情報科学類専門語学 A

知能情報メディア主専攻　201611350　江畑 拓哉
指導教員　櫻井哲也 (コンピュータサイエンス専攻)

提出日　2018 年 8 月 3 日

1　文献

Lars Elde n., Matrix Methods in Data Mining and Pattern Recognition., SIAM, 2007.

2　概要

“Matrix Methods in Data Mining and Pattern Recognition” は行列の固有値分解などの基礎的な手法からパターン認識やデータマイニングなどの問題に対する解法について説明されている文献である。今回はその中でも第 1 部である chapter1 から chapter9 までを取り上げて輪講を行った。以下はその内自分が担当した内容について説明したものである。

3　Chapter 5 QR 分解

QR 分解 という行列の圧縮手法がある。これは、ある行列 $\boldsymbol{A}$ を直交行列と三角行列に因数分解する手法である。これは 2 つの要素が両方共三角行列であることだけが要求されるという点で、LU 分解よりも広い範囲をカバーしている。[1]

3.1　直交変換を用いた三角行列への変換

直交変換の一つであるハウスホルダー変換を用いることで、任意の行列 $\boldsymbol{A} \in \mathbb{R}^{m \times n}\ where\ m \geq n$ について上三角行列 $\boldsymbol{R}$ と直交行列 $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ を用いた関係式を作ることが出来る。

$$\boldsymbol{A} \rightarrow \boldsymbol{Q}^T \boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix} \qquad where\ \boldsymbol{R}\ \in\ \mathbb{R}^{n \times n}$$

以下に具体的に分解を行った例を示す。

$\boldsymbol{A} \in \mathbb{R}^{5 \times 4}$ である場合
最初のステップでは 1 列目の上から 1 番目より下の要素をゼロにする。

[1] LU 分解には分解する行列が正則であるという適用条件がある。

$$\boldsymbol{H}_1\boldsymbol{A} = \boldsymbol{H}_1 \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} + & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{pmatrix}$$

$+$ の値は $\times$ に対して変形が行われ値が変わっているという事を示している。直交行列である $\boldsymbol{H}_1$ の適用は、ハウスホルダー変換を行うことに等しい。

次のステップでは変換した $\boldsymbol{A}$ に対して、前のステップと同様に 2 列目の上から 2 番目より下の要素をゼロにして、

$$\boldsymbol{H}_2 \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & + & + & + \\ 0 & 0 & + & + \\ 0 & 0 & + & + \\ 0 & 0 & + & + \end{pmatrix}$$

三番目のステップでも同様に変換して、

$$\boldsymbol{H}_3 \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & + & + \\ 0 & 0 & 0 & + \\ 0 & 0 & 0 & + \end{pmatrix}$$

四番目のステップでも同様にして、上三角行列 $\boldsymbol{R}$ を得ることが出来る。

この変換を要約すると、

$$\boldsymbol{Q}^T\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix} \qquad where\ \boldsymbol{Q}^T = \boldsymbol{H}_4\boldsymbol{H}_3\boldsymbol{H}_2\boldsymbol{H}_1$$

また、 $\boldsymbol{H}_i\ where\ \boldsymbol{A} \in \mathbb{R}^{m\times n}$ の構造は以下のようになる。

$$\boldsymbol{H}_1 = \boldsymbol{I} - 2\boldsymbol{u}_1\boldsymbol{u}_1^T \qquad where\ \boldsymbol{u}_1 \in \mathbb{R}^m$$

$$\boldsymbol{H}_2 = \begin{pmatrix} 1 & 0 \\ 0 & \boldsymbol{P}_2 \end{pmatrix} \qquad where\ \boldsymbol{P}_2 = \boldsymbol{I} - 2\boldsymbol{u}_2\boldsymbol{u}_2^T,\ \boldsymbol{u}_2 \in \mathbb{R}^{m-1}$$

$$\boldsymbol{H}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \boldsymbol{P}_3 \end{pmatrix} \qquad where\ \boldsymbol{P}_3 = \boldsymbol{I} - 2\boldsymbol{u}_3\boldsymbol{u}_3^T,\ \boldsymbol{u}_3 \in \mathbb{R}^{m-2}$$

このようにして、単位行列に連続して小さくなっていくハウスホルダー変換を埋め込んでいき、それと同時にベクトル $\boldsymbol{u}_i$ の次元も小さくなる。

3.2 QR 分解

Theorem 5.1 QR 分解

どのような行列 $\boldsymbol{A}$ $where$ $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m \geq n$ についても直交行列によって上三角行列に変形することが出来る。またこの変形は以下の行列の圧縮に等しい。

$$\boldsymbol{A} = \boldsymbol{Q}\begin{pmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{pmatrix} \qquad where\, \boldsymbol{Q} \in \mathbb{R}^{m\times m}\ is\ orthogonal$$

$$\boldsymbol{R} \in \mathbb{R}^{n\times n}\ is\ upper\ triangular$$

(orthogonal matrix=直交行列,upper triangular matrix=上三角行列)　もし $\boldsymbol{A}$ が列について線形独立であるならば $\boldsymbol{R}$ は正則である。

