

Risa/Asir による一般逆行列の計算と その連想記憶への応用

愛媛大学 工学部 水口寛之 (Hiroyuki Minakuchi)

愛媛大学 工学部 甲斐博 (Hiroshi Kai)

愛媛大学 工学部 野田松太郎 (Matu-Tarow Noda)

1. はじめに

今日、コンピュータや数式処理システムの発達によって今まで誤差を含む数値計算でしか計算できなかったような問題が数式処理によって誤差なく解かれるようになってきた。そこで、本研究では数式処理による一般逆行列の計算と、その応用例として一般逆行列の連想記憶への応用を取り上げる。すでに一般逆行列を数式処理を用いて求めようとする試みや、そのための各種アルゴリズムの比較については、いくつかの研究が発表されている [1],[2],[3] が、本研究ではさらに進んで一般逆行列の工学問題への応用について考察する。ここで取り扱う対象は画像処理等の分野で研究されている連想記憶の実現のための問題 [4],[5],[6] への応用である。連想記憶で用いられる記憶行列の作成には一般逆行列を計算することが必要である。特に、分散して記憶された情報の選択や統合には正確に一般逆行列を求めることが重要になる。

2. 一般逆行列

2.1. 一般逆行列の定義

まず、本稿の記述の一貫性を保つために、一般逆行列に対して知られているいくつかの性質や事項をまとめる。

A を $m \times n$ 行列、 G を $n \times m$ 行列とする。次の4つの方程式、

(i) $AGA = A$

(ii) $GAG = G$

(iii) $(AG)^T = AG$

(iv) $(GA)^T = GA$

において、行列 G が (i) をみたすとき、 G は A の一般逆行列 (A^-) とよばれる。そして、行列 G が (i)~(iv) をみたすとき、 G は A の Moore-Penrose 型一般逆行列 (A^+) とよばれる。本研究では単に一般逆行列といった場合、Moore-Penrose 型一般逆行列を表すものとする。

2.2. 一般逆行列の計算アルゴリズム

文献[1],[3]等に示されているように、一般逆行列を求めるためには多くのアルゴリズムが知られている。この大半は数式処理で一般逆行列を求めることを主に論じられているが、一般にはこれら以外に数値計算で一般逆行列を計算する場合が多い。数値計算のアルゴリズムとしては、特異値分解法を用いるのが一般である。しかし、悪条件問題(一般逆行列を必要とする問題の多くは悪条件であるが)に対しては、微小な特異値の処理等で特異値分解法による数値計算には常に不安がつきまとう。そこで、本研究では数式処理の算法の中、計算速度、メモリ効率が比較的優れ、かつプログラムが容易な Greville のアルゴリズムを用いる。このアルゴリズムは解析も容易であるので、将来、安定化理論[7]を適用するような場合にも困難はない。

Greville のアルゴリズムとは以下のようなものである。

入力行列を $A = [a_1^T, a_2^T, \dots, a_m^T]^T$
 $a_i \cdots m$ 次元行ベクトル

とする。 A_i を、

$$A_1 = a_1 \quad A_i = \begin{bmatrix} A_{i-1} \\ a_i \end{bmatrix}$$

と定義し、 $i = 1, 2, \dots, m$ において、

$$A_i^+ = [A_{i-1}^+ - b_i^T d_i \quad b_i^T]$$

を順次計算することにより、最終的に A_m^+ が A^+ となる。

なお、 d_i, c_i, b_i および初項 A_1^+ は、

$$\begin{aligned} d_i &= a_i A_{i-1}^+ \\ c_i &= a_i - d_i A_{i-1} \\ b_i &= \begin{cases} \frac{c_i}{c_i c_i^T} & (c_i \neq 0) \\ \frac{d_i (A_{i-1}^+)^T}{1 + d_i d_i^T} & (c_i = 0) \end{cases} \\ A_1^+ &= \begin{cases} \frac{a_1^T}{a_1 a_1^T} & (a_1 \neq 0) \\ a_1^T & (a_1 = 0) \end{cases} \end{aligned}$$

として計算する。アルゴリズム実現上の問題点は、行列の大きさを変化させていく必要のあることと、 b_i, A_i^+ の計算に必要な結果の零判定である。

2.3. 一般逆行列の計算と実行時間

実際に Risa/Asir 上で一般逆行列を計算した例をあげる。

● 例 1

$$\text{入力行列 } A = \begin{pmatrix} 32 & 11 & 15 & 1 \\ 53 & 22 & 3 & 1 \end{pmatrix}$$

```
[16] A=newmat(2,4,[[32,11,15,1],[53,22,3,1]]);
[ 32 11 15 1 ]
[ 53 22 3 1 ]
[17] geninv(A);
[ 544/592157 9175/592157 ]
[ -7315/592157 8338/592157 ]
[ 43593/592157 -25647/592157 ]
[ 1319/592157 -613/592157 ]
```

