メトロポリス・ヘイスティングス法とギブスサンプリング

樋口さぶろお https://hig3.net

龍谷大学大学院 理工学研究科 数理情報学専攻

理論物理学特論　L08 (2022-11-09 Wed)

最終更新: Time-stamp: "2022-11-09 Wed 16:55 JST hig"

今日の目標

- メトロポリス・ヘイスティングス法を説明できる
- ギブスサンプリングを説明できる

略解 I

L07-Q1

Quiz 解答: マルコフ連鎖の時間発展

1. $\vec{p}(1) = M\vec{p}(0) = \frac{1}{3}\binom{1}{2}$.
2. 転置推移確率行列 M の固有値 1 の固有ベクトル $\vec{u}$ を (あるなら) 求めればよい. $M\vec{u} = \vec{u}$ を解いて, $\vec{u} = \binom{3}{4}s \quad (s \neq 0)$. 定常分布は, 規格化された $\vec{u} = \frac{1}{7}\binom{3}{4}$.
3. 転置推移確率行列 M の固有値 (絶対値の大きさの順に) λ_1, λ_2, 対応する固有ベクトル $\vec{u}_1, \vec{u}_2$ を求めると,

$$\lambda_1 = 1, \lambda_2 = -\frac{1}{6}, \qquad \vec{u}_1 = \binom{3}{4}s, \vec{u}_2 = \binom{1}{-1}s \quad (s \neq 0).$$

固有方程式には解 $\lambda_1 = 1$ があることが最初からわかってるから, 因数分解は楽.

略解 II

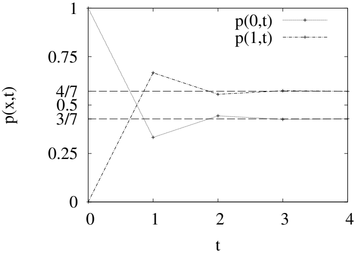
$\vec{u}_1, \vec{u}_2$ とも $s = 1$ に固定する (他の取り方をしても最終的には同じ $\vec{p}(t)$ が求まる). このとき, $\vec{p}(0) = a\vec{u}_1 + b\vec{u}_2$ で係数 a, b を決めると, $a = \frac{1}{7}, b = \frac{4}{7}$.

$$\begin{aligned} \vec{p}(t) =& M^t \vec{p}(0) \\ =& M^t (a\vec{u}_1 + b\vec{u}_2) \\ =& \lambda_1^t a\vec{u}_1 + \lambda_2^t b\vec{u}_2 \\ =& \tfrac{1}{7}\left(\begin{smallmatrix} 3 \\ 4 \end{smallmatrix}\right) + \tfrac{4}{7}\left(\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right)(-\tfrac{1}{6})^t. \end{aligned}$$

$\vec{u}_1$ の係数が $\frac{1}{7}$ である (=この項が確率ベクトルである) ので, $t \to +\infty$ でも $\vec{p}(t)$ が確率ベクトルでありつづけることが確認できる. (固有値が $\lambda = 1, -\frac{1}{6}$ となった時点で, 第 1 項が確率ベクトルになるはず, ということから, 1 次方程式を解かずに $a = \frac{1}{7}$ と決められたはずだった).

略解 III

時間変化. $|-\frac{1}{6}| < 1$ なので, $\vec{p}(t) \to \frac{1}{7}\binom{3}{4} \quad (t \to +\infty)$



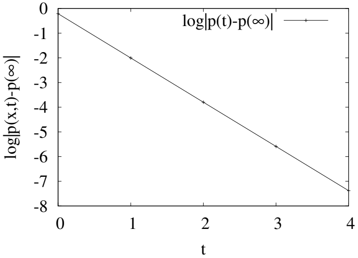
略解 IV

4. 極限分布 $\vec{p}(\infty) = \frac{1}{7}\begin{pmatrix} 3 \\ 4 \end{pmatrix}$ との差の大きさ (ベクトルの長さ) を考えると,

$$\begin{aligned} |\vec{p}(t) - \vec{p}(\infty)| &= \left| \frac{4}{7}\begin{pmatrix} 1 \\ -1 \end{pmatrix}\left(-\frac{1}{6}\right)^t \right| \\ &= \frac{4\sqrt{2}}{7}\left(\frac{1}{6}\right)^t \\ \log|\vec{p}(t) - \vec{p}(\infty)| &= t \times \log\frac{1}{6} + \log\frac{4\sqrt{2}}{7} \end{aligned}$$

よって, t 対 $\log|$差$|$ のグラフを描くと, 傾き $\log|$第 2 固有値$|(< 0)$ の直線になるはず.