Proof

単位ベクトルに変換されるベクトルがゼロベクトルであれば選ばれた直交変換が恒等行列に等しい、という条件のもとで、一般的なケースに適用することは容易である。　以下の行列の列について線形独立性を考える。

$$\begin{pmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{pmatrix}$$

$\boldsymbol{R}$ は上三角行列であることから、線形独立性より対角成分は非ゼロ。(もし対角成分にゼロがある場合には、その列の左側の列との線型結合があると言える) そして $\boldsymbol{R}$ の行列式が非ゼロならば、 $\boldsymbol{R}$ は正則である。

QR 分解を抽象的に示すと、以下のようになる。

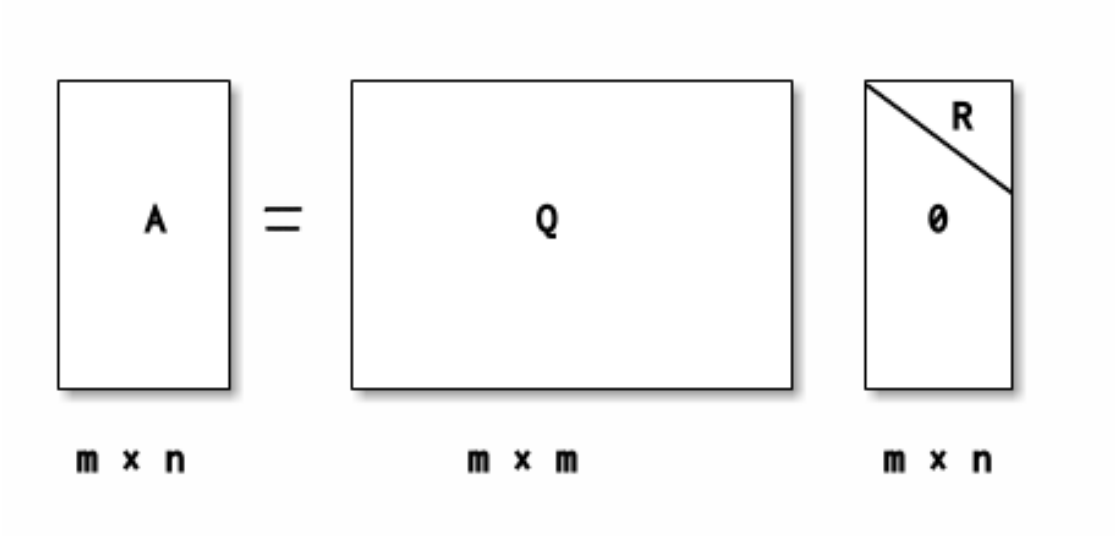

図 1: QR 分解の図

A の列の直交化に対応する Q の部分のみを用いる代替手段は分解を行う際にしばしば有用である。その薄い QR 分解は $\boldsymbol{Q} = (\boldsymbol{Q}_1\boldsymbol{Q}_2)\ where\, \boldsymbol{Q}_1 \in \mathbb{R}^{m\times n}$ に分割することで行われる。 $\boldsymbol{Q}_2$ が 0 に掛けられても 0 となることに注意しなければならない。

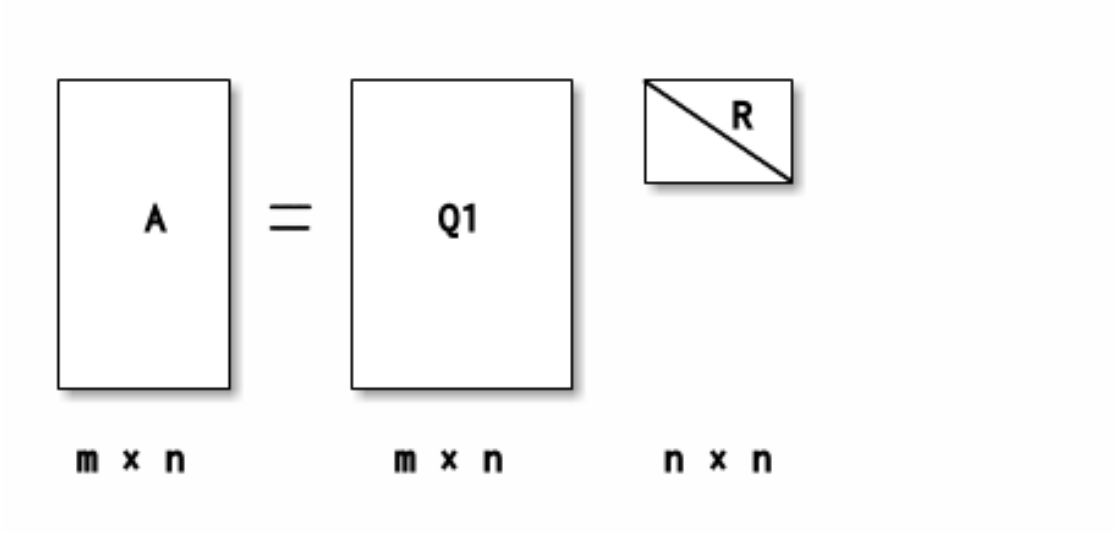

図 2: 薄い QR 分解の図

$$\boldsymbol{A} = (\boldsymbol{Q}_1 \boldsymbol{Q}_2) \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix} = \boldsymbol{Q}_1 \boldsymbol{R}$$

この方程式から、$\boldsymbol{R}(\boldsymbol{A}) = \boldsymbol{R}(\boldsymbol{Q}_1)$ がわかる。以上で値空間 $\boldsymbol{R}(\boldsymbol{A})$ の直交基底を計算したことになる。更に上式において j 列を書き出すと、

$$\boldsymbol{a}_j = \boldsymbol{Q}_1 \boldsymbol{r}_j = \Sigma_{i=1}^{j} r_{ij} \boldsymbol{q}_i$$

