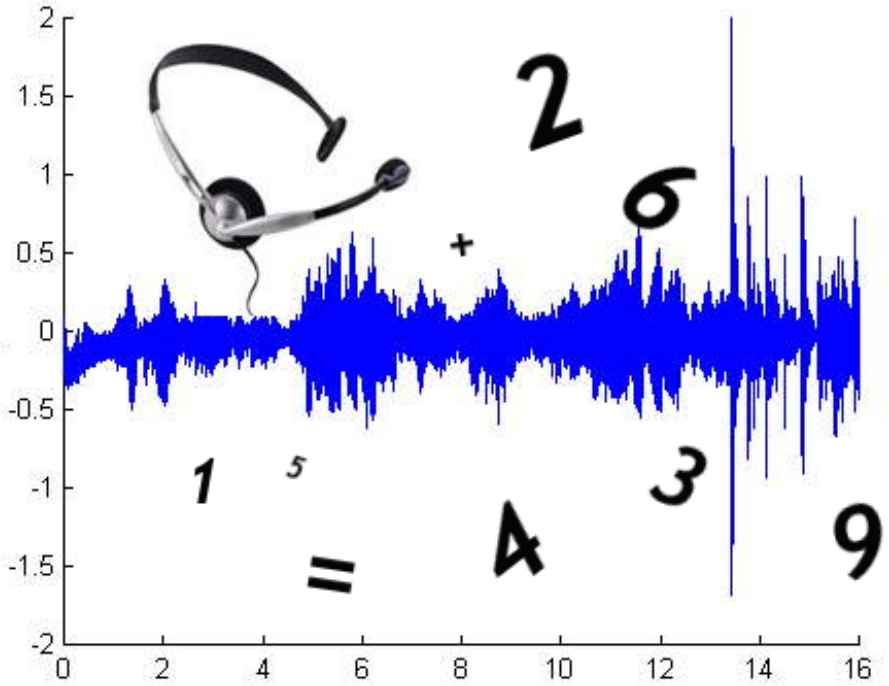
EPITA
14-16 rue Voltaire
94270 Kremlin-Bicêtre

SCIA PROMO 2005



RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

Calculatrice Vocale

Novembre 2004

par:
FRANCK BONNET
BENJAMIN DEVÈZE
MATHIEU FOUQUIN
JULIEN JEANY

responsable: **REDA DEHAK**

TABLE DES MATIÈRES

1 Introduction 1

2 Prétraitement 2

3 Monolocuteur 3

3.1 DTW . . . 3

3.2 LPC . . . 4

3.3 MFCC . . . 6

3.4 Résultats . . . 6

4 Multilocuteur 8

4.1 HMM . . . 8

4.1.1 Topologie utilisée . . . 8

4.1.2 Implémentation . . . 9

4.2 Résultats . . . 9

4.3 Parole continue . . . 11

4.4 Comparaison à la DTW . . . 11

Annexes

Bibliographie a

Glossaire a

Table des Figures b

Liste des Tableaux c

Chapitre 1

Introduction

Le but du projet que nous allons décrire dans ce rapport consiste en la réalisation d'une calculatrice vocale, le point intéressant étant la reconnaissance de la parole. Nous allons pour cela implémenter plusieurs méthodes, afin de les étudier en terme de performance et de qualité de reconnaissance.

Chapitre 2

Prétraitement

Nous avons décidé de réaliser la reconnaissance de la parole directement à partir de l'entrée microphone. Le premier travail effectué a donc consisté à capturer le signal, puis à normaliser celui-ci. Ceci a été implémenté à l'aide des fonctions fournies par l'API Win32.

