Putnam 2024 Problems

Vamshi Jandhyala

December 2024

Contents

1 Problem A5

Consider a circle Ω with radius 9 and center at the origin $(0,0)$, and a disk Δ with radius 1 and center at $(r,0)$, where $0 \leq r \leq 8$. Two points P and Q are chosen independently and uniformly at random on Ω. Which value(s) of r minimize the probability that the chord $\overline{PQ}$ intersects Δ?

1.1 Solution

Let the two random points on Ω be $P(9\cos\theta_1, 9\sin\theta_1)$ and $Q(9\cos\theta_2, 9\sin\theta_2)$ where $0 \leq \theta_1, \theta_2 \leq 2\pi$. The slope of the chord $\overline{PQ}$ is

$$\frac{9\sin\theta_2 - 9\sin\theta_1}{9\cos\theta_2 - 9\cos\theta_1} = -\cot\left(\frac{\theta_1+\theta_2}{2}\right). \tag{1.1}$$

The equation of the line passing through P and Q is given by

$$(y - 9\sin\theta_1) = -\cot\left(\frac{\theta_1+\theta_2}{2}\right)(x - 9\cos\theta_1). \tag{1.2}$$

The perpendicular distance d of the point from the center of disk Δ at $(r,0)$ to the above line is given by:

$$\begin{aligned} d &= \frac{\left|(r - 9\cos\theta_1)\cot\left(\frac{\theta_1+\theta_2}{2}\right) - 9\sin\theta_1\right|}{\sqrt{1+\cot^2\left(\frac{\theta_1+\theta_2}{2}\right)}} \\ &= \left|\sin\left(\frac{\theta_1+\theta_2}{2}\right)\right|\left|r\cot\left(\frac{\theta_1+\theta_2}{2}\right) - 9\csc\left(\frac{\theta_1+\theta_2}{2}\right)\cos\left(\frac{\theta_1-\theta_2}{2}\right)\right| \\ &= \left|r\cos\left(\frac{\theta_1+\theta_2}{2}\right) - 9\cos\left(\frac{\theta_1-\theta_2}{2}\right)\right|. \end{aligned} \tag{1.3}$$

For the chord $\overline{PQ}$ to intersect Δ, $d \leq 1$, i.e.

$$-1 \leq r\cos\left(\frac{\theta_1+\theta_2}{2}\right) - 9\cos\left(\frac{\theta_1-\theta_2}{2}\right) \leq 1. \tag{1.4}$$

Let $A(r,\theta_1,\theta_2)$ denote the area of the region given by Equation (1.4), the probability of the chord $\overline{PQ}$ intersecting Δ as a function of r is given by

$$P(r) = \frac{A(r,\theta_1,\theta_2)}{2\pi \cdot 2\pi} = \frac{A(r,\theta_1,\theta_2)}{4\pi^2}. \tag{1.5}$$

To be able to recast $A(r,\theta_1,\theta_2)$ as an integral, we use the substitution $\frac{\theta_1+\theta_2}{2} = u$ and $\frac{\theta_1-\theta_2}{2} = v$. We now have $0 \le u \le 2\pi, -\pi \le v \le \pi$ and the inequality Equation (1.4) is transformed into

$$-1 \le r\cos u - 9\cos v \le 1. \tag{1.6}$$

As $\theta_1 = u + v, \theta_2 = u - v$, the Jacobian matrix is

$$J = \begin{pmatrix} \frac{\partial\theta_1}{\partial u} & \frac{\partial\theta_1}{\partial v} \\ \frac{\partial\theta_2}{\partial u} & \frac{\partial\theta_2}{\partial v} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.7}$$

Therefore, $|\det(J)|$ for the transformation is 2. If we denote the region given by Equation (1.6) as $S(r,u,v)$, the probability of the chord $\overline{PQ}$ intersecting Δ as a function of r is given by

$$P(r) = \frac{S(r,u,v)}{2 \cdot 2\pi \cdot 2\pi} = \frac{S(r,u,v)}{8\pi^2}. \tag{1.8}$$

From Equation (1.6), we get

$$\begin{gathered} \frac{r\cos u + 1}{9} \ge \cos v \ge \frac{r\cos u - 1}{9} \\ \Rightarrow \arccos\left(\frac{r\cos u - 1}{9}\right) \ge v \ge \arccos\left(\frac{r\cos u + 1}{9}\right), 0 \le v \le \pi \end{gathered} \tag{1.9}$$

as the cosine function is one-to-one and decreasing in $[0,\pi]$. We can now write $S(r,u,v)$ as an integral,

$$S(r,u,v) = 2\int_0^{2\pi} \int_{\arccos(\frac{r\cos u+1}{9})}^{\arccos(\frac{r\cos u-1}{9})} 2dvdu = 16\int_0^{\frac{\pi}{2}} g(r,u)du \tag{1.10}$$

where

$$g(r,u) = \arccos\left(\frac{r\cos u - 1}{9}\right) - \arccos\left(\frac{r\cos u + 1}{9}\right). \tag{1.11}$$