略解 V



状態空間が 3 点以上からなり, (第 2 固有値より絶対値が小さい) 第 3 固有値以降がある場合も, t が大きいところではこの直線に近づいていく.
さらに t を大きくすると, $|\vec{p}(t)-\vec{p}(\infty)|$ が桁落ちしたり, 数値的に 0 とみなされるなどで, 直線から外れる.

5.

$$\vec{p}(t)=\frac{1}{7}\begin{pmatrix}3\\4\end{pmatrix}+\frac{1}{14}\begin{pmatrix}1\\-1\end{pmatrix}\left(-\frac{1}{6}\right)^t$$

略解 VI

L07-Q2
L07-Q5

ここまで来たよ

MCMC の復習と追加説明

どんな (定数 Z 不明の) $\frac{1}{Z}g(x)$ に対しても, 定常分布が $\frac{1}{Z}g(x)$ になるような, うまいマルコフ連鎖 M を $f(x)$ から作って, シミュレートして, t における状態を取り出すことで擬似乱数を得る方法.

- うまい:$X(t)$ から $X(t+1)$ を $\mathrm{U}(0,1)$ 擬似乱数だけで確率的に定められる.

Listing 1: MCMC

```
double x=xinitial;
  for(n=0;n<サンプルサイズ;n++){
    for(t=0;t<待ち時間;t++){
      x=getnext(x); // U(0,1)一様乱数を使い,Mにしたがう
    }
  // 定常分布に近づいただろう
  printf("%f\n",x);
}
```

要注意

- 最初のそれなりの個数の出力は, 初期値の影響を受けているかもしれないので=定常分布に十分に近づいていないかもしれないので, 捨てる (burn-in).
- 出力が定常分布にしたがっていても, 互いに独立になっていないかもしれないので, 待ち時間は十分大きくとる. 次回以降に出てくる, 自己相関係数で判定できる.

メトロポリス・ヘイスティングス法

getnext(x) を構成する方法.

下心

p が定数倍を除いてわかっているとき, 詳細つりあいの条件を満たすマルコフ連鎖 M を作れれば, その定常分布 (運が良ければ極限分布) として p が現れるはず. M を擬似乱数でシミュレートする.

命題 (メトロポリス・ヘイスティングス法 (Metropolis-Hastings))

与えられた $\vec{p}$ に対して,

$$M_{xy} = q_{xy} \min\{1, \frac{p_x q_{yx}}{p_y q_{xy}}\} + (\text{定数}) \times \delta_{xy}$$

は詳細つりあいの条件を満たす. 定数は, M が転置確率行列となるように調整.

提案分布 自由に決めていい転置推移確率行列 q_{xy}. 対称行列が選ばれ, $\min$ からは消去されることが多い.
直観的には, q_{xy} で x が提案されたなら, p_x が p_y より大きいときは, かならず x に移る. 大きくないときは, 比に応じて確率的に移す.

メトロポリス・ヘイスティングス法

転置推移確率行列であること列の和が 1 になるように, (定数) で対角成分を調節できる.
対称であること ($x \neq y$ の場合に示す)
$p_x q_{yx} \geq p_y q_{xy}$ のとき

$$M_{xy} = q_{xy} \cdot 1, \quad M_{yx} = q_{yx} \cdot \frac{p_y q_{xy}}{p_x q_{yx}}$$

よって, $M_{xy} p_y = q_{xy} p_y$. $M_{yx} p_x = q_{xy} p_y$.
$p_x < p_y$ のとき

$$M_{xy} = q_{xy} \cdot \frac{p_x q_{yx}}{p_y q_{xy}}, \quad M_{yx} = q_{yx} \cdot 1$$

よって, $M_{xy} p_y = p_x q_{yx}$. $M_{yx} p_x = p_x q_{yx}$.

