

シフト付き Cholesky QR 分解を用いた特異値分解の高速化

高田 雅美^{1,a)} 荒木 翔^{2,b)} 大澤 真之^{2,c)} 青木 雅奈^{2,d)} 木村 欣司^{2,e)} 中村 佳正^{2,f)}

概要: 本稿では、長方形の特異値分解においてボトルネックとなる 2 重対角化または帯行列化を高速に実行するために、シフト付き Cholesky QR 分解を提案する。Cholesky QR 分解は、level3 の BLAS を用いることができるため、高速に実行可能である。しかしながら、丸め誤差の影響により、悪条件な行列が与えられた場合、Cholesky 分解が破綻することがある。この問題を回避するために、シフトを導入する。

Acceleration of Singular Value Decomposition using Cholesky QR algorithm with Shift

MASAMI TAKATA^{1,a)} SHO ARAKI^{2,b)} MASAYUKI OSAWA^{2,c)} MASANA AOKI^{2,d)} KINJI KIMURA^{2,e)}
YOSHIMASA NAKAMURA^{2,f)}

1. はじめに

画像処理、情報検索、最小 2 乗問題を扱うアプリケーションにおいて、その特徴量を取得するために、特異値分解が利用されている。また、近年、ビッグデータの 1 つである顧客の購入情報が保存されたデータ行列に対する主成分分析や特徴量の取得において、有効な方法として利用する試みがなされている [18]。多くのビッグデータでは、行列の短辺に集めた情報の項目を与え、長辺に各個体を識別するための番号が並ぶ。この番号は、情報爆発により、非常に大きな数となることが多い。

特異値分解では、与えられた長方形データ行列が対角行列と 2 つの直交行列の積に分解される。対角行列の対角要素には、その行列の特異値が並ぶ。2 つの直交行列には、それぞれ、特異値に対応する特異ベクトルが並ぶ。多く

の特異値分解法において、与えられた長方形行列に対して Householder 変換 [3], [11] を適用することによって直交行列と正方形の 1 種である上 2 重対角行列に分解し、この上 2 重対角行列に対して特異値分解が行われる。上 2 重対角行列に対する特異値分解は、密行列に比べ、高速に計算することができる。しかしながら、長方形行列を 2 重対角行列に直接変換する Householder 変換の計算量が大きい。この問題を回避するため、与えられた長方形行列に Householder 変換による QR 分解 (Householder QR 分解) 法を適用し、得られた上三角行列に対して Householder 変換による 2 重対角行列化を行うことが適切であると考えられる。ただし、長方形の長辺が非常に長い場合には、Householder QR 分解法がボトルネックになる。

本稿では、高速性と高精度性を有する新たな特異値分解法を提案する。具体的には、多くの特異値分解法で用いられている Householder QR 分解法を適用するのではなく、Cholesky QR 分解 [10] を前処理に用いる特異値分解法を提案する。Cholesky QR 分解は、level-3 の BLAS [1] を適用することができるため、Householder QR 分解法に比べ、高速に実行することができる。しかし、丸め誤差の影響により、Cholesky 分解ができない場合がある [21]。この問題を解決するために、本稿では、シフトを導入する。

¹ 奈良女子大学
Nara Women's University, Nara, Nara 630-8506, JAPAN
² 京都大学
Kyoto University, Kyoto, Kyoto 606-8501, JAPAN
a) takata@ics.nara-wu.ac.jp
b) araki@amp.i.kyoto-u.ac.jp
c) osawa.masayuki.36z@st.kyoto-u.ac.jp
d) aoki.masana.36e@st.kyoto-u.ac.jp
e) kimura.kinji.7z@kyoto-u.ac.jp
f) ynaka@i.kyoto-u.ac.jp

2章では、特異値分解について紹介する。3章では、シフト付き Cholesky QR 分解を提案する。4章では、提案されたシフト付き Cholesky QR 分解と Householder QR 分解法の比較実験を行う。