We have

$$P(r) = \frac{2}{\pi^2}\int_0^{\frac{\pi}{2}} g(r,u)du \Rightarrow P'(r) = \frac{2}{\pi^2}\int_0^{\frac{\pi}{2}} \frac{\partial g(r,u)}{\partial r}du. \tag{1.12}$$

Now

$$\frac{\partial g(r,u)}{\partial r} = (\cos u)\left(\left(80 - 2r\cos u - r^2\cos^2 u\right)^{-\frac{1}{2}} - \left(80 + 2r\cos u - r^2\cos^2 u\right)^{-\frac{1}{2}}\right) \tag{1.13}$$

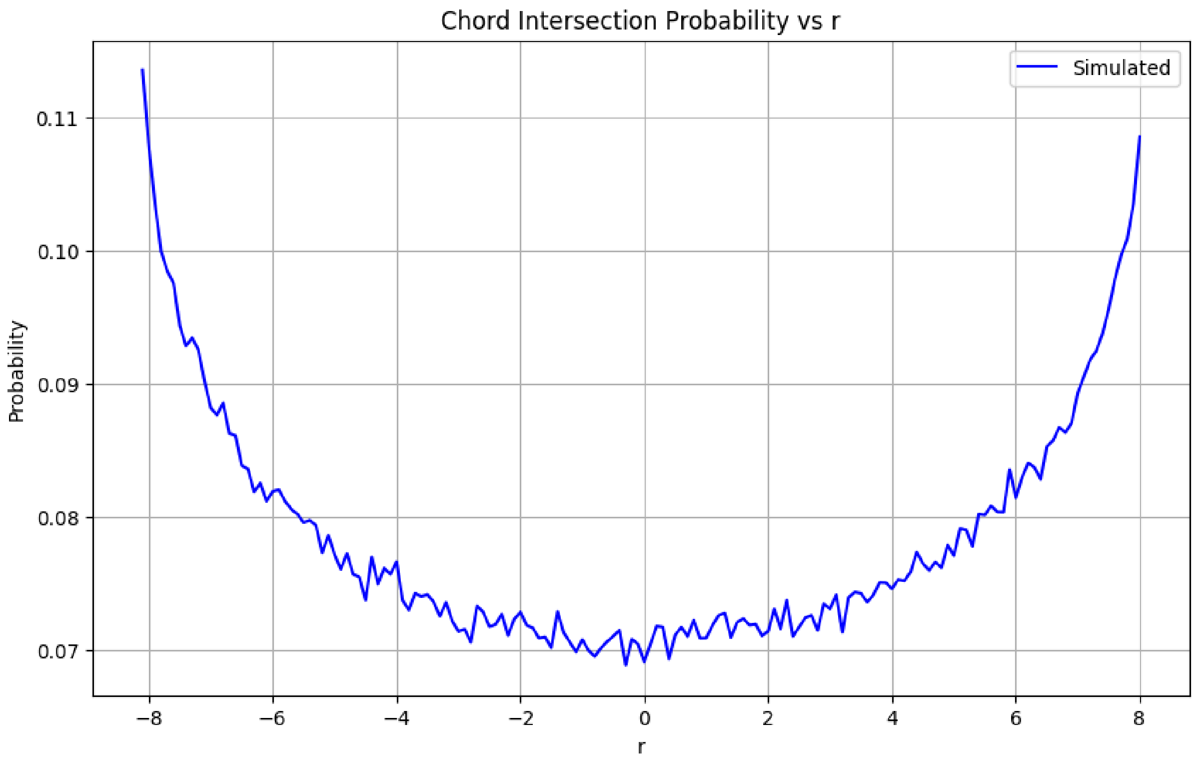
which, for $u \in \left(0, \frac{\pi}{2}\right)$, is zero for $r = 0$ and strictly positive for $r > 0$. It follows that $P'(0) = 0$ and $P'(r) > 0$ for $r \in (0, 8]$. Therefore, $P(r)$ is minimized for $\boldsymbol{r = 0}$.

1.2 Estimation of probability using Monte-Carlo

In the second approach, we follow the procedure below to estimate the probability of intersection:

(i) Choose two random points on the circumference of Ω.
(ii) Derive the equation of the line passing through the two points chosen above.
(iii) Calculate the perpendicular distance from the center $(r, 0)$ of Δ to the line using the formula $\left|\frac{ar+c}{\sqrt{a^2+b^2}}\right|$ where $ax + by + c = 0$ is the equation of the line.
(iv) Repeat the above steps multiple times and calculate the fraction of trials where the perpendicular distance is less than 1, the radius of Δ.

