Національний Технічний Університет України

«Київський Політехнічний Інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

Процеси Маркова

|  |  |
| --- | --- |
| Прийняв  Доц. Марковський О.П.  «\_\_»\_\_\_\_\_\_\_\_ 2014 р. | Виконала Студентка 2-ого курсу ФІОТ  групи ІО-32  Руденко Т.А. |

Марківський неперервний, 3 стани

Лістинг

**import** java.util.Arrays;

**import** org.apache.commons.math3.linear.Array2DRowRealMatrix;

**import** org.apache.commons.math3.linear.ArrayRealVector;

**import** org.apache.commons.math3.linear.LUDecomposition;

**import** org.apache.commons.math3.linear.RealMatrix;

**import** org.apache.commons.math3.linear.RealVector;

**import** org.apache.commons.math3.linear.DecompositionSolver;

**public** **class** Lab5 {

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

**double**[][] paths = {{0, 2, 1},{0, 0, 0.5},{0.25, 0, 0}};

**double**[][] coeffM = **new** **double**[3][3];

**for**(**int** i = 0; i < 3; i++)

coeffM[0][i] = 1;

**for**(**int** i = 1; i < 3; i++){

**for**(**int** j = 0; j < 3; j++){

**if**(j == i-1)

**for**(**int** k = 0; k < 3; k++)

coeffM[i][j] -= paths[j][k];

**else** coeffM[i][j] = paths[j][i-1];

}

}

RealMatrix coeff = **new** Array2DRowRealMatrix(coeffM);

DecompositionSolver solver = **new** LUDecomposition(coeff).getSolver();

RealVector constants = **new** ArrayRealVector(**new** **double**[] { 1, 0, 0 }, **false**);

RealVector solution = solver.solve(constants);

**double**[] probs = solution.toArray();

System.*out*.println("Static probabilities(theoretical):");

**for**(**int** i = 0; i < 3; i++)

System.*out*.print("P" + (i+1) + " = " + probs[i] + " ");

System.*out*.println();

**double** total = 0;

**double**[] totalV = **new** **double**[3];

**int** jumpTo = 0;

**int** prevJump = 0;

**double**[] t = **new** **double**[3];

**for**(**int** i = 0; i < 1000; i++){

total += t[jumpTo];

totalV[prevJump] += t[jumpTo];

prevJump = jumpTo;

**for**(**int** j = 0; j < 3; j++)

**if**(paths[jumpTo][j] != 0)

t[j] = (-1/paths[jumpTo][j])\*Math.*log*(Math.*random*());

**else** t[j] = 100000;

jumpTo = 0;

**for**(**int** j = 1; j < 3; j++){

**if** (t[j] < t[j-1])

jumpTo = j;

}

}

**for**(**int** i = 0; i < 3; i++){

probs[i] = totalV[i]/total;

}

System.*out*.println("Static probabilities(practical):");

**for**(**int** i = 0; i < 3; i++)

System.*out*.print("P" + (i+1) + " = " + probs[i] + " ");

}

}