

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

Лабораторна робота №4
«Статистичне прийняття рішень»

Виконав:
Студент групи ІВ-71
Мазан Я. В.
Залікова книжка №7109
Перевірив:
доцент Марковський О. П.

м. Київ
2018 р.

Варіант: $f_0(y) = E(\lambda_0)$, $f_1(y) = E(\lambda_1)$, прийняття рішення за критерієм Неймана-Пірсона

Теоретичні розрахунки:

За критерієм Неймана-Пірсона маємо:

$$\frac{f_1(y)}{f_0(y)} < \frac{PC_1}{(1-P)C_2}$$

Позначу умову в правій частині нерівності як k для компактності запису розрахунків:

$$k = \frac{PC_1}{(1-P)C_2}$$

$$\frac{\lambda_1 e^{\lambda_1 y}}{\lambda_0 e^{\lambda_0 y}} < k$$

$$e^{(\lambda_0 - \lambda_1)y} < k \frac{\lambda_0}{\lambda_1}; (\lambda_0 - \lambda_1)y < \ln\left(k \frac{\lambda_0}{\lambda_1}\right)$$

Маємо два випадки: 1) $\lambda_0 > \lambda_1$ та 2) $\lambda_0 < \lambda_1$. У випадку 2 маємо ділення нерівності на від'ємне число, тому її знак змінюється

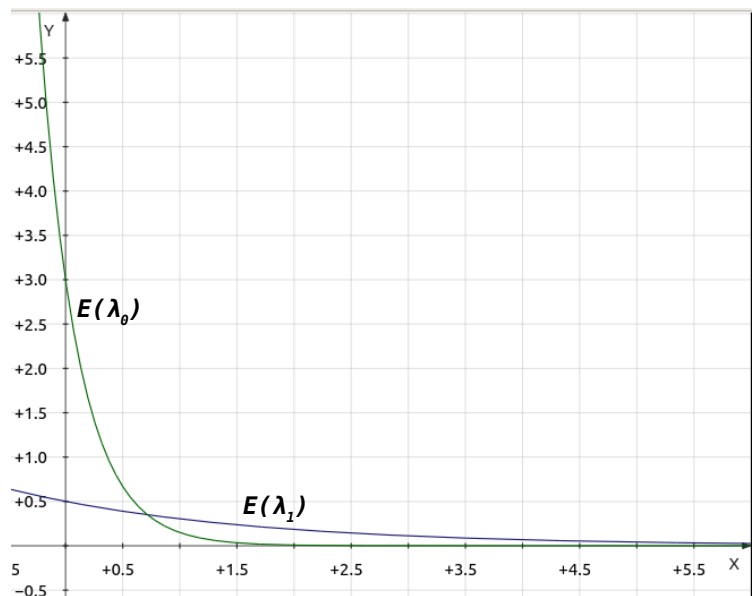
Кінцевий метод прийняття рішення:

У випадку $\lambda_0 > \lambda_1$:

Приймаємо гіпотезу $E(\lambda_0)$, якщо $y < \frac{\ln\left(k \frac{\lambda_0}{\lambda_1}\right)}{\lambda_0 - \lambda_1}$, інакше – гіпотеза $E(\lambda_1)$

У випадку $\lambda_0 < \lambda_1$:

Приймаємо гіпотезу $E(\lambda_0)$, якщо $y > \frac{\ln\left(k \frac{\lambda_0}{\lambda_1}\right)}{\lambda_0 - \lambda_1}$, інакше – гіпотеза $E(\lambda_1)$



