單元 4: 反轉換法
(課本 §4.1)

令隨機變數

$$X \sim p_j = P(X = x_j),\ \ j = 0, 1, 2, \ldots$$

且

$$\sum_{j=0}^{\infty} p_j = 1$$

亦即, X 爲離散隨機變數, 取值 x_j 的機率爲 p_j.

問 1. 如何生成 (或模擬) X?

答 1. 令隨機變數

$$U \sim \text{unif}(0, 1)$$

設定

$$X = \begin{cases} x_0, & \text{若 } U < p_0 \\ x_1, & \text{若 } p_0 \le U < p_0 + p_1 \\ x_2, & \text{若 } p_0 + p_1 \le U < p_0 + p_1 + p_2 \\ & \vdots \\ x_j, & \text{若 } \sum_{i=0}^{j-1} p_i \le U < \sum_{i=0}^{j} p_i \\ & \vdots \end{cases}$$

如此設定合理嗎? 合理, 因爲

(1) 由

$$p_i > 0 \text{ 且 } \sum_i p_i = 1$$

得

$$\sum_{i=0}^{j} p_i,\ j = 0, 1, 2, \ldots$$

爲 $(0,1)$ 的一個分割 (partition), 且

(2) 由

$$U \in (0,1)$$

得 U 必落在某一個子區間內.

故對於每一個隨機數 U, 一定會產生唯一的一個 X 值, 而合理地定義出 X, 亦即, 不會發生有了 U 值而無法產生出 X 值; 或一個 U 值產生出多個 X 值的情形.

另外, X 爲一隨機變數, 此乃因爲 X 的值乃由隨機變數 U 所決定而改變, 也就是說, X 爲隨機變數 U 的一個函

數. 同時, 根據 X 的定義, 對於 $j = 0, 1, 2, \ldots,$

$$\begin{aligned} P(X = x_j) &= P\left(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i\right) \\ &= \sum_{i=0}^{j} p_i - \sum_{i=0}^{j-1} p_i \\ &= p_j \end{aligned}$$

因此,

$$X \sim p_j = P(X = x_j),\ j = 0, 1, 2, \ldots$$

如所求.

最後, 得可執行的

演算法:

(1) 生成一個仿隨機數 U.

(2) 若 $U < p_0$, 則令 $X = x_0$ 且停止.

(3) 若 $U < p_0 + p_1$, 則令 $X = x_1$ 且停止.

(4) 若 $U < p_0 + p_1 + p_2$, 則令 $X = x_2$ 且停止.

$\vdots$

註 1. 看一特例: 若

$$x_0 < x_1 < x_2 < \cdots$$

則 X 的分布函數 (distribution function, cdf) $F(x)$ 如圖示, 且對於 $k = 0, 1, 2, \ldots$,

$$\begin{aligned} F(x_k) &= P(X \le x_k) \\ &= \sum_{i=0}^{k} P(X = x_i) \\ &= \sum_{i=0}^{k} p_i \end{aligned}$$

因此, 對於 $j = 0, 1, 2, \ldots$,

$$X = x_j$$

乃相當於

$$F(x_{j-1}) \le U < F(x_j)$$

亦即, X 的值是由 U 落在下列哪一個區間

$$[F(x_{j-1}), F(x_j)),\ j = 0, 1, 2, \ldots$$

所決定, 也就是說, 給定分布函數

$$F(x_j),\ j = 0, 1, 2, \ldots$$

決定隨機變數

$$X$$

與所熟悉的給定隨機變數

$$X$$

而決定出其分布函數

$$F(x_j),\ j = 0, 1, 2, \ldots$$

剛好呈現出相反 (互逆) 的運算. 所以, 生成 X 的方法稱爲離散反轉換法 (discrete inverse transform method, DITM).

註 2. 離散反轉換法的演算法如下.

演算法:

(1) 根據

$$p_0,\ p_1,\ p_2,\ \cdots$$

將 $(0,1)$ 分割成如下的子區間.

(2) 由第一個區間開始, 決定 U 落在哪一個子區間, 而得出對應的 X 值.