2. 特異値分解

ランク r の $m \times n$ ($m \geq n$) 次の長方形行列 $A \in \mathbb{R}^{m \times n}$ の特異値分解は、

$$A = U \Sigma V^T \quad (1)$$

と表される。ここで、列直交行列 $U \in \mathbb{R}^{m \times r}$ の列は左特異ベクトル、列直交行列 $V \in \mathbb{R}^{n \times r}$ の列は右特異ベクトル、対角行列 $\Sigma \in \mathbb{R}^{r \times r}$ は対角成分に特異値を持つ行列である。つまり、

$$A v_i = \sigma_i u_i, \quad (2)$$

$$A^T u_i = \sigma_i v_i \quad (i = 1, \dots, r), \quad (3)$$

$$U := [u_1, u_2, \dots, u_r] \in \mathbb{R}^{m \times r}, \quad (4)$$

$$V := [v_1, v_2, \dots, v_r] \in \mathbb{R}^{n \times r}, \quad (5)$$

$$\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}, \quad (6)$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0 \quad (7)$$

が成り立つ。また、 i 番目の特異値 σ_i の左特異ベクトルは u_i 、右特異ベクトルは v_i である。

特異値分解法として、Jacobi 法 [6], [7], [20] がある。この特異値分解法は、得られた行列の直交性に関する精度は良好であるが、計算時間が非常に長い。そのため、大規模行列には適していない。

そこで、計算時間を少なくするために、前処理として、Householder 変換により、与えられた長方形行列 A を適切な直交行列と上 2 重対角行列 B の積に変換する。この時、長方形行列 A と正方形行列 B の特異値は等しい。この 2 重対角行列 B のための特異値分解法として、QR 法 [3], [4], [9], [11], [12], I-SVD (Integrable-Singular Value Decomposition) 法 [14], [15], [19] 法、分割統治法 [2], [13], 拡大行列を用いた 2 分法と逆反復法による解法、拡大行列を用いた MR³ 法 [5], [8], [17] がある。これらの特異値分解法のいずれかを用いて特異値分解を行う場合、長方形行列の長辺が長ければ長いほど、Householder 変換がボトルネックとなる。

この問題を回避するために、長方形行列 A に Householder 変換を適用するのではなく、QR 分解 $A = QR$ の上三角行列 $R \in \mathbb{R}^{n \times n}$ に対して行う。ここで、 $Q \in \mathbb{R}^{m \times n}$ は、直交行列を表す。上三角行列 R の左特異ベクトル行列 $U_{(R)}$ 、右特異ベクトル行列 $V_{(R)}$ 、対角成分に特異値を持つ行列 $\Sigma_{(R)}$ とすると、

Algorithm 1 Cholesky 分解

```

1: for  $k = 1, \dots, n$  do
2:    $r_{k,k} := \sqrt{w_{k,k}^{(k)}}$ 
3:   for  $j = k + 1, \dots, n$  do
4:      $r_{k,j} := w_{k,j}^{(k)} / r_{k,k}$ 
5:   end for
6:   for  $j = k + 1, \dots, n$  do
7:     for  $i = j, \dots, n$  do
8:        $w_{j,i}^{(k+1)} := w_{j,i}^{(k)} - r_{k,j} r_{k,i}$ 
9:     end for
10:  end for
11: end for

```

$$R = U_{(R)} \Sigma_{(R)} V_{(R)}^T, \quad (8)$$

$$\begin{aligned} A &= QR \\ &= Q \left(U_{(R)} \Sigma_{(R)} V_{(R)}^T \right) \\ &= (QU_{(R)}) \Sigma_{(R)} V_{(R)}^T, \end{aligned} \quad (9)$$

が得られる。つまり、長方形行列 A の左特異ベクトル行列は $QU_{(R)}$ 、右特異ベクトル行列は $V_{(R)}$ 、対角成分に特異値を持つ行列は $\Sigma_{(R)}$ となる。

