6J-05

BLR 行列の MBGS アルゴリズムによる QR 分解の Python 実装とその性能評価

板垣　翼 †　　鈴木　智博 ‡

山梨大学大学院工学専攻コンピュータ理工学コース †　　山梨大学医学工学総合研究部 ‡

1　はじめに

科学技術アプリケーションでは大規模な行列計算が行われる．この様な大規模化は，メモリ消費量や計算量の増大を引き起こす．これに対して，行列を階層的に分割し，その部分行列を低ランク近似することで，メモリ消費量と計算量を抑える低ランク近似手法が存在する．その手法の一つである BLR 行列（Block Low Rank Matrices）は，階層的な構造を持たない最も単純なものである．

密行列や疎行列に対する効率的な行列分解の手法は既に整いつつあり，LAPACK（Linear Algebra Package）を始とする様々なアプリケーションで用いられている．しかし今のところ，BLR 行列などの低ランク近似行列に対する行列分解の研究は多くない．我々は，その効率的な行列分解の開発に取り組んでいる．BLR 行列は単純な格子構造を持つため，既存のブロックアルゴリズムの適用が期待できる．今回は，ブロック QR 分解を行うための MBGS アルゴリズム（Modified Block Gram-Schmidt Algorithm）[1] の BLR 行列への適用と Python による実装を行い，その性能評価と検討を行う．

2　BLR 行列

BLR 行列は，図 2.1 の様な格子状の構造を持ち，対角の小行列には密行列，非対角の小行列に低ランク行列を持つ．小行列のサイズは全て同一である．ここで低ランク行列とは，ある行列 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ に対して，2 つの行列 $\boldsymbol{U} \in \mathbb{R}^{m\times r}$，$\boldsymbol{V} \in \mathbb{R}^{r\times n}$（$r \ll m, n$）が以下を満たすときの行列 $\boldsymbol{UV}$ である．

$$\|\boldsymbol{A} - \boldsymbol{UV}\| \leq \varepsilon \quad (\varepsilon > 0)$$

ε は近似の度合を示す任意の正数であり，小さいほど元の行列を良く近似していると言える．

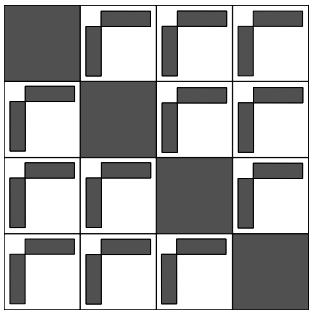


図 2.1　BLR 行列の構造例．対角の小行列に密行列，非対角の小行列に低ランク近似行列を持つ．小行列は全て同一サイズである．また色の付いた部分は行列の要素を表している．

2.1　2 種類の低ランク近似手法

行列の低ランク近似には様々な手法があり，ここでは有名な Truncated SVD（Singular Value Decomposition）と ACA（Adaptive Cross Approximation）の 2 つを紹介する．

Truncated SVD では，まず行列の特異値分解を行い，近似の度合 ε に従うように分解後の行列をスライスし，2 つの行列の積になるように特異値を左右どちらかの特異ベクトル行列に乗ずる．この手法は，与えられた行列に対して最も小さなランクで近似することが可能である一方，そのアルゴリズムに特異値分解を持つため計算量が非常に大きくなるという欠点を持つ．

ACA では，低ランク行列 $\boldsymbol{UV}$ を元の行列の列ベクトルや行ベクトルを選択しながら求めていく [3]．Truncated SVD に比べ計算量が非常に小さいため高速に低ランク行列を生成することが可能である．しかし，元の行列の特性によっては収束が満足に行われない場合もあることがわかっている [2]．

2.2　基本算法

BLR 行列の行列分解では，密行列と低ランク行列の 2 種類の小行列の混在した計算が生じる．計算結果の BLR 行列の構造が前のものと同様な構造を持つようにするためには，密行列どうしの通常算法に加え，低ランク行列に関する以下の 4 つの演算を定義する必要が有る．