$\boldsymbol{R}$ の j 列は直交基底の $\boldsymbol{a}_j$ の位置を保持することがわかる。

Example 5.2 単純な数を用いた QR 分解

以下に Octave (Matlab) での QR 分解の実行を示す。

Octave Code

```
A = [1 1 1;
     1 2 4;
     1 3 9;
     1 4 16];
[Q,R] = qr(A)
```

Output

```
Q =

  -0.50000   0.67082   0.50000   0.22361
  -0.50000   0.22361  -0.50000  -0.67082
  -0.50000  -0.22361  -0.50000   0.67082
```

```
  -0.50000  -0.67082   0.50000  -0.22361

R =

   -2.00000   -5.00000  -15.00000
    0.00000   -2.23607  -11.18034
    0.00000    0.00000    2.00000
    0.00000    0.00000    0.00000
```

薄い QR 分解は $qr(A,0)$ コマンドで実行する。

Octave Code

```
A = [1 1 1;
     1 2 4;
     1 3 9;
     1 4 16];
[Q,R] = qr(A,0)
```

Output

```
Q =

  -0.50000   0.67082   0.50000
  -0.50000   0.22361  -0.50000
  -0.50000  -0.22361  -0.50000
  -0.50000  -0.67082   0.50000

R =

   -2.00000   -5.00000  -15.00000
    0.00000   -2.23607  -11.18034
    0.00000    0.00000    2.00000
```

3.3 最小二乗問題

QR 分解を用いて、以下の最小二乗問題を 正規方程式[2]を形成することなく解くことが出来る。これを行うために、ユークリッドベクトルノルムは直交変換の元で変わらないという事実を利用する。

$$\min_x ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2 \qquad where\ \boldsymbol{A}\ \in\ \mathbb{R}^{m\times n},\ m \geq n \tag{1}$$

$$||\boldsymbol{Q}\boldsymbol{y}||_2 = ||\boldsymbol{y}||_2$$

[2] $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$

残差ベクトルに A についての QR 分解を用いて、

$$||\boldsymbol{r}||_2^2 = ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2^2 = ||\boldsymbol{b} - \boldsymbol{Q}\begin{pmatrix}\boldsymbol{R}\\0\end{pmatrix}\boldsymbol{x}||_2^2$$
$$= ||\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{b} - \begin{pmatrix}\boldsymbol{R}\\0\end{pmatrix}\boldsymbol{x})||_2^2 = ||\boldsymbol{Q}^T\boldsymbol{b} - \begin{pmatrix}\boldsymbol{R}\\0\end{pmatrix}\boldsymbol{x}||_2^2$$

ここで $\boldsymbol{Q} = (\boldsymbol{Q}_1\ \boldsymbol{Q}_2),\ where\ \boldsymbol{Q}_1 \in \mathbb{R}^{m\times n}$ と分割して以下の式を導く。

$$\boldsymbol{Q}^T\boldsymbol{b} = \begin{pmatrix}\boldsymbol{b}_1\\\boldsymbol{b}_2\end{pmatrix} := \begin{pmatrix}\boldsymbol{Q}_1^T\boldsymbol{b}\\\boldsymbol{Q}_2^T\boldsymbol{b}\end{pmatrix}$$

即ち残差ベクトルの式は以下のように変形できる。

$$||\boldsymbol{r}||_2^2 = ||\begin{pmatrix}\boldsymbol{b}_1\\\boldsymbol{b}_2\end{pmatrix} - \begin{pmatrix}\boldsymbol{R}\boldsymbol{x}\\0\end{pmatrix}||_2^2 = ||\boldsymbol{b}_1 - \boldsymbol{R}\boldsymbol{x}||_2^2 + ||\boldsymbol{b}_2||_2^2 \tag{1}$$

さらに A が線形独立であると仮定した場合、以下の式を満たす値を求めることで $||\boldsymbol{r}||_2$ を最小化する値を求めることが出来る。

$$\boldsymbol{R}\boldsymbol{x} = \boldsymbol{b}_1$$

ここで次の定理が成り立つことになる。

Theorem 5.3 QR 分解を用いた最小二乗問題の解

列についてフルランクであり、QR 分解によって $\boldsymbol{A} = \boldsymbol{Q}_1\boldsymbol{R}$ となる行列 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ の最小二乗問題 $min_x||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2$ は以下の唯一解を持つ。

$$\boldsymbol{x} = \boldsymbol{R}^{-1}\boldsymbol{Q}_1^T\boldsymbol{b}$$

Example 5.4 QR 分解を用いて最小二乗問題を解く

以下に Octave (Matlab) での QR 分解を用いた最小二乗問題の解法を示す。
尚 MATLAB では $x = A\backslash b$ とすると同じアルゴリズムで解を求めることが出来る。

Octave Code

```
A = [1 1;
     1 2;
     1 3;
     1 4;
     1 5];
b = [7.9700;
     10.2000;
     14.2000;
     16.0000;
     21.2000];
```

```
# thin QR
[Q1,R]=qr(A,0)
x=R\(Q1'*b)
```

