

マルコフ連鎖の数値計算のプログラムをチームで

樋口さぶろお

龍谷大学工学部数理情報学科

計算科学☆実習 B L06E(2020-05-19 Tue)

最終更新: Time-stamp: "2020-05-19 Tue 13:10 JST hig"

今日の目標

- チームで活動できる
- サンプルプログラムを元に、マルコフ連鎖の分布の時間発展を計算できる



マルコフ連鎖の時間発展の数値計算

状態 $x = 0, \dots, m-1$ の m 状態のマルコフ連鎖を考える.

分布 $\vec{p}(t), p(x, t) \rightarrow$

```

1  double p[m] = {1.0, 0.0, ..., 0.0}; /*配列. mは整数.*/
2  /* {p(0,t), p(1,t), p(2,t), ..., p(m-1,t)} */

```

以下, $m = 2$ で例示.

転置推移確率行列 $M = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.3 \\ 0.9 & 0.7 \end{pmatrix} \rightarrow$

```

1  double M[][m] = { {0.1, 0.3},
2                    {0.9, 0.7} }; /* 2次元配列 */

```

$\{ \{p_{00}, p_{01} \},$
 $\{p_{10}, p_{11} \} \}$

行列とベクトルの積

$$\vec{q} = M\vec{p} \rightarrow q_x = \sum_y M_{xy}p_y.$$

$\vec{p}(1) = M\vec{p}(0), \vec{p}(2) = M\vec{p}(1), \dots$ という計算を t に関する for loop にする.

```

1  p[] を p(x,0) で初期化;
2  p を出力;
3  for (t){
4      pn=M p; /*行列とベクトルの積*/
5      p=pn;
6      p を出力;
7  }
```

サンプル https://www.data.math.ryukoku.ac.jp/course/compscib_2020/p/markov01/src/markov01sample.c

```

1  /*
2   マルコフ連鎖の分布の時間発展の計算
3   Time-stamp: "2020-05-19 Tue 12:16 JST hig"
4  */
5  #define _CRT_SECURE_NO_WARNINGS /* Visual C++ に必要なおまじない
6  #include <stdio.h>
7
8  /* 状態数 m 要編集 */
9  #define NS 3
10
11 int multiply_trans(double pn[], double p[]);
12 int measure_dist(double p[], int t, int m);
13
14 int main(){
15     int t, tmax;
16     int x;
17     double p[NS]; /* 分布  $p(t)$  */
18     double pn[NS]; /* 分布  $p(t+1)$  */
19     int m=NS; /* 状態数 */
20
21
22     scanf("%d", &tmax);
23     printf("#T=%d\n", tmax);

```

```

24
25 t=0; p[0]=1.0; p[1]=0.0; p[2]=0.0; /* 初期分布 */
26 measure_dist(p,t,m); /* 分布の出力 */
27
28 for (t=0;t<tmax;t++){
29     multiply_trans(pn,p); /*  $pn=M*p$  で次の時刻の分布を求める */
30     for (x=0;x<m;x++){
31         p[x]=pn[x]; /* コピー  $p=pn$  */
32     }
33     measure_dist(p,t+1,m); /* 分布の出力 */
34 }
35 return 0;
36 }
37
38 /* 縦ベクトル  $p$  を右から行列  $M$  をかけてできるベクトル  $M p$  を  $q$  に
39 int multiply_trans(double q[], double p[]){
40     int x,y;
41     int m=NS;
42     /* 転置推移確率行列 */
43     double M[][NS]={ {0.5, 0.5, 0.0},
44                       {0.5, 0.5, 0.0},
45                       {0.0, 0.0, 1.0} }; /*要編集*/
46     for (x=0;x<m;x++){

```

```
47     q[x]=0;
48     for(y=0;y<m;y++){
49         q[x]+=M[x][y]*p[y];
50     }
51 }
52 return 1;
53 }
54
55 /* tとpを1行に出力 編集不要 */
56 int measure_dist(double p[], int t, int m){
57     int x;
58     printf("%d,", t);
59     for(x=0;x<m;x++){
60         printf("%f, ", p[x]);
61     }
62     printf("\n");
63     return 0;
64 }
```

上の説明と下のサンプルと課題 p061 markov01 との関係

最初の説明では $m = 2$. サンプルでは $NS = m = 3$. 課題ではそれなりの値 (正方行列の次元) になるので, ベクトル, 行列に相当する 1次元, 2次元配列のサイズを修正する必要がある.

今回は $NS = m$ (上で言っているもの) や, M や $p(0)$ を問題に合わせるだけでよい.

- 関数 `print_dist` の中身は変更しなくてよい.
- 関数 `multiply_trans` の中身は 2重配列 M の定義だけを変えればよい.

markov01 の出力の出力の縦横は, `ca1d02` と同じ, `rw21` とは異なる.
markov01 で描きたいグラフは, `ca1d02` とは異なる. 資料 `w07.pdf` に載ってるようなグラフを描きたい.

この課題 (とその後の数個) は重要

レポートはこれに類する結果を式とデータで説明するものになります。

必要な C 言語的予習

プログラミング及び実習ほかの授業資料を発掘しておいて。

- 1次元, 2次元配列の宣言と初期化
- 配列を引数とする関数

必要な線形代数的予習

- 固有値固有ベクトル

オンラインサブチーム課題のやり方 (計算科学☆演習 B) I

2-3 名程度の (サブ) チームでオンラインでやる課題のやりかたです.

- 欠席するときは事前にサブチームメンバーと教員に連絡します.
- 自分のサブチーム番号は, 資料 (または事前のメール) で確認します. Driver, Navigator の役目分担をしていることがあります.
- (課題の中で, 自分のサブチームのミッション番号の部分だけをやります.)
- 最初は教員または TA からサブチームメンバーにグループチャットをします. 音声と画面共有で進めます. 両方から書き込める白紙として Word を使います. Chat に添付した Word (docx) ファイルを, :アイコンから「ブラウザで開く」
- Driver だけが Codeanywhere を使います. Driver は Codeanywhere.com を画面共有し, タイプします.
- Navigator は口やテキストチャットでアドバイスします.

オンラインサブチーム課題のやり方 (計算科学☆演習 B) II

- 授業時間外の連絡やファイルの交換は、Teams の chat を基本とします。メンバーで合意すれば別の方法をとっても構いません。
- 終了時には *.c, *.xlsx などのファイルをテキストチャット添付で共有します。また Moodle に提出しておきます。締切までは何度でも削除提出ができるので、作業中のファイルの置き場所として使うこともできます。
- 自分のほうがよくわかってるっぽい分野は相手に説明し、相手のほうがよくわかってるっぽい分野は相手に説明してもらいましょう。
- 疑問はまずサブチーム内で解決を試みましょう。もし解決できなかったり、意見が一致しなかったりしたときは、質問と相談チャンネルで TA のアドバイスも受けてください。
- 教員にきく、クラスでいちばん分かってる人にきく、ほうが疑問解決には速いですが、この活動では、未知のメンバーとコミュニケーションして疑問を解決すること自体を目的にしています。

オンラインサブチーム課題のやり方 (計算科学☆演習 B) III

- Moodle への課題提出方法パターン 1: チームメンバーの誰かが提出すればそれが全員の提出となります。締切までに最終バージョンでいいことをチームメンバーが合意する必要があります。だれが提出するかをはっきりさせ、全員が合意してから最終版としましょう。