

Traitement Automatique de la Parole

N7 3SN-M

Jérôme Farinas
jerome.farinas@irit.fr

Institut de Recherche en Informatique de Toulouse
Université de Toulouse

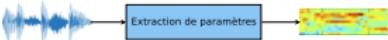
10 janvier 2025



Troisième partie III

Reconnaissance automatique de la parole

- Dans la dernière partie, nous avons eu un aperçu du signal que l'on doit traiter. Ce signal complexe a été simplifié afin de pouvoir être traité comme un vecteur d'observation évoluant dans le temps



- Nous allons maintenant proposer une modélisation statistique afin de traiter ces vecteurs d'observation afin d'obtenir une décision.

- Les données qui seront utilisées pour alimenter l'apprentissage de la modélisation détermineront la fonction du système : nous en verrons plusieurs dans les sections suivantes.

Plan

1 Reconnaissance des formes par approche statistique

• Détection d'activité vocale

• Reconnaissance automatique des sons

- Modélisation Gaussienne
- Modélisation par réseau de neurones profonds

• Identification des locuteurs

• Identification des langues

• Reconnaissance automatique de la parole

- Reconnaissance de mots isolés par programmation dynamique
- Reconnaissance par modèles de Markov cachés

- Réseaux de neurones profonds pour la parole

- Modélisation de langage

- Systèmes end-to-end (E2E)

- Réseaux antagonistes génératifs (GAN)

Reconnaissance des formes

une forme =

- un son, une note
- un mot
- une suite de mots
- une phrase

Ω : ensemble des observations $o \in \mathbb{R}^n$

K : ensemble des k classes

Une réalisation o de la classe k est un signal acoustique dans un espace \mathbb{R}^n

Exemples :

- voyelles du français
- scierie

Cadre Bayésien

• loi de réalisation de la classe : $P(o/k)$

• distribution à priori : $P(k)$

Problématique de la reconnaissance

• Soit $o \in \mathbb{R}^n$, de quelle classe est issu o ?

• Quelle est la classe la plus probable pour produire o ?

• $P(k/o)$?

Règle de Bayes

$$P(k/o) = \frac{P(o/k)P(k)}{P(o)}$$

Règle du maximum de vraisemblance

Plan pour résoudre chaque problème de RF

Formalisation

- quelles classes ? K ? $P(k)$?
- quelles observations ? $o \in \Omega$?
- quelles lois ? $P(o/k)$? quel type ? quel apprentissage ?

Mise en œuvre

- apprentissage supervisé
 - de chaque classe
 - choix algorithme
- test
 - règle de décision : maximum de vraisemblance ? minimisation du risque ?

Exercice I

Pour l'observation x , on sélectionne la classe C_1 qui maximise la probabilité $\Pr(C_1/x)$. Dans un cas à deux classes, la décision est C_1 si $\Pr(C_1/x) > \Pr(C_2/x)$ et C_2 sinon.

Une variable aléatoire X peut prendre 3 valeurs : 0, 1 ou 2. On définit deux classes C_1 et C_2 telles que :

$$\begin{aligned} \Pr(C_1) &= 0,4 & \Pr(x = 0/C_1) &= 0,2 & \Pr(x = 0/C_2) &= 0,2 \\ \Pr(C_2) &= 0,6 & \Pr(x = 1/C_1) &= 0,3 & \Pr(x = 1/C_2) &= 0,1 \\ && \Pr(x = 2/C_1) &= 0,2 & \Pr(x = 2/C_2) &= 0,8 \end{aligned}$$

- Développez la règle de décision avec la règle de Bayes
- Indiquez la décision prise lorsque $x = 0$, $x = 1$ ou $x = 2$ en utilisant la règle du maximum de vraisemblance

Règle du maximum de vraisemblance

$$\hat{k} = \arg \max_k P(k/o) = \arg \max_k \frac{P(o/k)P(k)}{\Pr(o)} = \arg \max_k P(o/k)P(k)$$

Donc pour arriver à prendre une décision sur $P(k/o)$ il est nécessaire d'apprendre $\forall k P(o/k)$ et $P(k)$.

Exercice II

Critère de décision par minimisation du risque (ou du coût moyen)

Soit $\lambda(A, B)$ une fonction quantifiant le coût ou le risque lié au fait de décider qu'on a une occurrence de la classe A alors qu'en réalité, on a une occurrence de la classe B. On établit que le coût moyen d'une décision d'occurrence de la classe C_i sachant qu'on a observé la valeur x est donné par : $\text{coût}(C_i/x) = \sum_j \lambda(C_i, C_j) P(C_j/x)$ La décision prise correspond au coût moyen de l'erreur le plus faible.

Soit la matrice suivante détaillant la fonction de coût de décision

décision C_1	décision C_2
C_1	0 4
C_2	15 0

- Indiquer la décision prise lorsque $x = 0$, $x = 1$ ou $x = 2$ en utilisant la règle de minimisation du coût moyen de l'erreur.

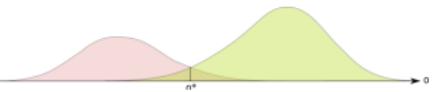
Modélisation Parole/Non Parole

- reconnaissance locale = décision sur une trame de N points (ex : 25 ms, $f_{ech} = 16KHz$, $N = 400$ pts)

- $o_t = \text{énergie}_t = \sum_{n=0}^{N-1} s_n^2$

- $\Pr(o_t/\text{Parole}) = N(o_t, \mu_p, \sigma_p) = \frac{1}{\sigma_p \sqrt{2\pi}} \exp^{-\frac{1}{2}(\frac{o_t - \mu_p}{\sigma_p})^2}$

- $\Pr(o_t/\text{NonParole}) = N(o_t, \mu_{np}, \sigma_{np})$



- Apprentissage = rechercher θ en apprentissage (il est inutile de trouver les paramètres des lois normales (μ, σ))

- Reconnaissance = rechercher $k \Rightarrow o \leq o^*$

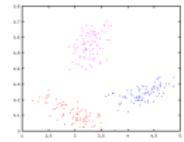
Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 161 / 339

Reconnaissance automatique des sons

- reconnaissance locale = décision sur une trame de 20 ms

- $o_t \in \mathbb{R}^n$ avec $n > 1$

- Quel est la classe ? Quel est donc le son à reconnaître ?



les trois classes à l'extrême du triangle vocalique :
/a/ (rouge) /u/ (bleu) et /i/ (magenta)

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 164 / 339

Plan

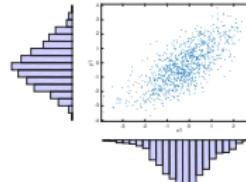
- ➊ Reconnaissance des formes par approche statistique
- ➋ Détection d'activité vocale
- ➌ Reconnaissance automatique des sons
 - ➌ Modélisation Gaussienne
 - ➌ Modélisation par réseau de neurones profonds
- ➍ Identification des locuteurs
- ➎ Identification des langues
- ➏ Reconnaissance automatique de la parole
 - ➏ Reconnaissance de mots isolés par programmation dynamique
 - ➏ Reconnaissance par modèles de Markov cachés
 - ➏ Réseaux de neurones profonds pour la parole
 - ➏ Modélisation de langage
 - ➏ Systèmes end-to-end (E2E)
 - ➏ Réseaux antagonistes génératifs (GAN)

Plan

- ➊ Reconnaissance des formes par approche statistique
- ➋ Détection d'activité vocale
- ➌ Reconnaissance automatique des sons
 - ➌ Modélisation Gaussienne
 - ➌ Modélisation par réseau de neurones profonds
- ➍ Identification des locuteurs
- ➎ Identification des langues
- ➏ Reconnaissance automatique de la parole
 - ➏ Reconnaissance de mots isolés par programmation dynamique
 - ➏ Reconnaissance par modèles de Markov cachés
 - ➏ Réseaux de neurones profonds pour la parole
 - ➏ Modélisation de langage
 - ➏ Systèmes end-to-end (E2E)
 - ➏ Réseaux antagonistes génératifs (GAN)

Loi normale multidimensionnelle

- le vecteur de paramètres peut être représenté par des lois normales multidimensionnelles



Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 165 / 339

Détection d'activité vocale

- Nous allons mettre en application la modélisation statistique par un premier système simple : un détecteur d'activité vocale. Il faudra différencier les zones de parole et de non parole.
- Nous allons considérer chaque observation de manière indépendante
- Pas de prise en compte de l'enchâinement temporel
- Une décision sera prise pour chaque vecteur d'observation
- vecteur d'observation : calcul de l'énergie de la parole

Plan

- ➊ Reconnaissance des formes par approche statistique
- ➋ Détection d'activité vocale
- ➌ Reconnaissance automatique des sons
 - ➌ Modélisation Gaussienne
 - ➌ Modélisation par réseau de neurones profonds
- ➍ Identification des locuteurs
- ➎ Identification des langues
- ➏ Reconnaissance automatique de la parole
 - ➏ Reconnaissance de mots isolés par programmation dynamique
 - ➏ Reconnaissance par modèles de Markov cachés
 - ➏ Réseaux de neurones profonds pour la parole
 - ➏ Modélisation de langage
 - ➏ Systèmes end-to-end (E2E)
 - ➏ Réseaux antagonistes génératifs (GAN)

Modélisation loi normale multidimensionnelle

Hypothèse

Les observations en apprentissage se regroupent sous un nuage Gaussien : $\Pr(o/k) = N(o, \mu_k, \Sigma_k)$

$$s(o) = \arg \max_k \Pr(o/k) \Pr(k)$$

$$s(o) = \arg \max_k \Pr(o/k)$$

car les classes sont équiprobables : $\Pr(k)$ identiques $\forall k$

$$s(o) = \arg \max_k \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_k)}} \exp^{-\frac{1}{2}(o - \mu_k)^T \Sigma_k^{-1} (o - \mu_k)}$$

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 166 / 339

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 167 / 339

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 168 / 339

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 169 / 339

Jérôme Farissat (IRIT UT) TAP 10 janvier 2025 170 / 339

Apprentissage loi normale multidimensionnelle

Apprentissage

$\text{Appk} = \{o_1^k, \dots, o_{N_k}^k\}$ ensemble des observations à utiliser pour

$$\text{l'apprentissage avec } o_i = \begin{bmatrix} o_i(1) \\ o_i(2) \\ \vdots \\ o_i(n) \end{bmatrix} \text{ et } i \in [1, 2, \dots, N_k]$$

afin de pouvoir estimer $N(\mu_k, \Sigma_k)$

$$\mu_k = \frac{1}{N_k} \sum_{p=1}^{N_k} o_p^k$$

$$\Sigma_k(r, s) = \frac{1}{N_k} \sum_{i=1}^{N_k} (o_i^k(r) - \mu_k) * (o_i^k(s) - \mu_k)$$

