

Modèles de Markov cachés principes et exemples

Master IVI,
module VisA

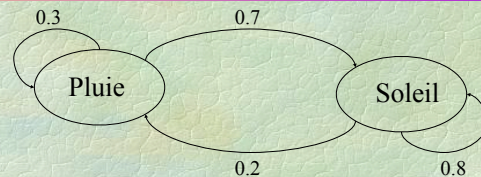
Principe

- Un ensemble d'états possibles : $\{s_1, s_2, \dots, s_N\}$
- Processus passe d'un état à l'autre, générant ainsi une séquence $s_{i1}, s_{i2}, \dots, s_{ik}, \dots$
- Principe des chaînes de Markov: probabilité d'occurrence d'un état dépend uniquement de l'état précédent:

$$P(s_{ik} | s_{i1}, s_{i2}, \dots, s_{ik-1}) = P(s_{ik} | s_{ik-1})$$

- Un modèle de markov est défini par les probabilités de transitions, $a_{ij} = P(s_i | s_j)$ and initial probabilities $\pi_i = P(s_i)$

Un petit exemple qui sert à rien



- Deux états : 'Soleil' et 'Pluie'.
- Probabilité de transition: $P(\text{'Pluie'}|\text{'Pluie'})=0.3$, $P(\text{'Soleil'}|\text{'Pluie'})=0.7$, $P(\text{'Pluie'}|\text{'Soleil'})=0.2$, $P(\text{'Soleil'}|\text{'Soleil'})=0.8$
- Proba initiales: $P(\text{'Pluie'})=0.4$, $P(\text{'Soleil'})=0.6$.
- La probabilité d'occurrence d'une séquence $\{\text{'Soleil'}, \text{'Soleil'}, \text{'Pluie'}, \text{'Pluie'}\}$:

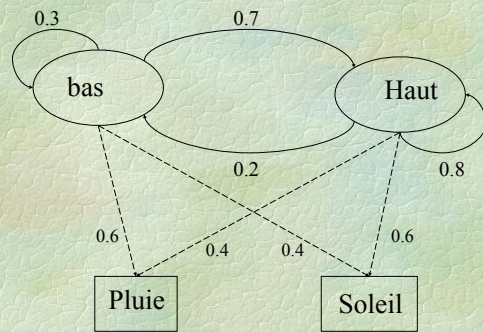
$$P(\text{'Pluie'}|\text{'Pluie'}) P(\text{'Pluie'}|\text{'Soleil'}) P(\text{'Soleil'}|\text{'Soleil'}) P(\text{'Soleil'}) = 0.3*0.2*0.8*0.6$$

Modèle de Markov cachés

- états: $\{s_1, s_2, \dots, s_N\}$
- Processus passe d'un état à un autre, générant une suite où les éléments appartiennent à l'ens. d'état : $s_{i1}, s_{i2}, \dots, s_{ik}, \dots$
- propriété de la chaîne de markov: probabilité dépend uniquement de l'état précédent:

$$P(s_{ik} | s_{i1}, s_{i2}, \dots, s_{ik-1}) = P(s_{ik} | s_{ik-1})$$
- les états ne sont pas visibles, mais chaque état génère un état observable parmi un certain nombre: $\{v_1, v_2, \dots, v_M\}$
- To define hidden Markov model, the following probabilities have to be specified: matrice des proba de transition $A=(a_{ij})$, $a_{ij}= P(s_i | s_j)$, matrice des proba des états observables $B=(b_i(v_m))$, $b_i(v_m)= P(v_m | s_i)$ et un vecteur de proba initiales $\pi=(\pi_i)$, $\pi_i = P(s_i)$. Le modèle est donné par $M=(A, B, \pi)$.

Le même petit exemple qui sert à rien, revisité



Le même petit exemple qui sert à rien, revisité

- Si l'on veut calculer la probabilité d'observation d'une séquence, par ex. {'Soleil', 'Pluie'}.

