Contents

Clase 5: Generación de Variables Aleatorias - Técnica de Aceptación y Rechazo

Objetivos:

- Comprender los fundamentos de la técnica de aceptación y rechazo para la generación de variables aleatorias.
- Identificar cuándo es apropiado utilizar la técnica de aceptación y rechazo en lugar de la transformada inversa.
- Aplicar la técnica de aceptación y rechazo para generar variables aleatorias de distribuciones continuas y discretas.
- Calcular la eficiencia de la técnica de aceptación y rechazo.

Contenido Teórico:

La técnica de aceptación y rechazo (también conocida como método de aceptación-rechazo) es un método general para generar variables aleatorias a partir de una distribución de probabilidad arbitraria, especialmente útil cuando la transformada inversa es difícil de derivar o computacionalmente costosa. Este método se basa en generar valores a partir de una distribución de probabilidad auxiliar (más simple) y luego aceptar o rechazar estos valores según un criterio definido.

Fundamentos:

1. **Distribución Objetivo *f(x)*:** Es la distribución de probabilidad de la variable aleatoria que deseamos generar. Debe ser conocida (al menos hasta una constante de normalización).

2. **Distribución Propuesta *g(x)*:** Es una distribución de probabilidad auxiliar, más fácil de generar, que "cubre" a la distribución objetivo. Formalmente, existe una constante *c* tal que *f(x)*   *c g(x)* para todo *x*. La elección de *g(x)* es crucial para la eficiencia del método.

3. **Constante *c*:** Es un factor de escala que asegura que la curva *c g(x)* envuelva completamente la curva *f(x)*. El valor de *c* debe ser lo más pequeño posible para maximizar la eficiencia.

Algoritmo General:

1. Generar un valor *x* de la distribución propuesta *g(x)*.
2. Generar un número aleatorio uniforme *u* entre 0 y 1.
3. Si *u*   *f(x) / (c * g(x))*, entonces aceptar *x* como un valor de la distribución objetivo *f(x)*.
4. Si *u > f(x) / (c * g(x))*, entonces rechazar *x* y volver al paso 1.

Eficiencia:

La eficiencia de la técnica de aceptación y rechazo se define como la probabilidad de aceptar un valor generado, que es igual a $1/c$. Por lo tanto, minimizar c maximiza la eficiencia. Una alta eficiencia implica que se requiere un menor número de iteraciones para obtener una muestra de tamaño dado de la distribución objetivo.

Ejemplo: Generación de una Variable Aleatoria con Densidad Triangular

Supongamos que queremos generar valores de una variable aleatoria con la siguiente función de densidad triangular:

- *f(x) = 2x* para 0   x   1
- *f(x) = 0* en otro caso.

Pasos:

1. **Distribución Propuesta:** Elegimos una distribución uniforme *g(x) = 1* para 0 x 1.
2. **Constante c:** Necesitamos encontrar *c* tal que *f(x) c g(x)* para todo *x*. El máximo de *f(x)* es 2 (en x=1), por lo tanto, *c = 2*.
3. **Algoritmo:**

   a. Generar *x* ~ U(0, 1). b. Generar *u* ~ U(0, 1). c. Si *u f(x) / (c * g(x)) = (2x) / (2 * 1) = x*, entonces aceptar *x*. d. Sino, rechazar *x* y volver al paso a.

Caso de Estudio: Generación de una Distribución Beta

La distribución Beta es utilizada frecuentemente para modelar probabilidades. Su función de densidad es:

f(x) = x $^{-1}$ *(1-x)* $^{-1}$ */ B(,)* para 0 x 1, donde B(,) es la función Beta.

Cuando y no son enteros, la transformada inversa es muy complicada. Aceptación y rechazo puede ser una buena opción. Usaremos una distribución uniforme [0,1] como distribución propuesta.

Problemas Prácticos y Ejercicios:

Problema 1: Generación de una Distribución Normal Truncada (0, ∞)

Diseñar un algoritmo de aceptación y rechazo para generar valores de una distribución normal estándar truncada en el intervalo (0, ∞). Utilizar una distribución exponencial como distribución propuesta.