The above procedure gives us the following plot of the estimate of the probability as a function of r:



From the symmetry of the plot we can see that the probability is minimized when $\boldsymbol{r = 0}$.

1.2.1 Python code

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm

def get_line_equation(p1, p2):
    """Returns a, b, c for line ax + by + c = 0"""
    a = p2[1] - p1[1]
    b = p1[0] - p2[0]
    c = p2[0]*p1[1] - p1[0]*p2[1]
    return a, b, c

def perpendicular_distance(line_coeffs, point):
    """Calculate perpendicular distance from point to line ax + by + c = 0"""
    a, b, c = line_coeffs
```

```
    x0, y0 = point
    return abs(a*x0 + b*y0 + c) / np.sqrt(a*a + b*b)

def simulate_probability(r, n_trials=100000):
    """Simulate using perpendicular distance method"""
    R = 9  # radius of large circle
    intersections = 0

    for _ in range(n_trials):
        # Generate random points on circle
        theta1, theta2 = np.random.uniform(0, 2*np.pi, 2)
        p1 = R * np.array([np.cos(theta1), np.sin(theta1)])
        p2 = R * np.array([np.cos(theta2), np.sin(theta2)])

        # Get line equation and calculate distance
        line_coeffs = get_line_equation(p1, p2)
        dist = perpendicular_distance(line_coeffs, [r, 0])

        if dist <= 1:  # radius of small circle is 1
            intersections += 1

    return intersections / n_trials

# Run simulation
r_values = np.arange(-8.1, 8.1, 0.1)
probabilities = []

for r in tqdm(r_values):
    prob = simulate_probability(r)
    probabilities.append(prob)