- On considère toutes les séquences possibles d'état caché, et leur proba:

$$P(\{\text{'Soleil'}, \text{'Pluie'}\}) = P(\{\text{'Soleil'}, \text{'Pluie'}\}, \{\text{'Bas'}, \text{'Bas'}\}) + P(\{\text{'Soleil'}, \text{'Pluie'}\}, \{\text{'Bas'}, \text{'Haut'}\}) + P(\{\text{'Soleil'}, \text{'Pluie'}\}, \{\text{'Haut'}, \text{'Bas'}\}) + P(\{\text{'Soleil'}, \text{'Pluie'}\}, \{\text{'Haut'}, \text{'Haut'}\})$$

Pour le premier terme :

$$\begin{aligned} P(\{\text{'Soleil'}, \text{'Pluie'}\}, \{\text{'Bas'}, \text{'Bas'}\}) &= \\ P(\{\text{'Soleil'}, \text{'Pluie'}\} | \{\text{'Bas'}, \text{'Bas'}\}) P(\{\text{'Bas'}, \text{'Bas'}\}) &= \\ P(\text{'Soleil'} | \text{'Bas'}) P(\text{'Pluie'} | \text{'Bas'}) P(\text{'Bas'}) P(\text{'Bas'} | \text{'Bas'}) &= \\ = 0.4 * 0.4 * 0.6 * 0.4 * 0.3 \end{aligned}$$

Classes de problèmes génériques

Evaluation. En considérant le HMM $M=(A, B, \pi)$ et la séquence d'observation $O=o_1 o_2 \dots o_K$, comment calculer la probabilité que M ait généré O .

• **Decodage.** En considérant le HMM $M=(A, B, \pi)$ et la séquence d'observation $O=o_1 o_2 \dots o_K$, calculer la séquence d'état caché la plus probable S_i qui génère la séquence observée O .

• **Apprentissage.** Etant donné des séquences observées de référence $O=o_1 o_2 \dots o_K$ et une structure générique de HMM (états cachés et observables), trouver les paramètres du HMM $M=(A, B, \pi)$ qui correspond au mieux aux séquences observées.

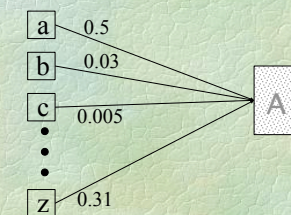
$O=o_1 \dots o_K$ est ici une séquence d'observation, donc $o_k \in \{v_1, \dots, v_M\}$.

Un exemple un peu plus utile: reconnaissance de mots (1)

- Reconnaissance de mots, chaque lettre séparée des autres.

Amherst

- La reconnaissance de caractère donne pour chaque modèle la probabilité de reconnaissance de la lettre, $P(\text{image} | \text{caractère})$.



Etats cachés

Observation

Un exemple un peu plus utile: reconnaissance de mots (2)

- Les états cachés du HMM = lettres.
- Observations = Les images segmentées depuis l'image globale du mot, on les note v_α . Il peut y avoir un nombre infini d'observations.

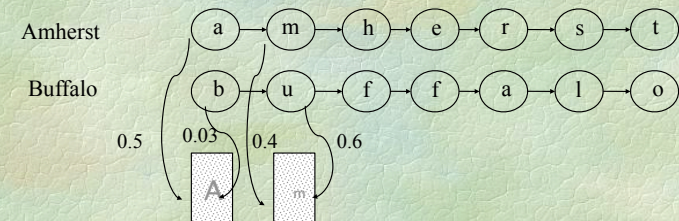
- Proba des observations = efficacité de la reconnaissance de caractère.

$$B = (b_i(v_\alpha)) = (P(v_\alpha | s_i))$$

- Les proba de transition vont être dérivées en deux exemples

Un exemple un peu plus utile: reconnaissance de mots (3)

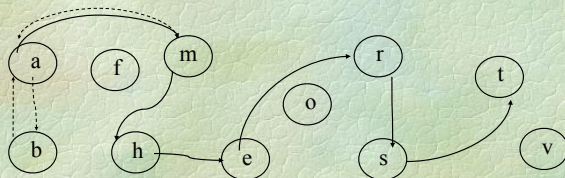
- Si on a un dictionnaire, on peut construire un HMM pour chaque mot



- reconnaissance du mot -> calcul de probabilité sur quelques HMM
- Problème d'**Evaluation**.

Un exemple un peu plus utile: reconnaissance de mots (4)

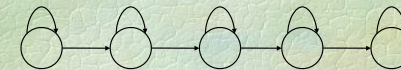
- Autre approche: un seul HMM pour l'ensemble des mots
- États cachés = les lettres de l'alphabet.
- Les proba de transition et proba initiales sont calculées à partir du modèle de langage.
- Observations et proba d'observation: inchangés.



- Ici on doit déterminer la meilleure séquence d'état caché qui a la plus forte probabilité d'avoir produit l'image du mot
- Application du problème de **Decodage**.