Effectuant la reconnaissance sur des mots isolés, le deuxième traitement consiste à découper le signal récupéré en mot. Chaque mot prononcé est espacé d'un silence pour permettre ce découpage. Il s'agit donc de trouver ces silences afin de délimiter chacun des mots. Pour ce faire nous calculons dans un premier temps l'énergie dont dispose un signal représentant un silence.

$$E = \sum_{x \in signal} |x|^2$$

Cette valeur nous sert alors de seuil (celui-ci est ensuite multiplié par un coefficient légèrement supérieur à 1, afin de bien différencier les silences). Nous découpons grossièrement le signal en fenêtre de 50 ms, puis nous calculons l'énergie de chacune de ces portions. Si celle-ci est inférieure au seuil, cela signifie qu'il s'agit d'un silence, sinon la portion contient un mot (ou une partie d'un mot). Ayant récupéré les portions contenant autre chose que du silence, nous recherchons le début "réel" du mot (ainsi que sa fin) parmi ces portions de signal de la même manière que précedemment mais avec des fenêtres de taille beaucoup plus petite.

Le taux de passage par 0 est également fréquemment utilisé pour le découpage d'un signal. En fait notre signal peut avoir quelque fois une énergie inférieure au seuil que l'on a fixé, sans qu'il s'agisse pour autant de silence. C'est donc le taux de passage par 0 qui permet de trancher pour savoir s'il s'agit ou non de silence. Nous n'avons pas utilisé ce taux dans notre processus de découpage, car en pratique le seuil sur l'énergie nous fournit des résultats très satisfaisants à lui tout seul.

A la fin de ce traitement nous avons une liste de morceaux du signal d'origine, chacun représentant un mot prononcé.

CHAPITRE 3

MONOLOCUTEUR

Sommaire

3.1 DTW 3
3.2 LPC 4
3.3 MFCC 6
3.4 Résultats 6

3.1 DTW

Les systèmes de reconnaissance de la parole basés sur l'algorithme de DTW essaient d'évaluer la distance entre une observation et une liste de références, ainsi la référence pour laquelle cette distance est minimale permet de dire de quel mot il s'agit.

Cependant un problème se pose pour le calcul de cette distance. Le fait est que l'on ne prononce jamais deux fois le même mot, ce qui est du à plusieurs facteurs, telle que la vitesse d'élocution. Il s'agit de déformation temporelle du signal. L'algorithme de DTW permet de résoudre ce problème, il applique un alignement temporel au signal de façon à reduire au maximum le coût entre celui-ci et le signal de référence considéré.