因此, 由上述的演算法知, 生成 X 所需的時間隨著所需檢查的子區間個數的增加而增加. 又一區間愈大, U 落在此區間的機率愈高. 所以, 修正原先的演算法, 可得較有效率 (快) 的演算法如下.

演算法:

(1) 將

$$p_j,\ \ j = 0, 1, 2, \ldots$$

由大到小排列; x_j 亦作對應的排列.

(2) 根據 (1) 分割 $(0,1)$ 成如下的子區間.

(3) 從第一個子區間開始檢查, 亦即, 從最易發生的子區間開始檢查, 並決定出對應的 X 值.

例 1. 令隨機變數

$$X \sim \begin{array}{ll} p_1 = 0.20, & p_2 = 0.15 \\ p_3 = 0.25, & p_4 = 0.40 \end{array}$$

其中

$$p_i = P(X = i),\ i = 1, 2, 3, 4$$

試模擬 X.

<解一> 直接根據 p_1 到 p_4 的順序將 $(0, 1)$ 分割, 如圖示, 得

演算法 1:

(1) 生成一個仿隨機數 U.

(2) 若 $U < 0.20$, 則令 $X = 1$ 且停止.

(3) 若 $U < 0.35$, 則令 $X = 2$ 且停止.

(4) 若 $U < 0.60$, 則令 $X = 3$ 且停止.

(5) 令 $X = 4$.

<解二> 先將 p_1 到 p_4 由大到小排列, 再根據排序後的結果將 $(0, 1)$ 分割, 如圖示, 得

演算法 2:

(1) 生成一個仿隨機數 U.

(2) 若 $U < 0.40$, 則令 $X = 4$ 且停止.

(3) 若 $U < 0.65$, 則令 $X = 3$ 且停止.

(4) 若 $U < 0.85$, 則令 $X = 1$ 且停止.

(5) 令 $X = 2$.

練習題: 試以程式說明演算法 2 較快.

問 2. 令隨機變數

$$X \sim \text{discrete unif}(n)$$

亦即,

$$X \sim P(X=i) = \frac{1}{n},\ i=1,\ldots,n$$

如何模擬 X?

答 2. 由離散反轉換法知, 將 $(0,1)$ 根據 X 的分布函數分割後, 如圖示, 得

$$X=j,\ 若\ \frac{j-1}{n} \le U < \frac{j}{n}$$

此乃相當於

$$X=j,\ 若\ j-1 \le nU < j \tag{1}$$

又

$$j-1 \le nU < j$$

乃表示當 nU 介於連續兩整數 $j-1$ 與 j 之間時, 其整數部分爲 $j-1$, 也就是說,

$$j-1 = \text{Int}(nU) \tag{2}$$

其中

$$\text{Int}(x) = [x] = \text{小於或等於 } x \text{ 的最大整數}$$

故, 合併 (1) 式與 (2) 式, 得

$$X = j, \text{ 若 } j = \text{Int}(nU) + 1$$

因此,

$$X = \text{Int}(nU) + 1$$

且對應的

演算法:

(1) 生成仿隨機數 U.

(2) 令

$$X = \text{Int}(nU) + 1$$

註. 雖然是根據離散反轉換法模擬離散均勻分布的隨機變數 X, 但經過等價的化簡步驟後, 不需逐一地檢查子區間即可得出 X, 故是一相當簡潔且有效率的模擬法.

離散均勻分布隨機變數的二個應用:

應用 1. **隨機排列 (random permutation) 的生成**. 給定

$$\{1, 2, \ldots, n\}$$

則共有 $n!$ 種排列. 試求一方法使得每種排列發生的機會均相等, 亦即,

$$P(\text{任一給定的排列}) = \frac{1}{n!}$$

另, 所得的排列稱爲隨機排列.

<解> 較直觀的演算法爲

演算法 1:

(1) 從 $\{1, 2, \ldots, n\}$ 中任取一數置放於第 n 個位置, 其機率爲 $\frac{1}{n}$.

(2) 從剩餘的 $(n-1)$ 個數中任取一數置放於第 $(n-1)$ 個位置, 其機率爲 $\frac{1}{n-1}$.

(3) 從剩餘的 $(n-2)$ 個數中任取一數置放於第 $(n-2)$ 個位置, 其機率為 $\frac{1}{n-2}$.