Reconnaissance de caractère par HMM .

- On fixe la structure des états cachés.



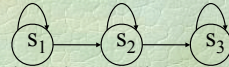
- Observations: *features* extraites de la structure de tranches verticales (par exemple).



- Mapping probabiliste depuis les états cachés vers les features : modèles de proba gaussienne, Features Quantifiées

Reconnaissance de caractère par HMM .

- Structure des états cachés:



- Observation = nombre de parties connexes dans la tranche.

- HMM pour la lettre 'A' :

$$\text{Proba des transition : } \{a_{ij}\} = \begin{pmatrix} .8 & .2 & 0 \\ 0 & .8 & .2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Proba des Observations : } \{b_{jk}\} = \begin{pmatrix} .9 & .1 & 0 \\ .1 & .8 & .1 \\ .9 & .1 & 0 \end{pmatrix}$$

A

- HMM pour 'B' :

$$\text{Proba des transitions : } \{a_{ij}\} = \begin{pmatrix} .8 & .2 & 0 \\ 0 & .8 & .2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Proba des observations : } \{b_{jk}\} = \begin{pmatrix} .9 & .1 & 0 \\ 0 & .2 & .8 \\ .6 & .4 & 0 \end{pmatrix}$$

B

Reconnaissance de caractère par HMM .

- Supposons par exemple que la séquence d'état observée soit la suivante, en 4 tranches:

{ 1, 3, 2, 1 }

- Lequel des deux HMM aura la plus forte probabilité de reconnaissance, celui associé à 'A', ou celui associé à 'B'?

Reconnaissance de caractère par HMM .

On évalue la probabilité de générer l'observation pour tous les HMM disponibles, en tenant compte de toutes les chaînes cachées possibles:

- HMM pour 'A' :

Sequence d'états cachés	Proba de transitions	proba d'observation
$S_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$	$.8 * .2 * .2$	$* .9 * 0 * .8 * .9 = 0$
$S_1 \rightarrow S_2 \rightarrow S_2 \rightarrow S_3$	$.2 * .8 * .2$	$* .9 * .1 * .8 * .9 = 0.0020736$
$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3$	$.2 * .2 * 1$	$* .9 * .1 * .1 * .9 = 0.000324$
Total = 0.0023976		

- HMM pour 'B' :

Sequence d'état cachés	Proba de transitions	Proba d'observation
$S_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$	$.8 * .2 * .2$	$* .9 * 0 * .2 * .6 = 0$
$S_1 \rightarrow S_2 \rightarrow S_2 \rightarrow S_3$	$.2 * .8 * .2$	$* .9 * .8 * .2 * .6 = 0.0027648$
$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3$	$.2 * .2 * 1$	$* .9 * .8 * .4 * .6 = 0.006912$
Total = 0.0096768		

Evaluation d'un HMM.

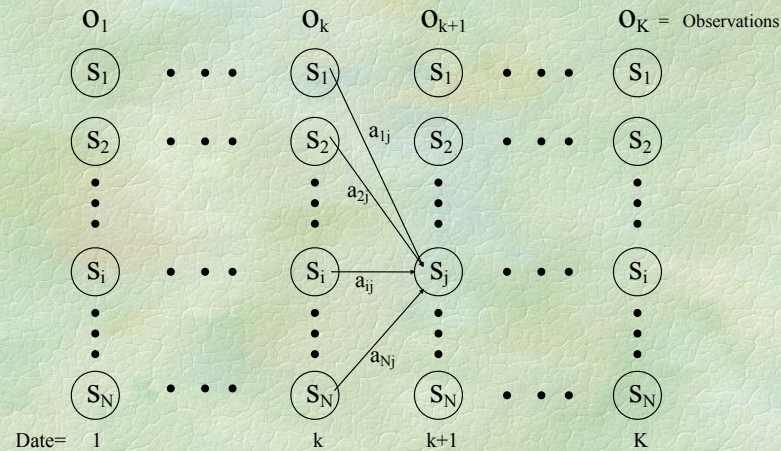
- **Problème d'évaluation.** Etant donné un HMM $M=(A, B, \pi)$ et une séquence observée $O=O_1 O_2 \dots O_K$, calculer la probabilité que M génère la séquence O .