3. シフト付き Cholesky QR 分解

Cholesky QR 分解とは、以下の手順で $m \times n$ 次の行列 A の QR 分解 $A = QR$ を行う手法である:

$$(1) W := A^T A,$$

$$(2) W = R^T R \text{ (Cholesky 分解)},$$

$$(3) Q := AR^{-1}.$$

ここで、 Q は $m \times n$ 次の列直交行列、 R は n 次上三角行列である。 n 次実対称行列 W の Cholesky 分解 $W = R^T R$ の基本的な手順は Algorithm 1 の通りである。ただし、 $w_{i,j}^{(1)}$ は W の (i, j) 成分であり、 $r_{i,j}$ が求める R の (i, j) 成分を表す。ここで、 W が正定値であっても、数値誤差によって反復の途中で $w_{k,k}^{(k)} < 0$ となってしまう、計算が破綻する可能性がある。これを回避するため、次の定理に基づくシフト戦略を考える。

定理 W_ℓ ($\ell = 1, \dots, n$) を W の ℓ 次首座小行列とし、その最小固有値を $\lambda_{\min}(W_\ell)$ とする。このとき、

$$w_{k,k}^{(k)} > 0 \quad (1 \leq k < \ell), \quad w_{\ell,\ell}^{(\ell)} < 0 \Rightarrow w_{\ell,\ell}^{(\ell)} < \lambda_{\min}(W_\ell), \quad (10)$$

$$w_{k,k}^{(k)} > 0 \quad (1 \leq k \leq \ell) \Leftrightarrow 0 < \lambda_{\min}(W_\ell) \quad (11)$$

が成り立つ。

証明 n 次実対称行列 W に対して Algorithm 1 を適用すると、1 行目から始まる反復が少なくとも $(\ell - 1)$ 回 ($\ell = 2, \dots, n$) までは上手く動くとする。すなわち、 $w_{k,k}^{(k)} > 0$ ($1 \leq k < \ell$) が成り立つとする。このとき、 $r_{\ell,\ell}$ が虚数になることも許すと、 $w_{\ell,\ell}^{(\ell)}$ の符号によらず、

