計算科学☆実習 B ファイナルトライアル (筆記)

樋口さぶろお[1]　配布: 2019-08-01 Thu 更新: Time-stamp: ”2019-08-04 Sun 10:22 JST hig”

ファイナルトライアル (筆記) 参加案内

1. 指定された用紙に解答しよう.
2. 過程も答えよう. 最終的な答えが正しいことがわかるような過程を記そう.
3. 問題文に現れない記号を使うときは, 定義を記そう.

1

過程不要

(座標が整数値のみをとる離散型の) 鴨のランダムウォークを考える. 座標は $x = 0, 1, 2, \ldots, 5$ に制限されているとする.

7 羽の鴨が, $x = 2$ に 1 羽, $x = 3$ に 4 羽, $x = 5$ に 2 羽いるとする.

1. ラグランジュ表現を用いたとき, 配列 `x[]` のサイズはどれだけ必要か. また, 配列の各要素はどのような値をとるか.
2. オイラー表現を用いたとき, 配列 `u[]` のサイズはどれだけ必要か. また, 配列の各要素はどのような値をとるか.

2

過程不要

この授業に関係するプログラムを書く際, オイラー表現とラグランジュ表現を対比して考えたとき, 正しいものに○, 正しくないものに×をつけよう.

ただし, この授業では, 1 人ウォーカー, 複数ウォーカーのランダムウォークの確率シミュレーションには, ラグランジュ表現を使っている.

1. オイラー表現では, ランダムウォーカーの座標が実数 (連続的) である場合は扱えない.
2. オイラー表現では, マルコフ連鎖の計算はできない
3. オイラー表現では, 複数のランダムウォーカーがいる場合に, 一人のウォーカーに着目して経路を追跡することはできない.
4. オイラー表現では, 「ある時刻のランダムウォーカーの位置が $a \le x < b$ である母比率」は計算できない.

hig@math.ryukoku.ac.jp, https://hig3.net(講義のページもここからたどれます), へや:1 号館 5 階 507

3

過程不要

以下の記述について, 正しいものに○, 正しくないものに×をつけよう.

1. ランダムウォークの確率シミュレーションのプログラムは, 母ナントカを推定するために標本を抽出している
2. 行列の積を用いたランダムウォークのマルコフ連鎖のプログラムは, 母ナントカを推定するために標本を抽出している
3. ランダムウォークの確率シミュレーションのプログラムの結果は, 丸め誤差など数値計算一般にある誤差に加えて, 統計誤差 (信頼区間で表現されるようなもの) を含む
4. ランダムウォークのマルコフ連鎖のプログラムの結果は, 丸め誤差など数値計算一般にある誤差に加えて, 統計誤差 (信頼区間で表現されるようなもの) を含む
5. ランダムウォークの確率シミュレーションのプログラムで, でいちばん外側にあるループ (反復処理) は, 時間 t についてのループである
6. ランダムウォークのマルコフ連鎖のプログラムで, いちばん外側にあるループ (反復処理) は, 時間 t についてのループである

4

過程不要

マルコフ連鎖で, m 次元の確率ベクトル $\vec{p}_1$ に, $m \times m$ の転置推移確率行列 M をかけて, 新しい確率ベクトル $\vec{p}_2 = M\vec{p}_1$ を得る計算を考える. ただし,

$$M = \begin{bmatrix} 0.7 & 0.1 & 0.0 & \cdots & \cdots & 0.0 & 0.2 \\ 0.2 & 0.7 & \ddots & & & 0.0 & 0.0 \\ 0.0 & \ddots & \ddots & & & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & & & \ddots & \ddots & 0.0 \\ 0.0 & 0.0 & & & \ddots & 0.7 & 0.1 \\ 0.1 & 0.0 & \cdots & \cdots & 0.0 & 0.2 & 0.7 \end{bmatrix}$$

とする.

これを実行する関数 void multiplytrans(double p2[], double p1[], int m) を定義しよう (main などプログラム全体を書く必要はない) ただし, 配列 p1 は入力 $\vec{p}_1$, 配列 p2 は出力 $\vec{p}_2$, m は次元である.

関数内で行列 M を 2 次元配列として定義することはせず, 1 重のループのみを使うこと.