Hypothèses de simplification I

Hypothèse

Supposons que la matrice de covariance est identique $\forall k \Sigma_k = \Sigma$

En utilisant une paramétrisation cepstralement, chaque dimension est a priori

$$\text{indépendante : } \Sigma = \begin{pmatrix} \sigma^2(1) & 0 & \dots & 0 \\ 0 & \sigma^2(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2(n) \end{pmatrix}$$

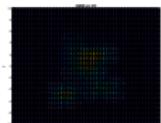
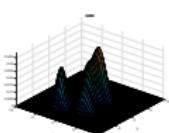
$$s^*(o) = \arg \max_k \frac{1}{\sqrt{(2\pi)^n \sqrt{\det(\Sigma)}}} \exp(-\frac{1}{2}(o - \mu_k)^t \Sigma^{-1}(o - \mu_k))$$

$$s^*(o) = \arg \max_k \exp(-\frac{1}{2}(o - \mu_k)^t \Sigma^{-1}(o - \mu_k))$$

Mé lange de lois normales (GMM) II

- Utilisation d'une combinaison linéaire de lois normales pour représenter les paramètres

- Gaussian Mixture Models (GMM)



Exercice 1

- Placez-vous dans le cas où $n = 1$. Simplifiez l'expression de la densité :

$$s(o) = \arg \max_k \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det(\Sigma_k)}} \exp(-\frac{1}{2}(o - \mu_k)^t \Sigma_k^{-1}(o - \mu_k))$$

- Que retrouvez-vous ?

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 167 / 339

Hypothèses de simplification II

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 168 / 339

$$s^*(o) = \arg \max_k -\frac{1}{2}(o - \mu_k)^t \Sigma^{-1}(o - \mu_k)$$

$$s^*(o) = \arg \min_k (o - \mu_k)^t \Sigma^{-1}(o - \mu_k)$$

$$s^*(o) = \arg \min_k \sum_{i=1}^n \frac{(o(i) - \mu_k(i))^2}{\sigma^2(i)}$$

et si $\Sigma = \sigma^2 * Id_n$ alors

$$s^*(o) = \arg \min_k \|o - \mu_k\|^2$$

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 169 / 339

Hypothèses de simplification II

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 170 / 339

Exercice 2

- Les ordinateurs n'ont pas une précision infinie, il y a donc de la perte de précision dans les manipulations qui sont réalisées sur les nombres flottants. Afin de limiter ces pertes, quand on implémente l'algorithme, on utilise donc une transformation qui va simplifier la complexité du problème et augmenter la précision des calculs : on passe au log ! Cette fonction étant croissante, elle ne change pas la décision.

- Reprenez la formule dans le cas général et simplifiez-là par un passage au log :

$$s(o) = \log \left(\arg \max_k \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{\det(\Sigma_k)}} \exp(-\frac{1}{2}(o - \mu_k)^t \Sigma_k^{-1}(o - \mu_k)) \right)$$

- Transcrivez ces calculs en code Matlab :

- calcul de la moyenne
- calcul de la matrice de covariance
- calcul de la décision

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 171 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 172 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 173 / 339

Jérôme Farissat (BRT UT)

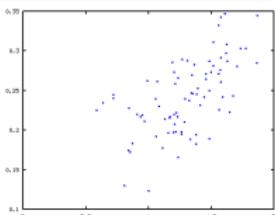
TAF

10 janvier 2025 174 / 339

Mé lange de lois normales (GMM) I

Hypothèse

Les observations en apprentissage se regroupent sous plusieurs nuages Gaussiens à cause de la variabilité : $\Pr(o/k) = \sum_{i=1}^I \pi_k^i N(o, \mu_k^i, \Sigma_k^i)$



Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 175 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 176 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 177 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 178 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 179 / 339

Apprentissage GMM I

Apprentissage

$\text{Appk} = \{o_1^k, \dots, o_{N_k}^k\}$ ensemble des observations à utiliser pour

$$\text{l'apprentissage avec } o_j = \begin{bmatrix} o_j(1) \\ o_j(2) \\ \vdots \\ o_j(n) \end{bmatrix} \text{ et } j \in [1, 2, \dots, N_k]$$

afin de pouvoir estimer π_k^i et $N(o, \mu_k^i, \Sigma_k^i)$

- Utilisation d'un algorithme EM (« Expectation Maximisation ») consistant à calculer par itérations successives la vraisemblance par modèle et à maximiser cette vraisemblance.

- Le modèle initial est obtenu par quantification vectorielle (QV).

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 180 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 181 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 182 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 183 / 339

Apprentissage GMM II

Initialisation

Algorithme de Quantification Vectorielle (Lloyd) :

- Dictionnaire de références : $D = \{d_1, d_2, \dots, d_I\}$
- $\mu^i = d_i$
- $\Sigma^i = \text{distortion des individus dans le nuage } d_i$
- $\pi^i = n_i/N$

$$\Pr(o/k) = \sum_{i=1}^I \pi_k^i N(o, \mu_k^i, \Sigma_k^i)$$

Remarque : c'est non optimisé car la QV est basée sur un critère de distortion : il ne s'agit pas d'une recherche de vraisemblance maximale comme dans l'hypothèse 1.

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 184 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 185 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 186 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 187 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 188 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 189 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 190 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 191 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 192 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 193 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 194 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 195 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 196 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 197 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 198 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 199 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 200 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 201 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 202 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 203 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 204 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 205 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 206 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 207 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 208 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 209 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 210 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 211 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 212 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 213 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 214 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 215 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 216 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 217 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 218 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 219 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 220 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 221 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 222 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 223 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 224 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 225 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 226 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 227 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 228 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 229 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 230 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 231 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 232 / 339

Jérôme Farissat (BRT UT)

TAF

10 janvier 2025 233 / 339

Jérôme Farissat (BRT UT)

Apprentissage GMM III

Itération par algorithme EM

$\pi_k^t N(o, \mu_k^t, \Sigma_k^t)$ correspond à un modèle de mélange de Gaussiennes (GMM : « Gaussian Mixture Model »). L'algorithme EM permet d'augmenter la vraisemblance du nuage.

$$\hat{Pr}(o/k) = \sum_{i=1}^I GMM_i$$

$$GMM = \arg \max_{GMM} \Pr(\text{App}_k | GMM) = \arg \max_{GMM} \prod_{k=1}^{N_k} \Pr(o_k | GMM)$$

en considérant les observations comme étant indépendantes les unes par rapport aux autres.

Hypothèse de résolution : à l'instant t, chaque observation o_k ne peut être générée que par une seule Gaussianne.

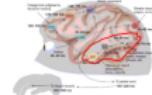
10 janvier 2025 176 / 339

Apprentissage GMM VI

Le critère d'arrêt peut-être basé sur :

- le nombre maximal d'itérations
- un seul sur la variation relative de la vraisemblance

Cerveau humain/Ordinateur



- 85×10^9 neurones
- 10^4 synapses par neurone
- 10 activations / seconde en moyenne par neurone
- 250 millions de neurones / mm^3
- 1,4 kg et 1,7 litres
- 25 Watts
- 180 000 km de connections



- 16×10^{12} opérations / s
- 4608 (petits) coeurs
- 280 Watts
- 2700 €
- Juste 10 000 fois moins puissant ?
- plutôt un facteur de 1 million : les synapses sont compliquées
- plus de 30 ans en suivant la loi de Moore \rightarrow 2046 ?

10 janvier 2025 179 / 339

Apprentissage GMM IV

1) Expectation step

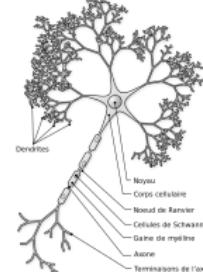
Calcul de la vraisemblance de chaque observation d'être émis par la Gaussianne j ($j \in [1, 2, \dots, I]$) :

$$\Pr(o_k | GMM(t)) = \sum_{i=1}^I \pi_i^{(t)} N(o_k, \mu_i^{(t)}, \Sigma_i^{(t)})$$

$$\Pr(o_k | j) = \frac{\pi_j^{(t)} N(o_k, \mu_j^{(t)}, \Sigma_j^{(t)})}{\Pr(o_k | GMM(t))}$$

Ces vraisemblances sont calculées pour chaque Gaussianne j

Neurone biologique



- Van Leeuwenhoek (1718) : première description fidèle
- Dutrochet (1824) : observation du corps cellulaire des neurones
- Valentin : découverte des dendrites
- Deiters (1865) : image actuelle de la cellule nerveuse
- Sherrington (1897) : les synapses ; les neuro-transmetteurs (première moitié du 20ème siècle)

10 janvier 2025 180 / 339

Apprentissage GMM V

2) Maximisation step

Ré-estimation des paramètres du GMM :

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{k=1}^N \Pr(o_k | j)$$

il s'agit du nombre moyen de fois où la j^{ème} Gaussianne est utilisée

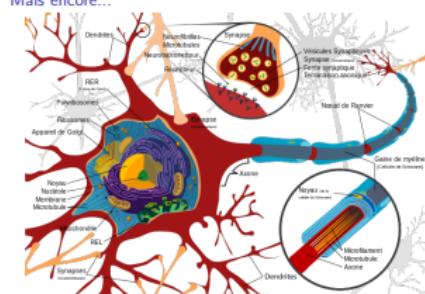
$$\mu_j^{(t+1)} = \frac{\sum_{k=1}^N o_k \Pr(o_k | j)}{\sum_{k=1}^N \Pr(o_k | j)}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{k=1}^N \Pr(o_k | j) (o_k - \mu_j^{(t+1)}) (o_k - \mu_j^{(t+1)})^T}{\sum_{k=1}^N \Pr(o_k | j)}$$

Ensuite $(t) \rightarrow (t+1)$ et les étapes sont recommandées.

A chaque étape la vraisemblance $\Pr(\text{App}_k | GMM)$ augmente.

Mais encore...