メトロポリス・ヘイスティングス法のアルゴリズム

Listing 2: MH in MCMC

```
//確率q(y,x)にしたがって提案yを選ぶ.
double getproposal(double x); // q(y,x)にしたがう提案
double getuniform(void);//[0,1)一様擬似乱数

double getnext(double x){ //x:現在の状態
  int y;  // 次の状態
  int yp; // 提案された状態
  double u;
    yp=getproposal(x);
    u=getuniform();
    if(u< min(1, (p(x)*q(ypropsed,x))/(p(ypropsed)*q(x,yp)) ){
      y=yp; // 提案を採択
    } else {
      y=x; // 提案を棄却
    }
  return y; // 次の状態
}
```

ここまで来たよ

7 マルコフ連鎖モンテカルロ法

8 メトロポリス・ヘイスティングス法とギブスサンプリング
- メトロポリス・ヘイスティングス法
- 混合分布のベイズ推定への応用
- 混合分布の事後分布からのギブスサンプリング
- ギブスサンプリングの一般論

混合ガウス分布のベイズ推定

混合ガウス分布 理論物理学特論 (2022)L05

$$f_{\mathrm{Y}}(y) = \theta f(y; \mu_0, \sigma_0) + (1-\theta) f(y; \mu_1, \sigma_1)$$

を考える．ここで，パラメタ $0 \le \theta \le 1$ を確率変数 X であるかのように扱う．$\mu_0, \sigma_0, \mu_1, \sigma_1$ は固定，既知．ガウス分布でなくても以下の話はできる．
事前分布 $f_\Theta(\theta)$ を決める．特に事前の情報がないとき，

$$f_\Theta(\theta) = \begin{cases} 1 & (0 \le \theta \le 1) \\ 0 & (\text{他}) \end{cases}$$

としてもよい．
Y のデータ $\{y_i\}_{i=1}^{n}$ が得られたとき，θ の情報を得る試みとして，事後分布 $f_{\Theta|\mathrm{Y}}(\theta|\{y_i\})$ を得たい．

ベイズの定理から

$$f_{\Theta|\mathrm{Y}}(\theta|\{y_i\}) = \frac{\prod_{i=1}^{n}(\theta f(y_i;\mu_0,\sigma_0) + (1-\theta)f(y_i;\mu_1,\sigma_1))}{\int_{-\infty}^{+\infty}\prod_{i=1}^{n}(\theta f(y_i;\mu_0,\sigma_0) + (1-\theta)f(y_i;\mu_1,\sigma_1))\mathrm{d}\theta}.$$

この確率密度関数を具体的に積分して確率や期待値を求めるのはたいへん.

今は提案行列 $q_{\theta\theta'} = 1$ でよい (一部の θ を重点的に選ぶようにすることが認められている).
分母を知らなくても, 採択確率に現れる比を

$$\frac{p_{\theta'}}{p_\theta} = \frac{f(\theta')}{f(\theta)} = \frac{\prod_{i=1}^n (\theta' f(y_i;\mu_0,\sigma_0) + (1-\theta')f(y_i;\mu_1,\sigma_1))}{\prod_{i=1}^n (\theta f(y_i;\mu_0,\sigma_0) + (1-\theta)f(y_i;\mu_1,\sigma_1))}$$

で, 実行すれば, θ の事後分布からサンプルできる.

応用: 物理の人が考えた画像生成モデル

確率変数 $x = \{s_{ij}\}$. $w \times h$ の 長方形状に並んだ画素. 簡単のため $0 \leq s_{ij} \leq 2^8 - 1$ (グレイスケール).

画像が得られる確率 $f(x) = \frac{1}{Z} \prod_{(i,j),(k,l)\text{が辺で連結}} g(s_{ij}, s_{k\ell})$.

辺の (pairwise な) 確率 $g(s, s') = (s, s'$が近い階調ほど大). (程度により画風が変わる).

