INSA | INSTITUT NATIONAL DES SCIENCES APPLIQUÉES Rouen

asi
Architecture des Systèmes d'Information

PRÉSENTATION DE PROJET EDTS RECONNAISSANCE DE PAROLE EN UTILISANT HMM BASÉ SUR CMU SPHINX

15 décembre 2015

Zhaolun Wang et Zenan Xu

Institut National des Sciences Appliquées

Sommaire

Introduction

Objectif

- Reconnaissance de parole française en utilisant HMM basé sur CMU Sphinx

Les Modèle de Markov Caché ou Hidden Markov Model

Définition

Un modèle de Markov caché (MMC) est un modèle statistique permettant de représenter un processus de Markov dont l'état est non observable.

Processus markov(Chaîne de Markov)

En mathématiques, un processus de Markov est un processus stochastique possédant la propriété de Markov. Dans un tel processus, toute l'information utile pour la prédiction du futur est contenue dans l'état présent du processus.

Applications de HMM

- Reconnaissance de la parole.
- Traitement automatique du langage naturel.
- Reconnaissance de l'écriture manuscrite.
- Bio-informatique, notamment pour la prédiction de gènes.

Un exemple illustratif de HMM

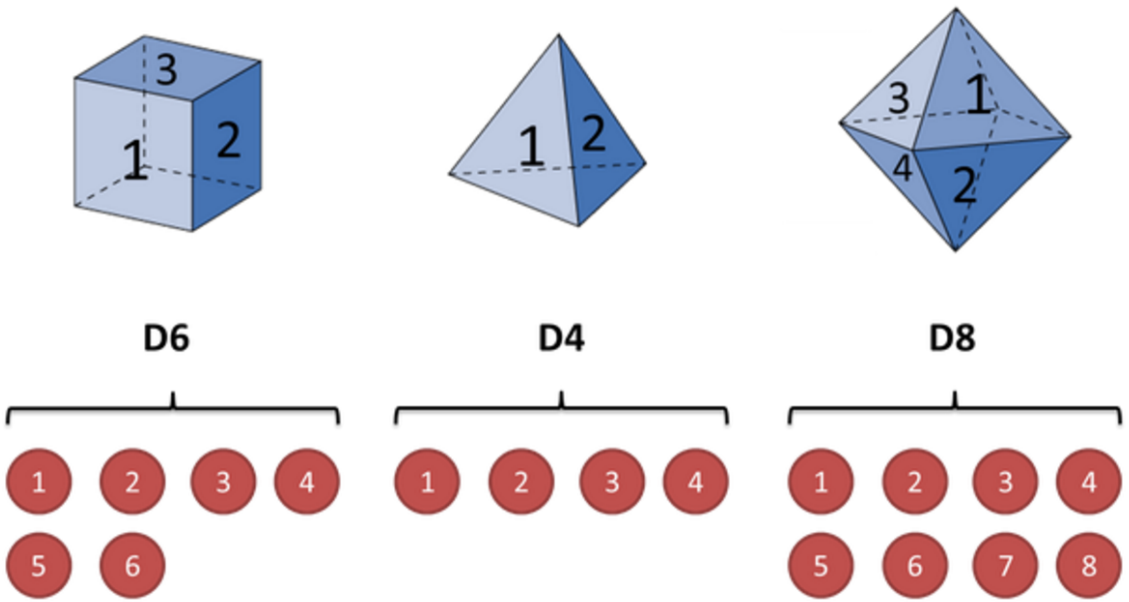

Figure 1: Un exemple de HMM

Un exemple illustratif de HMM

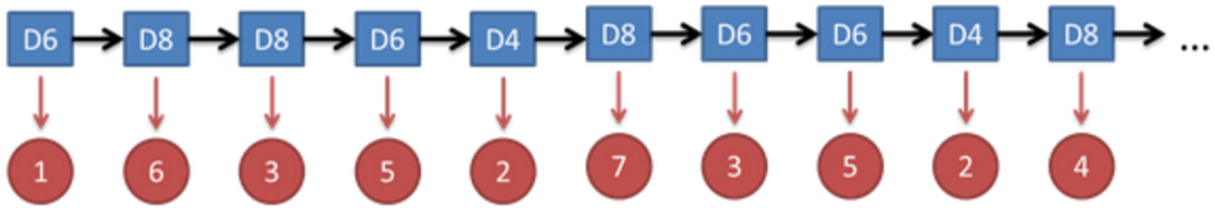

Figure 2: Un exemple de HMM

- Les états observés : 1 6 3 5 2 7 3 5 2 4
- Les états cachés : D6 D8 D8 D6 D4 D8 D6 D4 D8

L'Outil CMU Sphinx

Les CMU Sphinx comprend entre autres les outils suivants :

- Sphinx 2 : est un système de reconnaissance de la parole à grande vitesse. Il est habituellement employé dans des systèmes de dialogue et des système d'étude de prononciation.
- Sphinx 3 : est un système de reconnaissance de la parole légèrement plus lent mais plus précis.
- Sphinx 4 : Une réécriture complète du Sphinx en Java. Il offre à la fois la précision et la rapidité.
- Sphinxtrain : Une suite d'outils qui permet de créer le modèle acoustique .
- CMU-Cambridge Language Modeling Toolkit : Une suite d'outils qui permet de créer le modèle de langage.
- Sphinx Knowledge Base Tool : Un outil qui permet de créer le modèle de mots qui adapte son modèle de langage.