- Essayer de trouver la séquence $O=O_1 O_2 \dots O_K$ En considérant toutes les chaînes possibles d'état est pas le plus intelligent en pratique, car il y a N^K séquences de K symboles pris parmi N

- Algorithme du **Forward-Backward** plus efficace

- On définit la «forward variable» $\alpha_k(i)$ comme la probabilité d'observer la séquence d'observation partielle $O_1 O_2 \dots O_k$ AVEC l'état caché à la date k qui est S_i : $\alpha_k(i) = P(O_1 O_2 \dots O_k, q_k = S_i)$

Representation en treillis d'un HMM



Forward recursion for HMM

- Initialisation:

$$\alpha_1(i) = P(o_1, q_1 = s_i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N.$$

- récurivité «en avant»:

$$\begin{aligned} \alpha_{k+1}(i) &= P(o_1 o_2 \dots o_{k+1}, q_{k+1} = s_i) = \\ &= \sum_j P(o_1 o_2 \dots o_{k+1}, q_k = s_j, q_{k+1} = s_i) = \\ &= \sum_j P(o_1 o_2 \dots o_k, q_k = s_j) a_{ij} b_j(o_{k+1}) = \\ &= [\sum_j \alpha_k(j) a_{ij}] b_j(o_{k+1}), \quad 1 \leq j \leq N, 1 \leq k \leq K-1. \end{aligned}$$

- Terminaison:

$$P(o_1 o_2 \dots o_K) = \sum_i P(o_1 o_2 \dots o_K, q_K = s_i) = \sum_i \alpha_K(i)$$

- Complexité :

N²K operations.

Récurivité arrière pour les HMM

• On définit $\beta_k(i)$ comme la proba d'observer la séquence partielle $O_{k+1} O_{k+2} \dots O_K$ ET que l'état caché à la date k est S_i : $\beta_k(i) = P(o_{k+1} o_{k+2} \dots o_K | q_k = s_i)$

- Initialisation:

$$\beta_K(i) = 1, \quad 1 \leq i \leq N.$$

- Backward recursion:

$$\begin{aligned} \beta_k(j) &= P(o_{k+1} o_{k+2} \dots o_K | q_k = s_j) = \\ &= \sum_i P(o_{k+1} o_{k+2} \dots o_K, q_{k+1} = s_i | q_k = s_j) = \\ &= \sum_i P(o_{k+2} o_{k+3} \dots o_K | q_{k+1} = s_i) a_{ji} b_i(o_{k+1}) = \\ &= \sum_i \beta_{k+1}(i) a_{ji} b_i(o_{k+1}), \quad 1 \leq j \leq N, 1 \leq k \leq K-1. \end{aligned}$$

- Terminaison:

$$\begin{aligned} P(o_1 o_2 \dots o_K) &= \sum_i P(o_1 o_2 \dots o_K, q_1 = s_i) = \\ &= \sum_i P(o_1 o_2 \dots o_K | q_1 = s_i) P(q_1 = s_i) = \sum_i \beta_1(i) b_i(o_1) \pi_i \end{aligned}$$

Decodage

• **Probleme du décodage:** étant donné $M = (A, B, \pi)$ et la séquence observée $O = o_1 o_2 \dots o_K$,

déterminer la séquence d'état caché la plus probable qui a généré cette séquence.

• On cherche la séquence d'état $Q = q_1 \dots q_K$ qui maximise $P(Q | o_1 o_2 \dots o_K)$, ou, de manière équivalente $P(Q, o_1 o_2 \dots o_K)$.

• Evaluation brutale se fait en temps exponentiel. On peut aussi utiliser l'algorithme de **Viterbi**.

• Define variable $\delta_k(i)$ as the maximum probability of producing observation sequence $O_1 O_2 \dots O_k$ when moving along any hidden state sequence $q_1 \dots q_{k-1}$ and getting into $q_k = S_i$.

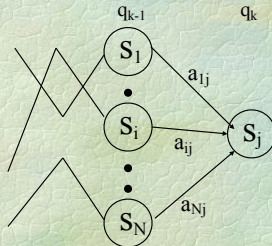
$$\delta_k(i) = \max P(q_1 \dots q_{k-1}, q_k = s_i, o_1 o_2 \dots o_k)$$

where max is taken over all possible paths $q_1 \dots q_{k-1}$.

Algo de Viterbi (1)

- **Principe:**

Si un meilleur chemin se terminant en $q_k = S_j$ passe par $q_{k-1} = S_i$ alors il devrait coïncider avec le meilleur chemin terminant en $q_{k-1} = S_i$.