Solución:

1. **Distribución Objetivo:** *f(x) = (2 / √(2)) * exp(-x*^{2} */ 2)*, x > 0 (Normal estándar truncada a la mitad derecha). El factor 2 viene de la normalización para que la integral de 0 a infinito sea 1.
2. **Distribución Propuesta:** *g(x) = exp(- x)*, x > 0 (Exponencial con parámetro). Debemos elegir de forma inteligente.
3. **Constante c:** Encontrar el valor óptimo de *c* requiere un poco de cálculo. Necesitamos maximizar la razón *f(x) / g(x)*. Derivando e igualando a cero, se puede demostrar que el valor de *x* que maximiza la razón es la solución de la ecuación x = . Por lo tanto, *c = f( ) / g( ) = (2 / √(2)) * exp(-* ^{2} */ 2) /* . Para simplificar y para asegurar que *c* sea finito y razonable, podemos fijar *= 1* (o cualquier otro valor positivo, pero =1 es común y razonable). En este caso, *c = 2 * exp(-0.5) / √(2) 0.4839. Nota: En realidad, la constante c debe ser el INVERSO de este valor. c 2.066*
4. **Algoritmo:**

   a. Generar *x* ~ Exponencial( ) (con =1). b. Generar *u* ~ U(0, 1). c. Si *u f(x) / (c * g(x)) = exp(-x*^{2} */ 2) / (c * exp(-x)) = exp(x - x*^{2}*/2) / c*, entonces aceptar *x*. d. Sino, rechazar *x* y volver al paso a.

Problema 2: Simulación en Python

Implementar en Python el algoritmo de aceptación y rechazo para generar 1000 valores de la distribución normal truncada del Problema 1. Calcular y mostrar la tasa de aceptación (número de valores aceptados / número total de intentos). Graficar un histograma de los valores generados y compararlo con la función de densidad teórica.

Solución (Código Python):

“‘python import numpy as np import matplotlib.pyplot as plt from scipy.stats import norm

def generar_normal_truncada(n, lambda_val=1): ”””Genera n valores de una normal truncada (0, inf) usando aceptación y rechazo.””” aceptados = [] intentos = 0 c = (2 / np.sqrt(2 * np.pi)) / lambda_val # Corrección: c es la INVERSA de esta cantidad while len(aceptados) < n: x = np.random.exponential(1/lambda_val) # Corregido: Parámetro de scale para np.random.exponential u = np.random.uniform(0, 1) if u <= (norm.pdf(x) / (c * (lambda_val * np.exp(-lambda_val * x)))): aceptados.append(x) intentos += 1 return aceptados, intentos n_samples = 1000 samples, total_intentos = generar_normal_truncada(n_samples) tasa_aceptacion = n_samples / total_intentos

print(f"Tasa de Aceptación: {tasa_aceptacion:.4f}")

Graficar Histograma

plt.hist(samples, bins=30, density=True, alpha=0.7, label="Muestra Generada")

Graficar la Densidad Teórica (Normal Truncada)

x = np.linspace(0, 5, 100) pdf_truncada = (2 / np.sqrt(2 * np.pi)) * np.exp(-x**2 / 2) plt.plot(x, pdf_truncada, 'r-', label="Densidad Teórica Normal Truncada")

plt.xlabel("x") plt.ylabel("Densidad") plt.title("Generación de Normal Truncada (Aceptación y Rechazo)") plt.legend() plt.show() "'

Materiales Complementarios Recomendados:

- **Libros de texto:** Cualquier libro de texto de simulación estocástica o modelado estocástico cubrirá la técnica de aceptación y rechazo en detalle. Busca secciones sobre generación de números aleatorios no uniformes.
- **Artículos:** Busca artículos científicos sobre la optimización de la técnica de aceptación y rechazo para distribuciones específicas.
- **Recursos en línea:** Khan Academy (para repaso de distribuciones de probabilidad) y documentación de bibliotecas de simulación en Python (NumPy, SciPy).