**Date: 6.03.17**

**SUPROTIK DEY**

**IT Hx-31**

**4TH SEM, 510815050**

**Write a program in C/C++/Java to simulate a discrete Markov chain given with transition probabilities.**

**SOURCE CODE:**

/\*Simulation of rolling of a biased dice a discrete Markov chain given with transition probabilities\*/

#include<stdio.h>

#define max 1000

#define numOfStates 6

int main()

{ float transitionalMatrix[numOfStates + 1][numOfStates] =

{0.16,0.20,0.21,0.15,0.11,0.17, 0.15,0.19,0.14,0.16,0.24,0.12, 0.23,0.14,0.17,0.16,0.08,0.22,

0.20,0.24,0.03,0.16,0.23,0.14, 0.11,0.16,0.21,0.24,0.13,0.15, 0.21,0.17,0.24,0.14,0.11,0.13,

0.18,0.12,0.11,0.26,0.14,0.19 };//rows will have prev state,cols will have next state..

int predictingInstance, history[max], iter, predictedState,iter2;

float predictMult, currStatePredict[max];

for(iter=0;iter<numOfStates + 1;iter++)

{ predictMult=0.0;//used as temp

for(iter2=0;iter2<numOfStates;iter2++)

predictMult=predictMult+transitionalMatrix[iter][iter2];

if(predictMult!=1.0)

{ printf("\nError in transitional matrix row:%d Sum=%f",iter+1,predictMult);

break; } }

printf("\nSimulating rolling of a biased dice:\n");

printf("\nEnter the instance you want to predict: ");

scanf("%d",&predictingInstance);

printf("\nEnter the history:\n");

for( iter = 0; iter < (predictingInstance - 1); iter++ )

{scanf("%d", &history[iter]); }

predictMult = 1.0; //prediction...

for( iter = 0; iter < (predictingInstance - 1); iter++ )

{if( iter == 0)

predictMult = predictMult \* (transitionalMatrix[0 ][ history[iter] - 1 ]) ;

else

predictMult = predictMult \* (transitionalMatrix[ history[iter - 1] ][ history[iter] - 1 ]) ; }

printf("\nProbabilities of states:\n");

for( iter = 0; iter < numOfStates; iter++ )

{ currStatePredict[iter] = predictMult \* (transitionalMatrix[history[predictingInstance - 2 ] ][ iter ]);

printf("\n%d -> %f",(iter+1),currStatePredict[iter]); }

predictedState = 1; //state prediction...

for( iter = 0; iter < numOfStates; iter++ )

{ if(currStatePredict[iter] > currStatePredict[predictedState - 1])

predictedState = iter + 1; }

printf("\nPredicted output is: %d.\n", predictedState);

return 0;}

**OUTPUT:**

Simulating rolling of a biased dice:

Enter the instance you want to predict: 5

Enter the history:

4 5 5 2

Probabilities of states:

1 -> 0.000084

2 -> 0.000051

3 -> 0.000062

4 -> 0.000058

5 -> 0.000029

6 -> 0.000080

Predicted output is: 1.