（1）低ランク行列 + 低ランク行列 = 低ランク行列
（2）低ランク行列 + 密行列 = 密行列
（3）低ランク行列 × 低ランク行列 = 低ランク行列
（4）低ランク行列 × 密行列 = 低ランク行列

特に初めの低ランク行列同士の加算にはランク数を減らすために特別な演算が必要であり，Rounded Addition という手法を用いる [1]．

3　QR 分解

QR 分解とは，行列 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ を直交行列 $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ と上三角行列 $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ に分解することであり，最小二乗問題や固有値問題などに用いられる．

QR 分解のブロックアルゴリズムには様々な手法が存在する．今回扱う MBGS アルゴリズム [1]の手順をアルゴリズム 3.1 に示す．

アルゴリズム 3.1 MBGS アルゴリズム

Input: $N_B \times N_B$ の行列 $\boldsymbol{A}$（N_B はブロック数）
Output: $N_B \times N_B$ の行列 $\boldsymbol{Q}$, $\boldsymbol{R}$

```
for j = 1 to N_B
    Q_{*j}, R_{jj} ← A_{*j}                ▷ 縦長行列の QR 分解
    for k = j + 1 to N_B
        R_{jk} ← Q_{*j}^T A_{*k}
        A_{*k} ← A_{*k} − Q_{*j} R_{jk}
    end for
end for
```

BLR 行列は単純なブロック構造を持つため，既存のブロックアルゴリズムを適用させやすい．アルゴリズム 3.1 に関しては，2 行目の縦長行列の QR 分解に修正を加えることで BLR 行列を QR 分解することができる [1]．そして 4・5 行目では 2.2 節で定めた演算を行う．

Python implementation of QR factorization for BLR matrices with MBGS algorithm

† Tsubasa Itagaki, Integrated Graduate School of Computer Science and Engineering, University of Yamanashi
‡ Tomohiro Suzuki, Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi

Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.

4 性能評価

以上で述べた性能を評価するため BLR 行列と MBGS アルゴリズムを用いた BLR 行列の QR 分解を Python で実装した．基本的なデータ構造やアルゴリズムには既存の NumPy パッケージを積極的に採用し可能な限り高速化を行った．性能評価で用いた実行環境は以下である．

OS Ubuntu 18.04.3 LTS
CPU Intel ® Core ™ i5-5200U CPU @ 2.20 GHz
メモリ 8 GB
Python version 3.7.2
NumPy version 1.17.3

性能評価に用いるテスト行列はヒルベルト行列とした．また生成する BLR 行列のブロックサイズは，[1]を参考に $10\sqrt{n}$ とし，近似の閾値 ε は 10^{-4} に設定した．

4.1 実行時間

まず MBGS アルゴリズムを用いた BLR 行列の QR 分解の実行時間を調べる．比較対象には NumPy の QR 分解ルーチン `numpy.linalg.qr` も加え，公平性のため BLR 行列の QR 分解の実行時間に関しては，BLR 行列の生成時間（圧縮時間）と QR 分解の実行時間，そして元の密行列へ戻す時間を合計したものとした．結果を図 4.1 に示す．

これより ACA を用いた BLR 行列の QR 分解が最も高速であることがわかる．これは Truncated SVD に比べ ACA による BLR 行列の生成の方が非常に高速であるためである．また `numpy.linalg.qr` の計算量がおよそ $\mathcal{O}(N^3)$ であるのに対し，BLR 行列の QR 分解の計算量はおよそ $\mathcal{O}(N^2)$ であることがわかる．

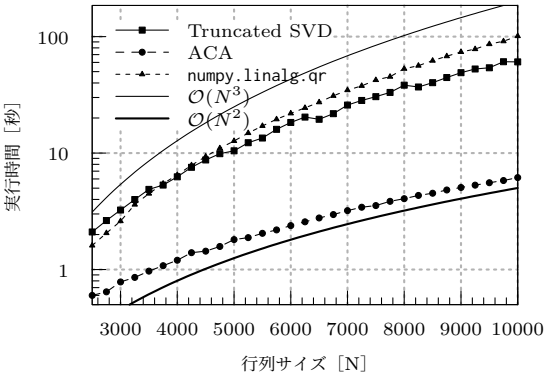


図 4.1 QR 分解の実行時間の比較.