- $\delta_k(i) = \max P(q_1 \dots q_{k-1}, q_k = S_j, o_1 o_2 \dots o_k) =$

$$\max_i [a_{ij} b_j(o_k) \max P(q_1 \dots q_{k-1} = S_i, o_1 o_2 \dots o_{k-1})]$$

- Pour remonter le chemin, garder l'info que le prédécesseur de S_j était S_i .

Algo de Viterbi (2)

- **Initialisation:**

$$\delta_1(i) = \max P(q_1 = S_i, o_1) = \pi_i b_i(o_1), 1 \leq i \leq N.$$

- **Récursion avant:**

$$\delta_k(j) = \max P(q_1 \dots q_{k-1}, q_k = S_j, o_1 o_2 \dots o_k) =$$

$$\max_i [a_{ij} b_j(o_k) \max P(q_1 \dots q_{k-1} = S_i, o_1 o_2 \dots o_{k-1})] =$$

$$\max_i [a_{ij} b_j(o_k) \delta_{k-1}(i)], \quad 1 \leq j \leq N, 2 \leq k \leq K.$$

- **Terminaison:** choix de la «meilleure fin» (proba maximale) à la date K

$$\max_i [\delta_K(i)]$$

- On reconstruit le meilleur chemin.

Algorithme similaire à la récursion avant du problème d'évaluation, en substituant \sum par \max , associé à une remontée de chemin.

Algo d'apprentissage (1)

- **Problème d'apprentissage:** Etant donné une séquence d'observation $O = o_1 o_2 \dots o_K$ et une structure de HMM (nombres d'états visibles et invisibles), déterminer les paramètres du HMM $M = (A, B, \pi)$ qui collent au mieux aux données d'apprentissage data, i.e. qui maximisent $P(O | M)$.

- Aucun algorithme existant produisant un optimal à ce problème.

- Solution: approche relaxation-maximisation pour maximiser localement $P(O | M)$ - Algo de **Baum-Welch**.

algo d'apprentissage (2)

- Si la séquence d'apprentissage donne aussi les séquences d'états cachés (ex: reconnaissance de mots), alors on utilise la probabilité maximale pour chaque paramètres:

$$a_{ij} = P(s_i | s_j) = \frac{\text{Nombre de transitions de } S_j \text{ à } S_i}{\text{Nombre de transitions sortantes de } S_j}$$

$$b_i(v_m) = P(v_m | s_i) = \frac{\text{Nombre de fois où l'observation } V_m \text{ survient à l'état } S_i}{\text{Nombre d'occurrence de l'état } S_i}$$

Algo de Baum-Welch

Principe:

$$a_{ij} = P(s_i | s_j) = \frac{\text{Nombre de transitions attendues de l'état } S_j \text{ à } S_i}{\text{Nombre de transitions sortantes attendues de } S_j}$$

$$b_i(v_m) = P(v_m | s_i) = \frac{\text{Nombre d'observation } V_m \text{ attendues dans l'état } S_i}{\text{Nombre d'occurrence attendues de l'état } S_i}$$

$$\pi_i = P(s_i) = \text{Fréquence attendue de } S_i \text{ à la date } k=1.$$

Algo de Baum-Welch : étape de relaxation

- On définit $\xi_k(i,j)$ comme la proba d'être à l'état caché S_i à la date k et à l'état S_j à la date $k+1$, en fonction de la séquence d'observation $O_1 O_2 \dots O_K$.
 $\xi_k(i,j) = P(q_k = s_i, q_{k+1} = s_j | O_1 O_2 \dots O_K)$

$$\begin{aligned} \xi_k(i,j) &= \frac{P(q_k = s_i, q_{k+1} = s_j, O_1 O_2 \dots O_K)}{P(O_1 O_2 \dots O_K)} = \\ &= \frac{P(q_k = s_i, O_1 O_2 \dots O_k) a_{ij} b_j(O_{k+1}) P(O_{k+2} \dots O_K | q_{k+1} = s_j)}{P(O_1 O_2 \dots O_K)} = \\ &= \frac{\alpha_k(i) a_{ij} b_j(O_{k+1}) \beta_{k+1}(j)}{\sum_i \sum_j \alpha_k(i) a_{ij} b_j(O_{k+1}) \beta_{k+1}(j)} \end{aligned}$$

Algo de Baum-Welch : étape de relaxation

- On définit $\gamma_k(i)$ comme la proba d'être dans l'état caché S_i à la date k , étant donné la séquence d'observation $O_1 O_2 \dots O_K$.