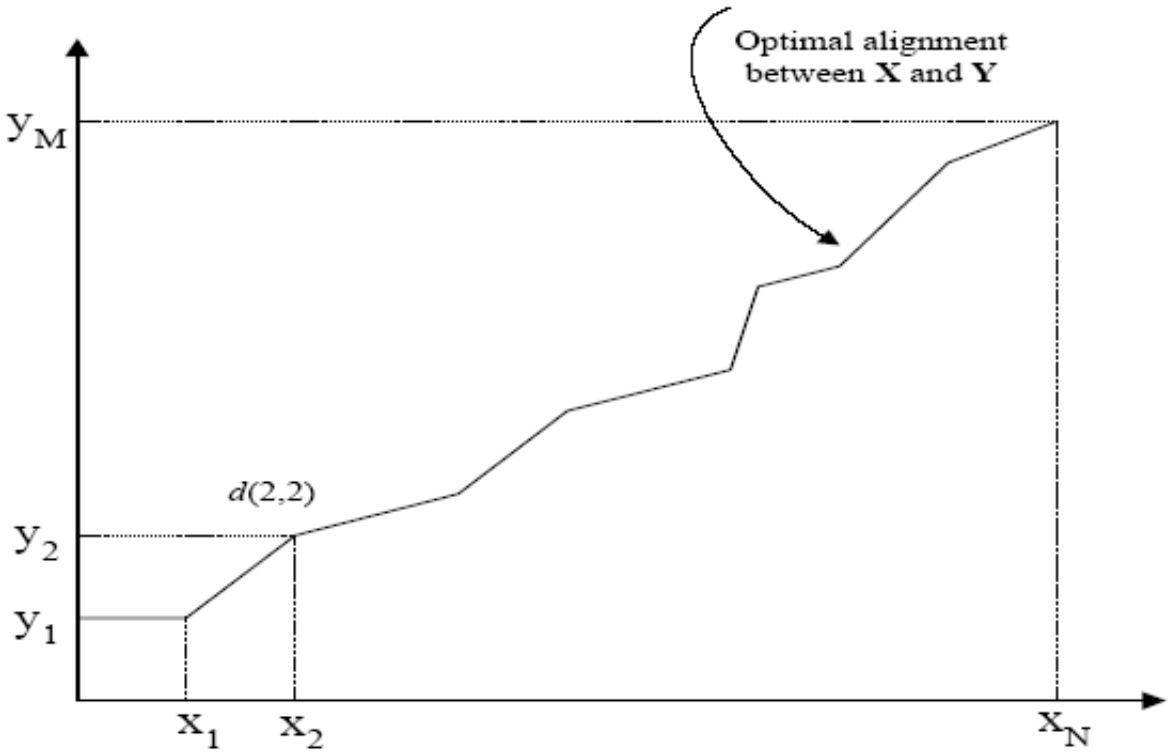


FIG. 3.1: *Alignement des signaux avec l'algorithme de DTW*

L'évaluation de la distance entre deux signaux ne s'effectue pas avec les signaux eux-mêmes. Cela ferait beaucoup trop de calculs. Il s'agit donc dans un premier temps de trouver une meilleure représentation des signaux. Nous allons donc étudier dans les parties suivantes la représentation par les coefficients LPC, puis par les coefficients MFCC.

3.2 LPC

Le codage LPC (Linear Predictive Coding) consiste à synthétiser des échantillons à partir d'un modèle de système de production vocal et d'excitation. Il s'agit d'une méthode très fréquemment utilisée pour l'analyse de la parole, l'encodage de la parole, ... Elle tire son nom du fait qu'elle permet de prédire une valeur future du signal à partir d'une combinaisons des valeurs précédentes.

Pour calculer les coefficients LPC, nous avons effectué les étapes suivantes :

- Découpage du signal en fenêtre de 30 ms, toutes les 10 ms
- Application d'une fenêtre de Hamming sur ces portions de 30 ms
- Calcul des coefficients LPC sur ces portions de 30 ms (à l'aide de la bibliothèque it++)

L'application de la fenêtre de Hamming permet de diminuer la distorsion spectrale.

Lors du calcul de la DTW entre 2 signaux on est amené à évaluer la distance entre 2 vecteurs de coefficients LPC. Dans un premier temps, nous effectuions simplement un calcul de distance euclidienne entre ces vecteurs.

Nous sommes ensuite passé à l'utilisation d'une autre distance : la distance d'Itakura-Saito, appelée également distance LLR (Log Likelihood Ratio).

Cette formule s'appuie sur le calcul de l'erreur (au carré) entre le signal et les valeurs déduites par la LPC :

$$E = \sum_{n=0}^{N-1}(s_n - \sum_{i=1}^{p} a_i s_{n-i})^2$$

Après calcul on a alors :

$$E = a^t R a$$

Avec a le vecteur de coefficient LPC, et R la matrice d'autocorrélation. La distance d'Itakura se définit alors par :

$$d(x, y) = log(|\frac{x^t R x}{y^t R y}|)$$

x étant les coefficients LPC du signal de référence, et y ceux du signal à tester. R est la matrice d'autocorrélation du signal à tester, il s'agit en fait de calculer la matrice de Toeplitz. On calcule dans un premier les coefficients composant la matrice :

$$r_k = \sum_{i=0}^{N-1-k} (s_i * s_{i+k})$$

la matrice est alors :

$$\begin{matrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ r_2 & r_1 & r_0 & \cdots & r_{p-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_0 \end{matrix}$$

Au niveau de l'implémentation il s'agit de calculer une matrice R pour chacune des fenêtres du signal (comme on l'a vu le signal est découpé en fenêtres qui se recoupent). On fait alors appel à la matrice R qui va bien, suivant le vecteur de coefficients LPC utilisé dans le calcul.