● 例 2

$$\text{入力行列 } A = \begin{pmatrix} 2a^2 - 1 & 2a \\ 10 & a \end{pmatrix}$$

```
[19] A=newmat(2,2,[[2*a^2-1,a*2],[10,a]]);
[ 2*a^2-1 2*a ]
[ 10 a ]
[20] geninv(A);
[ (1)/(2*a^2-21) (-2)/(2*a^2-21) ]
[ (-10)/(2*a^3-21*a) (2*a^2-1)/(2*a^3-21*a) ]
```

これらの一般逆行列の計算にかかる時間は以下のようなものである。計算はパソコン (Pentium 125MHz、メモリ 16MB) 上で稼働する数式処理システム Risa/Asir を用いた。数値行列の一般逆行列として、Givens 行列を取り上げた。これより、数値行列に対してはかなりの大きさの行列でも比較的高速に一般逆行列を求めることが出来ることがわかる。また、表 2 では入力行列として二つの変数をランダムに配置した要素を持つ $n \times n$ 行列を用い、それを 10 回計算した平均値を示している。

従来までは計算時間の問題が一般逆行列の応用上の障壁の一つになっていたが、もはやあまりその点を考慮する必要が無くなったと思われる。なお、記号行列に対する一般逆行列を求めようとする、メモリ消費問題など今後改良すべき点が多いこともわかる。

表 1. Givens 行列の一般逆行列計算にかかる時間

行列の 大きさ n	実行時間 (sec)
10	0.06
50	5.58
100	41.86
150	138.5
200	325.4
250	635.1
270	804.1

表 2. 入力に記号を含む場合の実行時間

行列の 大きさ n	実行時間 (sec)
2	0.04
3	0.30
4	1.99
5	4.21
6	6.11
7	20.54
8	35.26
9	91.04
10	135.78

3. 連想記憶

3.1. 連想記憶とは

本研究で用いる連想記憶とは不完全な入力パターンから、ある出力パターンを得るような写像のことをいう。入力パターンを

$$X = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_K), \quad \vec{x}_i \in R^n, i = 1, \dots, K$$

出力パターンを

$$Y = (\vec{y}_1, \vec{y}_2, \dots, \vec{y}_K), \quad \vec{y}_i \in R^n, i = 1, \dots, K$$

とする。線形連想写像は、

$$Y = MX$$

M : パターン間の変換行列

となる。ここで M は、

$$M = YX^+$$

と表される。

記憶行列の選択、統合をするために生成行列 G をつくる。

$$G = \sum_{i=1}^S a_i M_i$$

ただし

$$\sum_{i=1}^S a_i = 1, 0 < a_i < 1, \text{ for } i = 1, 2, \dots, S$$

G の固有ベクトルを ϕ_i とすると、固有ベクトルの直交性より、

$$\phi_i^T \phi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

となる。

3.2. 連想記憶に関する定理

連想記憶で用いる記憶行列は一般には巨大なものになる。大規模な行列を取り扱うより、いくつかの小規模行列に分散して記憶し、それらを統合したり選択したりして処理を進める方が効率が上がる。そこで、ここでは記憶行列を作る上で記憶情報の選択と統合を考える。記憶行列の選択とは、 S 個の連想記憶に共通に記憶されている情報 ($I_{add} = \cap_{i=1}^S I_i$) を $M_1, M_2, \dots, M_S$ から構成することであり、記憶情報の統合とは、 S 個の連想記憶に記憶されている情報全体 ($I_{or} = \cup_{i=1}^S I_i$) を $M_1, M_2, \dots, M_S$ から構成することである。すなわち、以下のような記憶行列の定理がある。[6]

記憶情報選択の定理

n 次元ベクトル ψ が記憶情報の集合 I_{add} の張る空間に含まれるならば、また、そのときの ψ が生成行列 G の大きさ 1 に対応する固有ベクトルとなる。

記憶情報統合の定理

n 次元ベクトル ψ が記憶情報の集合 I_{or} の張る空間に含まれるならば、また、そのときの ψ が生成行列 G の正の固有値に対応する固有ベクトルとなる。