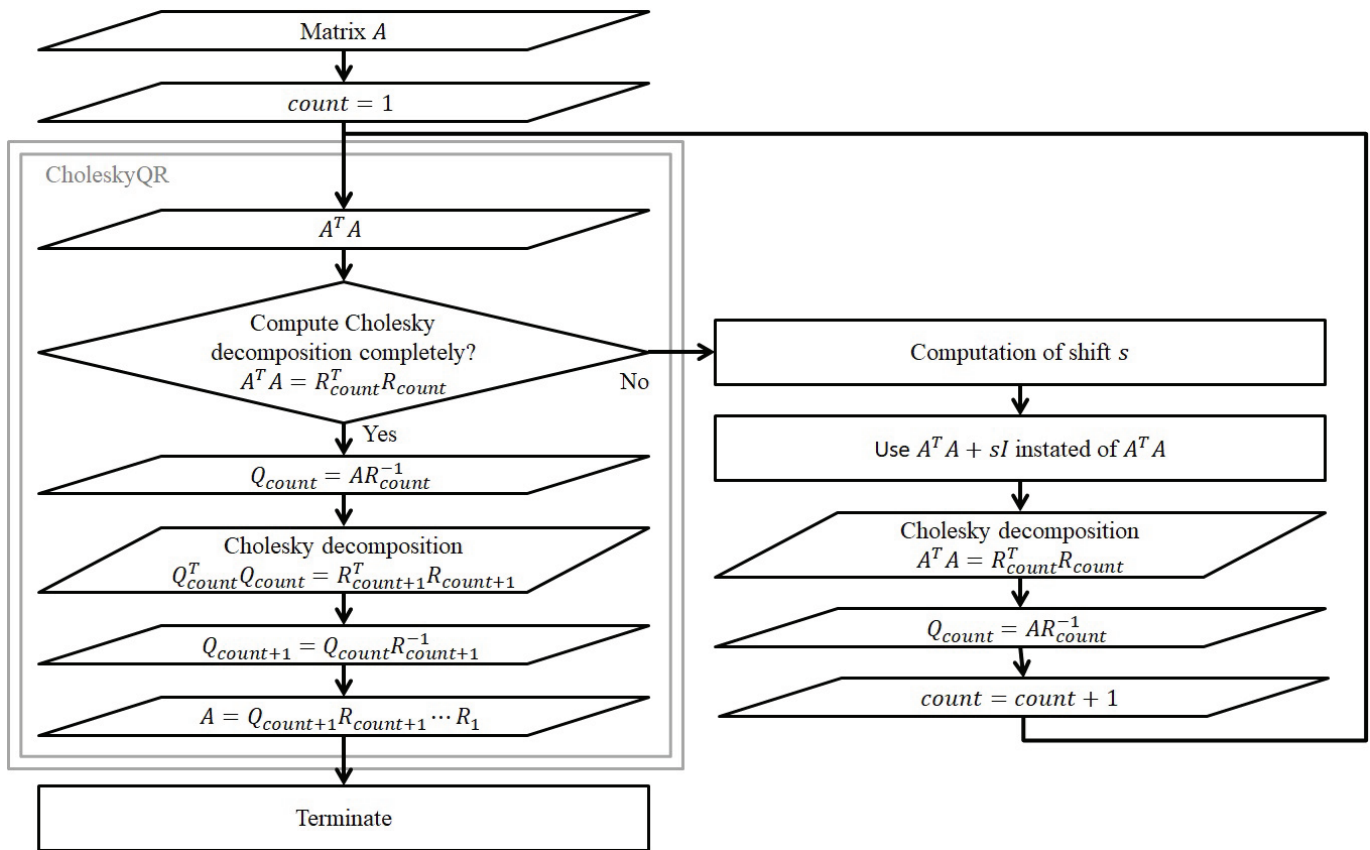


図 1 シフト付き Cholesky QR 分解の概念図

$r_{\ell,\ell} = \sqrt{w_{\ell,\ell}^{(\ell)}}$ を定めることができる。このとき、 R_j を R の j 次首座小行列とすると、

$$W_j = R_j^T R_j \quad (j = 1, \dots, \ell) \quad (12)$$

が成り立つ。したがって、

$$\det W_j = (\det R_j)^2 = \prod_{i=1}^j r_{i,i}^2 \quad (13)$$

が成り立ち、特に

$$\det W_\ell = (\det R_\ell)^2 = \prod_{i=1}^{\ell} r_{i,i}^2, \quad (14)$$

$$\det W_{\ell-1} = (\det R_{\ell-1})^2 = \prod_{i=1}^{\ell-1} r_{i,i}^2 \quad (15)$$

である。また、 $\lambda_i(X)$ と書いて、行列 X の大きい方から i 番目の固有値を表すものとする、

$$\det W_\ell = \prod_{i=1}^{\ell} \lambda_i(W_\ell), \quad (16)$$

$$\det W_{\ell-1} = \prod_{i=1}^{\ell-1} \lambda_i(W_{\ell-1}) \quad (17)$$

が成り立つ。(14),(15),(16),(17) より、

$$r_{\ell,\ell}^2 = \lambda_\ell(W_\ell) \prod_{i=1}^{\ell-1} \frac{\lambda_i(W_\ell)}{\lambda_i(W_{\ell-1})}. \quad (18)$$

さらに、分離定理より

$$\lambda_i(W_\ell) \geq \lambda_i(W_{\ell-1}) \geq \lambda_{i+1}(W_\ell) \quad (i = 1, \dots, \ell - 1) \quad (19)$$

が成り立つ。

ここで、

$$r_{k,k}^2 = w_{k,k}^{(k)} > 0 \quad (k = 1, \dots, \ell - 1) \quad (20)$$

という仮定のもとでは

$$\lambda_j(W_j) > 0 \quad (j = 1, \dots, \ell - 1) \quad (21)$$

が成り立つことを、帰納法によって示す。

(1) $j = 1$ のとき、(20) より

$$\lambda_1(W_1) = r_{1,1}^2 > 0. \quad (22)$$

(2) $j = k$ ($1 \leq k \leq \ell - 2$) で (21) が成り立つと仮定する。

(13) より

$$\det W_{k+1} = \prod_{i=1}^{k+1} r_{i,i}^2 \quad (23)$$

であるから、(20) より

$$\det W_{k+1} > 0 \quad (24)$$

が言える。ここで

$$\det W_{k+1} = \lambda_{k+1}(W_{k+1}) \prod_{i=1}^k \lambda_i(W_{k+1}) \quad (25)$$

であるが, (19) と仮定より

$$\lambda_i(W_{k+1}) \geq \lambda_k(W_k) > 0 \quad (i = 1, 2, \dots, k) \quad (26)$$

であるから

$$\lambda_{k+1}(W_{k+1}) > 0 \quad (27)$$

が成り立つ.