Le même travail pourrait être effectué en travaillant sur la matrice d'autocorrélation des signaux de référence, ce qui aurait pour avantage de ne pas recalculer les matrices à chaque fois, pour des raisons pratiques nous avons gardé l'utilisation du signal à tester pour calculer la matrice d'autocorrélation.

3.3 MFCC

L'étude des coefficients MFCC du signal permet d'extraire des caractéristiques de celui-ci autour de la FFT et de la DCT, convertis sur une échelle de Mel. Il s'agit de la méthode la plus utilisée pour représenter un signal en reconnaissance de la parole, car très robuste. Son principal avantage est que les coefficients obtenus sont décorréllés.

Le calcul des coefficients MFCC est réalisé de la manière suivante :

- Préaccentuation du signal, il s'agit de faire ressortir les hautes fréquences avec un filtre passe-haut de la forme $H(z) = 1 - 0.9z^{-1}$.
- Découpage du signal en fenêtre de 30 ms, toutes les 10 ms
- Application d'une fenêtre de Hamming sur ces portions de 30 ms
- Application d'une transformée de Fourrier sur chacune des portions, on obtient le spectre
- Création du banc de filtres, il s'agit de plusieurs filtres triangulaires qui vont chacun couvrir une fréquence, ils permettent de mieux simuler le fonctionnement de l'oreille humaine.
- Conversion en echelle de mel, à l'aide des filtres, de chacune des portions
- Application d'une DCT (Discrete Cosinus Transform) sur les portions, on obtient alors les coefficients cepstraux (MFCC).

En ce qui concerne l'évaluation de la distance entre 2 vecteurs de coefficients MFCC pour la DTW, nous utilisons simplement un calcul de distance euclidienne.

3.4 Résultats

Ci-dessous les taux de reconnaissances relevés pour la reconnaissance avec l'algorithme de DTW et représentation par coefficients LPC et MFCC.

Signal à reconnaître	taux de reconnaissance LPC	taux de reconnaissance MFCC
0	90 %	94 %
1	92 %	92 %
2	96 %	98 %
3	97 %	100 %
4	94 %	99 %
5	88 %	95 %
6	100 %	100 %
7	94 %	98 %
8	96 %	100 %
9	100 %	100 %
+	100 %	100 %
-	96 %	92 %
×	98 %	100 %
/	100 %	100 %
=	100 %	100 %

Tab. 3.1: *Taux de reconnaissance LPC-MFCC.*

Des problèmes de reconnaissance peuvent apparaître selon les conditions dans lesquelles le signal à tester est enregistré. Si le mot est prononcé plus ou moins proche du microphone les taux de reconnaissance peuvent varier grandement, malgré la normalisation du signal visant à empêcher ce phénomène.

Cependant si l'utilisateur prononce le mot toujours à la même distance et avec la même intensité, les taux de reconnaissance sont très satisfaisants.

Il résulte néanmoins que la représentation à l'aide des coefficients MFCC fournit de meilleurs résultats, et supporte mieux les limitations exposées liées au problème de la capture du signal. Les résultats obtenus par la représentation par coefficients LPC et par l'utilisation de la distance euclidienne sur ceux-ci ne sont pas exposés ici, il est à noter que les résultats n'étaient pas si mauvais que ça, mais toutefois moins bons qu'avec l'utilisation de la distance d'Itakura-Saito.

CHAPITRE 4

MULTILOCUTEUR

Sommaire

4.1 HMM **8**
4.1.1 Topologie utilisée 8
4.1.2 Implémentation 9
4.2 Résultats **9**
4.3 Parole continue **11**
4.4 Comparaison à la DTW **11**

4.1 HMM

Les modèles cachés de Markov consistent en une méthode statistique. Ils sont une extension des chaînes de Markov auxquelles ont été rajouté, en chaque état, des émissions de symboles, ces émissions étant soumis à des probabilités.

