

lab5-task1

August 28, 2024

1 Laboratorio 5 - Variables Aleatorias

1.1 Task 1

Genere muestras aleatorias a partir de una distribución de probabilidad discreta personalizada mediante el método de aceptación-rechazo. 1. Defina una distribución de probabilidad discreta objetivo con valores y probabilidades especificados. 2. Elija una distribución de propuesta con un soporte mayor que cubra el soporte de la distribución de destino. a. N.B: El “soporte” de una variable aleatoria o distribución de probabilidad se refiere al conjunto de valores para los cuales la variable aleatoria tiene una probabilidad distinta de cero. En otras palabras, es el rango de valores donde la distribución de probabilidad es positiva. 3. Calcule la constante C para acotar la relación entre el PMF objetivo y el PMF propuesto. 4. Implemente el método de aceptación-rechazo para generar muestras aleatorias a partir de la distribución discreta personalizada. 5. Genere una muestra aleatoria de tamaño 1000 a partir de la distribución personalizada. 6. Trace un histograma de la muestra generada y compárelo con el PMF objetivo.

```
[17]: import numpy as np
import matplotlib.pyplot as plt

# Paso 1: Definir la distribución de probabilidad discreta objetivo
valores_objetivo = np.arange(1, 11)
probabilidades_objetivo = np.array([0.05, 0.1, 0.15, 0.2, 0.25, 0.15, 0.05, 0.
    ↪0.25, 0.015, 0.01])

# Paso 2: Elegir una distribución de propuesta con soporte mayor
# Usaremos una distribución geométrica truncada como propuesta
p = 0.3
valores_propuesta = np.arange(1, 21) # La geométrica truncada hasta 20
probabilidades_propuesta = (1 - p) ** (valores_propuesta - 1) * p
probabilidades_propuesta /= probabilidades_propuesta.sum() # Normalizamos para
    ↪truncar

# Paso 3: Calcular la constante C
# Extendemos el PMF objetivo para el rango de la propuesta, llenando con ceros
    ↪donde no tiene soporte
probabilidades_objetivo_extendida = np.zeros_like(valores_propuesta,
    ↪dtype=float)
```

```

probabilidades_objetivo_extendida[:len(probabilidades_objetivo)] =
    ↪probabilidades_objetivo

C = max(probabilidades_objetivo_extendida / probabilidades_propuesta)

# Paso 4: Implementar el método de aceptación-rechazo
def generar_muestra(n):
    muestras = []
    while len(muestras) < n:
        propuesta = np.random.choice(valores_propuesta,
    ↪p=probabilidades_propuesta)
        u = np.random.uniform(0, 1)
        # Usamos las probabilidades normalizadas como "densidades" en este
    ↪contexto discreto
        if u < probabilidades_objetivo_extendida[valores_propuesta ==
    ↪propuesta] / (C * probabilidades_propuesta[valores_propuesta == propuesta]):
            muestras.append(propuesta)
    return np.array(muestras)

# Paso 5: Generar una muestra aleatoria de tamaño 1000
muestra = generar_muestra(1000)

# Paso 6: Trazar un histograma de la muestra generada y comparar con el PMF
    ↪objetivo
plt.hist(muestra, bins=len(valores_objetivo), density=True, alpha=0.6,
    ↪color='g', label="Muestra generada")
plt.plot(valores_objetivo, probabilidades_objetivo, 'ro-', label="PMF objetivo")
plt.xlabel("Valores")
plt.ylabel("Probabilidad")
plt.title("Histograma de la muestra generada vs PMF objetivo")
plt.legend()
plt.show()

```

