

# TP5 : Reconnaissance de la parole par Modèles de Markov Cachés (HMM)

Le but est de réaliser une calculatrice. La méthode de reconnaissance utilisée est fondée sur les Modèles de Markov Cachés (Hidden Markov Models – HMM). Vous utiliserez pour cela la boîte à outil HTK (Hidden ToolKit distribué par <http://htk.eng.cam.ac.uk/>).

## 1. Présentation du système

Des programmes ont été écrits pour vous faciliter l'utilisation du toolkit. Vous pouvez les télécharger sous Moodle : **BE\_TAP\_HMM.zip**. Ils ont également présents sur les machines de la MFJA dans le répertoire : **BE\_TAP\_HMM/**

Dans un premier temps, le système sera limité à 12 mots : les chiffres, l'addition et la soustraction. Pour plus de détails, voir le fichier « liste-mots » dans le répertoire « Modeles ».

La construction de la phrase (enchaînement des mots) doit être réalisée dans le fichier « grammaire.txt » (celui-ci se basant sur le « dictionnaire ») du répertoire « Modeles ».

Des enregistrements sont à votre disposition dans le répertoire « Signaux » : 10 fichiers par modèles (mots). Tous les signaux sont fournis au format Wave, mono, 16 bits, 16 kHz. Une liste de ces fichiers est présente dans le fichier « wav.all.lst » du répertoire « Listes ».

La **vérité terrain** (étiquetage manuel) est ici créée de manière automatique (les fichiers ne contenant qu'un seul mot !) via le script « Bin/creerlabels ». Cette annotation de référence permettra d'évaluer le système en étant comparée à l'annotation automatique lors de la phase de reconnaissance.

## 2. Topologie des modèles

Tout d'abord, déterminer le contenu de vos modèles de mots par HMM. Il est conseillé d'utiliser un état émetteur par état stable (phonème) au niveau du signal.

Pour les probabilités de transition, initialiser-les par exemple à 0,8 quand vous rebouclez sur l'état et 0,2 lorsque vous passez à l'état suivant.

Ensuite, **créer un fichier par modèle** dans le répertoire « Modeles/gabarits/ ». Le nom du fichier sera le nom du mot en toutes lettres.

Pour vous aider, regarder les modèles fournis : « 3etats », « 4etats », « 5etats », « 6etats », « 7etats ». Les modèles sont basés sur une paramétrisation cepstrale : pour plus de détails sur celle-ci regarder le fichier « htk.conf » dans le répertoire « Modeles ».

### 3. Apprentissage

La première étape consiste à **initialiser vos modèles** (script « Bin/initialisation ») grâce à l'algorithme de Viterbi.

La deuxième étape va permettre de **réaliser l'apprentissage** (script « Bin/apprentissage ») grâce à l'algorithme de Baum-Welsh.

### 4. Reconnaissance et évaluation

La reconnaissance consiste à **effectuer un décodage des mots** de la liste de test (script « Bin/reconnaissance ») grâce à l'algorithme de Viterbi. Dans un premier temps et afin de valider votre système, utiliser les mêmes fichiers en reconnaissance que ceux qui ont servi en apprentissage : fichier « wav.all.lst » du répertoire « Listes ».

L'évaluation des résultats revient à **calculer les scores sur la reconnaissance** (script « Bin/scores »).

### 5. Mise en pratique

**Modifier la grammaire** pour pouvoir prendre en compte la connexion des mots et ainsi réaliser des phrases correspondant à des commandes pour la calculatrice. Pour cela, **enregistrer des fichiers de test**.

### 6. Reconnaissance de l'opération en direct !

Utiliser la commande « `adinrec -lv 4000 Signaux/fichier.wav` » afin de détecter l'activité vocale au micro et enregistrer votre phrase dans le fichier « fichier.wav » (par exemple : 6+4). Il faudra ensuite appeler la reconnaissance. **Vérifier que votre système fonctionne** (résultats cohérents).

Remarque : le seuil d'activité vocale (4000) de « `adinrec` » doit être adapté au réglage de votre microphone ...

**Si besoin ajouter des enregistrements de votre voix** dans l'apprentissage. Ceci peut être effectué en utilisant le script « record », disponible sur Moodle. Ceci nécessite de passer un fichier contenant la liste des mots à enregistrer ainsi que le nombre d'occurrences de chacun.