4.1.1 Topologie utilisée

Pour chacun des mots de référence nous avons créé un HMM (discret). La topologie que nous utilisons pour ceux-ci est un modèle gauche-droite, contenant entre 5 et 10 états, selon que l'on est en mode monolocuteur ou multilocuteur. Il s'agit d'un modèle classique employé pour la reconnaissance de la parole.

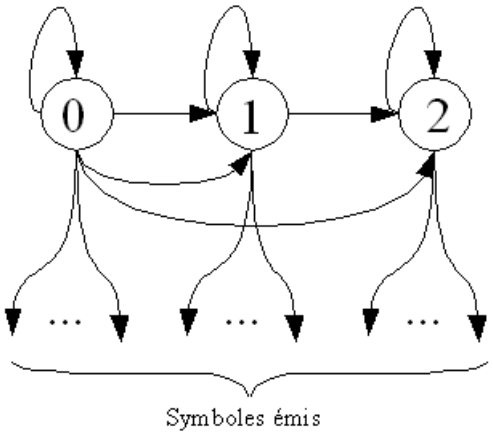


Fig. 4.1: *Topologie du HMM utilisé*

4.1.2 Implémentation

Le premier travail consiste à créer le "Codebook", c'est-à-dire la liste des symboles pouvant être émis, puisque nous voulons passer en cas discret. Pour cela nous réalisons un algorithme des k-means sur les vecteurs de coefficients représentant chacun des signaux de référence. Ce "Codebook" sera alors commun à tous les HMM de notre système. Nous avons fixé la taille de ce "Codebook" à 96 éléments (pour 15 signaux de référence). Des tailles inférieures nous donnaient des résultats moins satisfaisants, quand à prendre des tailles supérieures, les résultats étaient parfois dégradés et le temps de calcul s'avéraient bien plus long.

Pour l'apprentissage lui-même nous utilisons simplement l'algorithme de Baum-Welch (Forward-Backward). L'implémentation de base ne nous permettait pas d'utiliser un grand nombre d'exemple pour l'apprentissage, ceci à cause du fait que certaines probabilités devenaient trop faibles pour être codables sur la machine. Pour résoudre ce problème nous avons implémenter la méthode de normalisation des coefficients alpha et beta proposée par Rabiner.

Pour l'évaluation, il suffit simplement d'utiliser l'algorithme Forward. L'algorithme de Viterbi a également été implémenté, il nous permet d'étudier quelle séquence d'état est emprunté lors de l'évaluation d'une observation dans un HMM, et également de calculer la probabilité de la réalisation de cette observation par celui-ci. L'algorithme en soit ne nous est pas utile pour la reconnaissance de mots isolés, l'intérêt intervenant pour la reconnaissance en parole continue. Cependant les résultats fournis par l'algorithme restent cohérents avec ceux fournis par la procédure Forward.

4.2 Résultats

Voici les résultats obtenus avec la reconnaissance par HMM :

Signal à reconnaître	taux de reconnaissance HMM
0	100 %
1	96 %
2	99 %
3	100 %
4	100 %
5	100 %
6	100 %
7	100 %
8	98 %
9	100 %
+	100 %
-	92 %
×	96 %
/	100 %
=	100 %

Tab. 4.1: *Résultats monolocuteur.*

Signal à reconnaître	taux de reconnaissance HMM
0	75 %
1	55 %
2	59 %
3	72 %
4	76 %
5	67 %
6	82 %
7	76 %
8	69 %
9	57 %
+	81 %
-	82 %
×	54 %
/	73 %
=	75 %

