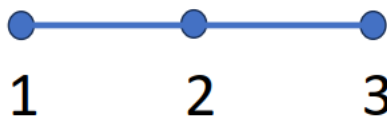


1. 三点问题



假设我们需使得粒子在这三个点上的概率分别为 0.2,0.5,0.3. 请设计随机游走过程加以实现, 其中, 每个粒子最多只能运动到和它当前位置直接相连的格点。

(1) 请给出算法步骤; (2) 编写相应程序; (3) 用程序游走 N 步 (设 $N=1000$), 统计不同格点出现的频数 (画出频数直方图即可)。

算法步骤:

- (1) 选择概率密度分布最大的位置 2 作为初始位置, 取位置 0 和位置 4 概率密度为 0 来处理边界;
- (2) 选择一个试探位置, 假定该点位置为 $x_{try} = x_n + \eta_n$, 其中 $\eta_n = \pm 1$, 可取 $[0,1]$ 上均匀分布的随机数 ξ_1 , 当 $\xi_1 > 0.5$ 时 $\eta = 1$, 否则 $\eta = -1$;
- (3) 计算 $r = \frac{f(x_{try})}{f(x_n)}$;
- (4) 如果 $r \geq 1$, 则接受这步游走, 取 $x_{n+1} = x_{try}$, 然后返回 (1) 开始下一步试探;
- (5) 如果 $r < 1$, 则产生另一个 $[0,1]$ 上均匀分布的随机数 ξ_2 ;
- (6) 如果 $\xi_2 \leq r$, 接受这步游走, 取 $x_{n+1} = x_{try}$, 回到 (1) 开始下一步游走;
- (7) 如果 $\xi_2 > r$, 拒绝这步游走, 仍留在 x_n ;
- (8) 返回步骤 (1), 开始对 x_{n+1} 的又一次试探。

MATLAB 代码:

```
N = 1000; % 进行 N 次游走
x = zeros(1, N); % 初始化坐标序列
x(1) = 2; % 选择初始位置为概率密度分布最大的点 2

i = 1;
while i < N
    % 选取步长-1 或 1
    xi_1 = rand;
    if xi_1 < 0.5
        eta = -1;
    else
        eta = 1;
    end
    % 进行试探
```

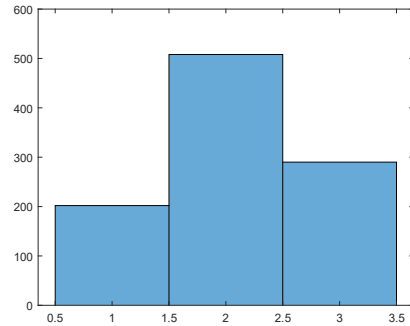
```

x(i + 1) = x(i) + eta;
r = f(x(i + 1)) / f(x(i));
% 如果 xi_2 > r 则拒绝游走，留在位置 i
if r < 1
    xi_2 = rand;
    if xi_2 > r
        x(i + 1) = x(i);
    end
end
i = i + 1;
end

histogram(x)

% 概率密度分布函数，当走出边界时概率密度为 0
function p = f(i)
    switch(i)
        case 0
            p = 0;
        case 1
            p = 0.2;
        case 2
            p = 0.5;
        case 3
            p = 0.3;
        case 4
            p = 0;
    end
end

```



PYTHON 代码:

```

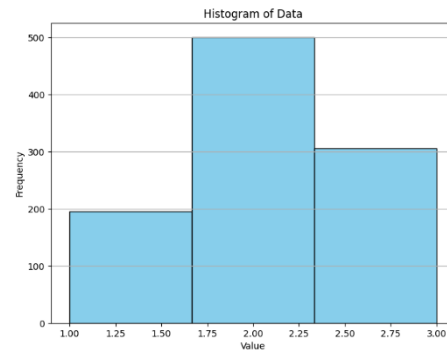
import random
from matplotlib import pyplot as plt

N = 1000
y0 = random.random()
if y0 < 0.2:
    x0 = 1
elif y0 < 0.7:
    x0 = 2
else:
    x0 = 3

Num = []

```

```
xn = x0
Num.append(x0)
for i in range(N):
    if xn < 2:
        xn = xn + 1
    elif xn > 2:
        xn = xn - 1
    else:
        y1 = random.random()
        if y1 < 0.6:
            xn = xn + 1
        else:
            xn = xn - 1
    Num.append(xn)
```



```
# 使用 matplotlib 绘制直方图
plt.figure(figsize=(8, 6)) # 设置图形大小
plt.hist(Num, bins=3, color='skyblue', edgecolor='black') # bins 指定分区数
plt.title('Histogram of Data') # 添加标题
plt.xlabel('Value') # X 轴标签
plt.ylabel('Frequency') # Y 轴标签
plt.grid(axis='y') # 添加网格线
plt.show()
```

评分标准 (10 分): 算法部分 5 分, 程序 3 分, 图 2 分。直方图三个位置的频数除 N 应等于该点的概率密度分布。