2003年度情報処理II　第12回

数学のためのコンピューター (4) 行列の解析的性質で遊ぶ

桂田 祐史 (かつらだ まさし)

2003年7月3日

「情報処理IIトップページ」[1]

1 はじめに

前々回は線形計算 (連立一次方程式や固有値問題など) のための方法と、そのための Octave というソフトウェア (MATLAB[2] という有名なソフトの互換ソフト) について紹介したが[3]、今回は Octave を使って行列の簡単な解析的性質の実験をしてみよう。

ここでいう解析的性質とは、行列の極限にかかわる性質のことを意味している。たとえば、行列の列 A_n が $n \to \infty$ のときの極限とか、無限級数 $\sum_{n=1}^{\infty} A_n$ とか、行列値関数 $f(A)$ の連続性などの話である。線型作用素の解析的性質は通常は関数解析で学ぶが、(行列の段階での) 初等的な解説は杉浦・横沼 [1] でも読める。

2 行列の等比数列

一つの正方行列 A の冪 A^n で作られる行列の列、いわば行列の等比数列

$$A^0 = I, A^1 = A, A^2, \cdots, A^n, A^{n+1}, \cdots$$

の極限はどうなるだろうか？既に学んだかもしれないが (行列の Jordan 標準形の簡単な応用である)、ここでは実験で試してみよう。

[1] http://www.math.meiji.ac.jp/~mk/syori2/

[2] http://www.mathworks.com/

[3] そのときの予定では、今回は行列の固有値計算の話 (例えば回転による相似変換を繰り返すことで行列を対角行列に近づける Jacobi 法など) をするつもりだったが、数学的解説を手短にする自信がなくなったので、今回はもっと素朴な話にした。

```
isc-xas05% octave
octave:1> n=5
octave:2> a=rand(n,n)
octave:3> a*a
octave:4> a^2
octave:5> a^100
octave:6> a^1000
```

Octave では、行列 a の n 乗は a^n で計算できることがわかるが、この例では n の増加と共に急速に大きくなり、あっと言う間にオーバーフローすることが分かる。

種明かしをすると、行列の固有値の絶対値 (これをスペクトル半径と呼ぶ) を調べると事情が見えてくる。

```
octave:7> eig(a)
octave:8> r=max(abs(eig(a)))
octave:9> a=a/r
octave:10> max(abs(eig(a)))
octave:11> a^100
octave:12> a^1000
```

Octave では max(abs(eig(a))) で a のスペクトル半径が計算できる。a をそのスペクトル半径 r で割ることで、スペクトル半径を 1 に縮めることができる。a^100, a^1000 ともにオーバーフローしていないが、それだけでなく何かが起こっていることに気がつくだろうか？(どうしてそうなるのか考えてみよう。)

今度はスペクトル半径が 1 より小さい行列の冪を調べてみよう。

```
octave:13> a=0.9*a
octave:14> a^100
octave:15> a^1000
octave:16> a^10000
```

3　行列の等比級数

今度は正方行列 A (ただし A のスペクトル半径は 1 より小さいとする) の等比級数

$$I + A + A^2 + A^3 + \cdots = \sum_{n=0}^{\infty} A^n \quad (I \text{ は単位行列})$$

を考える。この級数は解析学では Neumann 級数と呼ばれる。

以下の実験は前節の続きで行うことを想定している。Octave を終了してしまっている場合は、

この節の実験に先立ち必要なこと

```
n=5
a=rand(n,n)
r=max(abs(eig(a)))
a=a/r
a=0.9*a
```

を実行しておこう。

Neumann 級数の部分和

$$S_n = \sum_{k=0}^{n} A^k$$

を計算する Octave プログラム `neuamnn.m` を用意した。

neumann.m

```
% neumann.m --- 行列の Neumann 級数（等比級数）の第 N 部分和
function s = neumann(a,N)
  [m,n] = size(a);
  if m != n
    disp("a が正方行列でない！");
    return
  endif
  % 第 0 項  S_0 = I
  s = eye(n,n);
  % 第 1 項  S_1 = I + a
  t = a; s = s + t;
  % 第 2〜N 項まで加える (t が a^n になるようにしてある)
  for k=2:N
    t = t * a;
    s = s + t;
  end
```