$\vdots$

(n-1) 從剩下的 2 數中任取一數置放於第 2 個位置, 其機率為 $\frac{1}{2}$.

(n) 將剩下的最後一數置放在第 1 個位置, 其機率為 1.

因此,

$$P(\text{得到的排列}) = \frac{1}{n} \cdot \frac{1}{n-1} \cdots \frac{1}{2} \cdot 1 = \frac{1}{n!}$$

確實是一個隨機排列, 但不易寫程式執行. 想想看!

一個可執行的修正如下述的

演算法 2:

(1) 先設定任一排列

$$\pi_1,\ \pi_2,\ \ldots,\ \pi_n$$

如, 1, 2, ..., n.

(2) 先從第 1 至第 n 個位置中, 任選一位置且與第 n 個位置的值交換, 其機率爲 $\frac{1}{n}$.

(3) 從第 1 至第 $(n-1)$ 個位置中, 任選一位置且與第 $(n-1)$ 個位置的值交換, 其機率爲 $\frac{1}{n-1}$.

(4) 從第 1 至第 $(n-2)$ 個位置中, 任選一位置且與第 $(n-2)$ 個位置的值交換, 其機率爲 $\frac{1}{n-2}$.

$\vdots$

(n) 從第 1 與第 2 個位置中, 任選一位置且與第 2 個位置的值交換, 其機率爲 $\frac{1}{2}$.

(n+1) 保留第一個位置的值, 其機率爲 1.

由此導出,

$$P(\text{得到的排列}) = \frac{1}{n}\cdot\frac{1}{n-1}\cdots\frac{1}{2}\cdot 1 = \frac{1}{n!}$$

確實是一隨機排列.

更具體的演算法如下述的

演算法 2:

(1) 先選定一排列

$$w_1,\ w_2,\ \ldots,\ w_n$$

(2) 令 $k = n$.

(3) 生成仿隨機數 U, 且令

$$I = \text{Int}(kU) + 1$$

(4) 交換 w_I 與 w_k.

(5) 令 $k = k - 1$, 且若 $k > 1$ 回到 (GOTO) (3).

(6) 得

$$w_1,\ w_2,\ \ldots,\ w_n$$

爲一隨機排列.

例如, 若 $n = 4$ 且起始排列爲

$$2,\ 1,\ 4,\ 3$$

則

$$\text{第 1 個 } I = 2 \ \Rightarrow\ 2,\ 3,\ 4,\ 1$$

$$\text{第 2 個 } I = 3 \ \Rightarrow\ 2,\ 3,\ 4,\ 1$$

$$\text{第 3 個 } I = 1 \ \Rightarrow\ 3,\ 2,\ 4,\ 1$$

練習題: 試回答下各小題.

(1) 以不同的 n, 求長度爲 n 的隨機排列各 10 個.

(2) 從

$$\{1, 2, \ldots, n\}$$

中任選 r 個數的選法有 $\binom{n}{r}$ 種. 若任一組 r 個數字被選中的機率均相同, 亦即,

$$P(\text{選中任一指定的 } r \text{ 個數}) = \frac{1}{\binom{n}{r}}$$

則稱所得的 r 個數爲一大小爲 r 的隨機子集合 (random subset of size r). 問如何生成一大小爲 r 的隨機子集合?

提示: 由隨機排列開始, 當

$$\text{第 } (n-r+1) \text{ 個位置}, \ldots, \text{第 } n \text{ 個位置}$$

內的值被決定後, 此 r 個值的集合就會是一個隨機子集合. 爲何如此? 想想看, 爲何

$$\begin{aligned}
&P(\text{選中此 } r \text{ 個值}) \\
&\quad = \frac{r!}{n(n-1)\cdots(n-r+1)} \\
&\quad = \frac{1}{\frac{n!}{r!(n-r)!}} \\
&\quad = \frac{1}{\binom{n}{r}}
\end{aligned}$$

成立, 且必須成立, 才能說明此選中的 r 個值可形成一隨機子集合?

應用 2. 求平均值. 令

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a(i)$$

其中 n 相當大 (如, n 為 100 億) 且 $a(i)$ 是複雜並難計算.

問. 如何估計 $\overline{a}$?

