Kpi-best

Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота № 6

з теорії ймовірностей:

Виконали: студенти 2 курсу

ФІОТ, гр. ІО - 02

Агеєнко Юлія

та студент гр. ІО – 01

Ткаченко Ігор

Перевірив: викладач

Марковський О.П.

Київ

2011 рік

**Завдання**



Лістинг програми:

**package** Lab6;

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** java.util.Arrays;

**import** org.apache.commons.math.linear.Array2DRowRealMatrix;

**import** org.apache.commons.math.linear.RealMatrix;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**double**[][] switchTable = **null**;

**int** dim;

**int** startVertex = 0;

**int** cycleCount = 0;

**try**{

BufferedReader stream = **new** BufferedReader(**new** InputStreamReader(System.*in*));

System.*out*.println("Enter vertex count");

dim = Integer.*parseInt*(stream.readLine());

switchTable = **new** **double**[dim][dim];

**for**(**int** i = 0; i<switchTable.length; i++){

**for**(**int** j = 0; j<switchTable.length; j++){

**if**(i != j){

System.*out*.println("Enter lambda "+(i+1)+" to "+(j+1));

switchTable[i][j] = Double.*parseDouble*(stream.readLine());

}

}

}

System.*out*.println("Enter start vertex");

startVertex = Integer.*parseInt*(stream.readLine())-1;

//System.out.println("Enter cycles count");

//cycleCount = Integer.parseInt(stream.readLine());

cycleCount = 100000;

}**catch**(Exception e){

e.printStackTrace();

}

*printStationaryP*(switchTable);

*simulate*(switchTable, startVertex, cycleCount);

}

**private** **static** **void** printStationaryP(**double**[][] switchTable) {

/\*double p;

for(int i = 0; i<switchTable.length; i++){

p = 0;

for(int j = 0; j<switchTable.length; j++){

p+= switchTable[j][i];

}

System.out.println("P"+(i+1)+" = "+p/switchTable.length);

}\*/

**double**[][] array1 = **new** **double**[switchTable.length][switchTable.length];

**double**[][] array2 = **new** **double**[switchTable.length][1];

array2[0][0] = 1;

**for**(**int** i=0 ;i<switchTable.length; i++){

array1[0][i] = 1;

}

**for**(**int** i=1; i<switchTable.length; i++){

**for**(**int** j=0; j<switchTable.length; j++){

**if**(i == j){

**for**(**int** k = 0; k<switchTable.length; k++){

**if**(k != j){

array1[i][j] -= switchTable[j][k];

}

}

}**else**{

array1[i][j] = switchTable[j][i];

}

}

}

//System.out.println(Arrays.deepToString(switchTable));

//System.out.println(Arrays.deepToString(array1));

//System.out.println(Arrays.deepToString(array2));

RealMatrix mat1 = **new** Array2DRowRealMatrix(array1);

RealMatrix mat2 = **new** Array2DRowRealMatrix(array2);

mat1 = mat1.~~inverse~~();

**double**[][] result = mat1.multiply(mat2).getData();

**for**(**int** i = 0; i<result.length; i++){

System.*out*.println("P"+(i+1)+" = "+result[i][0]);

}

}

**private** **static** **void** simulate(**double**[][] switchTable, **int** vertex, **int** cycleCount) {

**double**[] vertexProbability = **new** **double**[switchTable.length];

System.*out*.println("Starting from vertex "+(vertex+1));

**for**(**int** i = 0; i<cycleCount; i++){

vertex = *decide*(switchTable[vertex], vertex, vertexProbability);

//System.out.println("Switched to "+(vertex+1));

}

**double** S = 0;

**for**(**int** i = 0; i<vertexProbability.length; i++){

S += vertexProbability[i];

//System.out.println(vertexProbability[i]);

}

**for**(**int** i = 0; i<vertexProbability.length; i++){

System.*out*.println("Have been in "+(i+1)+" vertex for "+(vertexProbability[i]/S)+" percents of time");

}

}

**private** **static** **int** decide(**double**[] switchTable, **int** currentVertex, **double**[] timeTable){

**double**[] time = **new** **double**[switchTable.length];

**for**(**int** i = 0; i<time.length; i++){

time[i] = -Math.*log*(Math.*random*())/switchTable[i];

//System.out.println(time[i]);

}

**int** result = *findMin*(time);

timeTable[currentVertex] += time[result];

**return** result;

}

**private** **static** **int** findMin(**double**[] time){

**int** result = 0;

**for**(**int** i = 0; i < time.length; i++){

**if**(time[i]<time[result]){

result = i;

}

}

**return** result;

}

}

**Результати роботи програми:**

Enter vertex count

5

Enter lambda 1 to 2

1

Enter lambda 1 to 3

0

Enter lambda 1 to 4

0

Enter lambda 1 to 5

0

Enter lambda 2 to 1

0

Enter lambda 2 to 3

2

Enter lambda 2 to 4

4

Enter lambda 2 to 5

0

Enter lambda 3 to 1

3

Enter lambda 3 to 2

0

Enter lambda 3 to 4

0

Enter lambda 3 to 5

5

Enter lambda 4 to 1

0

Enter lambda 4 to 2

0

Enter lambda 4 to 3

0

Enter lambda 4 to 5

7

Enter lambda 5 to 1

0

Enter lambda 5 to 2

0

Enter lambda 5 to 3

6

Enter lambda 5 to 4

0

Enter start vertex

3

P1 = 0.5040000000000001

P2 = 0.084

P3 = 0.16799999999999998

P4 = 0.04800000000000001

P5 = 0.19599999999999998

Starting from vertex 3

Have been in 1 vertex for 0.5047474435122025 percents of time

Have been in 2 vertex for 0.08500745681937784 percents of time

Have been in 3 vertex for 0.16647713743938466 percents of time

Have been in 4 vertex for 0.04874116346661113 percents of time

Have been in 5 vertex for 0.19502679876242382 percents of time