

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки

Лабораторна робота №5
з предмету Теорія ймовірності та математична статистика

Виконав: Радер Роман
Факультет: ІОТ
Група: ІО-02
Залікова книжка №0215

Київ - 2011

Варіант:

Розподіл f_0 - Нормальний, f_1 - Сімпсона

Критерій - Неймана-Пірсона

$$f_1 := \frac{2}{2 \cdot a} \cdot \left(1 - \frac{|m - a + m + a - 2x|}{2 \cdot a} \right) = \frac{1 - \frac{1}{2} \frac{|2m - 2x|}{a}}{a} \xrightarrow{\text{assuming real range}} - \frac{-a + |-x + m|}{a^2}$$

$$f_0 := \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\frac{(x-m)^2}{2 \cdot \sigma^2}} = \frac{1}{2} \frac{e^{-\frac{1}{2} \frac{(x-m)^2}{\sigma^2}} \sqrt{2}}{\sqrt{\pi} \sigma}$$

Критерій визначення вірності гіпотези

$$\frac{f_1}{f_0} \leq \frac{p \cdot C_1}{(1-p) \cdot C_2}$$

$$\frac{\left(1 - \frac{1}{2} \frac{|2m - 2x|}{a} \right) \sqrt{2} \sqrt{\pi} \sigma}{a e^{-\frac{1}{2} \frac{(x-m)^2}{\sigma^2}}} \leq \frac{p C_1}{(1-p) C_2}$$

$$\frac{f_1}{f_0}$$

$$\frac{\left(1 - \frac{1}{2} \frac{|2m - 2x|}{a} \right) \sqrt{2} \sqrt{\pi} \sigma}{a e^{-\frac{1}{2} \frac{(x-m)^2}{\sigma^2}}} \quad (2)$$

$\xrightarrow{\text{simplify symbolic}}$

$$- \frac{(-a + |-x + m|) \sqrt{2} \sqrt{\pi} \sigma e^{\frac{1}{2} \frac{(-x + m)^2}{\sigma^2}}}{a^2} \quad (3)$$

```

from __future__ import division
import random
from itertools import takewhile
from math import pi, sqrt, exp

count = 1000
C1 = 1
C2 = 1
p = 0.5
sigma = 5
m1 = 0
m2 = 0
a = 1

def normal_distribution(sigma, m):
    def normal_generator(sigma, m):
        while True:
            Y = sum([random.random() for x in xrange(12)]) - 6
            yield Y*sigma+m
    return normal_generator(sigma, m)

triangular = lambda x,a,m: -(-a+abs(-x+m))/a**2

def triangular_distribution(a, m):
    def triangular_generator(a, m):
        while True:
            r = random.random()
            S = 0
            e = 0.05
            R = 0.-e
            while S<r:
                R += e
                if R<m:
                    S = (1/2)*(R-(m-a))*triangular(R,a,m)
                else:
                    S = (1/2)*a*(1/a)+((1/2)*a*(1/a) - (1/2)*(m+a-R)*triangular(R,a,
m))
            yield R
    return triangular_generator(a, m)

def check(x):
    left = - ( (-a+abs(m2-x))*sqrt(2*pi)*sigma*exp( (1/2)*((-x+m1)**2/(sigma**2)) )
) / ( a**2 )
    right = (p*C1) / ((1-p) * C2)
    return left <= right

normal = normal_distribution(sigma, m1)
triang = triangular_distribution(a, m2)
norm_numbers = (normal.next() for x in xrange(count))
tri_numbers = (triang.next() for x in xrange(count))
pe1 = len(filter(lambda x: not check(x), norm_numbers))/count
pe2 = len(filter(check, tri_numbers))/count
print "Error 1: %f" % pe1
print "Error 2: %f" % pe2

```