

MDI 343
Apprentissage Statistique
TP "Modèles de Markov Cachés"

Laurence Likforman-Sulem, Marc Sigelle, Aurélien Garivier

Telecom ParisTech

Avril 2012

1. Préliminaires

1.1 Quelques Rappels

- pour vous mettre dans le répertoire de travail `mon_repertoire`, taper `setwd("mon_repertoire")`. Pour lancer un script : `source("mon_script.R")`
- pour afficher les variables utilisées, taper `ls()`.
- pour effacer les variables taper `rm(list=ls)`.
- les commentaires sont précédés de #
- $x \leftarrow c(1, 2, 5)$ crée le vecteur $x=(1, 2, 5)$.
- `length(x)` donne le nombre de composantes du vecteur x
- utiliser les flèches pour rappeler des commandes
- `help(commande)` affiche la documentation correspondant à une commande donnée.

1.2 Liste des opérations ou fonctions dont vous pouvez avoir besoin

- $s \leftarrow \text{matrix}(0, \text{nrow}=5, \text{ncol}=28)$ crée une matrice de zéros de 5 lignes et 28 colonnes
- $t(m)$ est la transposée de la matrice m
- $s \leftarrow \text{array}(0, 10)$ crée un vecteur s de dix zéros
- $\text{perm} \leftarrow \text{sort}(\text{Delta}, \text{index.return}=\text{TRUE})$ trie les valeurs du vecteur Delta de la plus petite à la plus grande. La commande `sort` retourne dans le vecteur `perm$x` les valeurs triées et dans le vecteur `perm$ix` les indices correspondants du vecteur Delta .
- $u \leftarrow \text{runif}(1)$ retourne dans la variable u un nombre aléatoire suivant la loi uniforme entre 0 et 1.
- `cumsum(x)` : calcule la somme cumulée des éléments du vecteur x
- $A \leftarrow \text{as.matrix}(\text{read.table}('A1.txt'))$ charge dans la matrice A les valeurs du fichier `A1.txt`

1.3 Fonctions d'affichage

- `xII()` ouvre une nouvelle fenêtre d'affichage
- `image(c(1:28),c(1:5),t(im))` affiche sous forme d'image la matrice transposée de la matrice `im`. l'échelle en x va de 1 à 28 et l'échelle en y de 1 à 5

2. 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

Les séquences d'observations sont discrètes et issues d'images de chiffres de la base MNIST.

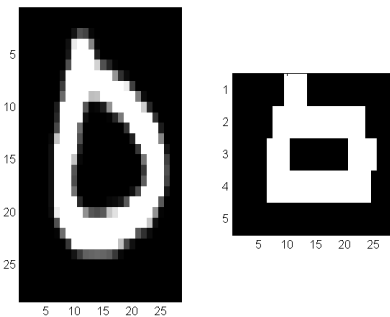


Fig. 1 : (gauche) image de chiffre (base MNIST) (droite) séquence d'observations représentant cette image. Les observations sont issues du dictionnaire de symboles.

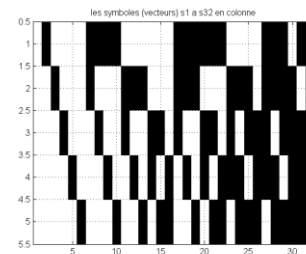


Fig. 2 : Chaque colonne représente un symbole du dictionnaire (32 symboles en tout).

3. 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 dictionnaire de 32 symboles se trouve dans le fichier `matrice_symboles.txt` (1 élément du dictionnaire par colonne).

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

3.2 écrire une fonction *etat_suivant* qui génère l'état suivant j à partir de l'état (précédent) i . L'argument de cette fonction sera la ligne de la matrice de transition $A[i, \cdot]$. On pourra utiliser la fonction *cumsum*.

3.3 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, puis on générera la séquence d'observations par le même procédé.

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

4. 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. La n ème séquence correspond à la n ème ligne du fichier.

Le script suivant extrait la 5^{ème} observation de la 3^{ème} séquence des chiffres 0 :

```
seq ← as.matrix(read.table('SeqTest0.txt'))
s ← 3
t ← 5
seq[s,t]
```

4.1 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 de la façon suivante. Pour la matrice A :

```
A <- log(A) ;
```

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