

### Démonstration 5 :

Date : le 14 Février 2003

Ratiba TERBAOUI

### Problème 1 : Chaîne de Markov

**États :**

1. Pluvieux (P)
2. Nuageux (N)
3. Ensoleillé(E)

**Matrice des probabilités de transition d'états : A**

Etats	P	N	E
P	0.4	0.3	0.3
N	0.2	0.6	0.2
E	0.1	0.1	0.8

**Question :**

Calculer la probabilité d'observer une séquence EEPPENE sachant qu'aujourd'hui c'est ensoleillé donc état E

Règle de probabilité de base :  $P(A,B) = P(A|B) * P(B)$

La règle de Markov

$$P(q_1, q_2, \dots, q_T) =$$

$$\begin{aligned} &= P(q_T | q_1, q_2, \dots, q_{T-1}) P(q_1, q_2, \dots, q_{T-1}) \\ &= P(q_T | q_{T-1}) P(q_1, q_2, \dots, q_{T-1}) \\ &= P(q_T | q_{T-1}) P(q_{T-1} | q_{T-2}) P(q_1, q_2, \dots, q_{T-2}) \\ &= P(q_T | q_{T-1}) P(q_{T-1} | q_{T-2}) \dots P(q_2 | q_1) P(q_1) \end{aligned}$$

Soit l'observation O :

$$O = (E, E, E, P, P, E, N, E)$$

En utilisant la règle on obtient :

$$\begin{aligned} P(O|\text{modèle}) &= P(E, E, E, P, P, E, N, E|\text{modèle}) \\ &= P(E) P(E|E) P(E|E) P(P|E) P(P|P) P(E|P) P(N|E) P(E|N) \\ &= ?_3 A_{33} A_{33} A_{31} A_{11} A_{13} A_{32} A_{23} \\ &= (1) (0.8) (0.8) (0.1) (0.4) (0.3) (0.1) (0.2) \\ &= 1.536 * 10^{-4} \end{aligned}$$

Le programme correspondant à ce modèle :

```
/*
 * Exemple de chaîne de Markov
 * Auteur e: Ratiba TERBAOUI
 * Date : 12 Février 2003
 */

import java.*;
import java.lang.*;
import java.util.*;

public class Prevision {

    public static int etatCourant = 1;

    public static void main(String args[] ) {

        double [][] MatriceTransition = initMatTransition();
        String unePrevision = generePrevision();
        // int[] unEtat= deSequenceEtat(unePrevision);
        int[] unEtat = {3,3,3,1,1,3,2,3};

        double probaSequence = probaSequence(unEtat, MatriceTransition) ;

        System.out.println("Voici La sequence      : " +unePrevision);
        System.out.print("Voici la sequence en etat : ");
        for (int i=0; i<unEtat.length; i++) {
            System.out.print(unEtat[i]);
        }
        System.out.print("\n");
        System.out.println("La probabilite finale est : "+probaSequence);
    }
}
```

```

public static double probaSequence(int[] s, double[][] mat) {
    double pi=0.0;
    double p = 1.0;
    double pinitiale = 1.0;
    System.out.println(pinitiale);
    for (int i=0;i<s.length-1;i++) {
        pi = pinitiale*mat[s[i]-1][s[i+1]-1];
        System.out.println(pi);
        p = p*pi;
    }
    return p;
}

```

```

public static int[] deSequenceEtat(String s) {
    char lettre = ' ';
    int etatS = 0;
    int [] etat = new int[s.length()];
    for (int i=0; i<s.length();i++) {
        lettre = s.charAt(i);
        switch(lettre) {
            case 'S':
                etatS = 3;
                break;
            case 'C':
                etatS = 2;
                break ;
            case 'R':
                etatS = 1;
                break;
            default:
                break;
        }
        etat[i] = etatS;
    }
    return etat;
}

```

```

public static String generePrevision() {
    int x = 0;
    StringBuffer observation = new StringBuffer();
    do {
        x = (int) ((Math.random()*20)+65);
        if ((x==82)||(x==83)||(x==67)) {
            char Obs = (char)(x);
            observation.append(Obs);
        }
    } while (observation.length() < 8);

    return observation.toString();
}

```

```

public static double[][] initMatTransition() {
    double[][] MatriceTransition = {{0.4,0.3,0.3} , {0.2,0.6,0.2}, {0.1,0.1,0.8}};

    for (int i=0; i<MatriceTransition.length;i++) {
        System.out.print("La ligne " +i+ ": ");
        for (int j=0; j<MatriceTransition.length;j++) {
            System.out.print(MatriceTransition[i][j] + ", ");
        }
        System.out.println();
    }
    return MatriceTransition;
}
}

```

### Résultat de l'exécution pour l'observation (E,E,E,P,P,E,N,E)

```

La ligne 0: 0.4, 0.3, 0.3,
La ligne 1: 0.2, 0.6, 0.2,
La ligne 2: 0.1, 0.1, 0.8,
1.0
0.8
0.8
0.1
0.4
0.3
0.1
0.2
Voici la sequence en etat : 33311323
La probabilite finale est : 1.5360000000000005E-4

```

### Modification du Modèle de Markov en chaîne de Markov Caché :

Si je veux savoir dans chaque état, l'intensité du temps autrement dit, quand c'est ensoleillé est-ce c'était très ensoleillé ou peu ensoleillé, j'attribue à l'état ensoleillé E deux sous états E+ pour dire très ensoleillé et E- si pas très ensoleillé.