これを `~re00018/neumann.m` から、Octave を実行するディレクトリィにコピーしよう。

こうしてコピーする

```
isc-xas05% cp ~re00018/neumann.m .
```

```
octave:17> c=eye(n,n)-a
octave:18> b=neumann(a,100)
octave:19> b*c
octave:20> b=neumann(a,1000)   ← もう少し精度を上げてみる
octave:21> b*c
octave:22> format long         ← お望みなら表示桁数を上げて
octave:23> b*c
```

Neumann 級数の (部分) 和 $S_N = \sum_{k=0}^{N} \mathrm{a}^k$ に $C = I - \mathrm{a}$ をかけた結果が単位行列に非常に近

いことが分かる。種明かしをすると、

$$\sum_{n=0}^{\infty} A^n = (I - A)^{-1}$$

が成り立つ。これは等比級数の和の公式

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r} \quad (\text{ただし } r \text{ は } |r| < 1 \text{ を満たす複素数})$$

の一般化である。

4　絶対値最大の固有値を求める — 冪乗法

簡単のため n 次正方行列 A が対角化可能で、その固有値を $\lambda_1, \cdots, \lambda_n$, それらに属する固有ベクトルを $u_1, \cdots, u_n$ とする。さらに $|\lambda_1|$ は他の固有値の絶対値よりも大きいとする。

$$|\lambda_1| > |\lambda_i| \quad (i = 2, 3, \cdots, n)$$

任意のベクトル $x \in \mathbf{C}^n$ は

$$x = c_1 u_1 + c_2 u_2 + \cdots c_n u_n$$

と展開できるが、A^k を作用させると

$$A^k x = c_1 A^k u_1 + c_2 A^k u_2 + \cdots C_n A^k u_n = c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + \cdots C_n \lambda_n^k u_n.$$

k が大きいとき、右辺第 1 項は右辺の他の項と比べて大きくなることが分かる (ただし $c_1 \neq 0$ とする)。k を十分大きくすると、右辺第 2 項以下は第 1 項と比べて無視できるほど小さくなるだろう。すると $A^k x$ は u_1 の定数倍、すなわち λ_1 に属する固有ベクトルに近くなるはずである。

以上のことを Octave による計算で確かめるためには、$A^k x$ がオーバーフローすることを防ぐため、代りにその長さで割った $\dfrac{A^k x}{\|A^k x\|}$ を作ればよい。

以下では素朴に A をかけていくことで $\dfrac{A^k x}{\|A^k x\|}$ を求めている。

```
octave:5>  x=ones(n,1)
octave:5>  for i=1:100
>  y=a*x
>  x=y/norm(y)
>  end
octave:5>  a*x ./ x        ← a*x の各成分を対応する x の成分で割ってみる
octave:5>  eig(a)          ← 念のため eig() で a の固有値を調べて比較
```

線形代数では、固有値を固有多項式の根として特徴づけるが、普通固有多項式を数値計算で解くのは難しいので、行列の問題のまま各種の反復法を用いることになる。上で見た方法は『冪乗法』と呼ばれ、多くの方法の基礎となっている。

5　課題 8

締め切りは 7 月 16 日。

- 今回は Octave を使った計算の感想を (5 行程度以上) 書いて下さい (これだけでレポートと認めます)。

- もし興味が生じて自分で何か実験を考案して実行した場合はそれについて書いてください。

- 具体的な挑戦課題が欲しいという人は `neumann.m` を参考に行列の指数関数

$$\exp A = \sum_{n=0}^{\infty} \frac{1}{n!} A^n$$

  を計算する関数 `exponential()` を作って、実験して下さい。

参考文献

[1] 杉浦光夫, 横沼健雄, Jordan 標準形・テンソル代数, 岩波書店 (1990).