5

時刻 $t = 0$ に, $x = 15$ から出発するランダムウォークの座標 $X(t)$ を考える.
各時刻 t での座標の増分 $R(t)$ は独立同分布にしたがう確率変数で,

$$\mathrm{E}[R(t)] = -2, \mathrm{V}[R(t)] = 9$$

を満たす.

1. $X(100)$ の母平均値を求めよう.
2. $X(100)$ の母分散を求めよう.
3. $X(100) \geq -200$ の母比率を正規分布の上側確率 $Q(u) = P(Z > u)$ を使って近似的に表そう. ただし, $0 < u$ の形に整理すること.

6

ある水平なゲーム盤上で, コマは発射装置で初速を与えて打ち出された後, 等速直線運動する. 発射装置からゴールまでは 90[cm] ある. 発射装置の与える初速の向きは, 正確にゴールに向かう向きであり, 速さ R は, 10.0[cm/s] から 15.0[cm/s] (連続値) を同じ確からしさで取る.
打ち出されてからゴールするまでの時間を T [s] とする.
大注意: 授業でやってた例は, T が R の単調増加関数だったけど, これは単調減少関数. この t はもちろんマルコフ連鎖やランダムウォークの離散時間の t でなく, 確率変数.

1. 母平均値 $\mathrm{E}[T]$ を求めよう.
2. 母比率 $P(T < \frac{15}{2})$ を求めよう.
3. T の確率密度関数 $f_{\mathrm{T}}(t)$ を求めよう. ただし, T が R の単調減少関数なので, 確率密度関数の間の関係は, 絶対値つきで,

$$f_{\mathrm{T}}(t) = \frac{1}{\left|\frac{dt}{dr}\right|} f_{\mathrm{R}}(r)$$

で与えられる.

7

確率密度関数

$$f_{\mathrm{R}}(r) = \begin{cases} \frac{1}{15}(12 - 2r) & (2 \leq r < 5) \\ 0 & (\text{他}) \end{cases}$$

に従う確率変数 R に対応する乱数を, 逆関数法を用いて $[0, 1)$ 一様乱数 y から $r = g(y)$ で作りたい. 関数 $g(y)$ を求めよう. プログラムとして書く必要はない.

8

$\boxed{\textbf{過程不要}}$

確率密度関数

$$f(x) = \begin{cases} \frac{1}{\pi}\sqrt{4-x^2} & (0 \le x < 2) \\ 0 & (\text{他}) \end{cases}$$

に従う乱数を生成したい.

1. 棄却法を利用して乱数を返す関数 `double getrandom()` を C で書こう. π は `M_PI` と書くこと (ヘッダーファイルの include は不要). 関数内で $[0, 1)$ 一様乱数を返す関数 `double getuniform()` を (答案内で定義せず) 呼び出してよいが, なるべく呼び出しの回数を減らすこと.
2. 解答した `getrandom()` を 1000 回呼び出すとき, `getuniform()` の呼び出しは, 平均して何回起きるか, 答えよう.

計算科学☆実習 B ファイナルトライアル (筆記) 略解

樋口さぶろお[2] 配布: 2019-08-01 Thu 更新: Time-stamp: ”2019-08-04 Sun 10:22 JST hig”

これは, 一部の過程のみ記した略解です. 参加者はすべての過程を記す必要があります.

配点 計 100 点.

1

1. 7 羽なのでサイズは 7.
   各要素は, `x[]={2,3,3,3,3,5,5};` (順序はこうとは限らない).
2. 座標が $x = 0, 1, 2, \ldots, 5$ の計 6 点なので, サイズは 6.
   各要素は `u[]={0,0,1,4,0,2};` (順序はこうである必要).

2

1. ◯
2. ×
3. ◯
4. ×

3

1. ◯
2. ×
3. ◯
4. ×
5. ×
6. ◯

4

周期境界条件なので, 両端 $x = 0, m-1$ を特別扱いする.