$$\gamma_k(i) = P(q_k = s_i | O_1 O_2 \dots O_K)$$

$$\gamma_k(i) = \frac{P(q_k = s_i, O_1 O_2 \dots O_K)}{P(O_1 O_2 \dots O_K)} = \frac{\alpha_k(i) \beta_k(i)}{\sum_i \alpha_k(i) \beta_k(i)}$$

Algo de Baum-Welch : étape de relaxation

- On a calculé $\xi_k(i,j) = P(q_k = s_i, q_{k+1} = s_j | O_1 O_2 \dots O_K)$
 et $\gamma_k(i) = P(q_k = s_i | O_1 O_2 \dots O_K)$

- Nombre de transitions attendues de S_i à S_j =
 $= \sum_k \xi_k(i,j)$
- Nombre de transitions sortantes de S_i = $\sum_k \gamma_k(i)$

- Nombre de fois où l'observation V_m survient dans l'état S_i =
 $= \sum_k \gamma_k(i)$, k est tel que $O_k = V_m$
- Fréquence attendue de S_i à la date $k=1$: $\gamma_1(i)$.

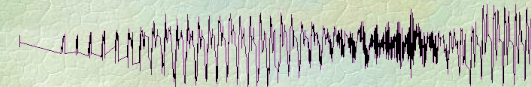
Algo de Baum-Welch : étape de maximisation

$$a_{ij} = \frac{\text{Nombre de transitions attendues de l'état } S_j \text{ à } S_i}{\text{Nombre de transitions sortantes attendues de } S_j} = \frac{\sum_k \xi_k(i,j)}{\sum_k \gamma_k(j)}$$

$$b_i(v_m) = \frac{\text{Nombre d'observation } V_m \text{ attendues dans l'état } S_i}{\text{Nombre d'occurrence attendues de l'état } S_i} = \frac{\sum_k \xi_k(i,j)}{\sum_{k, o_k=v_m} \gamma_k(i)}$$

$$\pi_i = (\text{Fréquence d'occurrence attendue de } S_i \text{ à la date } k=1) = \gamma_1(i).$$

Exemple d'utilisation des HMM: Reconnaissance de la voix



- Étant donné un signal acoustique O , quelle est la phrase la plus probable se cachant derrière, parmi toutes les phrase d'un langage L ?

Reconnaissance de la voix

- Étant donné un signal acoustique O , quelle est la phrase la plus probable se cachant derrière parmi toutes les phrase d'un langage L ?

- On décompose O en une séquence de trames :

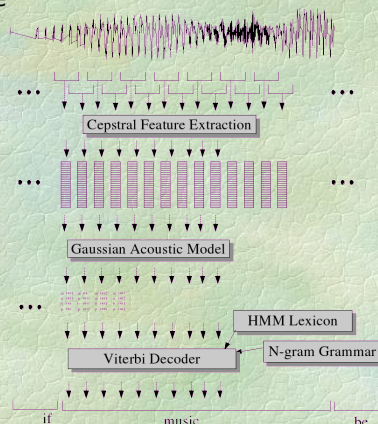
$$O = o_1, o_2, \dots, o_t$$

- On définit la phrase par une séquence de mots :

$$W = w_1, w_2, \dots, w_n$$

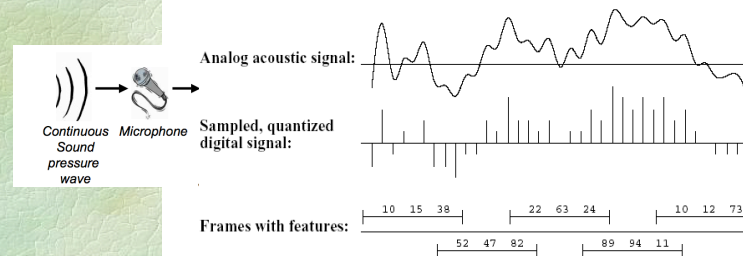
- Le problème de la reconnaissance vocale revient à trouver \hat{W} tel que

$$\hat{W} = \arg \max_{W \in L} P(W | O)$$



Reconnaissance de la voix

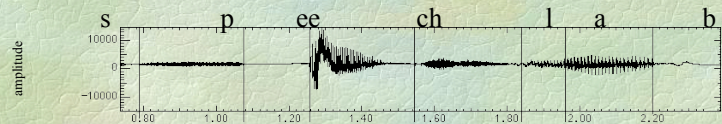
- Le signal brut provenant d'un microphone est divisé en segments non disjoints, chacun décrit par un vecteur caractéristique.