Output

```
Q1 =
  -4.4721e-01  -6.3246e-01
  -4.4721e-01  -3.1623e-01
  -4.4721e-01   2.7756e-17
  -4.4721e-01   3.1623e-01
  -4.4721e-01   6.3246e-01
R =
  -2.23607  -6.70820
   0.00000   3.16228
x =
   4.2360
   3.2260
```

4 Chapter 7 Krylov 部分空間法

次元削減を行うための手法として打ち切り SVD(Truncated SVD) や PCR(主成分回帰) を用いることがあるが、分解後の右辺の残差を減らすように基底を選択出来る手法として Lanczos-Golub-Kahan (LGK) 二重対角化がある。

LGK 二重対角化は線形代数の分野で用いられていおり、 Lanczos 二重対角化とも呼ばれる。また計量科学やその他の分野では部分的最小二乗法 (PLS(Partial Least Squares/Projection Latent structures)) に密接して関連した方法として知られている。Krylov 部分空間法 の中から来た手法であり、疎な線形問題 (sparse linear systems) や、固有値・特異値の計算で用いられる。

Krylov 部分空間法は再帰的であるが、始めに導出としてハウスホルダー変換を用いて行列を二重対角化する手法を扱う。

4.1 ハウスホルダー変換を用いた二重対角化

密行列 $\boldsymbol{C} \in \mathbb{R}^{m\times(n+1)}$ の SVD を計算するアルゴリズムの最初のステップは、左から右へハウスホルダー変換をし、上二重対角の形に $\boldsymbol{C}$ を変換することである。
$m > n$ を仮定すると、以下の形が求まる。

$$\boldsymbol{C} = \boldsymbol{P}\begin{pmatrix}\hat{\boldsymbol{B}}\\ \boldsymbol{0}\end{pmatrix}\boldsymbol{W}^T$$

$$where\ \boldsymbol{P}, \boldsymbol{W}\ is\ orthogonal$$

$$\hat{\boldsymbol{B}}\ is\ upperbidiagonal$$

この自己圧縮は $\Rightarrow$ 疎・密行列の最小二乗問題の近似解を求める際などでも有用である。
例として、 $\boldsymbol{C} \in \mathbb{R}^{6\times5}$ を用いて説明する。※ $m = 6,\ n+1 = 5$

まず 1 列目の非対角成分を、左側からかける変換行列 $\boldsymbol{P}_1^T \in \mathbb{R}^{6\times 6}$ を用いて 0 にする。

$$\boldsymbol{P}_1^T\boldsymbol{C} = \boldsymbol{P}_1^T \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix}$$

次に、別の右からかける変換行列 $\boldsymbol{W}_1$ を用いて 1 行目の 3 番目の要素から右側をすべて 0 にする。 $\boldsymbol{W}_1$ は以下のように表すことができる。

$$\mathbb{R}^{5\times 5} \ni \boldsymbol{W}_1 = \begin{pmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_1 \end{pmatrix} where\ \boldsymbol{Z}_1\ is\ Householder\ transformation$$

これを用いた変換は 1 列目の要素に対しては何も影響しない。つまり前に行った 1 列目の非対角成分を 0 にした変換の効果は打ち消されない。結果は以下のようになる。

$$\boldsymbol{P}_1^T\boldsymbol{C}\boldsymbol{W}_1 = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{pmatrix} \boldsymbol{W}_1 = \begin{pmatrix} \times & * & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix} =: \boldsymbol{C}_1$$

同様の手法を用いて 2 列目の対角成分よりも下の成分を 0 にする変換行列 $\boldsymbol{P}_2$ を $\boldsymbol{C}_1$ にかける 。但し 1 行目の要素については値が変わらないようにする。

すると、 $\boldsymbol{P}_2$ は以下のように表すことができる。

$$\mathbb{R}^{6\times 6} \ni \boldsymbol{P}_2 = \begin{pmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{P}}_2 \end{pmatrix} where\ \tilde{\boldsymbol{P}}_2 \in \mathbb{R}^{5\times 5}\ is\ Householder\ transformation$$

これを適用して、

$$\boldsymbol{P}_2^T\boldsymbol{C}_1 = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix}$$

そして同様に右からかける変換行列 $\boldsymbol{W}_2$ は以下のように表すことができる。

$$\boldsymbol{W}_2 = \begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 \end{pmatrix},\ \boldsymbol{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

これも $\boldsymbol{W}_1$ 同様 $\boldsymbol{P}_2$ で 0 にした部分を汚すことなく 2 列目の 4 行目以降の要素を 0 にする。

$$\boldsymbol{P}_2^T\boldsymbol{C}_1\boldsymbol{W}_2 = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & * & 0 & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix} =: \boldsymbol{C}_2$$

この操作を繰り返すことで最終的に以下の形を得る。

$$\boldsymbol{P}^T\boldsymbol{C}\boldsymbol{W} = \begin{pmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \\ & & & & \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{B}} \\ \boldsymbol{0} \end{pmatrix}$$

$$where\ \boldsymbol{P} = \boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_n \in \mathbb{R}^{m\times m}$$

$$\boldsymbol{W} = \boldsymbol{W}_1\boldsymbol{W}_2\cdots\boldsymbol{W}_{n-1} \in \mathbb{R}^{(n+1)\times(n+1)} \tag{2}$$

一般的に上で示された $\boldsymbol{P}$, $\boldsymbol{W}$ はハウスホルダー変換の積であり、

$$\hat{\boldsymbol{B}} = \begin{pmatrix} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \ddots & \ddots & \\ & & & \beta_n & \alpha_n \\ & & & & \beta_{n+1} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