ソースコード 1: 転置推移行列

```
void multiplytrans(double p1[], double p2[], int m){
  int x;

  p2[0]=  0.2*p1[m-1]+0.7*p1[0]+0.1*p1[0+1];
  for(x=1;x<m-1;x++){
    p2[x]=0.2*p1[x-1]+0.7*p1[x]+0.1*p1[x+1];
```


hig@math.ryukoku.ac.jp, https://hig3.net(講義のページもここからたどれます), へや:1 号館 5 階 507.

```
    }
    p2[m-1]=0.2*p1[m-2]+0.7*p1[m-1]+0.1*p1[0];

    return;
}
```

5

1. $\mathrm{E}[X(100)] = \mathrm{E}[15 + R(1) + \cdots + R(100)] = 15 + 100 \times \mathrm{E}[R(1)] = -185.$
2. $\mathrm{V}[X(100)] = \mathrm{V}[5 + R(1) + \cdots + R(100)] \overset{\text{独立}}{=} 100 \times \mathrm{V}[R(1)] = 900.$
3. 「中心極限定理から」, 近似的に $X(100) \sim \mathrm{N}(-185, 900)$. よって, $Z = \frac{X(100)+185}{\sqrt{900}}$ とすると, $Z \sim \mathrm{N}(0, 1^2)$.
   よって, $P(X(100) \geq -200) = P(Z \geq \frac{-200+185}{30}) = 1 - Q(\frac{1}{2})$.

6

1. $R \sim \mathrm{U}(10, 15)$.
   $T = 90/R$ なので, $\mathrm{E}[T] = 90\mathrm{E}[\frac{1}{R}] = 90 \int_{10}^{15} \frac{1}{5}\frac{1}{r}\mathrm{d}r = 18[\log r]_{10}^{15} = 18(\log 3 - \log 2) = 7.30[\mathrm{s}]$.
2. $P(T < \frac{15}{2}) = P(R > 12) = \frac{3}{5}$.
3. $10 \leq R < 15, 6 < T \leq 10$ の範囲に限定すると,

$$f_{\mathrm{T}}(t) = \frac{1}{\left|\frac{dt}{dr}\right|} f_{\mathrm{R}}(r)$$

   より

$$f_{\mathrm{T}}(t) = \frac{1}{+\frac{90}{R^2}} \frac{1}{5} = \frac{18}{t^2}.$$

   よって,

$$f_{\mathrm{T}}(t) = \begin{cases} 18/t^2 & (6 \leq t < 9) \\ 0 & (\text{他}) \end{cases}$$

講評　チームプロジェクト 2 で, $T = g(R)$ のときの $\mathrm{E}[T] = \mathrm{E}[g(R)]$ は息するように計算するようになったかと思ってたんだけど, $P(T < t) = P(g(R) < t)$ は息するように計算するようになったかと思ってたんだけど, 80 分制限のプレッシャーのある試験だと, $\mathrm{E}[T] = g(\mathrm{E}[R])$ とか, (前半やってた 1 次関数の g の時にしか正しくない)$R \sim U(6, 9)$ なら $T \sim U(10, 15)$ってやっちゃう人もいるよね〜

それとは別に, T が単調減少なので, $P(T < t) = P(R > g^{-1}(t))$ と不等号の向きが変わるというもありました.

7

$$
\begin{aligned}
f_{\mathrm{R}}(r) =& \frac{1}{\frac{\mathrm{d}r}{\mathrm{d}y}} f_{\mathrm{Y}}(y)\mathrm{d}y \\
\frac{1}{15}(12-2r)\mathrm{d}r =& 1\mathrm{d}y \\
-\frac{1}{4}\frac{1}{15}(12-2r)^2 =& y + C
\end{aligned}
$$

$y = 1$ のとき $r = 5$ より, $C = -\frac{16}{15}$. $y = 0$ のとき $r = 2$ より, 符号を選んで, $r = g(y) = 6 - (16 - 15y)^{1/2}$.

8

ソースコード 2: 棄却法

1.

```
#include <math.h>
extern double getuniform();

double getrandom(){
  double x,y;
  double M=2.0/M_PI; /* max f(x) */

  while(1){
    x=2.0*getuniform()*2.0; /* U(0,2) */
    y=M*getuniform();       /* U(0,M) */
    if( y<=1.0/M_PI*sqrt(4-x*x)){ /* y<=f(x) */
      break;
    }
  }
  return x;
}
```

2. getuniform() を2回呼び出して (x,y) を作ったとき, x が return される確率は, 扇型と長方形の図形の面積の比を考えて, $\frac{\mathrm{E}[1]}{\frac{2}{\pi}\cdot 2} = \frac{\pi}{4}$. よって, 回数の平均値は $\frac{2\times 1000}{\frac{\pi}{4}} = \frac{8000}{\pi}$ 回.