Contents

Clase 4: Transformada Inversa - Aplicaciones y Distribuciones Discretas

1. Objetivos de la Clase:

- Comprender en profundidad la técnica de la transformada inversa para la generación de variables aleatorias.
- Aplicar la transformada inversa para generar variables aleatorias a partir de distribuciones continuas específicas.
- Adaptar la técnica de la transformada inversa para generar variables aleatorias discretas.
- Analizar las ventajas y desventajas de la transformada inversa en comparación con otros métodos.

2. Contenido Teórico Detallado:

2.1. Repaso de la Transformada Inversa:

La técnica de la transformada inversa es un método fundamental para generar variables aleatorias a partir de una distribución de probabilidad dada. Se basa en la función de distribución acumulada (FDA) *F(x)* de la variable aleatoria deseada. El principio es simple:

1. Generar un número aleatorio u uniformemente distribuido en el intervalo (0, 1).
2. Calcular $x = F^{-1}(u)$, donde F^{-1} es la función inversa de la FDA. Este valor x es una realización de la variable aleatoria con la distribución deseada.

La validez de este método se basa en que *F(x)* mapea los valores de la variable aleatoria x al intervalo (0, 1), y su inversa, $F^{-1}(u)$, realiza el mapeo inverso. Como u está uniformemente distribuido en (0, 1), los valores resultantes de x seguirán la distribución definida por *F(x)*.

2.2. Aplicación a Distribuciones Continuas:

La transformada inversa es particularmente útil cuando la FDA *F(x)* tiene una forma analítica conocida y su inversa $F^{-1}(u)$ puede ser calculada explícitamente. Veamos algunos ejemplos comunes:

- **Distribución Exponencial:** Como ya vimos en la clase anterior, la FDA es *F(x) = 1 - exp(- x)*, para *x 0*. Su inversa es $F^{-1}(u) = -ln(1-u)/$. Como u y (1-u) están ambas uniformemente distribuidas en (0, 1), podemos simplificar a $x = -ln(u)/$. Esto significa que para generar una variable aleatoria exponencial, simplemente generamos un número aleatorio uniforme u y aplicamos esta fórmula.
- **Distribución Uniforme Continua (a, b):** La FDA es *F(x) = (x - a) / (b - a)*, para *a x b*. Su inversa es $F^{-1}(u) = a + u(b - a)$. Por lo tanto, generar un número aleatorio uniforme u y aplicar esta fórmula produce una variable aleatoria uniformemente distribuida entre a y b.
- **Distribución de Weibull:** La FDA es $F(x) = 1 - exp(-(x/)^k)$, para $x \quad 0$, donde es el parámetro de escala y k es el parámetro de forma. Su inversa es $F^{-1}(u) = (-ln(1-u))^{1/k}$. Nuevamente, como u y (1-u) son ambas uniformemente distribuidas en (0, 1), podemos simplificar a $x = (-ln(u))^{1/k}$.

2.3. Aplicación a Distribuciones Discretas:

La transformada inversa también se puede aplicar a distribuciones discretas. La adaptación requiere manejar el hecho de que la FDA es una función escalonada en lugar de continua.

El proceso es el siguiente:

1. Calcula la FDA acumulada *F(x)* para cada valor posible x_i de la variable aleatoria discreta.
2. Genera un número aleatorio u uniformemente distribuido en (0, 1).
3. Encuentra el valor x_i tal que $F(x_{i-1}) < u \quad F(x_i)$. Este valor x_i es la realización de la variable aleatoria discreta.

En la práctica, esto se implementa buscando el primer valor x_i cuya FDA es mayor o igual que u.

- **Distribución de Bernoulli:** Esta distribución tiene dos posibles valores: 0 (con probabilidad 1-p) y 1 (con probabilidad p). La FDA es:
  - F(0) = 1-p
  - F(1) = 1 Para generar una variable aleatoria de Bernoulli:
  - Generar u.
  - Si u *1-p*, entonces retornar 0.
  - Si $u >$ *1-p*, entonces retornar 1.
- **Distribución de Poisson:** La FDA se calcula como $F(x) = \Sigma_{i=0}^{x} (e^{-} {}^{i})/i!$, donde es el parámetro de la distribución de Poisson. Para generar una variable aleatoria de Poisson, generamos u y buscamos el menor valor de x tal que *F(x)* u. Debido a que la FDA de Poisson no tiene una forma cerrada simple, la búsqueda generalmente se realiza iterativamente.

2.4. Ventajas y Desventajas:

- **Ventajas:**
  - Es un método conceptualmente simple y fácil de entender.
  - Es exacto: genera variables aleatorias que siguen precisamente la distribución deseada (siempre que se pueda calcular la inversa de la FDA).
- **Desventajas:**
  - Requiere conocer la forma analítica de la FDA y poder calcular su inversa, lo cual no siempre es posible.
  - Para distribuciones discretas donde la FDA debe ser evaluada iterativamente (como la Poisson), puede ser computacionalmente costoso.

