal matrix with not only desired singular values but also desired singular vectors, $Algorithm_G$ and $Algorithm_J$ require $O(m^2)$ and $O(m^3)$ execution time, respectively. $Algorithm_G$ runs faster than $Algorithm_J$.

Acknowledgement

This work is partially supported by JSPS Grant-in-Aid for Young Scientists(B) Japan, No.18700025.

参考文献

[1] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *J. SIAM. Numer. Anal.*, Ser. B 2, pp.205–224, 1965.

[2] R.T. Gregor and D.L. Karney. *A Collection of Matrices for Testing Computational Algorithms Interscience.*, Wiley-Interscience, 1969.