HES-SO
Engineering and Architecture
Faculty
MSE

Enseignant: Stefano Carrino
Enseignant: Antonio Ridi

Multimodal Processing, Recognition and Interaction

Série 3 du 13 avril 2015
Reconnaissance de la parole – HMMs

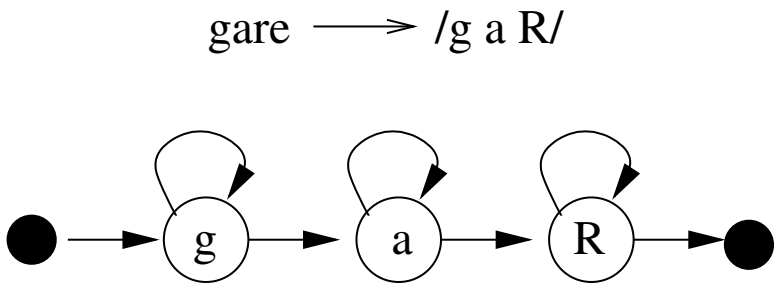

FIGURE 1 – Exemple d'un HMM pour le mot *gare*. Le mot est traduit en sa séquence de phonèmes et chaque phonème est associé à un état du HMM. Le premier et dernier état du HMM (ronds en noir) sont des états "non-émetteurs", càd qu'aucune probabilité d'émission ne leur est associée. La séquence de phonèmes est précédée et suivie par un état associé au silence.

Contexte

Nous allons illustrer les cours sur les HMM par un petit exemple de construction de modèles pour les chiffres de 1 à 5 (avec trois exemples d'entraînement par chiffre). Pour ce faire, nous vous donnons un ensemble de scripts en Matlab que vous allez utiliser pour entraîner et tester les modèles. Dans ces scrips, les HMMs ont été implémentés de la façon suivante (les notations correspondent aux noms des variables) :

- Le nombre d'états N est défini comme le nombre de phonèmes dans chaque chiffre plus les deux états de silence plus les deux états non-émetteurs de début et de fin (à déterminer).
- Le modèle de Markov caché est du type strictement gauche droite avec $a_{ij} \neq 0$ pour $j = i$ et $j = i + 1$, comme illustré sur la figure ci-contre.
- Les probabilités d'émission sont calculées par des fonctions de densité de probabilités modélisée par une Gaussienne de dimension P :

$$b_j[\mathbf{o}(t)] = \mathcal{N}(\mathbf{o}(t); \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{\sqrt{(2\pi)^P |\boldsymbol{\Sigma}_j|}} e^{-\frac{1}{2}(\mathbf{o}(t)-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{o}(t)-\boldsymbol{\mu}_j)}, \quad (1)$$

où $\boldsymbol{\mu}_j$ représente la moyenne, $\boldsymbol{\Sigma}_j$ est la matrice de covariance (ici forcée à être diagonale pour simplifier les calculs) et $|\boldsymbol{\Sigma}_j|$ dénote son déterminant.
Un modèle HMM est défini par trois matrices :

a) **A** (N x N) la matrice des transitions entre les états
b) **MI** (P x N) la matrice des moyennes de tous les états pour les probabilités d'émission. La première et la dernière colonne de cette matrice sont vides vu que le premier et le dernier états ne sont pas émetteurs.
c) **SIGMA** (P x N) la matrice des écarts types (standard deviation) de tous les états pour les probabilités d'emission.

Etapes

a) **Lecture des scripts :**
Le script *train_test_hmm.m* est le script principal qui appelle les autres procédures Matlab. Lisez ce script ainsi que les autres procédures et essayez d'en comprendre le fonctionnnement. Les notations des différentes variables sont les mêmes que celles décrites ci-dessus. Dans les différents points ci-dessous, il vous est demandé de faire de légère modifications des scripts afin d'afficher certaines valeurs. Le but n'est pas de comprendre les scripts dans les détails mais de comprendre les principes généraux et de pouvoir faire des modifications rudimentaires des scripts.

b) **Préparation des données :**
Vous devez enregistrer les données qui vont servir à entraîner et à tester les HMMs. Pour ce faire, utilisez un outil de votre choix qui vous permet d'enregistrer des fichiers audio (format Microsoft wav, 16kHz, 16 bits). Vous avez besoin de 4 fichiers audio par chiffre, 3 pour entraîner, le dernier pour tester. Veuillez nommer les fichiers audio *'1_1.wav' '1_2.wav' '1_3.wav' '2_1.wav' '2_2.wav'... '5_3.wav'* pour les fichiers d'entraînement et *'1t.wav' '2t.wav' ... '5t.wav'* pour les fichiers de test. Vous devez ensuite déterminer le nombre d'état de chaque modèle HMM en effectuant les transcriptions phonétiques de chaque chiffre (complétez le tableau ci-après). Modifiez le script *train_test_hmm.m* en fonction des différentes valeurs de N (n'oubliez pas d'inclure les états non-émetteurs et les états réservés pour le silence).

