2012 年度 大問 3

hari64boli64 (hari64boli64@gmail.com)

2023 年 5 月 17 日

1　問題

ポアソン過程 計算問題
(注意力が無い人間には捨て問だと思う。私は注意力が無さ過ぎて三日掛かった)

2　解答

(1)

$$a(P(t,n-1)-P(t,n))+b((n+1)P(t,n+1)-nP(t,n))$$

(2)

$$M(t)=\frac{a}{b}+\Big(\lambda-\frac{a}{b}\Big)e^{-bt}$$

(3)

$$\frac{\partial G(t,s)}{\partial s}=(s-1)\left(aG(t,s)-b\frac{\partial G(t,s)}{\partial s}\right)$$

$\bar{G}(s)=1$ より、

$$\bar{G}(s)=e^{\frac{a}{b}(s-1)}$$

(4)

$G(0,s) = e^{\lambda(s-1)}$ より、

$$K(s-1) = e^{\left(\lambda - \frac{a}{b}\right)(s-1)}$$
$$K(x) = e^{\left(\lambda - \frac{a}{b}\right)x}$$

$$\begin{aligned} P(t,n) &= \frac{1}{n!}\frac{\partial^n}{\partial s^n} G(t,s)\Big|_{s=0} \\ &= \frac{1}{n!} M(t)^n e^{-M(t)} \end{aligned}$$

図 1 は、シミュレーション値と理論値が一致しすぎて、青と赤が重なり紫になっている。

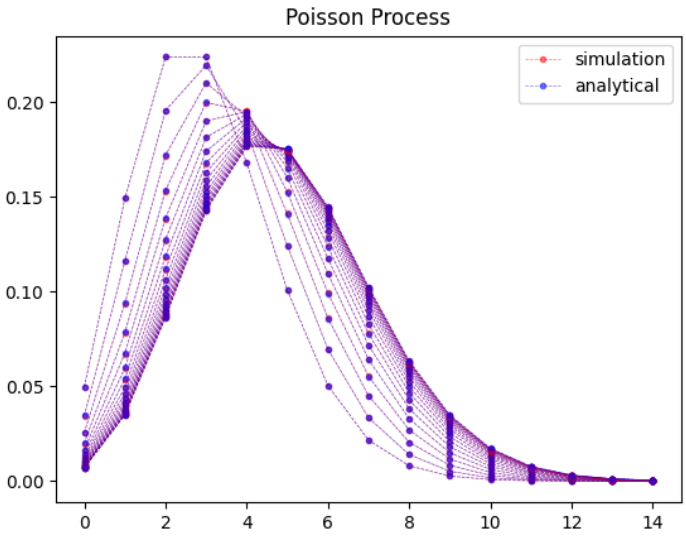

図 1　simulation

3　知識

なし

4　おまけ

ソースコード 1　simulation

```python
import matplotlib.pyplot as plt
import numpy as np

LAMBDA = 3
A = 10.0
B = 2.0
maxN = 15
timeStep = 100
timeLen = 2
dt = 1 / timeStep

N = np.arange(0, maxN, 1, dtype=np.int64)
P = (
    np.exp(-LAMBDA)
    * np.power(LAMBDA, N)
    / np.array([np.math.factorial(i) for i in range(maxN)])
).tolist()
Ps = [P]

for t in range(timeLen * timeStep):
    newP = [0 for _ in range(maxN)]
    for n in range(maxN):
        if n + 1 < maxN:
            newP[n + 1] += P[n] * A * dt
        if n - 1 >= 0:
            newP[n - 1] += P[n] * n * B * dt
        newP[n] += (1 - A * dt - n * B * dt) * P[n]
    Ps.append(newP)
    P = newP
Ps = np.array(Ps)

for t in range(0, timeLen * timeStep, timeLen * timeStep // 20):
    Mt = A / B + (LAMBDA - A / B) * np.exp(-B * (t / timeStep))
    analysis = [Mt**n / np.math.factorial(n) * np.exp(-Mt) for n
        in range(maxN)]
    plt.plot(
        N,
        Ps[t],
        label="simulation",
        linestyle="dashed",
        marker="o",
        markersize=3,
        linewidth=0.5,
        alpha=0.5,
        color="red",
    )
```

```
    plt.plot(
        N,
        analysis,
        label="analytical",
        linestyle="dashed",
        marker="o",
        markersize=3,
        linewidth=0.5,
        alpha=0.5,
        color="blue",
    )

handles, labels = plt.gca().get_legend_handles_labels()
by_label = dict(zip(labels, handles))
plt.legend(by_label.values(), by_label.keys())
plt.title("Poisson Process")
plt.savefig("3.png")
```