10 janvier 2025 181 / 339

10 janvier 2025 182 / 339

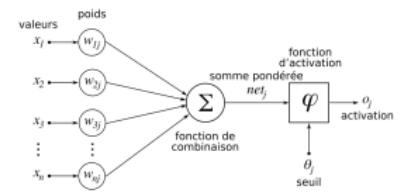
10 janvier 2025 183 / 339

10 janvier 2025 184 / 339

10 janvier 2025 185 / 339

Neurone formel

- proposé par McCulloch et Pitts en 1943



$$o = \varphi(\text{entrées}) = \varphi\left(\left(\sum_{i=1}^n x_i \cdot w_i\right) - \theta\right)$$

Apprentissage par entropie-croisée (CE)

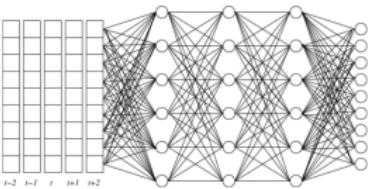
- « Cross Entropy training » [Vazely 2013]

$$E = - \sum_i t_i \log y_i, \quad \frac{\partial E}{\partial x_i} = y_i - t_i$$

- Critère prédominant pour la classification au niveau des trames
- Descente de gradient stochastique sur des mini séquences de 200-500 trames
- Tirage aléatoire des trames [Seide 2011]

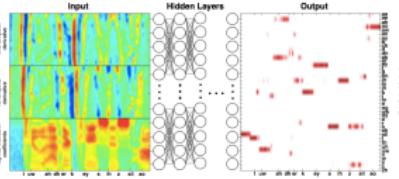
Exemple 1 architecture DNN

- Entrées : trames fMML/fMPME consécutives
- 5 à 7 couches cachées avec 2048 neurones, fonction sigmoïde
- Sortie : 2000-9000 neurones (1 neurone par état CD-HMM), fonction softmax



Exemple 2 architecture DNN

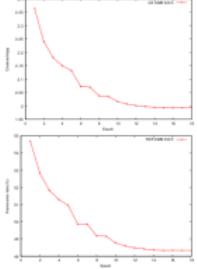
- Input layer: 11 frames of 2D spectrograms, Mel filter bank coefficients + deltas
- 5 sigmoid hidden layers: 256 nodes each; fully connected feed-forward
- Softmax output layer: 41 nodes for 40 phonemes and silence; context independent



Comparaison GMM/DNN

LVCST task	GMM	DNN
English BN SA	14.5%	12.6%
English CTS 300h SI	18.9%	14.1%
English CTS 200h SA	15.1%	12.5%
Levantine RATS 300h SA	46.7%	37.7%

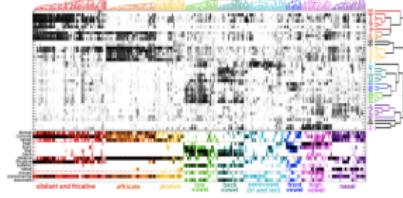
[Saon 2013]



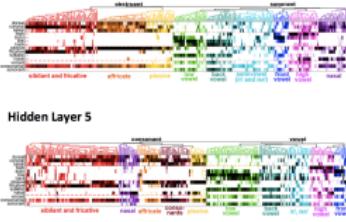
Courbes apprentissage des DNN

DNN = boite noire ? I

Hidden Layer 1

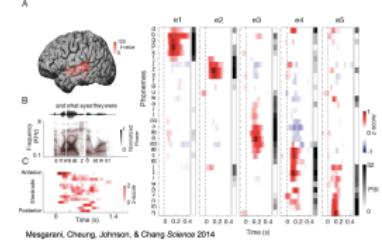


Hidden Layer 1



Et le cerveau ? → ElectroCorticoGraphy (ECG)

Phonetic feature selectivity- single electrodes



Plan

- ➊ Reconnaissance des formes par approche statistique
- ➋ Détection d'activité vocale
- ➌ Reconnaissance automatique des sons
 - ➌ Modélisation Gaussienne
 - ➌ Modélisation par réseau de neurones profonds
- ➍ Identification des locuteurs
- ➎ Identification des langues
- ➏ Reconnaissance automatique de la parole
 - ➏ Reconnaissance de mots isolés par programmation dynamique
 - ➏ Reconnaissance par modèles de Markov cachés
 - ➏ Réseaux de neurones profonds pour la parole
 - ➏ Modélisation de langage
 - ➏ Systèmes end-to-end (E2E)
 - ➏ Réseaux antagonistes génératifs (GAN)

Identification des langues

Définition

Détecter la langue parlée à partir de quelques secondes d'un échantillon sonore

Objectif

Aiguiller vers un système de reconnaissance de la parole multilingue, aiguiller vers standardiste parlant la langue pour un numéro urgence (ex : numéro 911), central téléphonique hôtelier, bornes interactives multilingues, indexation multimédia, renseignement militaire, etc.

Contraintes

Nombre limité de langue connues ou bien pas de limite (rejet), décision rapide (dès les premières secondes)

Sources d'information

- Différentes sources d'informations sont exploitables pour l'IAL :
- Acoustiques** les sons et leur fréquences d'apparition varient d'une langue à l'autre
- Phonotactiques** les enchaînements entre les sons et leur fréquence d'apparition caractérisent les langues
- Lexicales** les mots sont souvent propres aux langues. Source d'information peu intéressante si l'on veut pouvoir rajouter une langue au système sans connaissances a priori
- Prosodiques** le rythme et l'intonation varient d'une langue à l'autre.

Source d'information pour IAL : acoustique

- L'inventaire des sons varient d'une langue à l'autre (UPSID [Vallée 94])
- Même si une langue partage les mêmes sons avec une autre, il est fort peu probable que leur fréquence d'apparition soit identique.
- Nécessite des décodeurs acoustico-phonétiques ou bien une segmentation au niveau phonétique ou infra phonétique

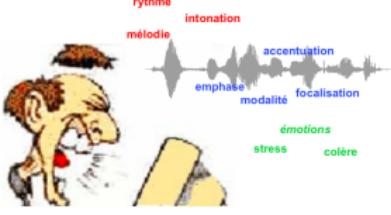
Source d'information pour IAL : phonotactique

- L'enchaînement des sons est particulier aux langues
- Certains enchaînements ne se retrouvent pas dans d'autres langues
- Leur fréquence d'apparition est également unique

Source d'information pour IAL : lexique

- Chaque langue possède son propre lexique
- Difficulté : la frontière entre les mots n'est pas facile à trouver quand on ne connaît pas la langue
- Utiliser l'inventaire des mots d'une langue impose de disposer d'importantes ressources lexicales, qui ne sont pas forcément faciles à obtenir (langues rares ou bien langues ne disposant pas de transcriptions textuelles)
- Si l'on veut pouvoir rajouter une langue facilement à un système, cette source d'information n'est pas privilégiée car elle demande des ressources coûteuses ou bien demandant l'utilisation d'expertises
- Quelques travaux ont été réalisés en utilisant partiellement cette ressource ([Hieronymous 96], [Adda 98])

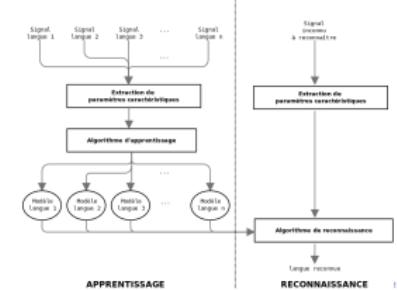
Source d'information pour IAL : prosodie



Typologie rythmique des langues

- Langues accentuelles (les syllabes peuvent avoir des durées différentes, mais le temps compris entre deux syllabes accentuées est approximativement constant)
 - ▶ Anglais
 - ▶ Allemand
 - ▶ Néerlandais
 - ▶ Polonais
 - ▶ ...
- Langues syllabiques (la prononciation de chaque syllabe prend approximativement le même temps, si bien que la durée effective de chacune d'elles dépend de la situation)
 - ▶ Espagnol
 - ▶ Italien
 - ▶ Français
 - ▶ Cantonais
 - ▶ ...
- Langues moraïques (rythme comparable à celui des langues syllabiques, mais dont l'unité rythmique de base est la more)
 - ▶ Japonais

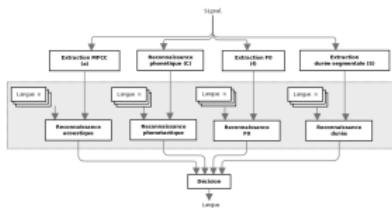
IAL : apprentissage et reconnaissance



IAL : modélisation par approche probabiliste

Décomposition du problème en sous problèmes [Hazen-Zue 1994]

$$L^* = \arg \max_i \Pr(L_i, C, S, a, f)$$



Résultats systèmes IAL

Système	Tâche	Test	LID Performance	Référence
GMM 40 mélènes	GMM-TS LID	IS95	95.5% LID recognition rate	[Zissman 1995]
GMM-SAC 512 émettrice	CALLFRIEND evalt		98.4% equal error rate (EER)	[Singer 2008]
GMM-SAC 1 2048 émettrices avec filtre de la forme	GMM-TS LID	45s	98.4% LID recognitions	[Alesis 2008]
GMM-MCE	GMM-TS IC	45s	93.1% open set test, 98.4% close set test	[Qas 2008]
SVR-GSC	2003 NIST LRE	30s	9.1% EER	[Campbell 2004]
GMM-SVH	2003 NIST LRE	30s	9.1% EER	[Yang 2003]
Anchor GMM 512 mélènes	2003 NIST LRE	30s	9.1% EER	[Noda 2003]
Jointe Faisant 1/GMM-SAC with filtre de la forme, RUEA et phonetic subphonetic	2007 NIST LRE	Conseil-set 30s; 15s; 3s	0.93%; 1.48%; 13.22%	[Torres 2008]

Reconnaissance de mots isolés par DTW

Phase d'apprentissage

- Constitution de dictionnaire de référence (ou modèles) : R_1, R_2, \dots, R_N

Phase de reconnaissance

- Recherche de la référence R_m la plus proche de l'image acoustique du mot M à identifier à l'aide d'une distance D :

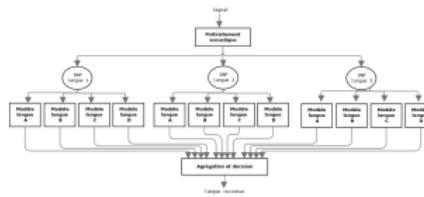
$$m = \arg_{1 \leq i \leq N} \min D(O, R_i)$$

Remarques :

- la distance utilisée dépend de la nature des paramètres
- dans tous les cas il s'agit d'un problème d'alignement temporel

IAL : modélisation PRLM

Parallel Phone Recognition followed by Language Modeling [Zissman 1996]



Fusion de systèmes IAL

Equal Error Rate (EER) of various systems in NIST LRE 2003 30s tasks.

