

INFMDI 341

TP "Modèles de Markov Cachés"

Laurence Likforman-Sulem, Marc Sigelle, Chloé Clavel, Alexandre Gramfort

Telecom ParisTech

Juin 2015

1. Introduction

L'objectif de ce TP est de

- générer des séquences d'observations suivant un modèle de Markov Caché donné
- de calculer la vraisemblance d'une séquence d'observations suivant un modèle de Markov Caché donné

Le modèle de Markov est de type discret. Les classes de caractères (classes 0, 1, 7) sont modélisées chacune par un modèle à $Q=5$ états de type gauche-droite. Les états 1 et 5 correspondent à des colonnes de pixels de type fond de l'image (niveau 0). Les états 2, 3 et 4 correspondent au début, milieu et fin du caractère respectivement. Les transitions entre états sont indiquées dans la matrice de transitions A de taille $Q \times Q$. Les vecteurs π sont tous égaux à $\pi=(1 \ 0 \ 0 \ 0 \ 0)$. Les séquences d'états commencent donc toujours par l'état $q_1=1$.

Les séquences d'observations sont discrètes et issues d'images de chiffres de la base MNIST. Les séquences d'observations consistent en séquences d'index (symboles) des éléments du dictionnaire. Ce dictionnaire est stocké sous forme matricielle (matrice v) dans le fichier `matrice_symboles`. L'élément numéro i d'une séquence d'observations correspond au symbole i et donc à la colonne i de la matrice v . Un symbole correspond à une configuration de colonne de 5 pixels (binaires : noir/blanc). Il y a $2^5=32$ configurations, et donc symboles possibles.

Une séquence d'observations correspondant à l'image simplifiée de la fig. 1 est :

[1 1 1 1 1 14 23 23 27 18 18 18 12 12 12 12 12 12 23 23 23 14 4 1 1 1]

La concaténation des éléments du dictionnaire correspondant aux index de la séquence d'observations peut être visualisée sous forme d'image en remplaçant chaque index par le vecteur de pixels correspondant dans le dictionnaire (Fig. 2).

Les probabilités des observations dans chaque état sont indiquées dans la matrice B (32 lignes, 5 colonnes).

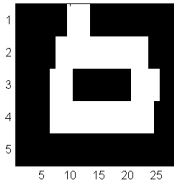
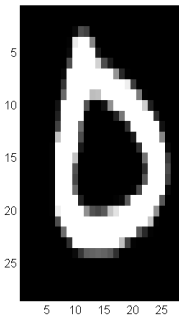


Fig. 1 : (gauche) image de chiffre (base MNIST, taille 28x28) (droite) image simplifiée (taille 5x28).

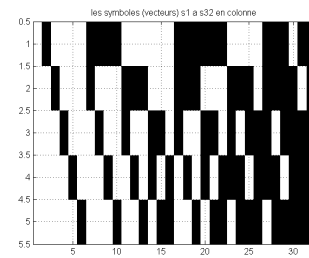


Fig. 2 : Chaque colonne représente un symbole du dictionnaire (32 symboles en tout).

2. Génération de séquences d'observations

Les matrices A0.txt, B0.txt, vect_p0.txt contiennent les matrices A, B, et le vecteur π correspondant au modèle de Markov caché du chiffre 0. Le fichier matrice_symboles.txt contient le dictionnaire. Charger ces fichiers depuis la page :

http://perso.telecom-paristech.fr/~lauli/TPHMM_MDI343/

Le dictionnaire de symboles se trouve dans la matrice v qui se charge en utilisant :

```
import numpy as np
filename='matrice_symboles.txt'
v=np.loadtxt(filename)
```

2.1 A quoi correspondent les zéros de la matrice B ? et ceux de la matrice A et du vecteur π ?

2.2 Ecrire une fonction *etat_suivant* qui génère un état q_{t+1} (à $t+1$) à partir de l'état courant q_t (à t) à l'aide de la matrice de transitions et de la fonction de répartition *cumsum*.

Afficher la fonction de répartition pour une ligne de la matrice de transition et expliquer son rôle pour la génération de l'état à $t+1$.

2.3 Générer une séquence d'observations suivant le modèle de Markov Caché du chiffre 0. On commencera par générer une séquence d'états suivant ce modèle à l'aide la fonction *etat_suivant*. Puis on générera la séquence d'observations par le même procédé.

2.4 Visualiser le résultat sous forme d'image. Générer des séquences pour le chiffre 7 et le chiffre 1 (matrices B1.txt, B7.txt, etc...)

```
import matplotlib.pyplot as plt
```

```

im=[]
# les x contiennent les index dans le dictionnaire
for t in range (0,len(stateSeq)) :
    im_col=v[:, x[t]-1]
    im.append(im_col)

im = np.array(im).T # now make an array
plt.imshow(im*255)

```

3. Calcul de la vraisemblance de séquences d'observations

Les fichiers SeqTest0.txt, SeqTest1.txt, SeqTest7.txt contiennent chacun 10 séquences d'observations de chiffres des 3 classes 0, 1 et 7, disposés en ligne. Le script suivant extrait la 5^{ème} observation de la 3^{ème} séquence des chiffres 0.

```

filename='SeqTest0.txt'
TestChiffres=np.loadtxt(filename)
nex=2
seq= TestChiffres[nex, : ]
seq[4]

```

3.1 Calculer la vraisemblance de ces séquences suivant chacun des modèles (0, 1 et 7) par l'algorithme de Viterbi (on implémentera la version logarithmique de cet algorithme). Pour cela les matrices A, B et π seront converties en logarithmes (utiliser np.log).

3.2 Donner le résultat de la classification des images de test en considérant un problème à trois classes : 0, 1 et 7.