は上二重対角行列である。
これらは、この章の残りの部分で用いられる直交行列を生成するための構造を持っている。

Proposition 7.3

二重対角分解 (2) 式における $\boldsymbol{P}$ の列を $\boldsymbol{p}_i, i = 1, 2, \ldots, m$ として表すと以下のように表すことが出来る。

$$\boldsymbol{p}_1 = \boldsymbol{\beta}_1\boldsymbol{c}_1,\ \boldsymbol{W} = \begin{pmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} \end{pmatrix}\ where\ \boldsymbol{Z} \in \mathbb{R}^{n\times n}\ is\ orthogonal$$

c_1 は $\boldsymbol{C}$ の最初の列である。

Proof

$i = 1$ の場合は、 $\boldsymbol{P}^T\boldsymbol{c}_1 = \beta_1\boldsymbol{e}_1$ であったことから明らかである。これ以降の場合では、 $\boldsymbol{W}_i$ が以下の構造で表されていたことからわかる。

$$\boldsymbol{W}_i = \begin{pmatrix} \boldsymbol{I}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_i \end{pmatrix}$$

$$where\ \boldsymbol{I}_i \in \mathbb{R}^{i\times i}\ is\ identicaly\ matrices$$

$$\boldsymbol{Z}_i are\ orthogonal$$

ハウスホルダー変換を用いた二重対角化への削減は $4mn^2 - 4n^3/3$ flops かかる。
もし $m \gg n$ ならば、 $\boldsymbol{A}$ を上三角行列にして、 $\boldsymbol{R}$ 要素を二重対角化したほうが良い。[3]

最小二乗問題 $\min_x ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2,\ where\ \boldsymbol{A} \in \mathbb{R}^{m\times n}$ を解く場合について考える。
二重対角化において $\boldsymbol{C} = (\boldsymbol{b}\ \boldsymbol{A})$ とした場合、同等の二重対角化最小二乗問題 (bidiagonal least squares problem) を得ることが出来る。式 (3) と命題 7.3 より以下を得る。

$$\boldsymbol{P}^T\boldsymbol{C}\boldsymbol{W} = \boldsymbol{P}^T \begin{pmatrix} \boldsymbol{b} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}^T\boldsymbol{b} & \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{Z} \end{pmatrix} = \begin{pmatrix} \beta_1 & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \tag{3}$$

$$where$$

$$\boldsymbol{B} = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \in \mathbb{R}^{(n+1)\times n}$$

そして $\boldsymbol{y} = \boldsymbol{Z}^T\boldsymbol{x}$ として残差の 2 ノルムを以下のように書くことができる。

$$\begin{aligned} ||\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}||_2 &= || \begin{pmatrix} \boldsymbol{b} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{x} \end{pmatrix} ||_2 = ||\boldsymbol{P}^T \begin{pmatrix} \boldsymbol{b} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{y} \end{pmatrix} ||_2 \\ &= || \begin{pmatrix} \boldsymbol{P^T b} & \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{X} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{y} \end{pmatrix} ||_2 = ||\beta_1\boldsymbol{e}_1 - \boldsymbol{B}\boldsymbol{y}||_2 \end{aligned} \tag{4}$$

もし平面回転の操作によって $\boldsymbol{B}$ が上二重対角行列に変換される場合、この二重対角最小二乗問題 $\min_y ||\beta_1\boldsymbol{e}_1 - \boldsymbol{B}\boldsymbol{y}||_2$ は $O(n)$ flops で解くことが出来る。

4.2 LGK 二重対角化

ここでは前で述べた二重対角化手法ではない別の手法を扱う。この手法は式 (3) の計算を再帰的に解く。これを LGK 二重対角化 という。式 (3) の最後の式は以下のように書くことが出来る。

$$\boldsymbol{P}^T\boldsymbol{A} = \begin{pmatrix} \boldsymbol{B}\boldsymbol{Z}^T \\ \boldsymbol{0} \end{pmatrix} where\ \boldsymbol{B}\boldsymbol{Z}^T \in \mathbb{R}^{(n+1)\times n}$$

これは更に詳しく書けば以下のようになる。

$$\begin{aligned} \boldsymbol{A}^T \begin{pmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_{n+1} \end{pmatrix} &= \boldsymbol{Z}\boldsymbol{B}^T \\ &= \begin{pmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_n \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_2 & & & & & & \\ & \alpha_2 & \beta_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & & \beta_i & & & \\ & & & & \alpha_i & & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \alpha_n & \beta_{n+1} \end{pmatrix} \end{aligned}$$

[3] R とは LU 分解で言う U 要素 (LU 分解の別名は LR 分解)

両辺の i 列 $(i \geq 2)$ を比較すると以下のような式が考えられる。

$$\boldsymbol{A}^T \boldsymbol{p}_i = \beta_i \boldsymbol{z}_{i-1} + \alpha_i \boldsymbol{z}_i$$

変形して

$$\alpha_i \boldsymbol{z}_i = \boldsymbol{A}^T \boldsymbol{p}_i - \beta_i \boldsymbol{z}_{i-1} \tag{5}$$