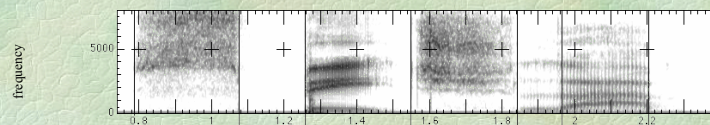
- Un segment (trame) dure 25-30 ms et est saisi à toutes les 10 ms.

Analyse spectrale

- Fréquence = ton; amplitude = volume
 - Échantillonnage à ~8 kHz téléphone, ~16 kHz microphone

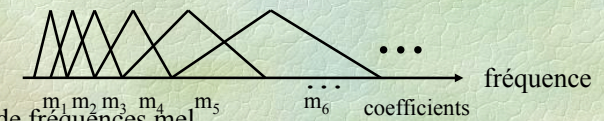


- Transformée de Fourier : structure d'une trame en termes de composantes de différentes fréquences.
- Succession des transformées de Fourier = spectrogramme
 - Les zones sombres indiquent des maxima d'énergie (formants)



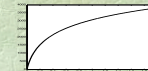
Extraction des traits acoustiques

- Banque de filtres
 - Réduit le nombre de paramètres FFT à déterminer par filtrage suivant ~20 filtres triangulaires uniformément espacés dans l'échelle mel
 - Chaque filtre fournit un coefficient qui donne l'énergie du signal dans la bande couverte par le filtre



- Échelle de fréquences mel
 - Modélise la non-linéarité de la perception audio humaine au niveau des fréquences

$$\text{mel}(f) = 2595 \log_{10}(1 + f / 700)$$



- À peu près linéaire jusqu'à 1kHz, ensuite compression logarithmique

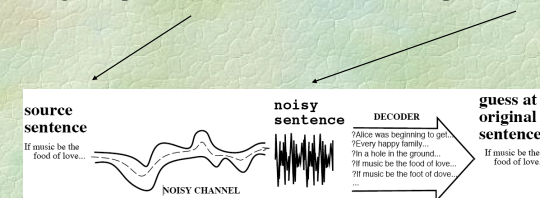
Vecteur des traits acoustiques

- Transformée en cosinus inverse du logarithme des coefficients tirés de la banque de filtres
- Donne les « Mel Frequency Cepstral Coefficients (MFCC) ». Seuls les ~12 premiers coefficients sont retenus.
- Les MFCC sont presque indépendants (à l'encontre des coefficients de la banque de filtres)
- On utilise aussi le Delta (vitesse / dérivée) et Delta² (accélération / dérivée seconde) des MFCC (+ ~24 traits)
- Et aussi le logarithme de l'énergie de la trame et son Delta et Delta², pour un total de 39 traits

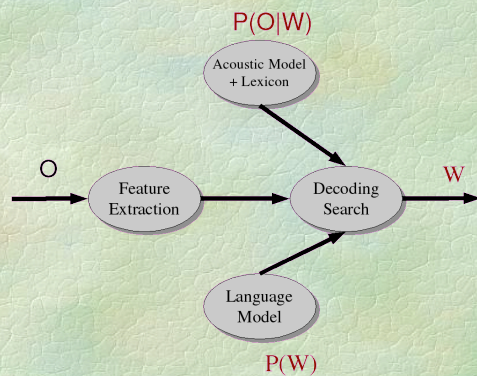
$$c_i = (2/N)^{1/2} \sum_{j=1}^N \log m_j \cos\left(\frac{\pi i}{N} (j - 0.5)\right)$$

Le problème de la reconnaissance

- Trouver la séquence la plus vraisemblable de « mots » **w** étant donnée une séquence d'observations acoustiques.
- Revient à considérer deux aspects :
 - Modèle linguistique : P(W)
 - Modèle acoustique : P(O|W)



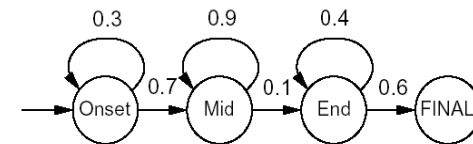
Architecture de reconnaissance vocale



Un HMM par phone ou phonème

HMM donne le modèle acoustique ($P(O|W)$)

Phone HMM for [m]:

