# HMM DICE

## Alphabet

**double**[] **initialPick**

Initial probabilities array **initialPick** such that **initialPick**[i] stores the probability that state die-i is picked at the beginning.

**double**[][] **transitionProb**

Transition matrix **transitionProb** such that **transitionProb**[i] [j] stores the transition probability of transiting from state die-i to state die-j.

**double[][] emissionProb**

Emission matrix **emissionProb** such that **emissionProb**[i] [k] stores the probability of observing label-k from state die-i.

**int**[] **ObservationSeq**

A sequence of observations such that **ObservationSeq**[t] stores the label that is observed at time t.

**int**[] **dieSeq**

A sequence of dice such that **dieSeq**[t] stores the die that is used at time t.

**double**[][] **PathProbability**

Path probabilities matrix such that **PathProbability**[t] [i] stores the probability of the path ends with state die-i at time t.

**int**[][] **Parent**

Parent Nodes matrix such that **Parent**[t] [i] stores which is the state in time t-1 when state in time t is state die-i.

## Viterbi Algorithm

1. *Initialize*

//t = 0

For each state die-i, find the probability that state i shows **ObservationSeq**[0], and store the probability in **PathProbability**[0] [i].

**PathProbability**[0] [i] = **emissionProb**[i] [**ObservationSeq**[0]] **\* initialPick**[i]

1. *Iteration*

For t from 1 to n (n is the length of the **ObservationSeq**):

For each current state die-**j** at time t:

For each parent state die-**i** at time t-1:

**PathProbability[t] [j]** = Max

(PathProbability[t-1][i] \* transitionProb[i][j] \* emissionProb[j][ObservationSeq[t]])

**Parent[t] [j]** = the **i** that let **PathProbability[t] [j]** become max

1. *Find the optimal path*

**dieSeq**[n] (n is the length of the **ObservationSeq**) = the i that let **PathProbability[n] [i]** become max.

**dieSeq**[t-1] = **Parent**[t] [**dieSeq**[t]]

## How to run the code

**HMM\_DICE. Java**

1. **public** **static** **void** main (String[] args) {
2. HMM\_DICE hmm\_dice = **new** HMM\_DICE();
3. String filename = "InputFiles/InputFile8.txt";
4. System.out.println(filename);
5. fileReader.readFileByLines(filename,hmm\_dice);
6. hmm\_dice.evaluate(hmm\_dice.getObservationSeq());
7. }

Input the filename in:

String filename = "InputFiles/InputFile8.txt";

## Result

**For InputFile8.txt, the result is: (dice D1 D2 D3)**

InputFiles/InputFile8.txt

sequence probability: 1.1838001748121298E-46

Most possible dies sequence:

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1

D1 D1 D1 D1 D1 D1 D1 D1 D1 D1