Esercizio 15* del Foglio 2

Dennis Parolin 2113203, Tommaso Ceron 2101045, Lorenzo Soligo 2101057

Marzo 2025

Dimostrazione dell'esercizio 14

Sia $N \in \mathbb{N}$, $q \in (0,1)$. Definiamo la distribuzione binomiale:

$$p_{Bin(N,q)}(k) = \binom{N}{k} q^k (1-q)^{N-k}, \quad \text{per } k \in \{0, \dots, N\}$$

Parte 1

Dimostriamo che per $k \in \{0, \dots, N-1\}$:

$$p_{Bin(N,q)}(k+1) = \frac{q}{1-q} \cdot \frac{N-k}{k+1} \cdot p_{Bin(N,q)}(k)$$

Partiamo dalla definizione:

$$p_{Bin(N,q)}(k+1) = \binom{N}{k+1} q^{k+1} (1-q)^{N-k-1}$$

Usiamo la relazione tra coefficienti binomiali:

$$\binom{N}{k+1} = \frac{N-k}{k+1} \binom{N}{k}$$

E riscriviamo:

$$p_{Bin(N,q)}(k+1) = \frac{N-k}{k+1} \cdot \binom{N}{k} \cdot q \cdot q^k \cdot \frac{(1-q)^{N-k}}{1-q}$$

$$= \frac{q}{1-q} \cdot \frac{N-k}{k+1} \cdot \binom{N}{k} q^k (1-q)^{N-k} = \frac{q}{1-q} \cdot \frac{N-k}{k+1} \cdot p_{Bin(N,q)}(k)$$

Parte 2

Verifichiamo che:

$$p_{Bin(N,q)}(k) = p_{Bin(N,1-q)}(N-k)$$

Infatti:

$$p_{Bin(N,q)}(k) = \binom{N}{k} q^k (1-q)^{N-k}$$

$$p_{Bin(N,1-q)}(N-k) = \binom{N}{N-k} (1-q)^{N-k} q^k = \binom{N}{k} q^k (1-q)^{N-k}$$

Poiché $\binom{N}{k} = \binom{N}{N-k}$, le due espressioni coincidono.

Esercizio 15

Supponiamo che $N + M$ persone votino, con M a favore del candidato A, e N voti casuali (variabile $S_N \sim Bin(N, 1/2)$).

Vogliamo la probabilità:

$$P\left(S_N + M > \frac{N+M}{2}\right) = P\left(S_N > \frac{N-M}{2}\right) = \sum_{k=\lfloor \frac{N-M}{2} \rfloor + 1}^{N} p_{Bin(N,1/2)}(k) \tag{1}$$

Giustificazione

Il termine $\lfloor \frac{N-M}{2} \rfloor + 1$ rappresenta il numero minimo di voti casuali che il candidato A deve ottenere per superare il 50% del totale.

Pseudocodice

```
Input: N + M (numero totale di elettori), M (voti certi per A)
Output: Probabilità che A vinca

1. N = (N + M) - M
2. k_min = floor((N - M)/2) + 1
3. Calcola la somma delle probabilità binomiali da k_min a N
4. Output della somma
```

Codice Python (semplificato e commentato)

Questo codice è quello definitivo con tutti i miglioramenti che impediscono l'overflow e velocizzano i calcoli.

```
from scipy.stats import binom
import numpy as np
import matplotlib.pyplot as plt

def probability_A_wins(N, M):
    k_min = (N - M) // 2 + 1
    return binom.sf(k_min - 1, N, 0.5)

N_plus_M = 10**6
M_values = np.arange(0, 5001, 10)
probabilities = [probability_A_wins(N_plus_M - M, M) for M in M_values]

plt.plot(M_values, probabilities)
plt.xlabel("M")
plt.ylabel("Probabilità")
plt.title("Probabilità di vittoria elettorale in funzione di M")
plt.grid(True)
plt.savefig("probabilita_vittoria.pdf")
```

Grafico

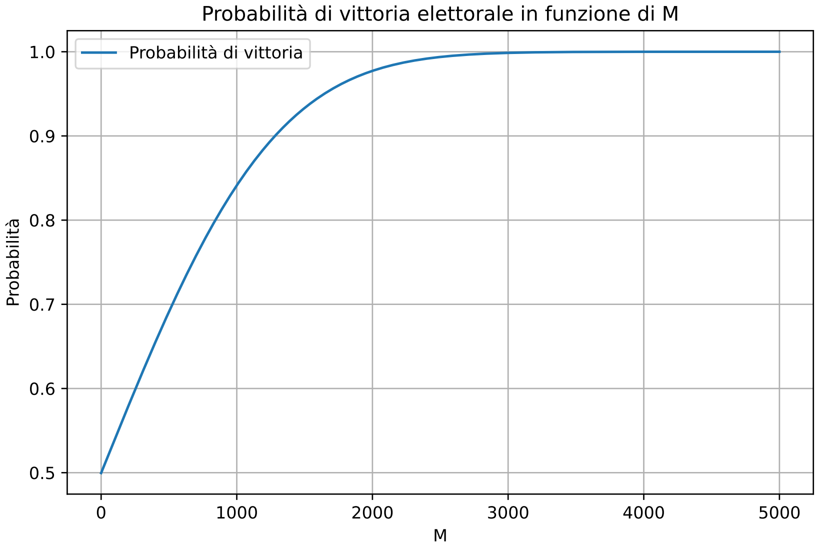

Figure 1: Probabilità di vittoria elettorale in funzione di M

Versioni Precedenti

Applicazione alla lettera

Questo codice segue alla lettera (1), provocando però overflow per la grande dimensione di M

```
import math
import random

def binomial_pmf(N, k):
    if 0 <= k <= N:
        return math.comb(N, k) * (0.5 ** k) * (0.5 ** (N - k))
    return 0

def probability_A_wins(N, M):
    k_min = math.floor((N - M) / 2) + 1
    return sum(binomial_pmf(N, k) for k in range(k_min, N + 1))

N_plus_M = 10**6  # Numero minimo di elettori
M = random.randint(0, 5000)  # Genera M casualmente tra 0 e 5000
N = N_plus_M - M

print(f"Numero di elettori già decisi (M): {M}")
print(f"Probabilità che vinca A: {probability_A_wins(N, M):.6f}")
```

Applicazione con Approssimazione della distribuzione

```
def probability_A_wins(N, M):
    # Approssimazione normale della distribuzione binomiale
    mu = N / 2  # Media della distribuzione binomiale
    sigma = math.sqrt(N / 4)  # Deviazione standard della distribuzione binomiale
    k_min = math.floor((N - M) / 2) + 1

    # Calcoliamo la probabilità cumulativa dalla distribuzione normale
```

```
    return 1 - norm.cdf(k_min, loc=mu, scale=sigma)

N_plus_M = 10**6  # Numero minimo di elettori
M = random.randint(0, 5000)  # Genera M casualmente tra 0 e 5000
N = N_plus_M - M

print(f"Numero di elettori già decisi (M): {M}")
print(f"Probabilità che vinca A: {probability_A_wins(N, M):.6f}")
```