LID SYSTEM	EER%
Primary system 1 (MFCC)	11.9
Primary system 2 (Pitch+Intensity)	25.3
Primary system 3 (MFCC+Pitch+Intensity)	9.2
Primary system 4 (FM)	21.9
Primary system 5 (PRLM)	14.6
GMM fusion system (incl. all primary systems)	7.5

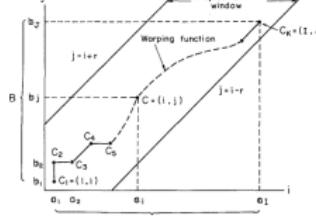
Plan

- Reconnaissance des formes par approche statistique
- Détection d'activité vocale
- Reconnaissance automatique des sons
 - Modélisation Gaussienne
 - Modélisation par réseau de neurones profonds
- Identification des locuteurs
- Identification des langues
- Reconnaissance automatique de la parole
 - Reconnaissance de mots isolés par programmation dynamique
 - Reconnaissance par modèles de Markov cachés
 - Réseaux de neurones profonds pour la parole
 - Modélisation de langage
 - Systèmes end-to-end (E2E)
 - Réseaux antagonistes génératifs (GAN)

- Reconnaissance automatique de la parole
 - Reconnaissance de mots isolés par programmation dynamique
 - Reconnaissance par modèles de Markov cachés
 - Réseaux de neurones profonds pour la parole
 - Modélisation de langage
 - Systèmes end-to-end (E2E)
 - Réseaux antagonistes génératifs (GAN)

Méthode de la programmation dynamique (1)

- proposé par Bellman en 1957
- puis adapté à la parole par Sakoe & Chiba en 1978



Méthode de la programmation dynamique (2)

- Soient $I = [1, N]$ et $J = [1, M]$ les supports temporels de A et B
- Soit $d(c) = d(a_i, b_j)$ pour tout $c = (i, j)$ de $I \times J$ (distance locale)
- Évaluer la dissemblance entre A et B revient à déterminer dans $I \times J$ un chemin $C = c_1, c_2, \dots, c_K$ tel que $c_k = (i_k, j_k)$
- Les dissemblances entre A et B sont cumulées le long de ce fichier selon :

$$D(A, B, C) = \sum_{k=1, \dots, K} w_k d(c_k)$$

avec $c_1 = (1, 1)$ et $c_K = (N, M)$ et w_k le poids attaché à l'arc (c_{k-1}, c_k)

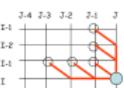
- Les chemins C doivent vérifier les conditions suivantes :
 - $c_1 = (1, 1)$ et $c_K = (N, M)$ (coincidence des extrémités)
 - $c_k < c_{k-1}$ (monotonie : croissance temporelle)
 - continuité (i.e. pas de saut ou de « trou » sur le chemin)

Contraintes

Dans le domaine discret, il n'est pas possible d'introduire des contraintes de continuité : on introduit des notions de contraintes locales et globales qui permettent de limiter le nombre de chemins possible « autour » de la diagonale.

Contraintes locales

Ex :



Pour accéder au point (i, j) , on devra venir obligatoirement d'un des 5 points suivants :
 $\{(i-1, j-1); (i-2, j-1); (i-1, j-3); (i-1, j-2); (i-1, j-1)\}$

Contraintes globales

On limite le chemin à une certaine enveloppe autour de la diagonale

Pondération des chemins et normalisation des distances (1)

- Il existe des pondérations symétriques :

- le poids associé à un arc est la somme des progressions des deux indices
- on convient que $w_1 = 2$ (poids de l'arc allant de $(0, 0)$ à $(1, 1)$)

- Exemples (les $g(i, j)$ sont les cumuls des $w_k \cdot d(c_k)$)

$$g(i, j) = \min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + 2 * d(i, j) \\ g(i-1, j-1) + 2 * d(i, j) + d(i, j) \end{cases}$$

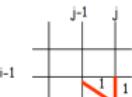
$$g(i, j) = \min \begin{cases} g(i-2, j-1) + 2 * d(i-1, j) + d(i, j) \\ g(i-1, j-1) + 2 * d(i, j) \\ g(i-1, j-2) + 2 * d(i-1, j-1) + d(i, j) \end{cases}$$

Pondération des chemins et normalisation des distances (2)

- Il existe des pondérations asymétriques :

- le poids associé à un arc est la somme des progressions de l'un des deux indices

- on convient que $w_1 = 1$ (poids de l'arc allant de $(0, 0)$ à $(1, 1)$)



$$g(i, j) = \min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + d(i, j) \\ g(i-1, j-1) + d(i, j) + d(i, j) \end{cases}$$

$$g(i, j) = \min \begin{cases} g(i-1, j) + d(i, j) \\ g(i-1, j-1) + d(i, j) \\ g(i-1, j-1) + d(i, j) + d(i, j) \end{cases}$$

Plan

① Reconnaissance des formes par approche statistique

② Détection d'activité vocale

③ Reconnaissance automatique des sons

- Modélisation Gaussienne
- Modélisation par réseau de neurones profonds

④ Identification des locuteurs

⑤ Identification des langues

⑥ Reconnaissance automatique de la parole

- Reconnaissance de mots isolés par programmation dynamique
- Reconnaissance par modèles de Markov cachés
- Réseaux de neurones profonds pour la parole
- Modélisation de langage
- Systèmes end-to-end (E2E)
- Réseaux antagonistes génératifs (GAN)

Algorithmme de programmation dynamique

```

1 début
2    $g(i, 0) \leftarrow 0$  ;
3   pour  $j \leftarrow 1$  à  $J$  faire
4      $| g(0, j) \leftarrow +\infty$  ;
5   fin
6   pour  $i \leftarrow 1$  à  $I$  faire
7      $g(i, 0) \leftarrow +\infty$  ;
8     pour  $j \leftarrow 1$  à  $J$  faire
9       /* recherche du chemin minimal */           +
10       $g(i, j) \leftarrow \min(g(i-1, j) + d(i, j),$ 
11         $g(i-1, j-1) + 2 * d(i, j), g(i, j-1) + d(i, j))$  ;
12    fin
13  fin
14   $D = g(I, J)/(I + J)$  ;
15 fin

```

Reconnaissance automatique de la parole

Soit une séquence de vecteurs d'observation :

$$\mathcal{O} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_{t_1}, \mathbf{o}_{t_1+1}, \dots, \mathbf{o}_{t_2}, \dots, \mathbf{o}_T)$$

On cherche à trouver la séquence de mots la plus probable :

$$\hat{\mathcal{W}} = (\hat{w}_1, \hat{w}_2, \dots, \hat{w}_m, \dots, \hat{w}_M)$$

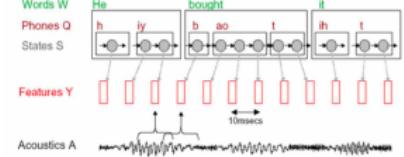
$$\hat{w}_m = \arg \max_i \Pr(w_i | \mathcal{O}_{t_1, t_2})$$

avec w_i représentant le i ème mot du vocabulaire.

En utilisant la règle de Bayes :

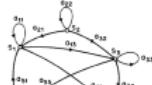
$$\Pr(w_i | \mathcal{O}_{t_1, t_2}) = \frac{\Pr(\mathcal{O}_{t_1, t_2} | w_i) \Pr(w_i)}{\Pr(\mathcal{O}_{t_1, t_2})}$$

Exemple de reconnaissance



Introduction aux processus de Markov (1)³

- Système de N états distincts : $\{S_1, S_2, \dots, S_N\}$



Temps t	1	2	3	4	5	\dots
Etat	s_1	s_2	s_3	s_4	s_5	\dots

3. Réf : Lawrence R. Rabiner, « A tutorial on hidden Markov models and selected applications in speech recognition », Proc. of IEEE, Vol. 77, n°2, février 1989.

Introduction aux processus de Markov (2)

- Soit un processus où les transitions entre états sont indépendantes du temps, i.e. :

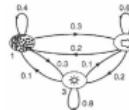
$$a_{ij} = \Pr(q_i | q_{t-1} = S_j) \text{ avec } 1 \leq i, j \leq N$$

$$a_{jj} > 0 \forall j, i$$

$$\sum_{i=1}^N a_{ji} = 1 \forall j$$

Exercice : prédiction météo par modèle de Markov (1)

- On construit un modèle de Markov en observant quel temps il fait tous les jours à midi.
- Le temps est classé en trois catégories :
 - pluvieux (état 1 : S_1)
 - nuageux (état 2 : S_2)
 - ensOLEillé (état 3 : S_3)
- L'observation du temps sur une année permet de déterminer les probabilités de transition du temps qu'il fait d'un jour à l'autre.



Exercice : prédiction météo par modèle de Markov (2)

- Traduisez les transitions entre les états de l'automate en une matrice $A = (a_{ij})$ avec a_{ij} la probabilité de passer de l'état i à l'état j
- En sachant que la météo du jour 1 est ensOLEillé, quelle est la probabilité (de ce modèle) pour que le temps pour les 7 prochains jours soit : « ensOLEillé-ensOLEillé-pluvieux-pluvieux-ensOLEillé-nuageux-ensOLEillé » ?
- Quelle est la probabilité d'observer 5 jours de soleil consécutifs ?
- Sachant que le modèle est dans un état connu, quelle est la probabilité qu'il reste dans cet état pour exactement j jours ?