これより, (19) に注意すれば

$$\lambda_i(W_{\ell-1}) > 0 \quad (i = 1, \dots, \ell-1) \quad (28)$$

であり,

$$\frac{\lambda_i(W_\ell)}{\lambda_i(W_{\ell-1})} \geq 1 \quad (i = 1, \dots, \ell-1) \quad (29)$$

が成り立つ.

以上より, $r_{\ell,\ell}^2 = w_{\ell,\ell}^{(\ell)} < 0$, すなわち

$$r_{\ell,\ell}^2 = \lambda_\ell(W_\ell) \prod_{i=1}^{\ell-1} \frac{\lambda_i(W_\ell)}{\lambda_i(W_{\ell-1})} < 0 \quad (30)$$

を仮定すれば

$$\lambda_\ell(W_\ell) < 0 \quad (31)$$

となる. (29),(31) より

$$\lambda_\ell(W_\ell) \geq \lambda_\ell(W_\ell) \prod_{i=1}^{\ell-1} \frac{\lambda_i(W_\ell)}{\lambda_i(W_{\ell-1})} = r_{\ell,\ell}^2 \quad (32)$$

であるから, 結局

$$w_{\ell,\ell}^{(\ell)} = r_{\ell,\ell}^2 \leq \lambda_\ell(W_\ell) = \lambda_{\min}(W_\ell) \quad (33)$$

が成り立つ.

一方, $r_{\ell,\ell}^2 = w_{\ell,\ell}^{(\ell)} > 0$, すなわち

$$r_{\ell,\ell}^2 = \lambda_\ell(W_\ell) \prod_{i=1}^{\ell-1} \frac{\lambda_i(W_\ell)}{\lambda_i(W_{\ell-1})} > 0 \quad (34)$$

を仮定すれば, (29) より

$$\lambda_\ell(W_\ell) > 0, \quad (35)$$

が成り立つ.

定理によると, Algorithm 1 の反復の過程で, ある k に対し $w_{k,k}^{(k)} < 0$ となった場合, シフトによって新たに $W := W - w_{k,k}^{(k)}I$ (I は n 次単位行列) と取り直して Algorithm 1 を適用すれば, $w_{k,k}^{(k)} > 0$ となると期待される. このようなシフトを, コレスキー分解が破綻することなく完了するまで繰り返し行う. ただし, $w_{k,k}^{(k)}$ がちょうど 0 になった場合には, それまでの累積シフト量を s として, $s := s(1 + \epsilon)$ (ϵ はマシンイプシロン) を新た

表 1 実験環境

CPU	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz (2 cores)
RAM	128GB
OS	Ubuntu 16.04.3 LTS
Compiler	icc version 18.0.0, ifort version 18.0.0
Options	-fast
Software	Intel Math Kernel Library 2018

な累積シフト量とする. その際, $s = 0$ である場合には, $s := \max\{w_{k,k} \mid k = 1, \dots, n\}$ を新たな累積シフト量とする. 以上が, シフト付き Cholesky QR 分解のシフト戦略である. 実装では, 前回 $w_{k,k}^{(k)} < 0$ となった k の値 k_0 を保持しておき, 再度, $w_{k_0,k_0}^{(k_0)} < 0$ となった場合には, $s := 2s$ とする.

つまり, 本提案の概念図は, 図 1 のようになる.

4. 実験

提案手法の有効性を確認するために, Householder QR 分解法とシフト付き Cholesky QR 分解の比較実験を行う. 性能を比較するために, 表 1 を用いる. 実験に用いる行列は, 次の 6 種類である.