Donc à chaque fois que je serais dans un état ensoleillé, je ne saurais pas de quel état précisément il provient l'état.

## Problème 2 : Chaîne de Markov Caché

Soit une observation  $O = (o_1, o_2, o_3, \dots, o_T)$  et le modèle ? trouver une séquence d'états optimale  $q = (q_1, q_2, \dots, q_T)$

**Solution:** Trouver le chemin  $(q_1, q_2, \dots, q_T)$  qui maximise la probabilité  $P(q_1, q_2, \dots, q_T | O, ?)$

**Exemple :**

	Etat1	Etat2	Etat3
P(P)	0.5	0.75	0.25
P(F)	0.5	0.25	0.75

Les probabilités de la matrice de transitions entre états est de  $1/3$  (A)

Probabilités de l'état initial est de  $1/3$  ( $?_i$  avec  $1 \leq i \leq N$ )

1. On observe  $O = (P, P, P, P, F, P, F, F)$ . Quel est l'état de séquence la plus probable?

Solution avec l'algorithme de Viterbi

**Initialisation :**

$$\begin{aligned} \pi_1(i) &= \pi_i b_i(o_1), \quad 1 \leq i \leq N \\ \pi_1(i) &= 0 \end{aligned}$$

**Réurrence :**

$$\begin{aligned} \pi_t(j) &= \max_{1 \leq i \leq N} [\pi_{t-1}(i) a_{ij}] b_j(o_t), \\ \pi_t(j) &= \arg \max_{1 \leq i \leq N} [\pi_{t-1}(i) a_{ij}] \\ & \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \end{aligned}$$

**Terminaison :**

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} [\pi_T(i)] \\ q_T^* &= \arg \max_{1 \leq i \leq N} [\pi_T(i)] \end{aligned}$$

**Chemin (état de séquence) chaînage arrière :**

$$q_t^* = \pi_t(j) (q_{t+1}^*), \quad t=T-1, T-2, \dots, 1$$

### Étapes de calcul :

#### Initialisations :

$$\gamma_1(1) = \gamma_1 b_1(P) = 1/3 * 1/2$$

$$\gamma_1(2) = \gamma_2 b_2(P) = 1/3 * 3/4$$

$$\gamma_1(3) = \gamma_3 b_3(P) = 1/3 * 1/4$$

$$\gamma_1(1) = \gamma_1(2) = \gamma_1(3) = 0$$

$$\max = \gamma_1(2) = \gamma_2 b_2(P) = 1/3 * 3/4$$

#### Réurrences :

$$\gamma_2(1) = 1/3 * 3/4 * 1/3 * 1/2$$

$$\gamma_2(2) = 1/3 * 3/4 * 1/3 * 3/4$$

$$\gamma_2(3) = 1/3 * 3/4 * 1/3 * 1/4$$

$$\max = \gamma_2(2) = 1/3 * 3/4 * 1/3 * 3/4$$

etc...

Donc la séquence d'état  $Q = (\text{Etat2}, \text{Etat2}, \text{Etat2}, \text{Etat2}, \text{Etat3}, \text{Etat2}, \text{Etat3}, \text{Etat3})$  maximise la probabilité d'avoir la séquence  $O = (P, P, P, P, F, P, F, F)$

**Question2 :** Quelle est la probabilité que l'observation vienne complètement de l'état1 ?

**Question3 :** Soit l'observation  $O' = (P, F, F, P, F, P, P, F, F, P)$ , quelle sera la séquence d'états la plus probable ?

En reprenant les calculs avec Viterbi, la réponse sera :  $Q' = (\text{Etat1}, \text{Etat3}, \text{Etat3}, \text{Etat2}, \text{Etat3}, \text{Etat2}, \text{Etat2}, \text{Etat3}, \text{Etat3}, \text{Etat2})$

**Question4 :** Si la matrice de transitions entre les états devient  $A'$  comme suit :

	Etat1	Etat2	Etat3
Etat1	0.9	0.45	0.45
Etat2	0.05	0.1	0.45
Etat3	0.05	0.45	0.1

De quelle façon, ce nouveau modèle affectera la recherche du chemin de Viterbi pour chacune des Observations citées ci-dessus  $O$  et  $O'$  ?