MH の提案確率

$$q_{x'x} = \begin{cases} \frac{1}{wh}\frac{1}{2^8} & (x' \text{ は, } x \text{ の 1 画素 } s_{ij} \text{ だけを } 0 \leq s'_{ij} < 2^8 \text{ に変更したもの}) \\ 0 & (\text{他}) \end{cases}$$

MH の採択確率

$$q_{x'x} \times \min\{1, \bullet\} = q_{x'x} \times \min\{1, \frac{g(s'_{ij},\text{隣})g(s'_{ij},\text{隣})g(s'_{ij},\text{隣})g(s'_{ij},\text{隣})}{g(s_{ij},\text{隣})g(s_{ij},\text{隣})g(s_{ij},\text{隣})g(s_{ij},\text{隣})}\}$$

ここまで来たよ

7 マルコフ連鎖モンテカルロ法

8 メトロポリス・ヘイスティングス法とギブスサンプリング
- メトロポリス・ヘイスティングス法
- 混合分布のベイズ推定への応用
- 混合分布の事後分布からのギブスサンプリング
- ギブスサンプリングの一般論

ギブスサンプリング

混合ガウス分布では, まず X, Y の同時分布を考えた後, その Y の周辺分布を考え, $X = 0, 1$ (コインの裏表) は和を取って消していた.
計算の便利のため, X を復活させよう.

定義 (ギブスサンプリング)

モデルに, 互いに関係する θ, X があるとき, 交互に条件付き分布からサンプルする (のが容易である) アルゴリズム.

Listing 3: Gibbs Sampling

```
double theta=thetainitial;
int n=サンプルサイズ;
int x[n]; // 0 or 1, コインの裏表
double y[]={,,,} ;// 測定されたn個のデータ

while(true){
  // X
  for(int i=0; x<n; i++){
    x[i]=getx(theta,y);
  }

  // theta
  theta=gettheta(x); // beta分布

  print("%f\n",theta);
}
```

int getx(double theta) 次の条件付き確率で 0 または 1 をランダムに返す.

$$P(X_i = x_i|\theta, y_i) = \begin{cases} \frac{\theta f(y_i;\mu_1,\sigma_1)}{\theta f(y_i;\mu_1,\sigma_1)+(1-\theta) f(y_i;\mu_0,\sigma_0)} & (x_i = 1) \\ \frac{(1-\theta) f(y_i;\mu_0,\sigma_0)}{\theta f(y_i;\mu_1,\sigma_1)+(1-\theta) f(y_i;\mu_0,\sigma_0)} & (x_i = 0) \end{cases}$$

double gettheta(int x[]) 次の確率密度関数にしたがって $\theta \in [0, 1]$ をランダムに返す. $k = \sum_i x_i$.

$$f(\theta) = \frac{1}{Z}\theta^k (1-\theta)^{n-k}$$

(Beta 分布といわれるもので, そのような乱数は $\mathrm{U}(0, 1)$ 擬似乱数だけから作れる)

L08-Q1

Quiz(ギブスサンプリング)

乱数やサイコロは使えず, コイン (ただし, 表のでる確率 p は自由に設定できる), すなわち $\mathrm{B}(1,p)$ の乱数だけが使える状況を考えよう.

離散型確率変数の同時分布 $f(x,y) =$

$y\backslash x$	1	2
10	1/10	2/10
20	3/10	4/10

を考える.

1. 初期値が $x=1$ だったとき, ギブスサンプリングのループの 1 回目が終わった時点で得られる標本の分布を求めよう.
2. 上の同時分布が, ギブスサンプリングの定めるマルコフ連鎖の定常分布になっていることを確かめよう.

ここまで来たよ

ギブスサンプリングの一般論

2 つの確率変数 X, Θ

- x, θ に初期値を代入
- while(1)
  - $X \sim X|\theta, \{y_i\}$ からサンプルして x を更新
  - $\Theta \sim \Theta|x, \{y_i\}$ からサンプルして θ を更新

ギブスサンプリングのもっと一般論

k 個の確率変数 X_i.

- $x_1, \ldots, x_k =$ 初期値を代入
- while(true)
  - $X_1 \sim X_1 | \quad x_2, \ldots, x_k$ からサンプルして x_1 を更新
  - $X_2 \sim X_2 | x_1, \quad x_3, \ldots, x_k$ からサンプルして x_2 を更新
  - $\vdots$
  - $X_k \sim X_k | x_1, x_2, \ldots, x_{k-1} \quad$ からサンプルして x_k を更新