同様に i 列について、$(\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{A}\boldsymbol{Z} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{Z}^T\boldsymbol{Z})$

$$\boldsymbol{A}\boldsymbol{Z} = \boldsymbol{A}\begin{pmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_n \end{pmatrix}$$

$$= \boldsymbol{P}\boldsymbol{B} = \begin{pmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_{n+1} \end{pmatrix} \begin{pmatrix} \alpha_1 & & & & & & \\ \beta_2 & \alpha_2 & & & & & \\ & \ddots & \ddots & & & & \\ & & & \alpha_i & & & \\ & & & \beta_{i+1} & & & \\ & & & & \ddots & \ddots & \\ & & & & & \beta_n & \alpha_n \\ & & & & & & \beta_{n+1} \end{pmatrix}$$

同様に以下のような式が考えられる。

$$\boldsymbol{A}\boldsymbol{z}_i = \alpha_i \boldsymbol{p}_i + \beta_{i+1} \boldsymbol{p}_{i+1}$$

変形して

$$\beta_{i+1} \boldsymbol{p}_{i+1} = \boldsymbol{A}\boldsymbol{z}_i - \alpha_i \boldsymbol{p}_i \tag{6}$$

等式 $\beta_1 \boldsymbol{p}_1 = \boldsymbol{b}$ を初期値として等式 (5), (6) を再帰的に解くことが出来る。

LGK 二重対角化

---

1. $\beta_1 \boldsymbol{p}_1 = \boldsymbol{b}, \boldsymbol{z}_0 = 0$
2. $for\ i\ =\ 1\ :\ n$
$\backslash\ \backslash\ \backslash\ \alpha_i \boldsymbol{z}_i = \boldsymbol{A}^T \boldsymbol{p}_i - \beta_i \boldsymbol{z}_{i-1}$
$\backslash\ \backslash\ \backslash\ \beta_{i+1} \boldsymbol{p}_{i+1} = \boldsymbol{A}\boldsymbol{z}_i - \alpha_i \boldsymbol{p}_i$
3. end

係数 α_{i-1}, β_i は $||\boldsymbol{p}_i|| = ||\boldsymbol{z}_i|| = 1$ となるように決定する。

再帰の終点は α_i 又は β_i がゼロと等しくなったときである。最小二乗問題の解法においてこれは明確に定義されている特別な場合であることがわかっているので、終点とみなしても問題がない。

正確な算術手続きを考えれば、再帰的二重対角化の手続きはハウスホルダー変換を用いた二重対角化と等しくなる。従ってこの過程で生成された $(\boldsymbol{p_i})_{i=1}^n\,(\boldsymbol{z}_i)_{i=1}^n$ は $\boldsymbol{p}_i^T \boldsymbol{p}_j = 0,\ \boldsymbol{z}_i^T \boldsymbol{z}_j = 0\ if\ i \neq j$ を満たす。

しかし浮動点計算を考慮すると、生成されるベクトルは再帰の過程で直交性を失ってしまう。

5 Chapter 9 非負行列の因数分解

ある行列 $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ が与えられ、非負の要素を持ように制限された k ランクの近似を行いたいと考えた時、言い換えれば $\boldsymbol{W} \in \mathbb{R}^{m \times k}$ と $\boldsymbol{H} \in \mathbb{R}^{k \times n}$ を仮定して以下の式を解きたい場合について考える。

$$\min_{\boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0} ||\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}||_F$$
$$where\ ||\boldsymbol{A}||_F = \sqrt{\Sigma_i \Sigma_j |a_{ij}|^2}\ means\ Frobenius\ norm \tag{7}$$

同時に $\boldsymbol{W}$ と $\boldsymbol{H}$ を最適化しようとした時、この問題は非線形になる。

しかし、行列の一つが既にわかっている場合、例えば $\boldsymbol{W}$ がわかっている場合で言えば、この $\boldsymbol{H}$ を求める問題は、非負の制限がついた右側の行列についての最小二乗問題であると言える。

従って元の問題に対する最も一般的な解決法は、交互最小二乗法 (ALS) を使うことである。

5.1 Alternating nonnegative least squares のアルゴリズム

Alternating nonnegative least squares algorithm

1. 初期値 $\boldsymbol{W}^{(1)}$ を与える。2. $k = 1, 2 \ldots$ と収束するまで繰り返す。- $\min_{H \geq 0} ||\boldsymbol{A} - \boldsymbol{W}^{(k)}\boldsymbol{H}||_F$ を解き、$\boldsymbol{H}^{(k)}$ を得る- $\min_{W \geq 0} ||\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}^{(k)}||_F$ を解き、 $\boldsymbol{W}^{(k+1)}$

しかし、近似 $\boldsymbol{W}\boldsymbol{H}$ は単一のものではない。ここで正の対角要素を持つ任意の対角行列 $\boldsymbol{D}$ とその逆行列を要素間に適用しすることが出来る。

$$\boldsymbol{W}\boldsymbol{H} = (\boldsymbol{W}\boldsymbol{D})(\boldsymbol{D}^{-1}\boldsymbol{H})$$

ある要素の増大と他の要素の減衰を防ぐために、毎反復ごとにそれらの一つを正規化する必要がある。一般的な正規化は、 $\boldsymbol{W}$ の各列の最大要素が 1 になるように $\boldsymbol{W}$ の列をスケーリングすることである。