すなわち、記憶行列選択の定理は

n 次元ベクトル ψ が記憶情報の集合 I_{add} の張る空間に含まれる。

$\Longleftrightarrow$

ψ が G の大きさ 1 に対応する固有ベクトル

を意味しており、記憶行列統合の定理は

n 次元ベクトル ψ が記憶情報の集合 I_{or} の張る空間に含まれる。

$\Longleftrightarrow$

ψ が G の正の固有値に対応する固有ベクトル

を意味している。

3.3. 連想記憶における問題点

入力と出力が同じである自己想起型連想記憶を考える。記憶行列 M の計算

$$M = XX^+$$

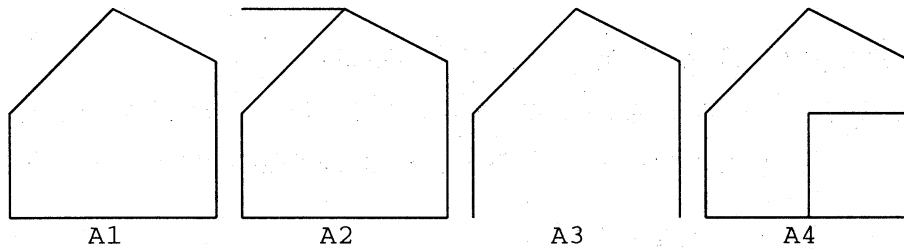
において、一般逆行列を数値的に計算した場合 M において数値誤差がでる。

また、それによって生成行列 G の固有値、固有ベクトルの計算において数値誤差がでる。このため X^+ を正確に計算する必要性が生じる。そこで、数式処理のアルゴリズムを用いることが重要となる。

4. 計算例

連想記憶の計算において、以下の例をあげる。

以下の4つの図形を与える。



これを 4×4 行列 A_1, A_2, A_3, A_4 で表す。

$$A_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

A'_1, A'_2, A'_3, A'_4 は $A_i (i=1 \sim 4)$ の第1行の要素を左から右へ、次に第2行の要素を左から右に、... とならべ、16次の列ベクトルにしたものとする。すなわち、その転値ベクトルは

$$A_1^T = (0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1)$$

と書かれる。

そして以下の3つの記憶情報を与える。

$$X_1 = (A'_1, A'_2) \quad X_2 = (A'_1, A'_3) \quad X_3 = (A'_1, A'_4)$$

このようにして作られた16行2列の行列 X_1, X_2, X_3 には全て A'_1 が共通に含まれている。したがって記憶情報の選択を行なうと、 A_1 が出力されるはずである。

記憶行列の選択を行うためにはすでに述べたように、自己想起型連想記憶の場合には、記憶行列 $M = XX^+$ を求めなければならない。ここで一般逆行列を用いるが、以下ではこれを Greville のアルゴリズムを数式処理によって計算して求める。比較のために、同じ Greville のアルゴリズムを浮動小数計算で求めた場合と特異値分解法による数値計算とによる場合も計算した。なお、数値計算は Risa/Asir 上で倍精度で計算を行った。

これらの計算で得られた記憶行列から、生成行列 G を求める。今の場合、

$$G = \sum_{i=1}^3 a_i M_i = \sum_{i=1}^3 a_i X_i X_i^+$$

において、 $a_1 = a_2 = a_3 = 1/3$ とした。記憶行列の選択のためには、このようにして得られた G の固有値のうち、固有値の大きさ1に対応する固有ベクトルを求める必要がある。そこで、生成行列 G の固有値を3種の計算法で求めた結果は以下のようになった。なお、数値計算で行った固有値の計算は Mathematica による。

● Greville のアルゴリズムを用いた数式処理

$$\lambda = 1, 1/3$$

● Greville のアルゴリズムを用いた数値計算

$$\begin{aligned}\lambda = & 1.000000, 0.333292, 0.333354 \pm 0.0000353633i, \\ & -0.0316241, 0.0336045, -0.0276826 \pm 0.0154931i, -0.0166718 \pm 0.0273275i, \\ & -0.00096999 \pm 0.032440i, 0.0156822 \pm 0.0290072i, 0.028652 \pm 0.0172079i\end{aligned}$$

● 特異値分解法を用いた数値計算

$$\begin{aligned}\lambda = & 1.000000, 0.333333, 0.333333, 0.333334, 0.0298483, -0.0286648, \\ & -0.0250105 \pm 0.0141351i, -0.0148663 \pm 0.0247964i, -0.000584958 \pm 0.0291795i, \\ & 0.0142747 \pm 0.0258095i, 0.0255954 \pm 0.0151599i\end{aligned}$$