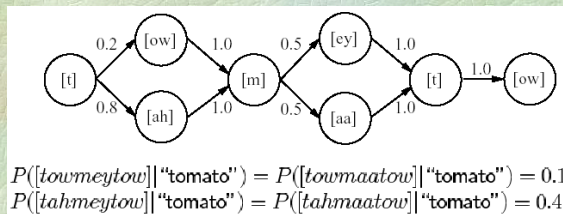


Output probabilities for the phone HMM:

Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

Modèle acoustique plus réaliste

- Coarticulation et variations dialectiques



Réseau de reconnaissance final

- Passe par la définition d'une grammaire :

```

/*
 * Task grammar
 */
$WORD = YES | NO;
( { START_SIL } [ $WORD ] { END_SIL } )
    
```

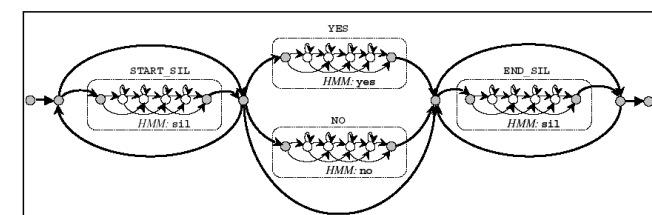


Fig. 7: Recogniser = Network + Dictionary + HMMs

Reconnaissance de la parole

- Etape 1 : L'observable est le signal de parole
 - Le HMM modélise un phonème en une suite d'états
 - un HMM par phonème
- Etape 2 : L'observable est une suite de phonèmes
 - le HMM modélise un mot en une suite de phonèmes
 - Un HMM par mot du dictionnaire
- Etape 3 : L'observable est une suite de mots
 - Le HMM modélise une phrase en une suite de mots

Reconnaissance du texte écrit

- Etape 1 : L'observable est le signal issu d'une tablette graphique
 - Le HMM modélise une lettre en une suite d'états
 - un HMM par lettre
- Etape 2 : L'observable est une suite de lettres
 - le HMM modélise un mot en une suite de lettres
 - Un HMM par mot du dictionnaire
- Etape 3 : L'observable est une suite de mots
 - Le HMM modélise une phrase en une suite de mots

Le geste

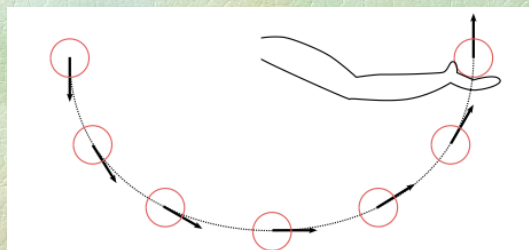


Figure 1. Movement of the example arc gesture.



Figure 2. States of the example arc gesture (when modeled by an HMM with 7 emitting states).

Conclusions

- HMM = technique de référence dans de nombreux domaines
- Bons résultats malgré les hypothèses (indép., stationnarité)
- Apprentissage coûteux
- Il existe :
 - D'autres méthodes d'apprentissage (ex: RNA)
 - D'autres principes d'apprentissage existent (spécialisation/généralisation)
 - Autres architectures (factorial HMM, input/output HMM, parallel LR HMM, etc ...)
 - Autres modèles (ex: HMM auto-regressif)

Quelques références

- [BGC00] L. Bréhelin, O. Gascuel, G. Caraux, *Hidden Markov Models with patterns and their application to integrated circuit testing*, ECML 2000.
- [Bier97] C. Biernacki, *Choix de modèles en classification*, thèse de l'Université Technologique de Compiègne, 1997.
- [BPSW70] L-E Baum, T. Petrie, G. Soules and N. Weiss, *A maximization technique occurring in statistical analysis of probabilistic functions in Markov chains*, The annals of Mathematical Statistics, 41(1):164-171, 1970.
- [DEKM98] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological sequence analysis probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.
- [KHB88] A. Kundu, Y. He, P. Bahl, *Recognition of handwritten word: First and second order Hidden Markov Model based approach*, in the proceedings of CVPR 88, pp 457-462, 1988.
- [Rab89] L.R. Rabiner, *A tutorial on Hidden Markov Models and selected applications in speech recognition*, In the proceedings of IEEE, 77(2):257-285, 1989.
- [Raph98] C. Raphael, *Automatic segmentation of acoustic musical signals using Hidden Markov Models*. IEEE PAMI, 21(4):360-370, 1998.