Tab. 4.2: *Résultats multilocuteur.*

Comparé à la reconnaissance par DTW, les HMM supportent mieux les problèmes liés à la capture du signal (problème d'intensité avec laquelle est prononcé le mot, distance par rapport au microphone, ...). Les résultats sont en général bien meilleurs, cependant lorsqu'un problème de reconnaissance apparaît pour un certain mot, les résultats sont très mauvais : soit le mot est bien reconnu soit il ne l'est jamais, dans quel cas il faut refaire la base d'apprentissage. Ces problèmes arrivent le plus fréquemment quand deux mots ont une prononciation très proche : un et moins, étoile et égal, ces quatre mots sont ceux qui présente le plus de problèmes, les autres sont en général très bien reconnus. En mode monolocuteur, les résultats sont excellents.

Pour ce qui est de l'aspect multi-locuteur les résultats sont moins satisfaisants, de plus pour des voix vraiment différentes de celles utilisées pour la base d'apprentissage, les résultats sont très bas. C'est pourquoi il faut diversifier la base d'apprentissage au mieux en terme de différence de voix pour les signaux de référence.

4.3 Parole continue

La reconnaissance en parole continue n'a pas été entièrement implémentée, des problèmes subsistant au niveau de la modélisation du système.

Le principe étudié consiste à relier les HMM utilisés pour reconnaître les mots isolés entre eux. On insère en premier lieu deux états : une source et un collecteur. La source est alors relié à chacun des états initiaux des HMM précédents, puis chacun des états finaux de ces HMM sont reliés au collecteur, qui lui-même est relié à la source.

Le problème que nous rencontrons avec l'implémentation de cette méthode repose sur le fait que les transitions entrantes et sortantes de la source et du collecteur sont des transitions "nulles". Cela signifie qu'il n'y a pas d'émission de symbole lorsque ces transitions sont empruntées, on a alors une deuxième matrice qui permet de garder ces transitions A'. Cependant le système doit répondre à la contrainte suivante :

$$\sum A_{ij} + A'_{ij} = 1$$

Les algorithmes Forward et Viterbi nécéssite également quelques changements pour prendre en compte ces transitions nulles. Le fait est que l'application de ces transitions nulles ne nous donnent pas de résultats.

4.4 Comparaison à la DTW

On a vu que les résultats obtenus sont meilleurs avec les HMM, surtout dans le cas où la capture audio est effectuée dans des conditions différentes : bruit de fond, distance au microphone, ... Cependant si le signal était capturé dans exactement les mêmes conditions, il est probable que les résultats seraient équivalents entre les 2 méthodes.

Cependant en terme de temps de calcul, il est bien plus long pour les HMM, mais reste tout de même raisonnable. Pour bien faire, il faudrait utiliser un seul HMM pour tous les mots, et utiliser l'algorithme de Viterbi afin de trouver la séquence la plus probable empruntée par un signal à reconnaître dans le HMM, ce qui nous permettrait de déduire quel est de façon la plus probable le mot reconnu. Dans ce cas, il est évident que le temps de calcul serait inférieur même à la DTW.

Annexes

BIBLIOGRAPHIE

[Cha02] Maurice Charbit. Reconnaissance de mots isolés. 2002.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. 1989.

[XH01] Hsiao-Wuen Hon Xuedong Huang, Alex Acero. *Spoken Language Processing*. Prentice Hall, 2001.

GLOSSAIRE

DCT Discrete Cosinus Transform.

DTW Dynamic Time Warping, algorithme de programmation dynamique classique qui permet de trouver un alignement temporel de coût minimal.

FFT Fast Fourrier Transform.

HMM Hidden Markov Model.

LPC Linear Predictive Coding.

MFCC Mel-Frequency Cepstrum Coefficients.

TABLE DES FIGURES

3.1 Alignement des signaux avec l'algorithme de DTW 4

4.1 Topologie du HMM utilisé 9

LISTE DES TABLEAUX

3.1 Taux de reconnaissance LPC-MFCC. 6

4.1 Résultats monolocuteur. 10
4.2 Résultats multilocuteur. 10