- A_1 (次元数: 1000 万 \times 100)
- A_2 (次元数: 1000 万 \times 200)
- A_3 (次元数: 1000 万 \times 300)
- A_4 (次元数: 1000 万 \times 100, 条件数: 2.62×10^{297})
- A_5 (次元数: 1000 万 \times 200, 条件数: 2.62×10^{297})
- A_6 (次元数: 1000 万 \times 300, 条件数: 2.62×10^{297})

行列 A_1, A_2, A_3 は, すべての要素を乱数で与えているため, 条件数は明らかではない

QR 分解を前処理とする特異値分解の性能を比較する. Householder QR 分解法および Cholesky QR 分解を用いた特異値分解を行うために, LAPACK [16] の DGESVD.f を基として開発を行う. 表 2 は, QR 分解を前処理とする特異値分解の結果を表す. HQR は Householder QR 分解法, CQR はシフト付き Cholesky QR 分解を意味する. 図 2 は, QR 分解を前処理とする特異値分解の結果を示す. 表 2 および図 2 より, 得られた特異ベクトル行列の直交性は, 条件数にかかわらず, Householder QR 分解法と比べ, シフト付き Cholesky QR 分解は良好である. 計算時間に関して, 乱数で作成された行列の場合, シフト付き Cholesky QR 分解の方が速い. 特に, 行列 A_3 においては, 2 倍以上の高速化が確認できる. 条件数が非常に大きな行列 A_4, A_5, A_6 に関して, Householder QR 分解法と同じような計算時間となっている. 条件数が非常に大きな行列が与えられた場合, 丸め誤差の影響が大きくなる. そのため, シフト付き Cholesky QR 分解において, 導入されたシフト量が適切ではないために, Cholesky 分解が破綻し, 再度シフト量を大きくして計算をやりなおす回数

表 2 特異値分解に関する性能

	HQR	CQR
A_1		
$\ U^T U - I\ _F$	$4.52 * 10^{-14}$	$3.89 * 10^{-14}$
$\ V^T V - I\ _F$	$3.03 * 10^{-14}$	$2.84 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$0.90 * 10^{-14}$	$0.19 * 10^{-14}$
Computation time[s]	5.28	3.20
A_2		
$\ U^T U - I\ _F$	$7.34 * 10^{-14}$	$6.53 * 10^{-14}$
$\ V^T V - I\ _F$	$5.55 * 10^{-14}$	$6.51 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$1.16 * 10^{-14}$	$0.27 * 10^{-14}$
Computation time[s]	11.00	8.43
A_3		
$\ U^T U - I\ _F$	$10.48 * 10^{-14}$	$9.13 * 10^{-14}$
$\ V^T V - I\ _F$	$8.35 * 10^{-14}$	$8.86 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$1.29 * 10^{-14}$	$0.85 * 10^{-14}$
Computation time[s]	33.23	14.80
A_4		
$\ U^T U - I\ _F$	$8.92 * 10^{-14}$	$3.50 * 10^{-14}$
$\ V^T V - I\ _F$	$2.83 * 10^{-14}$	$2.48 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$0.66 * 10^{-14}$	$0.66 * 10^{-14}$
Computation time[s]	4.34	4.13
A_5		
$\ U^T U - I\ _F$	$15.94 * 10^{-14}$	$5.99 * 10^{-14}$
$\ V^T V - I\ _F$	$4.75 * 10^{-14}$	$4.76 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$1.17 * 10^{-14}$	$0.79 * 10^{-14}$
Computation time[s]	11.93	12.11
A_6		
$\ U^T U - I\ _F$	$15.47 * 10^{-14}$	$8.66 * 10^{-14}$
$\ V^T V - I\ _F$	$7.66 * 10^{-14}$	$7.35 * 10^{-14}$
$\ A - U\Sigma V^T\ _F$	$1.14 * 10^{-14}$	$0.85 * 10^{-14}$
Computation time[s]	27.20	23.01