Код програми:

```
public class Main {
    public static void main(String[] args) {
        ExpDistr zero = new ExpDistr(0.5);
        ExpDistr one = new ExpDistr(3);
        ChoiceMaking choose = new ChoiceMaking(zero, one, 0.4, 1, 1);
        choose.make_choice();
        choose.test();
    }
}

import java.util.Random;
public class ExpDistr {
    private double lambda;
```

```

private Random generator;
public ExpDistr(double lambda) {
    this.lambda = lambda;
    generator = new Random();
}
public double nextNum() {
    double r = generator.nextDouble();
    return -1.0/lambda * Math.log(1-r);
}
public double lambda() {
    return lambda;
}
public double definite_integral(double start, double end) {
    return Math.exp(-lambda*start)-Math.exp(-lambda*end);
}
}

public class ChoiceMaking {
    private ExpDistr distr_0;
    private ExpDistr distr_1;
    private enum criterias {zero_less_one, zero_greater_one};
    private double priory_probability;
    private int c_1;
    private int c_2;
    private double min_distr_0;
    private double max_distr_0;
    private double min_distr_1;
    private double max_distr_1;
    public ChoiceMaking(ExpDistr zero, ExpDistr one, double P, int C1, int C2){
        distr_0 = zero;
        distr_1 = one;
        priory_probability = P;
        c_1 = C1;
        c_2 = C2;
    }
    public void make_choice() {
        double k = priory_probability * c_1 / ((1-priory_probability) * c_2);
        criterias cases = distr_0.lambda() < distr_1.lambda() ? criterias.zero_less_one :
criterias.zero_greater_one;
        double a = Math.log(k * distr_0.lambda() / distr_1.lambda()) / (distr_0.lambda() -
distr_1.lambda());
        if (a < 0)
            a = 0;
        switch (cases) {
            case zero_less_one : {
                min_distr_0 = a;
                max_distr_0 = Double.POSITIVE_INFINITY;
                min_distr_1 = 0;
                max_distr_1 = a;
                System.out.println("Range for choosing E(λ0): y ∈ [" + a + "; +inf];");
                System.out.println("Range for choosing E(λ1): y ∈ [0; " + a + "];");
                System.out.println("Type I error probability: p1 = " +
distr_0.definite_integral(0,a));
                System.out.println("Type II error probability: p2 = " + distr_1.definite_integral(a,
Double.POSITIVE_INFINITY));
                break;
            }
            case zero_greater_one : {
                min_distr_0 = 0;
                max_distr_0 = a;
                min_distr_1 = a;
                max_distr_1 = Double.POSITIVE_INFINITY;
                System.out.println("Range for choosing E(λ0): y ∈ [0; " + a + "];");
                System.out.println("Range for choosing E(λ1): y ∈ [" + a + "; +inf];");
                System.out.println("Type I error probability: p1 = " + distr_0.definite_integral(a,
Double.POSITIVE_INFINITY));
                System.out.println("Type II error probability: p2 = " +
distr_1.definite_integral(0,a));
                break;
            }
        }
    }
    public void test() {

```

```

int numbers_num = 1000;
double[] generated_0 = new double[numbers_num];
double[] generated_1 = new double[numbers_num];
int errors_1_num = 0;
int errors_2_num = 0;
for (int i = 0; i < numbers_num; i++) {
    generated_0[i] = distr_0.nextNum();
    generated_1[i] = distr_1.nextNum();
    if ((generated_0[i] > max_distr_0) || (generated_0[i] < min_distr_0))
        errors_1_num++;
    if (generated_1[i] > max_distr_1 || generated_1[i] < min_distr_1)
        errors_2_num++;
}
System.out.println("Experimentally defined error type I probability: p1 = " +
(double)errors_1_num/numbers_num);
System.out.println("Experimentally defined error type II probability: p2 = " +
(double)errors_2_num/numbers_num);
}
}

```

Тестування програми для $\lambda_0 = 3$, $\lambda_1 = 0.5$, $P = 0.5$, $C_1 = C_2 = 1$:

```

Range for choosing E( $\lambda_0$ ): y ∈ [0; 0.716703787691222];
Range for choosing E( $\lambda_1$ ): y ∈ [0.716703787691222; +inf];
Type I error probability: p1 = 0.11647118646192986
Type II error probability: p2 = 0.3011728812284208
Experimentally defined error type I probability: p1 = 0.111
Experimentally defined error type II probability: p2 = 0.277

Process finished with exit code 0

```