4.2 近似精度と直交性

次に BLR 行列の QR 分解により得られる $\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{R}}$ の近似精度（相対誤差）と $\tilde{\boldsymbol{Q}}$ の直交性を調べる．結果をそれぞれ図 4.2 と図 4.3 に示す．

これより，QR 分解の近似精度は初めに与えた近似閾値 $\varepsilon = 10^{-4}$ に従うように近似されていることがわかる．また Truncated SVD は，ACA よりも低ランク行列のランクが小さくなるため，近似精度が悪くなったと言える．一方，直交性に関しては，どちらの手法も良好でない．これは今回のテスト行列であるヒルベルト行列が条件数の非常に大きい行列なためであると考えられる．

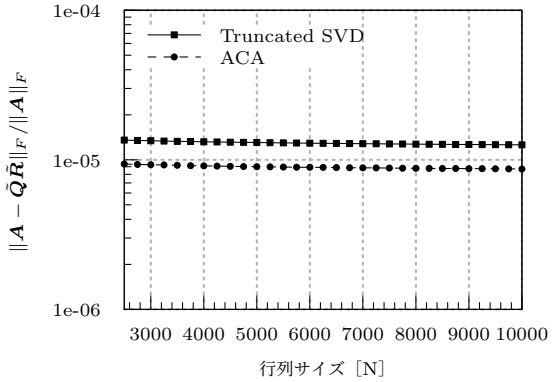


図 4.2 BLR 行列の QR 分解の近似精度の比較.

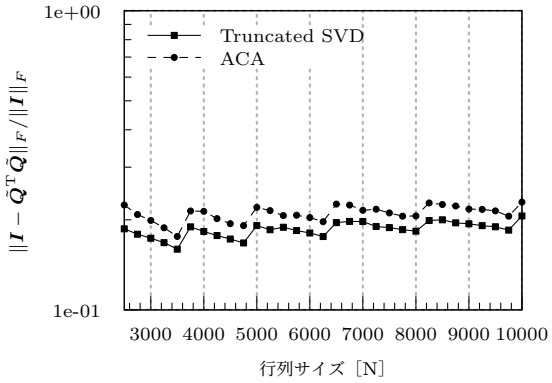


図 4.3 BLR 行列の QR 分解の直交性の比較.

5 まとめ

今回は MBGS アルゴリズムを用いた BLR 行列の QR 分解を実装しその性能評価を行った．ACA を用いた BLR 行列では，既存の QR 分解ルーチン `numpy.linalg.qr` に比べおよそ 10 倍程度高速化することがわかった．また QR 分解の近似精度も予め与える近似の度合 ε に従い近似されることが確認できた．しかし条件数の悪い行列に対しては，$\tilde{\boldsymbol{Q}}$ の直交性が悪くなるという問題を確認した．今後の課題は，悪条件の行列に対しても $\tilde{\boldsymbol{Q}}$ の直交性が失われないようなアルゴリズムを開発である．そのため現在は，$\tilde{\boldsymbol{Q}}$ の再直行化やハウスホルダー変換を用いたブロック QR 分解の適用を検討している．

参考文献

[1] Akihiro Ida, Hiroshi Nakashima, Tasuku Hiraishi, Ichitaro Yamazaki, Rio Yokota, and Takeshi Iwashita. QR Factorization of Block Low-rank Matrices with Weak Admissibility Condition. *Journal of Information Processing*, Vol. 27, pp. 831–839, 2019.

[2] Yang Liu, Wissam Sid-Lakhdar, Elizaveta Rebrova, Pieter Ghysels, and Xiaoye Sherry Li. A Parallel Hierarchical Blocked Adaptive Cross Approximation Algorithm. 2019.

[3] Kezhong Zhao, M.N. Vouvakis, and Jin-Fa Lee. The Adaptive Cross Approximation Algorithm for Accelerated Method of Moments Computations of EMC Problems. *Electromagnetic Compatibility, IEEE Transactions on*, Vol. 47, pp. 763 – 773, 12 2005.

Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.