増加することによるものであると考えられる。しかしながら、シフトを導入することによって、条件数が非常に大きい行列に対して、Householder QR 分解法と同等の計算時間で実行を終了させることができ、計算精度についても同等である。また、Cholesky QR 分解を用いる方法は、条件数がそれほど大きくない行列において、Householder QR 分解法よりも高速に計算できる。ゆえに、提案されたシフト付き Cholesky QR 分解は有効である。

5. まとめ

本稿では、長方形の特異値分解を高速化するために、シフト付き Cholesky QR 分解を提案している。従来の方法では、長方形を上 2 重対角行列に変換するための前処理がボトルネックとなっている。この問題を解決するために、Cholesky QR 分解を適用する。ただし、既存の Cholesky QR 分解は、丸め誤差の影響により、Cholesky 分解が破綻する場合があるため、実用化されていない。そこで、本稿では、シフト付き Cholesky QR 分解に改良する。シフトを 1 度導入するだけでは、条件数が大きい場合、再度破綻する

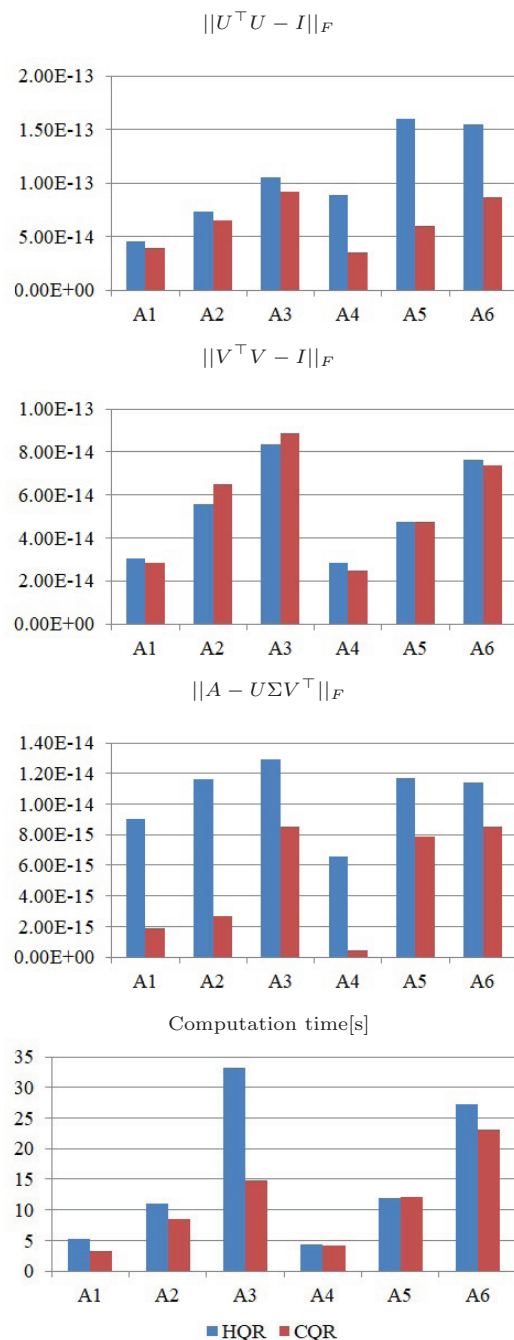


図 2 特異値分解に関する性能

場合がある。これに対処するための、シフト戦略を提案している。提案法の有効性を確認するために、Householder QR 分解法と比較実験を行った。実験の結果、特異値分解で得られたベクトル行列の直交性は、与えられた行列の性質により若干異なるが、大きな違いは確認されていない。計算時間に関して、条件数が非常に大きい行列を与えた場合はほぼ同じであるが、乱数行列を与えた場合、シフト付き Cholesky QR 分解の方が 2 倍以上速くなっている。また、実験で用いた行列に対して、2つの解法は、同等の計算精度を持つ特異値分解を得ることができている。ゆえに、提案手法は、有効であると考えられる。

今後の課題として、計算量を可能な限り削減するために、1 回目の Cholesky 分解の破綻を検出した直後に、適切なシフト量を算出するための方法を確立すべきである。