$\boldsymbol{A}$ と $\boldsymbol{H}$ の列要素として、 $\boldsymbol{a_j}$ と $\boldsymbol{h_j}$ を置く。各行の要素を一つずつ書き下すと、この最小二乗問題は以下の n 個の独立したベクトルの最小二乗問題と等しいものとみなすことが出来る。

$$\min_{h_j \geq 0} ||\boldsymbol{a}_j - \boldsymbol{W}^{(k)}\boldsymbol{h}_j||_2\ where\ j = 1, 2, \ldots, n$$

この行列を転置することで、 $\boldsymbol{W}$ を決定するの最小二乗問題は、独立な m このベクトルの最小二乗問題に変換される。即ち ALS アルゴリズムのコア部分は擬似的な MATLAB コードで以下のように表すことが出来る。

疑似 MATLAB コード

```
while (not converged)
    [W] = normalize(W);
    for i = 1:n
        H(;,i) = lsqnonneg(W, A(:,i));
    end
    for i = 1:m
        w = lsqnonneg(W, A(:, i));
        W(i,:) = w';
    end
end
```

非負値行列因子分解のためのアルゴリズムは多くの種類がある。前頁のアルゴリズムは、非負な最小二乗法のためのアクティブセット法にかなり時間がかかってしまうという欠点がある。より簡易な代替手段として、部分 QR 分解 $\boldsymbol{W} = \boldsymbol{QR}$ を用いることで非負という制約がない最小二乗解を得ることが出来る。そして $\boldsymbol{H}$ におけるすべての負の要素はゼロと等しいとみなすことが出来る。これは $\boldsymbol{W}$ の計算においても同様の議論をすることが出来る。

$$\boldsymbol{H} = \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{A}$$

疑似 MATLAB コード (上の例に基づけば $\boldsymbol{V} = \boldsymbol{A}$)

```
while (not converged)
    W = W.*(W >= 0);
    H = H.*(W'*V)./((W'*W)*H+epsilon);
    H = H.*(H>=0);
    W = W.*(V*H')./(W*(H*H')+epsilon);
    [W,H] = normalize(W, H);
end
```

$epsilon$ は極小の値であり、これはゼロ除算を避けるために用いられている。 $.*$ や $./$ で表される行列操作はそれぞれの構成要素についての演算で、

$$\boldsymbol{H}_{ij} := \boldsymbol{H}_{ij}\frac{(\boldsymbol{W}^T\boldsymbol{A})_{ij}}{(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H})_{ij}+\epsilon},\ \boldsymbol{W}_{ij} := \boldsymbol{W}_{ij}\frac{(\boldsymbol{A}\boldsymbol{H}^T)_{ij}}{(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T)_{ij}+\epsilon}$$

尚このアルゴリズムは勾配降下法と考えることが出来る。

非負値行列分解には非常に多くの重要な用途があるため、これのアルゴリズム開発は活発に行われている。例えば、反復法を用いた終了基準を見つける問題は未だ良い解決策を見つけたとは言い難い。

非負値行列分解 $\boldsymbol{A} \approx \boldsymbol{W}\boldsymbol{H}$ はクラスタリングにも用いられている。各データを表すベクトル $\boldsymbol{a}_j$ は、もし $\boldsymbol{h}_{ij}$ が $\boldsymbol{H}$ の j 列の最大の要素であるなら、それはクラスタ i に割り当てられる。

さらにこの分解法は文書分類、電子メールの監視[4]、バイオインフォマティクス、スペクトル分析のような分野でも用いられている。

5.2 初期化

非負値行列分解のアルゴリズムにはいくつかの問題がある。それは全体での最適解が求まる保証がないということである。このアルゴリズムではしばしば収束が遅いことや順最適解 (厳密解ではない) になってしまうことがある。良好な初期の近似を計算するための効率的な手法として、 A の SVD に基づいて行うというものがある。最初の k 個の特異の三つの組 $(\boldsymbol{\sigma}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)_{i=1}^k$ は、フロベニウスノルムにおいて A の最適なランク k の近似を与える。

もし A が非負な行列であったならば、 $\boldsymbol{u}_i$ や $\boldsymbol{v}_1$ が非負であることは明らか。(Section 6.4)

つまりもし $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ が $\boldsymbol{A}$ の SVD であるならば、特異ベクトル $\boldsymbol{u}_1$ を $\boldsymbol{W}^{(1)}$ の最初の列であるとすることが出来る。 (同様に以降のアルゴリズムのため、 $\boldsymbol{v}_1^T$ を初期近似 $\boldsymbol{H}^{(1)}$ の 第 1 行 であるする。)

次の最良なベクトル $\boldsymbol{u}_2$ は直交性が満たされているために負の成分を有する可能性が非常に高い。しかし行列 $\boldsymbol{C}^{(2)} = \boldsymbol{u}_2\boldsymbol{v}_2^T$ を計算しすべての負の成分をゼロにすることで非負な行列 $\boldsymbol{C}_+^{(2)}$ を得ると、この行列の最初の特異ベクトルは非負であることがわかる。さらにそれは、これが $\boldsymbol{u}_2$ の合理的で良い近似であると考えること

[4]楽譜にすること

が出来るので、$\boldsymbol{W}^{(1)}$ の第 2 列として取り上げることが出来る。

前頁の手続きを MATLAB を使って簡潔に書き下すと以下のようになる。[5]

MATLAB Code

```
[U,S,V] = svds (A, k) % Compute only the k largest singular
W (:,1) = U (:,1);    % values and the corresponding vectors
for j = 2:k
    C = U (:,j)*V (:,j)';
    C = C .* (C>=0);
    [u, s, v] = svds (C, 1);
    W (:,j) = u;
end
```