Exercice : correction I

$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

- Définissons la séquence d'observation
 $O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$

$$\begin{aligned} \Pr(O|\text{Modèle}) &= \Pr(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|\text{Modèle}) \\ &= P(S_3)P(S_3|S_3)^2P(S_1|S_3)P(S_1|S_1)P(S_3|S_1) \\ &\quad P(S_2|S_3)P(S_2|S_2) \\ &= \pi_3(a_{33})^2a_{31}a_{11}a_{13}a_{32}a_{23} \\ &= 1(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 0.0001536 \end{aligned}$$

Exercice : correction II

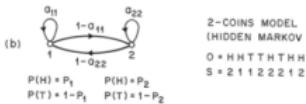
$$\begin{aligned} \Pr(O|\text{Modèle}) &= \Pr(S_3, S_3, S_3, S_3|\text{Modèle}) = \pi_3 P(S_3|S_3)^4 \\ &= 1 * 0.4096 \\ O &= \{S_1, S_1, S_1, \dots, S_1, S_j \neq S_i\} \\ t &= 1, 2, 3, \dots, d, d+1 \\ \Pr(O|\text{Modèle}, q_1 = S_i) &= (a_{ii})^{d-1}(1 - a_{ii}) \end{aligned}$$

$$\pi_i = \Pr(q_1 = S_i) \text{ avec } 1 \leq i \leq N$$

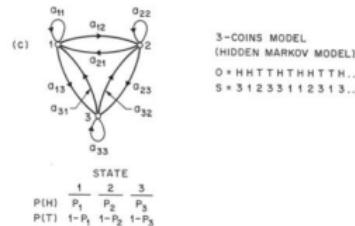
Introduction au MMC : jets de pièces I

- Une personne possédant une, deux puis trois pièces de monnaie les lance dans une pièce fermée. La seule chose que vous voyez est le résultat de chaque lancé (par exemple HHTTHHTHHTH...) appelé séquence des observations.

Introduction au MMC : jets de pièces II

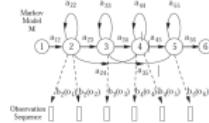


Introduction au MMC : jets de pièces III



Modèles de Markov Cachés

L'estimation de $\Pr(O_{t_1:t_2} | w_t)$ peut-être réalisée en utilisant des modèles de Markov cachés (HMM : « Hidden Model Markov »)



- HMM = automate probabiliste contrôlé par deux processus stochastiques

- le premier débute sur l'état initial et se déplace d'état en état en respectant les transitions autorisées par la topologie de l'automate
- le second génère une observation dans chaque état du HMM

Hypothèses simplificatrices en parole II

Hypothèse 1

Les probabilités $\Pr(w_i)$ peuvent être calculées séparément, sans données acoustiques (modèle de langage appris sur un grand corpus de texte)

Hypothèse 2

La probabilité que le modèle de Markov soit dans l'état i au temps t ne dépend que de l'état du modèle de Markov au temps $t-1$ (HMM du premier ordre)

$$\Pr(q_t | q_{t-1}, q_{t-2}, \dots, q_0) = \Pr(q_t | q_{t-1})$$

Paramètres d'un HMM

En prenant en compte les hypothèses simplificatrices, un modèle de Markov cache du premier ordre à N états finis est défini par la connaissance des paramètres $\Lambda = \{\Pi, A, B\}$:

- l'ensemble $\Pi = \{\pi_i, 1 \leq i \leq N\}$ des probabilités initiales π_i : probabilité d'être dans l'état i à l'instant initial
- la matrice de transition A de taille $N \times N$: l'élément a_{ij} de A est la probabilité de transition de l'état i vers l'état j
- l'ensemble des lois d'émission $B = \{b_i(o), 1 \leq i \leq N\}$: $b_i(o)$ est la probabilité d'émettre l'observation o sachant que le processus markovien est dans l'état i

Problématique des HMM

- choix des paramètres du modèle :
 - quelle topologie adopter (nombre d'états, transitions) ?
 - quelles lois estimer (discretées, continues) ?
- apprentissage :
 - étant donné un ensemble de J séquences d'observations O_j associées à chacun des modèles de Markov M_j comment choisir les paramètres Λ_j de ces modèles afin de maximiser la probabilité qu'un modèle M_j engendre la suite d'observation O_j

$$\arg \max_{\Lambda} \prod_{j=1}^J \Pr(O_j | M_j, \Lambda_j)$$

avec Λ l'ensemble des paramètres de tous les modèles.

- reconnaissance :

- étant donné une suite d'observation de longueur O_K et un ensemble de HMM, quelle est la séquence de ces modèles qui maximise la probabilité de générer O_K

Hypothèses simplificatrices en parole III

Hypothèse 3

La chaîne de Markov est stationnaire :

$$\Pr(q_t = j | q_{t-1} = i) = \Pr(q_{t+v} = j | q_{t+v-1} = i)$$

Hypothèse 4

La probabilité qu'un vecteur soit émis au temps t ne dépend pas des vecteurs précédents émis (indépendance des observations) :

$$\Pr(o_t | q_0, q_1, \dots, q_{t-1}) = \Pr(o_t | q_0, q_1, \dots, q_{t-1})$$

Reconnaissance par HMM

On cherche à calculer $\Pr(O|w)$ avec un HMM.

Il y a deux façons d'aborder le problème :

- considérer que tous les chemins sont possibles : algorithme Baum-Welch (application EM aux HMM)
- ne considérer que le meilleur chemin : algorithme de Viterbi

Hypothèses simplificatrices en parole I

- Soit $O = (o_1 o_2 \dots o_T)$ une suite d'observation de longueur T , par exemple une suite de vecteurs de paramètres MFCC.
- Soit $Q = (q_0 q_1 \dots q_T)$ une séquence d'états : au temps t , le HMM est dans l'état q_t et engendre l'observation o_t . q_0 est l'état initial avant l'émission de la première observation.

Hypothèse de base

On suppose que le signal de parole est produit par une suite d'états, chaque état étant gouverné par une loi statistique. Chaque unité de parole (phone, diphone, triphone, mot) est associé à une modèle de Markov et la concaténation de tels modèles permet d'obtenir des mots ou des phrases

Hypothèses simplificatrices en parole IV

Hypothèse 5

La probabilité qu'un vecteur soit émis au temps t ne dépend pas des états précédemment visités mais uniquement de l'état courant :

$$\Pr(o_t | q_0, q_1, \dots, q_{t-1}) = \Pr(o_t | q_t)$$

Hypothèse 6

La distribution des probabilités d'émission est approchée par un mélange de lois normales (lois gaussiennes).

$$b_j(o_t) = \Pr(o_t | N(\alpha_j, \mu_j, \Sigma_j))$$

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_j)}} \exp\left(-\frac{1}{2}(o_t - \mu_j)^T \Sigma_j^{-1} (o_t - \mu_j)\right)$$

Algorithme de Viterbi : conditions

Conditions d'application de l'algorithme de Viterbi :

- s'applique sur les treillis (graphe orienté acyclique = arbre dont les noeuds peuvent avoir plusieurs parents)
- chaque élément observé doit être aligné avec exactement un élément non observé
- le meilleur chemin en t ne dépend que des observations en t et $t-1$ (hypothèse de Markov)

Complexité : linéaire au cours du temps.

Algorithme de Viterbi I

Algorithme de Viterbi

- ❶ Initialisation : $\delta_0(i) = \pi_i$
- ❷ Récursion : $\delta_t(i) = \max_j (\delta_{t-1}(j) * a_{ij} * b_i(o_t))$
- ❸ Terminaison : $\Pr(O|w) = \max_i \delta_T(i) * a_{iT}$

avec δ_t et δ_{t-1} deux vecteurs contenant les vraisemblances cumulées sur un chemin

Vraisemblance obtenue par Viterbi :

$$\Pr(o|w) = \max_q \Pr(O|w, q)$$

Viterbi fournit aussi un alignement temporel : le meilleur chemin est trouvé par récursion arrière.

Exercice : décodage HMM par Viterbi I

- ❶ On suppose qu'un prétraitement acoustique a permis d'étiqueter chaque trame du signal à partir du dictionnaire suivant
 $D = \{X, V, F, N, U, C, S\}$.
- ❷ Signification des symboles :
 - X bruit non voisé
 - V bruit voisé
 - F zone formantique
 - N zone consonantique nasale
 - U zone consonantique voisée
 - C silence
 - S silence avec un voisement
- ❸ Nous souhaitons réaliser un petit système de reconnaissance de la parole. Nous nous intéresserons dans cet exercice uniquement aux mots : « quatre », « cent » et « six »

Jérôme Farissat (IRIT UT1) TAP 10 janvier 2025 248 / 339

Algorithme de Viterbi II

```

1 début                                /* Initialisation
2   pour chaque état i faire           +
3     | δ(i, 1) ← πi ;               |
4   fin                                 +
5   /* Récursion                      +
6   pour chaque observation i faire   +
7     pour chaque état j faire       |
8       | δ(i, j) ← Bj(i) * maxk(δ(k, i - 1)) * A(k, j) ; |
9     fin                               +
10    /* Terminaison                   +
11   Pr(O|w) ← arg maxk(δ(k, i - 1)) ; |
12 fin

```

Exercice : décodage HMM par Viterbi II

- ❶ Nous allons utiliser la version en -log de l'algorithme de Viterbi pour trouver le mot le plus probable.
- ❷ Vous utiliserez les lois d'émissions suivantes (en -log) pour représenter les états à considérer :

	X	V	F	N	U	C	S
- log (P_X)	0	2	3	3	2	1	2
- log (P_V)	2	0	3	3	3	2	1
- log (P_F)	3	3	0	2	1	3	3
- log (P_N)	3	3	2	0	1	3	3
- log (P_U)	2	3	1	1	0	3	3
- log (P_C)	1	2	3	3	3	0	1
- log (P_S)	2	1	3	3	3	1	0

- ❸ Ecrivez ces mots en phonétiques et traduisez-les avec le dictionnaire de symboles proposé.

Jérôme Farissat (IRIT UT1) TAP 10 janvier 2025 252 / 339

Algorithme de Viterbi (version -log)

L'algorithme de Viterbi peut-être formulé de manière logarithmique, ce qui permet de contrôler les risques « d'underflow » (dépassagement de capacité par le bas) qui peuvent survenir en raison de la multiplication itérée de probabilités. Cela nous permettra également de faciliter les calculs pour une résolution manuelle.

Algorithme de Viterbi (version -log)

- ❶ Initialisation : $\delta_0(i) = \ln \pi_i$
- ❷ Récursion : $\delta_t(i) = \min_j (-\ln \delta_{t-1}(j) - \ln (a_{ij}) - \ln (b_i(o_t)))$
- ❸ Terminaison : $\Pr(O|w) = \min_i (\delta_T(i) - \ln (a_{iT}))$

avec δ_t et δ_{t-1} deux vecteurs contenant les vraisemblances cumulées sur un chemin

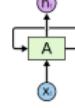
On se ramène à une minimisation d'un « coût » (il ne s'agit plus de trouver le maximum de probabilité).