3. Ejemplos y Casos de Estudio:

Ejemplo 1: Simulación de tiempos de espera en un call center.

Supongamos que el tiempo de espera en un call center sigue una distribución exponencial con una media de 5 minutos (= 1/5). Podemos usar la transformada inversa para simular estos tiempos de espera. Generamos $u \sim$ U(0,1) y calculamos x = -*5*ln(u)*. Repitiendo este proceso muchas veces, obtenemos una muestra de tiempos de espera que podemos usar para analizar el rendimiento del call center.

Ejemplo 2: Simulación del número de llegadas a una tienda por hora.

Supongamos que el número de clientes que llegan a una tienda por hora sigue una distribución de Poisson con un parámetro de = 10. Podemos simular el número de llegadas utilizando la transformada inversa, calculando iterativamente la FDA de Poisson hasta que exceda un número aleatorio u.

4. Problemas Prácticos y Ejercicios con Soluciones:

Problema 1: Implementar un generador de variables aleatorias de Weibull usando la transformada inversa en Python.

“‘python import numpy as np

def generar_weibull(lam, k, n): """ Genera n números aleatorios de una distribución de Weibull usando la transformada inversa.

Args: lam: Parámetro de escala (lambda). k: Parámetro de forma. n: Número de variables aleatorias a generar.

Returns: Un array de numpy con n números aleatorios de Weibull. """ u = np.random.uniform(0, 1, n) x = lam * (-np.log(u))**(1/k) return x

Ejemplo de uso

lam = 2.0 # Parámetro de escala k = 1.5 # Parámetro de forma n = 1000 # Número de variables aleatorias a generar

datos_weibull = generar_weibull(lam, k, n)

Opcional: Graficar un histograma para verificar la distribución

import matplotlib.pyplot as plt plt.hist(datos_weibull, bins=30, density=True) plt.title('Histograma de Variables Aleatorias de Weibull (Transformada Inversa)') plt.xlabel('Valor') plt.ylabel('Frecuencia') plt.show() “‘

Solución: El código genera n números aleatorios uniformes entre 0 y 1, y luego aplica la fórmula de la transformada inversa para la distribución de Weibull para obtener n variables aleatorias de Weibull.

Problema 2: Implementar un generador de variables aleatorias de Bernoulli usando la transformada inversa.

“‘python import random

def generar_bernoulli(p, n): """ Genera n números aleatorios de una distribución de Bernoulli usando la transformada inversa.

```
Args:
  p: Probabilidad de éxito (valor 1).
  n: Número de variables aleatorias a generar.

Returns:
  Una lista con n números aleatorios de Bernoulli (0 o 1).
"""
resultados = []
for _ in range(n):
    u = random.random()
    if u <= (1 - p):
        resultados.append(0)
    else:
        resultados.append(1)
return resultados
```

Ejemplo de uso

p = 0.3 # Probabilidad de éxito n = 1000 # Número de variables aleatorias a generar

datos_bernoulli = generar_bernoulli(p, n)

Imprimir algunos resultados

print(datos_bernoulli[:10])

Calcular la frecuencia de éxitos (valor 1) para verificar

frecuencia_exitos = sum(datos_bernoulli) / n print(f"Frecuencia de éxitos: {frecuencia_exitos}") "'

Solución: El código genera n números aleatorios uniformes entre 0 y 1, y luego compara cada número con (1-p). Si el número aleatorio es menor o igual que (1-p), retorna 0; de lo contrario, retorna 1.

Problema 3: Discuta las situaciones en las que la transformada inversa no sería el método más eficiente para generar variables aleatorias. ¿Qué alternativas existen?

Solución: La transformada inversa no es eficiente cuando:

- La FDA no tiene una forma analítica conocida o su inversa es difícil de calcular.
- La evaluación de la FDA (especialmente para distribuciones discretas) es computacionalmente costosa.

Alternativas:

- **Aceptación y Rechazo:** Este método es útil cuando se puede encontrar una distribución "envolvente" más simple que la distribución deseada. Lo veremos en la siguiente clase.
- **Métodos Especializados:** Algunas distribuciones tienen algoritmos de generación específicos que son más eficientes que la transformada inversa. Por ejemplo, existen algoritmos muy rápidos para generar variables aleatorias normales (Box-Muller, Marsaglia polar method).

5. Materiales Complementarios Recomendados:

- **Libro:** "Simulation Modeling and Analysis" por Averill M. Law. (Capítulos sobre generación de números aleatorios y variables aleatorias).
- **Artículo:** "The Art of Computer Programming, Volume 2: Seminumerical Algorithms" by Donald Knuth. (Sección sobre generadores de números aleatorios y pruebas estadísticas).
- **Recursos en línea:** Tutoriales sobre la transformada inversa en Khan Academy, MIT OpenCourseware. Documentación de las funciones de generación de números aleatorios en la biblioteca `numpy` de Python.