謝辞 本研究は JSPS 科研費 17H02858 の助成を受けている。

参考文献

- [1] Basic Linear Algebra Subprograms, Netlib (online), 入手先 (<http://netlib.org/blas/index.html>) (accessed 2017-11-10)
- [2] Cuppen, J.J.M.: A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numerische Mathematik*, Vol.36, pp. 177–195 (1981).
- [3] Demmel, J.: *Applied Numerical Linear Algebra*, SIAM, Philadelphia (1997).
- [4] Demmel, J. and Kahan, W.: Accurate singular values of bidiagonal matrices, *SIAM J. Sci. Sta. Comput.*, Vol.67, pp.191–229 (1994).
- [5] Dhillon, I., and Parlett, B.: Orthogonal eigenvectors and relative gaps, *SIAM J. Matrix Anal. Appl.*, Vol.25, No.3, pp.858–899 (2004).
- [6] Drmac, Z. and Veselic, K.: *New fast and accurate Jacobi SVD algorithm: I*, LAPACK WORKING NOTE 169.
- [7] Drmac, Z. and Veselic, K.: *New fast and accurate Jacobi SVD algorithm: II*, LAPACK WORKING NOTE 170.
- [8] Fernando, K.: On computing an eigenvector of a tridiagonal matrix, part 1: basic results, *SIAM J. Matrix Anal. Appl.*, Vol.18, No.4, pp.1013–1034 (1997).
- [9] Francis, J.G.F.: The QR transformation a unitary analogue to the LR transformation—part 1, *Computer J.*, Vol.4, pp.265–271 (1961).
- [10] Fukaya, T., Nakatsukasa, Y. and Yanagisawa, Y.: CholeskyQR2: A Simple and Communication-Avoiding Algorithm for Computing a Tall-Skinny QR Factorization on a Large-Scale Parallel System, *In Proceedings of 2014 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, pp.31–38 (2014).
- [11] Golub, G. and Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix, *SIAM J. Numer. Anal.*, Vol. 2, pp.205–224 (1965).
- [12] Golub, G. and Reinsch, C.: Singular value decomposition and least squares solutions, *Numer. Math.*, Vol.14, pp.403–420 (1970).
- [13] Gu, M. and Eisenstat, S.C.: A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, *SIAM Journal on Matrix Analysis and Applications*, Vol.16, No.1, pp. 172–191 (1995).
- [14] Iwasaki, M. and Nakamura, Y.: On the convergence of a solution of the discrete Lotka-Volterra system, *Inverse Problems*, Vol.18, pp.1569–1578 (2002).
- [15] 岩崎雅史, 阪野真也, 中村佳正: 実対称 3 重対角行列の高精度ツイスト分解とその SVD への応用, *応用数学会論文誌*, No.15, Vol.3, pp.461–481 (2005).
- [16] LAPACK-Linear Algebra PACKage 入手先 (<http://www.netlib.org/lapack/>) (accessed 2017-11-10)
- [17] Parlett, B., and Dhillon, I.: Fernando’s solution to wilkinson’s problem: An application of double factorization, *Lin. Alg. Appl.*, Vol.267, pp.247–279 (1981).
- [18] Takata, M., Araki, S., Kimura, K., Fujii, Y. and Nakamura, Y.: Implementation of Computing Singular Pairs for Large Scale Matrices using ARPACK, *In Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, pp.349–355 (2016).
- [19] 高田雅美, 木村欣司, 岩崎雅史, 中村佳正: 高速特異値分解のためのライブラリ開発, *情報処理学会論文誌コンピューティングシステム*, Vol.47, No.SIG7 (ACS14), pp.91–104 (2006).
- [20] Veselic, K. and Hari, V.: A Note on a One-sided Jacobi Algorithm, *Numer. Math.*, Vol.56, pp.627–633 (1989).
- [21] Yamamoto, Y., Nakatsukasa, Y., Yanagisawa, Y. and Fukaya, T.: Roundoff error analysis of the CholeskyQR2 algorithm in an oblique inner product, *JSIAM Letters*, Vol.8, pp.5–8 (2016).