Exercice : décodage HMM par Viterbi III

- ❶ Proposez une topologie de modèle de Markov cachés pour chacun des mots. Modélisez chaque son stable par un état de Markov. Prévoyez toutes les transitions possibles. Affectez des -log probabilités à chacune des transitions (0 : très probable, 1 : probable 2 : peu probable 3 : très peu probable).
- ❷ Appliquez l'algorithme de Viterbi (en -log) sur chacun des modèles de mots pour déterminer la séquence d'observation suivante :
 $O = \{\text{XXXUFEX}\}$. Quel alignement a été réalisé entre les observations et les états des modèles de Markov ? Quelle est la décision de reconnaissance ?

Recurrent Neural network (RNN) 1/2

- ❶ Réseaux de neurones récurrents introduits par Rumelhart, Hinton et Williams en 1986 [18]
- ❷ RN feed-forward = les neurones ne sont connectés que dans un sens (entrée → sortie)
- ❸ Avec RNN, les neurones de sortie par exemple peuvent voir leur sortie utilisée comme entrée d'un neurone d'une couche précédente
- ❹ Plus grande complexité
- ❺ Prise en compte des enchainements temporels (par le réseau et non des entrées avec des représentations temps-fréquence)



Algorithm de Baum Welch

- ❶ Recherche de tous les chemins ayant généré la suite d'observation $O = (o_1, o_2, \dots, o_T)$
- ❷ Repose sur le calcul de deux fonctions :
 - la passe avant (« forward »)
 - la passe arrière (« backward »)

Jérôme Farissat (IRIT UT1) TAP 10 janvier 2025 254 / 339

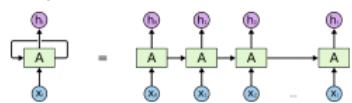
Plan

- ❶ Reconnaissance des formes par approche statistique
- ❷ Détection d'activité vocale
- ❸ Reconnaissance automatique des sons
 - ❶ Modélisation Gaussienne
 - ❷ Modélisation par réseau de neurones profonds
- ❹ Identification des locuteurs
- ❺ Identification des langues
- ❻ Reconnaissance automatique de la parole
 - ❶ Reconnaissance de mots isolés par programmation dynamique
 - ❷ Reconnaissance par modèles de Markov cachés
 - ❸ Réseaux de neurones profonds pour la parole
 - ❹ Modélisation de langage
 - ❺ Systèmes end-to-end (E2E)
 - ❻ Réseaux antagonistes génératifs (GAN)

Jérôme Farissat (IRIT UT1) TAP 10 janvier 2025 255 / 339

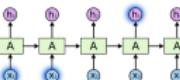
Jérôme Farissat (IRIT UT1) TAP 10 janvier 2025 256 / 339

- réseau déplié :

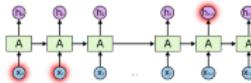


- permet naturellement de modéliser les listes et séquences

- Parfois il est nécessaire de modéliser des dépendances à court terme...



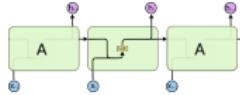
- et parfois il faut tenir compte de dépendances à long terme...



- ... or en pratique cela se révèle compliquer à apprendre [20]
 - le gradient disparaît lors de la propagation des erreurs
 - ou bien est renforcé à chaque itération
- cela rend les RNN très instable pour l'apprentissage !
- mais il existe des architectures particulières qui permettent de répondre à ce genre de problème

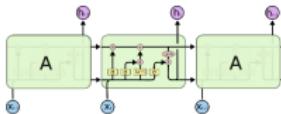
Long Short Term memory (LSTM) 1/5

- réseau de neurones récurrents à mémoire court-terme et long terme
- architecture particulière de RNN
- introduit par Hochreiter et Schmidhuber en 1997 [19]
- très utilisé pour la reconnaissance de la parole, la modélisation de langage, l'analyse de sentiments et la prédition de texte
- répond au problème de disparition du gradient
- RNN classique :



Long Short Term memory (LSTM) 2/5

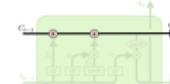
- LSTM :



- basé sur 4 sous parties

Long Short Term memory (LSTM) 3/5

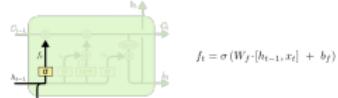
- état de la cellule



- LSTM a la possibilité de rajouter ou d'enlever des informations, c'est réglé par les sous-couches

Long Short Term memory (LSTM) 3/5

- sous couche d'oubli (forget gate)



- $\sigma \in [0, 1]$ régule l'entrée (0 : rien ne passe, 1 : tout passe)

Long Short Term memory (LSTM) 4/5

- sous couche d'entrée (input gate)

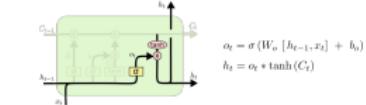


- $\sigma \in [0, 1]$ régule l'entrée (0 : rien ne passe, 1 : tout passe)

h_{t-1} représente la sortie de la dernière sortie LSTM

Long Short Term memory (LSTM) 5/5

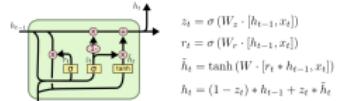
- calcul de la sortie



- \tanh permet de remplacer la sortie entre 0 et 1

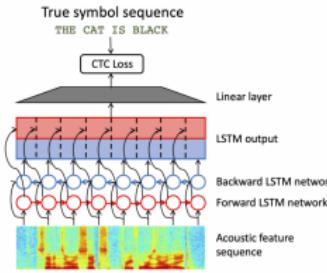
Gated Recurrent Unit (GRU)

- Variante LSTM introduite par Cho, et al. (2014) [21]
- Combinaison de la couche d'oubli et des entrées dans une seule porte de mise à jour
- Plus simple que les LSTM standards et assez populaire

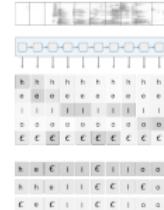


- Comparaison de variantes LSTM : Greff et al. [22] Jozefowicz [23]

Connectionist Temporal Classification (CTC)



Exemple CTC



The input is in fact an RNN,
for example:

The labels given $p_{\theta}(x|T)$,
a distribution over the outputs
(e.g. 1 to 10) for each input step

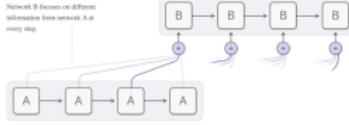
With the per time-step method
detailed above, we can compute
the probability of different sequences

By marginalizing over alignments,
we get a distribution over outputs.

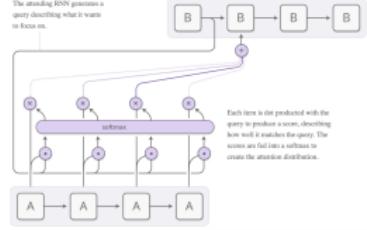
Nettoyage CTC



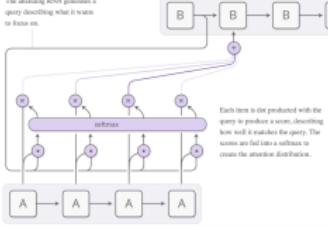
Mécanisme d'attention (1/2)



Mécanisme d'attention (2/2)



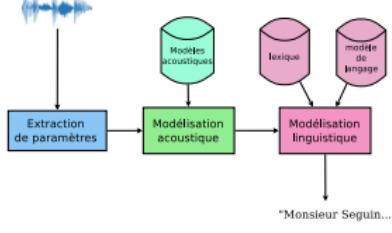
CTC/Attention système



Plan

- Reconnaissance des formes par approche statistique
- Détection d'activité vocale
- Reconnaissance automatique des sons
 - Modélisation Gaussienne
 - Modélisation par réseau de neurones profonds
- Identification des locuteurs
- Identification des langues
- Reconnaissance automatique de la parole
 - Reconnaissance de mots isolés par programmation dynamique
 - Reconnaissance par modèles de Markov cachés
 - Réseaux de neurones profonds pour la parole
 - Modélisation de langage
 - Systèmes end-to-end (E2E)
 - Réseaux antagonistes génératifs (GAN)

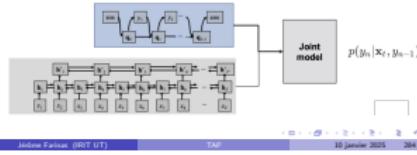
Schéma d'un système de reconnaissance de la parole classique



RNN transducteur (RNN-T)

- Extension des CTC en considérant la dépendance avec la sortie précédente
- Combine RNN en entrée et auto-regressive RNN en sortie pour alimenter une distribution jointe
 - bonnes performances avec alignement raisonnable
 - implémentation compliquée, lent, applications limitées

[Graves 2013]



Plan

- Reconnaissance des formes par approche statistique
- Détection d'activité vocale
- Reconnaissance automatique des sons
 - Modélisation Gaussienne
 - Modélisation par réseau de neurones profonds
- Identification des locuteurs
- Identification des langues
- Reconnaissance automatique de la parole**
 - Reconnaissance de mots isolés par programmation dynamique
 - Reconnaissance par modèle de Markov cachés
 - Réseaux de neurones profonds pour la parole
 - Modélisation de langage
 - Systèmes end-to-end (E2E)
- Réseaux antagonistes génératifs (GAN)**

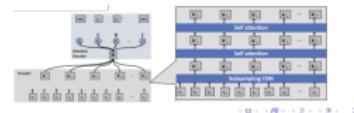
GAN : exemples d'application en parole

- génération de la parole
 - débruitage
 - analyse
 - synthèse de la parole
 - conversion de voix
- reconnaissance de la parole
 - transcription
 - reconnaissance du locuteur
 - reconnaissance des émotions
 - lecture sur les lèvres

Transformeur

- Remplace toutes les connexions récurrentes dans le codeur-décodeur basé sur l'attention par un bloc d'auto-attention (peut capturer la dépendance à très longue distance)
- toutes les opérations temporelles sont bien parallélisées
 - très bonnes performances (alignement raisonnable), apprentissage rapide, de nombreuses applications (ASR, TTS, NMT), implémentation relativement simple
 - inférence lente

[Wu et al. 2017, Dong 2018]