# Plot results
plt.figure(figsize=(10, 6))
plt.plot(r_values, probabilities, 'b-', label='Simulated')
plt.grid(True)
plt.xlabel('r')
plt.ylabel('Probability')
plt.title('Chord Intersection Probability vs r')
plt.legend()
plt.show()
```

2 Problem B4

Let n be a positive integer. Set $a_{n,0} = 1$. For $k \geq 0$, choose an integer $m_{n,k}$ uniformly at random from the set $\{1, 2, ..., n\}$, and let

$$a_{n,k+1} = \begin{cases} a_{n,k} + 1 \text{ , if } m_{n,k} > a_{n,k}; \\ a_{n,k} \text{ , if } m_{n,k} = a_{n,k}; \\ a_{n,k} - 1 \text{ , if } m_{n,k} < a_{n,k}. \end{cases} \tag{2.14}$$

Let $E(n)$ be the expected value of $a_{n,n}$. Determine $\lim_{n\to\infty} \frac{E(n)}{n}$.

2.1 Solution

Firstly, we notice that $1 \leq a_{n,n} \leq n$. Dropping the subscript n we can write the equation Equation (2.14) as

$$a_{k+1} = a_k + X \tag{2.15}$$

where X is a random variable that takes the values $1, 0$ and -1 when $m_k > a_k, m_k = a_k$ and $m_k < a_k$ respectively. Taking expectation on both sides of Equation (2.15) we get

$$\mathbb{E}[a_{k+1}] = \mathbb{E}[a_k] + \mathbb{E}[X]. \tag{2.16}$$

To calculate $\mathbb{E}[X]$, we make use of the **Law of Iterated Expectations**,

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|a_k]] = \mathbb{E}\left[1 \cdot \frac{n - a_k}{n} + (-1) \cdot \frac{a_k - 1}{n}\right] = \left(-\frac{2}{n}\right)\mathbb{E}[a_k] + \frac{n+1}{n}. \tag{2.17}$$

Substituting the value of $\mathbb{E}[X]$ from Equation (2.17) in Equation (2.16), we get

$$\mathbb{E}[a_{k+1}] = \frac{n-2}{n}\mathbb{E}[a_k] + \frac{n+1}{n}. \tag{2.18}$$

Setting $b(k) = \mathbb{E}[a_k]$ in Equation (2.18), we get the following recurrence relation

$$b(k+1) = \frac{n-2}{n}b(k) + \frac{n+1}{n} \tag{2.19}$$

with $b(0) = 1$. We use generating functions to solve the above linear recurrence relation. Let

$$G(z) = \sum_{k=0}^{\infty} b(k) z^k \tag{2.20}$$

Multiplying both sides of the recurrence relation by z^k and summing up both sides of the equation term by term over the nonnegative integers, we get

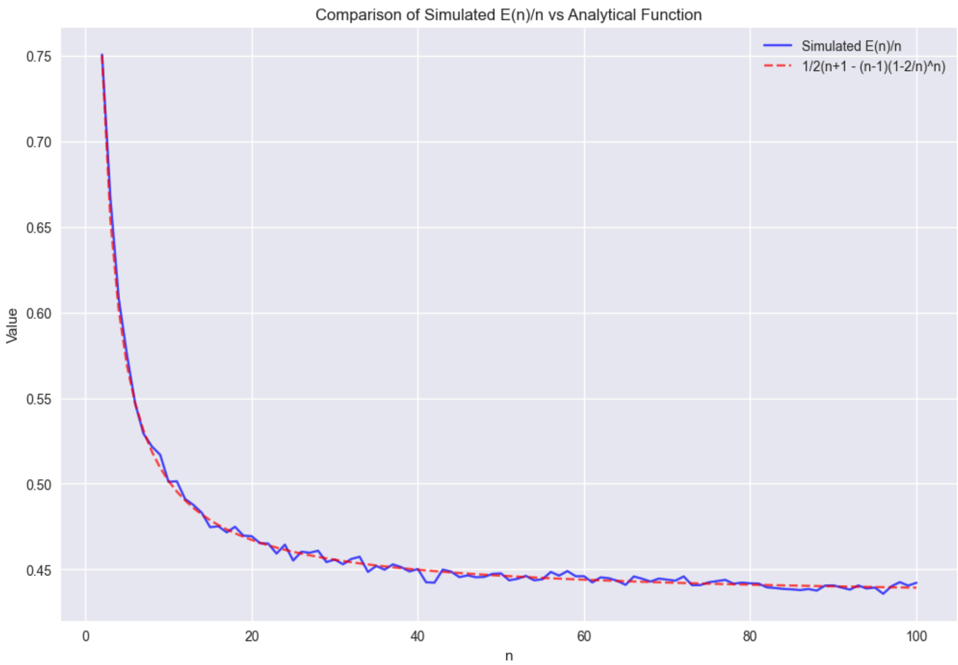
$$
\begin{aligned}
&\sum_{k=0}^{\infty} b(k+1)z^k + \frac{2-n}{n}\sum_{k=0}^{\infty} b(k)z^k = \frac{n+1}{n}\sum_{k=0}^{\infty} z^k \\
\Rightarrow &\sum_{k=0}^{\infty} b(k+1)z^{k+1}z^{-1} + \frac{2-n}{n}G(z) = \frac{n+1}{n(1-z)} \\
\Rightarrow &(G(z)-b(0))z^{-1} + \frac{2-n}{n}G(z) = \frac{n+1}{n(1-z)} \\
\Rightarrow &G(z) = \frac{n+z}{(n-(n-2)z)(1-z)} \\
\Rightarrow &G(z) = \frac{1}{2}\sum_{k=0}^{\infty}\left(n+1-(n-1)\left(\frac{n-2}{n}\right)^k\right)z^k.
\end{aligned} \tag{2.21}
$$

Therefore,

$$
\lim_{n\to\infty}\frac{E(n)}{n} = \lim_{n\to\infty}\frac{b(n)}{n} = \lim_{n\to\infty}\frac{1}{2}\frac{n+1-(n-1)\left(\frac{n-2}{n}\right)^n}{n} = \boldsymbol{\frac{1}{2}(1-e^{-2})}. \tag{2.22}
$$

2.2 Computational verification

From the code below we get the following plot for the simulated value of $\frac{E(n)}{n}$ against $\frac{b(n)}{n}$ calculated analytically:



2.2.1 Python Code

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

def simulate_process(n, num_trials=1000):
    final_values = []
    for _ in range(num_trials):
        a = 1
        for _ in range(n):
            m = np.random.randint(1, n+1)
            if m > a:
                a += 1
            elif m < a:
                a -= 1
        final_values.append(a)
    return np.mean(final_values)

def analytical_function(n):
    return 0.5 * ((n+1 - (n-1)*((n-2)/n)**n)/n)

# Generate data points
n_values = np.arange(2, 101)  # n from 1 to 100
simulated_values = []
analytical_values = []

# Calculate both simulated and analytical values
print("Running simulation...")
for n in n_values:
    if n == 1:  # Special case for n=1
        simulated_values.append(1)
    else:
        en = simulate_process(n)
        simulated_values.append(en/n)
    analytical_values.append(analytical_function(n))

# Create the plot
plt.figure(figsize=(12, 8))
plt.plot(n_values, simulated_values, 'b-', label='Simulated E(n)/n', alpha=0.7)
plt.plot(n_values, analytical_values, 'r--', label='1/2(n+1 - (n-1)(1-2/n)^n)',
alpha=0.7)
plt.xlabel('n')
plt.ylabel('Value')
plt.title('Comparison of Simulated E(n)/n vs Analytical Function')
plt.grid(True)
plt.legend()
plt.show()
```