Example 9.4

ランク 2 の行列 $\boldsymbol{A}$ の非負値分解の例を図 3 に示す。ここでは初期化をランダムな値で行ったものと、 SVD ベースで行ったものを比較している。ランダムな値で初期化したものは収束がより遅くなており、10 回反復させても収束したとは言い難い。これに対して SVD ベースで初期化したものの相対近似誤差は 0.574 であることがわかる。 (尚 k-means 法においてこの誤差は 0.596 であった) 更にいくつかのケースでランダムに初期化したものは最適でない準最適な値に収束してしまった。

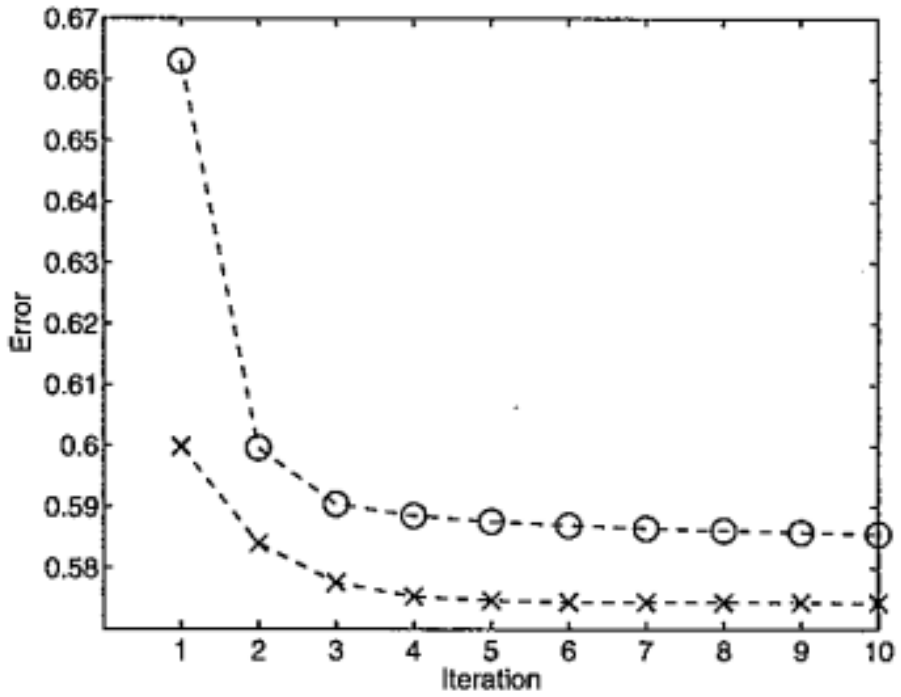

図 3: 反復回数を横軸とした相対近似誤差のグラフ。上のカーブは初期化をランダムに行ったもの、下のカーブは SVD ベースで初期化を行ったものである。

SVD ベースの初期化をするこの分解は具体的に以下のようになった。

[5] $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}]$ = svds (A, k) は Lanczos 法を用いることで、k 個の最大特異値及び対応する特異ベクトルとを計算する。標準な SVD 関数である $svd(A)$ はすべての分解を計算するがこれはかなり遅く、特に行列が大きな疎行列のときはより遅くなる。

$$\boldsymbol{WH} = \begin{pmatrix} 0.3450 & 0 \\ 0.1986 & 0 \\ 0.1986 & 0 \\ 0.6039 & 0.1838 \\ 0.2928 & 0 \\ 0 & 0.5854 \\ 1.0000 & 0.0141 \\ 0.0653 & 1.0000 \\ 0.8919 & 0.0604 \\ 0.0653 & 1.0000 \end{pmatrix} \begin{pmatrix} 0.7740 & 0 & 0.9687 & 0.9120 & 0.5251 \\ 0 & 1.0863 & 0.8214 & 0 & 0 \end{pmatrix}$$

前頁のそれは分解の処理を打ち切ることが出来る。最初の四つの文書は基底ベクトルによって表されており、これは Google-related keywords のための大きな要素を持っている。これに対して最後の文書は 1 つめの基底ベクトルによって表されているが、この座標値は先述の四つの文書に比べて小さくなっている。

この手法では、ランク 2 の近似は Google-related contents を強調するが、 ”football-document” は強調しない。

対して、ランク 3 の近似を計算した時には以下の値を得ることが出来る。

$\boldsymbol{W}$ の三番目のベクトルは、本質的に ”football” についての基底であり、その一方で他の 2 つのベクトルは Google-related document を示している基底である。

$$\boldsymbol{WH} = \begin{pmatrix} 0.2516 & 0 & 0.1633 \\ 0 & 0 & 0.7942 \\ 0 & 0 & 0.7942 \\ 0.6924 & 0.1298 & 0 \\ 0.3786 & 0 & 0 \\ 0 & 0.5806 & 0 \\ 1.0000 & 0 & 0.0444 \\ 0.0589 & 1.0000 & 0.0007 \\ 0.4237 & 0.1809 & 1.0000 \\ 0.0589 & 1.0000 & 0.0007 \end{pmatrix} \begin{pmatrix} 1.1023 & 0 & 1.0244 & 0.8045 & 0 \\ 0 & 1.0815 & 0.8315 & 0 & 0 \\ 0 & 0 & 0.1600 & 0.3422 & 1.1271 \end{pmatrix}$$