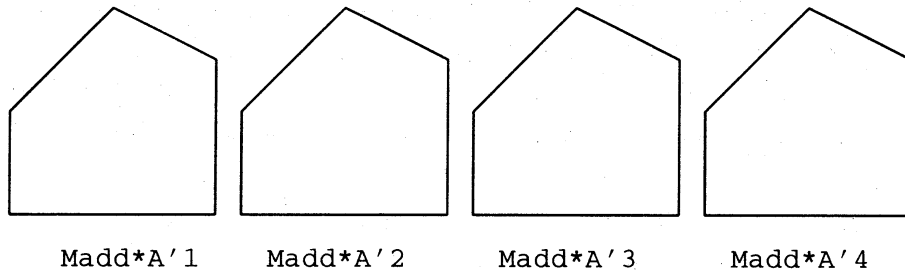
次に数式処理を行なった場合において、固有値1の固有ベクトル ϕ_1 の計算を行なう。

$$\phi_1 = \left(0 \ 0 \ 1/3 \ 0 \ 0 \ 1/3 \ 0 \ 1/3 \ 1/3 \ 0 \ 0 \ 1/3 \ 1/3 \ 1/3 \ 1/3 \ 1/3 \right)^T$$

$X = (A'_1, A'_2, A'_3, A'_4)$ 、 $M_{add} = \phi_1 \phi_1^T$ とすると、

$$M_{add}X^T = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 7/9 & 0 & 0 & 7/9 & 0 & 7/9 & 7/9 & 0 & 0 & 7/9 & 7/9 & 7/9 & 7/9 & 7/9 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

となる。これを図形的に表すと、以下ようになる。



以上のように各記憶情報 X_1, X_2, X_3 の共通部分である A_1 が出力としてあらわれる。

ここで用いた例は極めて簡単なもので、数式処理で求めた結果は当然期待する結果を与えるものの、他の2つの数値計算結果でも、生成行列 G の固有値として、高い精度で大きさ1を持つ部分を計算することが可能であり、実用上は近似的に記憶行列選択の定理を満たすため、さして問題は生じないように思われる。しかし、数値計算で求めた固有値の結果として、多くの共役複素根が表れているように、少し問題が複雑になったり、対象パターンの大きさが大きくなったりすると、伝統的な数値計算のみでは満足に記憶行列選択の定理を満たさなかったり、結果が不安定になることが予想される。

5. 結び

本論では一般逆行列を連想記憶の構成に応用することを考えた。現在までに得られた知見その他を以下にまとめる。

- 表1より、 200×200 のような大きな行列の一般逆行列でも十分実用的な時間で計算することが出来る。
- 連想記憶において、数式処理を用いて一般逆行列を計算することにより、正確な記憶行列 M 、および生成行列 G を求めることができる。そしてそれにより G の正確な固有値、固有ベクトルを計算することができる。
- さらに生成行列 G の固有値1の固有ベクトルを計算することにより、入力データの共通部分を取り出すことが出来る記憶行列をつくり出すことができる。

以上をふまえて本論での考察を更に深め、数式処理に基づく一般逆行列計算の連想記憶への応用をより大次元記憶行列に対して行うことが可能になったと思われる。当然記憶行列の形によっては非常に悪条件な逆行列計算も生じるであろうから数式処理のより技術的な応用につながるであろう。その過程で、

- 悪条件連立1次方程式の解法の見直し
- 固有値を正確に求める方法の確立
- 数値誤差の安定化

等の問題を克服することが必要になるであろう。

参 考 文 献

- [1] 野田松太郎、泉田正則、越智正明：一般逆行列の数式処理システムによる直接解法とその評価、情報処理学会論文誌 Vol.30,no.11、pp.1376-1384 (1989)。
- [2] Frawley,W.J:Computer Generation of Symbolic Generalized Inverse and Application to Physics and Data Analysis,in Application of Computer Algebra(Pavelle,R.ed),pp.415-426,Kluwer Academic Pub(1972)。
- [3] 齋藤敏明、牧野潔夫：一般逆行列の計算アルゴリズムとその証明、数理解析研究講究録
- [4] T.Kohonen:Associative Memory,Springer-Verlag,1977
- [5] 村上研二、武智宏親、泉田正則：複数の連想記憶に記憶された記憶情報の選択と結合、電子情報通信学会論文誌 J76-D-II,pp.2607-2614,1993
- [6] C.W.Therrien:Eigenvalue properties of projection operators and their application to the subspace method of feature extraction,IEEE Trans. Comput.C-24,pp.944-948,1975
- [7] Kiyoshi Shirayanagi,Moss Sweedler: A Theory of Stabilizing Algebraic Algorithms, pp.1-92,preprint,1995