Comparaison des approches

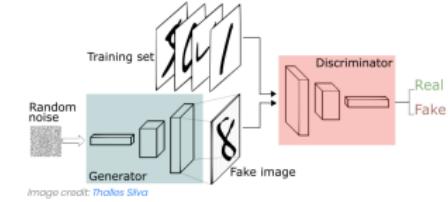
- Performance :**
 - Transformer > Joint C/A ~ RNN-T > ATT > CTC
- Vitesse d'apprentissage et d'inference :**
 - CTC > ATT > Joint C/A ~ RNN-T > Transformer
 - (l'apprentissage des transformes est rapide)
- Applications :**
 - ATT ~ Transformer ~ RNN-T > CTC, Joint C/A
- Facilité d'implémentation :**
 - CTC > ATT > Transformer > Joint C/A > RNN-T

Réseaux antagonistes génératifs (GAN)

- en anglais : Generative Adversarial Networks
- introduit par Goodfellow en 2014 [24]
- apprentissage non supervisé
- compétition entre deux modèles (cf. théorie des jeux, jeu à somme nulle) :
 - générateur : génère un échantillon (ex : image)
 - discriminateur : détermine si l'échantillon est vrai ou faux
- apprentissage difficile : gros problèmes de convergence
- de nombreuses déclinaisons : cf. Zoo⁴

4. <https://github.com/hindupuravinash/the-gan-zoo>

Architecture GAN



Plan

- Méthodes d'évaluation**
 - Corpus, ressources
 - Ressources disponibles
 - Sur et sous apprentissage
 - Campagnes d'évaluation et associations
- Comparaison HSR et ASR**
- Implémentation de systèmes de reconnaissance de la parole**
 - Boîtes à outils
 - Etat de l'art
 - Secteur industriel
 - Hype Curve Cycle de Gartner

Quatrième partie IV

Méthodes d'évaluation et implantations

❶ Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

❷ Comparaison HSR et ASR

❸ Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner

❶ l'apprentissage supervisé de modèles statistiques nécessite de grandes quantités de ressources

❷ séparation de l'ensemble des ressources :

- ▶ apprentissage : utilisé pour l'initialisation et l'apprentissage des modèles
- ▶ développement : utilisé pour le réglage des seuils de fonctionnement
- ▶ test : proche du développement, utilisé pour tester sur des données nouvelles

❸ Exemples de ressources audio :

- ▶ BREF 80 et 120 locuteurs : textes lus du journal LeMonde dans les années 1980
- ▶ Technolangue/ESTER : journaux d'information radiophoniques dans les années 2000
- ▶ EPAC : parole conversationnelle extraite de ESTER

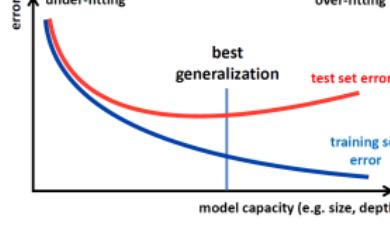


- Crée en 1992
- Distribution de corpus linguistiques
- Ressources orales et textuelles
- Favoriser les recherches et les développements sur les technologies de la langue
- Environ 30 corpora ajoutés chaque année
- Adhésion :
 - ▶ de \$1000 à \$2400 par an pour les organisations non lucratives
 - ▶ de \$24 000 à \$27 500 pour les organisations commerciales



- European Language Resources Association créée en 1995
- Distributeur de ressources linguistiques
- Adhésion de 750 à 5000€
- Ex : ESTER 1 (100h radios transcrits au niveau phrase)
 - ▶ usage recherche : 300€ (membre) / 2 000€ (non membre)
 - ▶ usage commercial : 25 000€

- Collecter des enregistrements oraux de textes
- Distribuer des modèles générés à partir de ces enregistrements
- Utilisables par les moteurs de reconnaissance vocale Open Source
- Multilingue : EN, GE, SP, FR, IT, NE, PO, GR, TU...
- Ressources très variables en fonction des langues
- A voir également :
 - ▶ LibriVox (<https://librivox.org/>)
 - enregistrement vocaux de chapitres de livres
 - beaucoup de ressources en EN, de nombreuses langues
 - ▶ OpenSLR : plus de 1000h transcrits EN (projet LibriSpeech <http://www.openslr.org/>)



❶ Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

❷ Comparaison HSR et ASR

❸ Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner

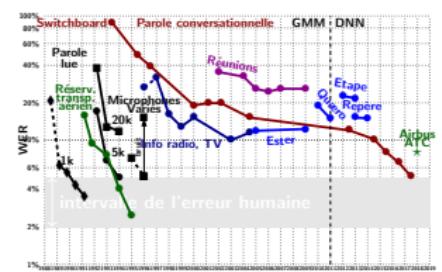
❶ Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

❷ Comparaison HSR et ASR

❸ Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner



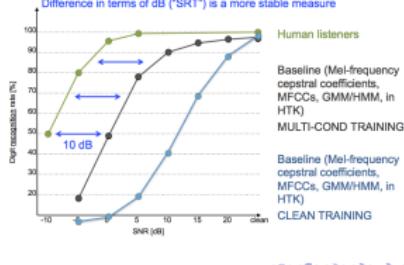
- Institut de standardisation USA
- Nombreuses campagnes d'évaluation internationales :
 - reconnaissance de la parole
 - identification des langues
 - identification du locuteurs
 - traduction automatique
- Outils d'analyse de résultats, de métriques (courbes DET, segmentation...)
- <http://www.itl.nist.gov/iad/mig/tests/>

- International Speech Communication Association
- Promouvoir le domaine de la science et de la technologie en communication parlée
- Organisation :
 - des conférences INTERSPEECH et ICSLP
 - d'ateliers (Speech Prosody, Odissey...)
 - d'écoles d'été
- lettre d'information ISCApad
- Vidéos en ligne (keynotes...)
- <http://www.isca-speech.org/>

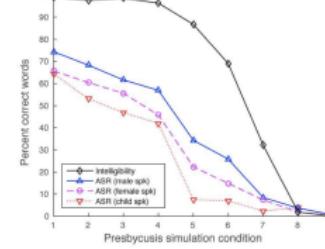


- Association Francophone de la Communication Parlée
- Special Interest Group ISCA
- Soutien, développement, diffusion et promotion sciences de la communication parlée
- Organisateur Journées d'Etude sur la Parole (actes en ligne)
- base de données des doctorats français sur la parole
- liste de diffusion parole : inscription parole-subscribe@listes.afcp-parole.org
- <http://www.afcp-parole.org/>

HSR vs ASR : différence en dB

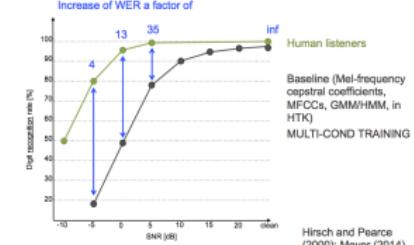


HSR vs ASR : influence du genre

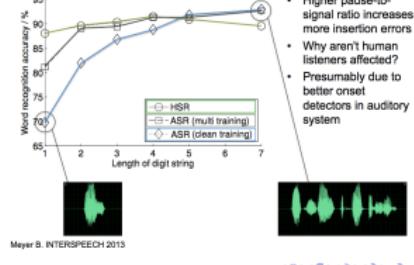


[Fontan et al., JSLHR, 2017]

HSR vs ASR



Importance longue des mots



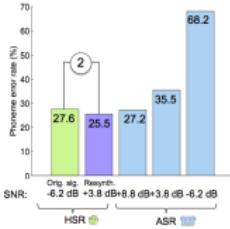
Comparaison Homme-Machine I

- Logatomes: Simple phoneme combinations(vowel-consonant-vowel, consonant-vowel-consonant)
- Suitable for HSR and ASR experiments
- 150 different logatomes
- 50 speakers
- Sources for intrinsic variability
 - Speaking rate (fast vs. slow)
 - Speaking effort (loudly and softly spoken utts.)
 - Speaking style (rising pitch/question and normal)
 - Dialect and accent
- Freely available at <http://medi.uni-oldenburg.de/ollo>
(find a list of free resources available online at the end of this tutorial)



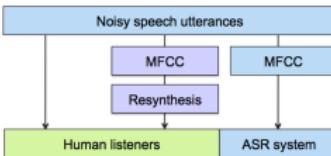
[Bernd T. Meyer 2015]

Comparaison Homme-Machine IV

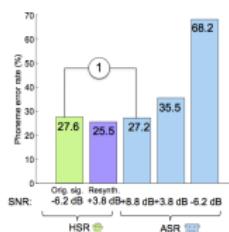


- ① Human-machine gap: ASR reaches human performance level when SNR is increased by 15dB
- ② Information loss due to feature extraction amounts to 10 dB → MFCCs do not contain all information relevant for SR

Comparaison Homme-Machine II

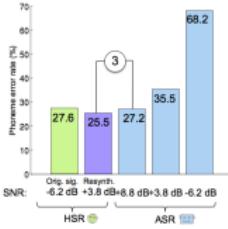


Comparaison Homme-Machine III



- ① Human-machine gap: ASR reaches human performance level when SNR is increased by 15dB

Comparaison Homme-Machine V



- ① Human-machine gap: ASR reaches human performance level when SNR is increased by 15dB
- ② Information loss due to feature extraction amounts to 10 dB → MFCCs do not contain all information relevant for SR
- ③ Using the same information for HSR and HSR: Gap of 5 dB (can be attributed to HMM)

Plan

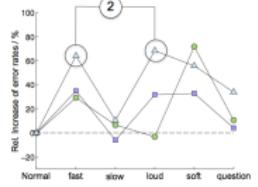
- Méthodes d'évaluation
 - Corpus, ressources
 - Ressources disponibles
 - Sur et sous apprentissage
 - Campagnes d'évaluation et associations
- Comparaison HSR et ASR

Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hypo Curve Cycle de Gartner

Comparaison Homme-Machine VII

— HSR (Org. signals, -6.2dB SNR)
— HSR (Resynth. signals, +3.8dB SNR)
△ ASR (+8.6dB SNR)



- ① Variability increases error rates (HSR and ASR): Rel. increase of up to 70 %
- ② ASR: Very high degradation for fast speaking rate and high speaking effort → analysis of effect of speaking rate

Conclusion comparaison H/M

TAP : moins performant que les humains !