答. 令隨機變數

$$X \sim \text{discrete unif}(n)$$

則

$$\begin{aligned} E[a(X)] &= \sum_{i=1}^{n} a(i)p(X=i) \\ &= \sum_{i=1}^{n} a(i)\cdot\frac{1}{n} \\ &= \overline{a} \end{aligned}$$

亦即, $\overline{a}$ 可表成 X 的函數的期望值.

此外, 若

$$X_1, X_2, \ldots, X_k \overset{\text{iid}}{\sim} X$$

則

$$a(X_1), a(X_2), \ldots, a(X_k) \overset{\text{iid}}{\sim} a(X)$$

且共同的期望值爲

$$E[a(X)] = \overline{a}$$

因此, 當 k 夠大時 (即使遠小於 n), 由 SLLN, 得

$$\frac{1}{k}\sum_{i=1}^{k} a(X_i) \approx E[a(X)] = \overline{a}$$

以及對應的

演算法:

(1) 生成 k 個仿隨機數

$$u_1,\ u_2,\ \ldots,\ u_k$$

(2) 令

$$x_1 = \mathrm{Int}(nu_1) + 1$$

$$x_2 = \mathrm{Int}(nu_2) + 1$$

$$\vdots$$

$$x_k = \mathrm{Int}(nu_k) + 1$$

得 k 個 discrete unif(n) 隨機變數的值.

(3) 以

$$\frac{1}{k}\sum_{i=1}^{k} a(x_i) \approx \overline{a}$$

註. 因爲 $k \ll n$, 故是可行的.

問 3. 令隨機變數

$$X \sim \text{geometric}(p)$$

亦即,

$$P(X=i) = pq^{i-1},\ i = 1, 2, \ldots$$

亦相當於 X 爲獨立實驗中, 得到第一個成功, 所需的實驗次數, 其中

$$P(\text{成功}) = p,\ q = 1-p$$

如何模擬 X?

答 3. 由離散反轉換法知, 對於 $j \geq 1$,

$$X = j$$

當

$$\sum_{i=1}^{j-1} P(X=i) \le U < \sum_{i=1}^{j} P(X=i) \qquad (3)$$

此乃相當於

$$1 - P(X > j-1) \le U < 1 - P(X > j)$$

亦相當於

$$\begin{aligned} &1 - P(\text{前面 } j-1 \text{ 次均失敗}) \\ &\qquad \le U < 1 - P(\text{前面 } j \text{ 次均失敗}) \end{aligned}$$

也就是說,

$$1 - q^{j-1} \le U < 1 - q^{j}$$

化簡, 得

$$q^{j} < 1 - U \le q^{j-1} \qquad (4)$$

因爲

$$q < 1 \text{ 且 } q^{j-1} \ge 1 - U$$

故 (4) 式相當於

$$j \text{ 爲最小的 } j \text{ 使得 } q^{j} < 1 - U \qquad (5)$$

因此, 合併 (3) 式與 (5) 式, 得

$$
\begin{aligned}
X &= \min\{j : q^j < 1-U\} \\
&= \min\{j : j\log(q) < \log(1-U)\} \\
&= \min\left\{j : j > \frac{\log(1-U)}{\log(q)}\right\} \\
&= \text{大於 } \frac{\log(1-U)}{\log(q)} \text{ 的最小整數} \\
&= \mathrm{Int}\left(\frac{\log(1-U)}{\log(q)}\right) + 1 \\
&= \mathrm{Int}\left(\frac{\log(U)}{\log(q)}\right) + 1
\end{aligned}
$$

其中第三個等號成立乃因為

$$q < 1,\ \log(q) < 0$$

所致, 最後一個等號成立乃因為

$$U \sim \mathrm{unif}(0,1)$$

可導出

$$1 - U \sim \mathrm{unif}(0,1)$$

故可以同分布的 U 取代 $1-U$ 所致.

對應的

演算法:

(1) 生成仿隨機數 U.

(2) 令

$$X = \text{Int}\left(\frac{\log(U)}{\log(q)}\right) + 1$$

註. 同樣地, 經過等價的化簡後, 不需逐一地檢查子區間即可產生幾何隨機變數 X.