c) **Extraction des paramètres :**
La paramétrisation des fichiers *.wav est faite avec le script *melcepst.m* qui est directement appelé par le script principal *train_test_hmm.m*. Nous utiliserons pour l'exercice les coefficients MFCC (Mel Filtered Cepstral Coefficient). Modifiez le script *train_test_hmm.m* pour afficher la durée des fichiers d'entraînement (en millisecondes) ainsi que le nombre de vecteurs acoustiques qui en sont extraits (compléter le tableau ci-dessous avec les valeurs moyennes des 3 fichiers d'entraînement). L'écart entre deux fenêtres d'analyse est de 8 ms. Vérifier que le nombre de vecteurs acoustiques correspond bien à ce qui est attendu étant donné l'écart entre deux fenêtres d'analyse.

d) **Entraînement des modèles :**
Pour l'entraînement des modèles, les fonctions suivantes sont appellées depuis le script principal *train_test_hmm.m* :

— *initemis.m* : initialise les moyennes et les écarts types d'un HMM à N états en utilisant comme "bootstrap" le premier fichier d'entraînement disponible. Les densités de probabilités d'émission sont des gaussiennes avec des matrices de covariance diagonales. Les "Inputs" du script *initemis* sont la séquence d'observations et le nombre d'états. Les "Outputs" sont les matrices **MI** et **SIGMA**. Tous les états émetteurs sont initialisés aux valeurs des moyennes et des déviations standards calculés sur base d'une segmentation linéaire d'un fichier wav.

— *inittran.m* : initialise la matrice de probabilités de transitions. Comme "Input", cette fonction prend le nombre d'état N et donne comme "Output" la matrice **A** initialisée. La probabilité de passer au prochain état ou de rester sur le même état est ici initialisée à 0.5.

— *vit_reestim.m* : Effectue l'entraînement des HMMs à travers une réestimation des paramètres des modèles en utilisant trois fichiers pour chaque chiffre. L'entraînement s'effectue avec le critère de Viterbi : on cherche l'alignement optimal en fonction des anciennes valeurs des gaussiennes et on calcule des nouvelles valeurs des gaussiennes sur base des nouveaux alignements. Les "Inputs" du script sont : les trois séquences d'observations et les anciennes matrices **A**, **MI** et **SIGMA**. Les Outputs sont : les nouvelles matrices **A**, **MI** et **SIGMA**, ainsi que la probabilité moyenne P_{tot} obtenues lors de la réestimation.

e) **Reconnaissance :**
La reconnaissance est effectuée par la fonction *viterbi_log.m* qui implémente l'algorithme de Viterbi dans le domaine logarithmique. Les "Inputs" de cette fonction sont la séquence d'observation et les matrices **A**, **MI** et **SIGMA**. Les "Outputs" sont la probabilité Viterbi et le vecteur X qui détermine la séquence d'états optimale étant donné la séquence d'observations et le modèle.

Points à résoudre

TABLE 1 – Veuillez compléter les transcriptions phonétiques, le nombre d'états du HMM, la durée moyenne des 3 fichiers wav ainsi que le nombre moyen de vecteurs acoustiques pour les 3 fichiers.

Mot	Transcription phonétique	N	durée (ms)	n vect acoust
un				
deux				
trois				
quatre				
cinq				

a) Copier le fichier "T4_HMMs.zip" depuis le site web des séries d'exercices.

b) Enregistrer votre propre voix comme indiqué dans les sections précédentes.

c) Compléter le tableau ci-dessus et effectuer les modifications des scripts comme indiqué.

d) Entraîner les modèles des 5 chiffres (1-5) avec le script *train_test_hmm.m*.

e) Observez l'évolution de la probabilité Ptot lors des itérations d'entraînement pour chaque chiffre. Que pouvez-vous conclure de cette évolution ?

f) Construisez un tableau 5×5 avec les valeurs de probabilités obtenues pour les observations de test étant donné les cinq modèles HMM. Quelles sont les modèles reconnus pour vos 5 fichiers de test ?

g) Enregistrer le mot "peu" et faire le test de reconnaissance. Quelles valeurs de probabilités obtenez-vous dans ce cas ? Quel est le modèle gagnant ?

h) Demander à un collègue de vous fournir ses enregistrements de test et faites la reconnaissance sur ces fichiers avec vos modèles entraînés. Comme au point précédent, construisez également un tableau 5×5 récapitulant les probabilités obtenues. Que pouvez-vous conclure de ces résultats ?

Que faut-il rendre ? Dans une archive zip :

— **un** document pdf contenant les réponses aux questions
— les fichiers sons *.wav
— les scripts *.m modifiés