Simulación

Tarea 3: Metodo de Monte Carlo

Manuel Almuna Cofré
Renato Rivera Mohana.

16 de noviembre de 2011

1. Usando montecarlo evaluar $\int_0^1 \int_0^1 e^{x+y}\,\mathrm{d}x\mathrm{d}y$

Este ejercicio corresponde al caso 4 estudiado, para integrales múltiples, en el que se obtiene θ mediante $\hat{\theta} = E[g(u_1, u_2, ..., u_n)]$ donde $u_1, u_2, ...u_n$ sucesion de v.a.i.i.d. $U(0,1)$:

$$\hat{\theta} = \frac{\sum_{i=1}^{k} g(u_1^i), g(u_2^i), ..., g(u_n^i)}{k}$$

Para realizar el cálculo se realizó el siguiente codigo en python:

```python
from random import random
from math import sqrt

def random_point(lo, hi):
    return [lo[i] + random() * (hi[i]- lo[i]) for i in range(len(lo))]

def integrate(f, lo, hi, N):
    dim = len(lo)
    MCsum = 0L
    MCsumsq = 0L
    for i in range(N):
        point = random_point(lo,hi)
        fx = f(point)
        MCsum += fx
        MCsumsq += fx*fx
    volume = 1L
    MCsum /= N
    MCsumsq /= N
    for i in range(dim):
        volume *= (hi[i] - lo[i])
    error = volume * sqrt(abs(MCsumsq - pow(MCsum,2))/N)
    return [volume * MCsum, error]
```

La función integrate recibe como parametros una función a integrar, una lista con los limites inferiores y una con los limites superiores de todas las dimensiones, también recibe la cantidad de puntos a iterar N. Al finalizar retorna el valor calculado, junto con el error.
En este caso la función a integrar corresponde a e^{x+y} en el intervalo [0,1] para x e y:

```
def function(point):
    x=point[0]
    y=point[1]
    return pow(e,(x+y)*(x+y))

lo = [0,0] # Limites inferiores
hi = [1,1] # Limites superiores
```

y evaluamos la integral para distintos N:

```
>>> integrate(function, lo, hi, 10000) # N = 1e4
[4.895196670852154, 0.05879102980391075]

>>> integrate(function, lo, hi, 100000) # N = 1e5
[4.903027602380265, 0.018741319706710415]

>>> integrate(function, lo, hi, 1000000) # N = 1e6
[4.90064772198951, 0.005956517639587255]

>>> integrate(function, lo, hi, 10000000) # N = 1e7
[4.8989669316208, 0.0018845365325794 3]

>>> integrate(function, lo, hi, 100000000) # N = 1e8
[4.899391481167716, 0.0005960138670439928]
```

Finalmente la mayor precisión se alcanza con $N = 1 \times 10^8$, obteniendose:

$$\hat{\theta} = \int_0^1 \int_0^1 e^{x+y} \,\mathrm{d}x\mathrm{d}y = 4{,}899391481167716 \pm 0{,}0005960138670439928$$

2. Caso 6: Derive un método aproximado para calcular θ, vía el método Monte Carlo y proponga un algoritmo

$$\theta = \int_{-\infty}^{+\infty} g(x)\,\mathrm{d}x$$

Una forma para calcular este tipo de integrales es dividir la expresión en dos partes:

$$\theta = \int_{-\infty}^{+\infty} g(x)\,\mathrm{d}x = \int_{-\infty}^{0} g(x)\,\mathrm{d}x + \int_{0}^{+\infty} g(x)\,\mathrm{d}x$$

realizamos las siguientes transformaciones de variables para cada parte:

$$y = \frac{1}{1-x}\,, \quad dy = \frac{dx}{(1-x)^2} = y^2 dx \quad -\infty \leq x \leq 0$$

$$z = \frac{1}{1+x}\,, \quad dz = \frac{dx}{(x+1)^2} = -z^2 dx \quad 0 \leq x \leq \infty$$

entonces,

$$\theta = \int_0^1 \frac{g(1-1/y)}{y^2}\,\mathrm{d}y + \int_0^1 \frac{g(1/z-1)}{z^2}\,\mathrm{d}z$$

Para el calculo de la integral se propone utilizar el algoritmo realizado en la pregunta 1, junto con un complemento que ejecute las transformaciones antes de usar la función de integración. Se crea una función de transformación para cada tramo:

```
def gy(point):
    y=point[0]
    x = (1/y) - 1
    return function([x])/(y*y)


def gz(point):
    z=point[0]
    x = 1 - (1/z)
    return function([x])/(z*z)
```

gy realiza la transformación para los valores $-\infty \leq x \leq 0$ y gz para $0 \leq x \leq \infty$, ambas luego de realizar la transformación ejecutan function (función a integrar) con las nuevas coordenadas. Finalmente el valor de la integral se calcula con la función:

```
def iintegrate(N):
    a = integrate(gy, lo, hi, N)
    b = integrate(gz, lo, hi, N)
    return [a[0]+b[0], a[1]+b[1]]
```

retornando el valor con N puntos aleatorios en $([0, 1], [0, 1])$ junto a su error.

3. Usando el método para resolver el caso 6 determine el valor de θ, con 15 decimales

$$\theta = \int_{-\infty}^{+\infty} \frac{1}{1+x^2} \, \mathrm{d}x$$

Se debe calcular

$$\theta = \int_0^1 \frac{g(1-1/y)}{y^2} \, \mathrm{d}y + \int_0^1 \frac{g(1/z-1)}{z^2} \, \mathrm{d}\,, \quad g(x) = \frac{1}{1+x^2}$$

Usando el algoritmo definido en 2 definimos la función a integrar como:

```
def function(point):
    x = point[0]
    return 1/(1+x*x)
```

y utilizamos la función iintegrate:

```
>>> iintegrate(10000) # N = 1e4
[3.1457439213553453, 0.006428512822300107]

>>> iintegrate(100000) # N = 1e5
[3.140125016884107, 0.0020347821049338534]

>>> iintegrate(1000000) # N = 1e6
[3.1420605531452677, 0.0006425576529390718]
```

```
>>> iintegrate(10000000) # N = 1e7
[3.141863916995568, 0.00020337431790602427]

>>> iintegrate(100000000) # N = 1e8
[3.141591843708557, 6.430943160832697e-05]
```

El valor obtenido corresponde a:

$$\theta = 3{,}141591843708557 \pm 0{,}0000643094316083 \approx \pi$$