- perte d'informations dans la représentation du signal
- perte d'informations dans la modélisation acoustique
- nombreuses sources d'erreurs dans le modèle de langage et le lexique
- moins bonne flexibilité et adaptabilité que les humains
- moins bonne résistance aux bruits
- a besoin de nombreuses adaptations

TAP : reste intéressant !

- reproductibilité des résultats
- alignements
- production de probabilités acoustiques, linguistiques
- produit de bon résultats dans des conditions contrôlées

● Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

● Comparaison HSR et ASR

● Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner

- Carnegie Mellon University
- open source, versions embaquée
- plusieurs moteurs de reconnaissance (pocketsphinx, sphinx 3 et 4)
- disponibilité de modèles acoustiques et linguistiques EN, FR (LIUM), RU, SP, GE, MA
- <http://cmusphinx.sourceforge.net/wiki/>

- Hidden Markov Model ToolKit
- logiciel créé par Université Cambridge, exploité par Entropic puis Microsoft
- logiciel très reconnu pour travailler sur les HMM (parole mais aussi biologie...)
- modèles acoustiques EN
- mise à jour DNN
- <http://htk.eng.cam.ac.uk/>

Kaldi

Speech brain

Hugging faces

● open source, optimisé (CUDA)

- HMM mais aussi DNN
- MMI, boosted MMI, MCE, fMPE
- modèles acoustiques EN avec corpus, recettes diverses pour de nombreux corpora
- très utilisé dans le monde académique de la recherche
- <http://kaldi-asr.org/>

- développement récent, open source, orienté GPU
- se veut simple, flexible et bien documenté
- développement sous PyTorch
- modélisations E2E : CTC, CTC+attention, transducers, transformers
- <https://speechbrain.github.io>

● Hugging Face, Inc. est une société privée américaine qui développe des outils pour la création d'applications utilisant l'apprentissage automatique

- Objectif : construire, former et déployer des modèles de pointe alimentés par la source ouverte de référence en matière d'apprentissage automatique
- Exemple (en octobre 2022) :
 - classification audio (175 modèles)
 - classification images (927 modèles)
 - détection d'objets (65 modèles)
 - Question Answering (2335 modèles)
 - Résumé autoamétique (576 modèles)
 - classification de texte (10461 modèles)
 - traduction (1745 modèles)

<https://huggingface.co>

Plan

Lien vers les systèmes actuels

Plan

● Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

- Etat de l'art des performances et lien vers les systèmes https://github.com/syhw/we_are_we
- Papier avec du code <https://paperswithcode.com/task/speech-recognition>
- WER we are and WER we think we are <https://arxiv.org/abs/2010.03432>

● Méthodes d'évaluation

- Corpus, ressources
- Ressources disponibles
- Sur et sous apprentissage
- Campagnes d'évaluation et associations

● Comparaison HSR et ASR

● Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner

● Comparaison HSR et ASR

● Implémentation de systèmes de reconnaissance de la parole

- Boîtes à outils
- Etat de l'art
- Secteur industriel
- Hype Curve Cycle de Gartner

- Google
- Amazon
- Microsoft
- IBM
- Apple
- Facebook
- Panasonic
- AT&T
- Bell Labs
- Philips
- Intel
- Vocapia
- Authôt
- Acapela
- Speech Ocean
- Vecsys
- Voxxygen
- Parrot
- Telisma
- Nuance
- Archean Technologies
- Audiogaming

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 329 / 339

Bibliographie III

- Georges Shaoen, Jen-Tsung Chen, « Large-Vocabulary Continuous Speech Recognition Systems : A Look at Some Recent Advances », Signal Processing Magazine, IEEE, vol. 29, n°6, pp. 18-33, nov 2012
- Geoffrey Hinton & co. « Deep Neural Networks for Acoustic Modeling in Speech Recognition : The Shared Views of Four Research Groups », Signal Processing Magazine, IEEE, vol. 29, n°6, pp. 82-97, nov 2012
- Stern, Morgan, « Hearing Is Believing : Biologically Inspired Methods for Robust Automatic Speech Recognition », Signal Processing Magazine, IEEE, vol. 29, n°6, pp. 34-43, nov 2012
- Daniel P. W. Ellis, « PLP, RASTA, MFCC and inversion in Matlab », <http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>, 2005
- Williams, Ronald J.; Hinton, Geoffrey E.; Rumelhart, David E. (October 1986). "Learning representations by back-propagating errors". *Nature*. 323 (6088) : 533–536. doi:10.1038/323533a0. ISSN 1476-4687.

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 332 / 339

Bibliographie VI

- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K. et Bengio, Y. "Attention-based models for speech recognition," in Advances in Neural Information Processing Systems (NIPS), 2015, pp. 577-585.
- Chan, W., Jaitly, N., Le, Q. et Vinyals, O. "Listen, attend and spell : A neural network for large vocabulary conversational speech recognition," in ICASSP. IEEE, 2016, pp. 4960-4964.
- Kim, S., Hori, T. et Watanabe, S. "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in ICASSP. IEEE, 2017, pp. 4835-4839.
- Hori, T., Watanabe, S., Zhang, Y. et Chan, W. "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in InterSpeech, 2017.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. et Polosukhin, I. "Attention is all you need," in NIPS, 2017, pp. 5998-6008.

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 335 / 339

Bibliographie I

- Calliope, « La parole et son traitement automatique », Collection technique et scientifique des télécommunications, Masson, Paris, 1989
- Lawrence R. Rabiner, « A tutorial on Hidden Markov Models and selected applications in speech recognition », Proc. of IEEE, Vol. 77, n°2, février 1989
- Joseph Mariani, « Reconnaissance automatique de la parole : progrès et tendances », Traitement du Signal, ISSN 0765-0019, Vol. 7, N° 4-NS, p. 239-266, 1990
- Lawrence Rabiner, Biing-Hwang Juang, « Fundamentals of Speech Recognition », Prentice Hall, Etats-Unis, 1993
- Jean-Claude Junqua, Jean-Paul Haton, « Robustness in Automatic Speech Recognition : Fundamentals and Applications », Springer, 1996
- Joseph Mariani, « Analyse, synthèse et codage de la parole », Hermès, Lavoisier, juillet 2002
- Joseph Mariani, « Reconnaissance de la parole », Hermès, Lavoisier, juillet 2002

Bibliographie IV

- Hochreiter, Sepp et Schmidhuber, Jürgen, ■ Long Short-Term Memory ■ Neural Computation, vol. 9, no 8, 1997, p. 1735-1780
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM : A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), 2222-2232
- Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, June). An empirical exploration of recurrent network architectures. In International Conference on Machine Learning (pp. 2342-2350)
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Bin Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. ■ Generative Adversarial Networks ■, in Advances in Neural Information Processing Systems 27, 2014

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 333 / 339

Bibliographie VII

- Dong L. et Xu, S. "Speech-transformer : A no-recurrence sequence-to-sequence model for speech recognition," in ICASSP. IEEE, 2018.
- Alexei Baevski, Yuhan Zhou, Abdelrahman Mohamed, Michael Auli "wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations", NeurIPS, 2020
- Ravanelli, Zhong, Pascol, Swietojanski, Monteiro, Trmal, & Bengio (2020, May). Multi-task self-supervised learning for robust speech recognition. In ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing , pp. 6989-6993
- Hsu, Bolte, Tsai, Lakhotia, Salakhutdinov, Mohamed (2021). Hubert : Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 3451-3460

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 335 / 339

Bibliographie II

- René Boite, Hervé Bourlard, Thierry Dutoit, Joël Hancq et Henri Leich, « Traitement de la parole », collection électricité, presses polytechniques et universitaires romandes, Lausanne, 2000
- Jean-Paul Haton, Christophe Cerisara, Dominique Fohr, Yves Laprie, Kamel Smali, « Reconnaissance automatique de la parole : du signal à son interprétation, Dunod, Paris, 2006
- Hess, « Pitch determination of speech signals », Springer, 1983
- Miller et Weibel, « Measurements of the fundamental frequency of a speech using a delay line », JASA n°28(A), 1956
- Martin, « Extraction de la fréquence fondamentale par intercorrélation avec une fonction peigne », XXII Journées d'Etudes sur la Parole, pp. 221-232, Montréal, 1981
- Povey, Ghoshal, Boulianne, Burget, Gembek, Goel, Hannemann, Motlicek, Qian, Schwarz, Silovsky, Stemberg, Vesely. « The Kaldi speech recognition toolkit », IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hawaii, US, dec 2011

Bibliographie V

- Graves, A., Fernandez, S., Gomez, F. et Schmidhuber, J. "Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks," in International Conference on Machine learning (ICML), 2006, pp. 369-376.
- Graves, A., Mohamed, A.-r. et Hinton, G. "Speech recognition with deep recurrent neural networks," in ICASSP. IEEE, 2013, pp. 6645-6649.
- Graves A. et Jaitly, N. "Towards end-to-end speech recognition with recurrent neural networks," in ICML, 2014, pp. 1764-1772.
- Miao, Y., Gowayyed, M. et Metze, F. "EESEN : End-to-end speech recognition using deep RNN models and WFST-based decoding," in IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015, pp. 167-174.

Bibliographie VIII

- Szymański, Źelasko, Morzy, Szymczak, Żyła-Hoppe, Banaszczak, Augustyniak, Mizzgajski, Carmiel (2020). WER we are and WER we think we are, arXiv :2010.03432, <https://doi.org/10.48550/arXiv.2010.03432>

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 337 / 339

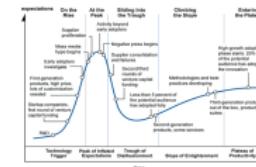
Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 335 / 339

Jérôme Farissat (BRT UT)

TAP 10 janvier 2025 337 / 339

Hype Curve Cycle de Gartner



5 phases qui composent le cycle :

- ➊ Lancement de la technologie
- ➋ Pic des espérances
- ➌ Phase de désillusionnement
- ➍ Pente d'éclaircissement
- ➎ Plateau de productivité

Méthodes d'évaluation

- ➊ Corpus, ressources
- ➋ Ressources disponibles
- ➌ Sur et sous apprentissage
- ➍ Campagnes d'évaluation et associations

Comparaison HSR et ASR

Implémentation de systèmes de reconnaissance de la parole

- ➊ Boîtes à outils
- ➋ Etat de l'art
- ➌ Secteur industriel
- ➍ Hype Curve Cycle de Gartner