Les Modèle de Markov Caché en speech to text

Nous considérons un signal acoustique S, le principe de la reconnaissance peut être expliqué comme le calcul de la probabilité P(W|S) avec W qui est une suite de mots (ou phrase) correspond au signal acoustique S, et de déterminer la suite de mots qui peut maximiser cette probabilité. En utilisant la formule de Bayes, P(W|S) peut s'écrire :

$$\mathbf{P(W|S) = P(W).P(S|W)/P(S)}$$

- P(W) est la probabilité a priori de la suite de mots W
- P(S|W) est la probabilité de signal acoustique S, étant donné la suite de mots W
- P(S) est la probabilité du signal acoustique
- P(S|W) est nommé Modèle Acoustique, et P(W) est nommé Modèle de Langage

Principe de la reconnaissance de la parole

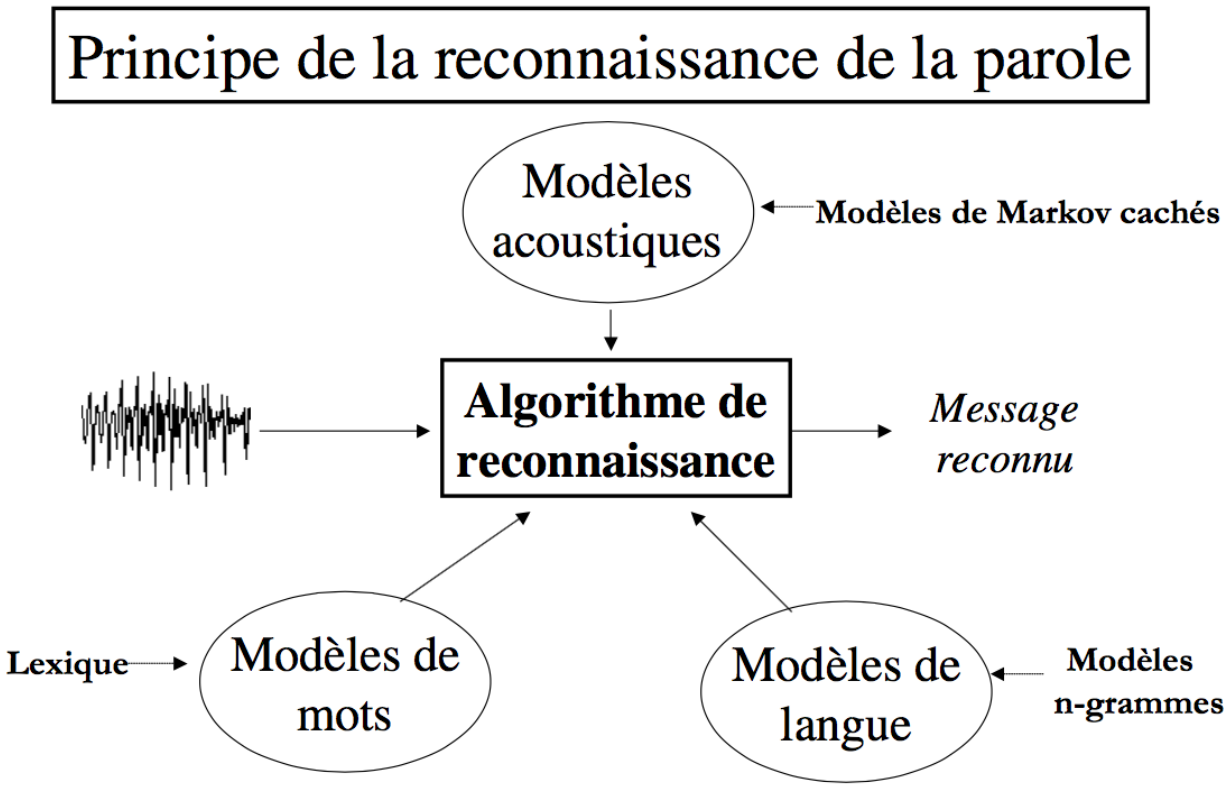

Figure 3: Principe de la reconnaissance de la parole

Modèles acoustiques

Traitement acoustique : extraction de paramètres

- MFCC(Mel Frequency Cepstral Coeficients)
- LPCC(Linear Predictive Cepstral Coeficients)
- PLP(Perceptuqal Linear Predictive analysis)

MFCC

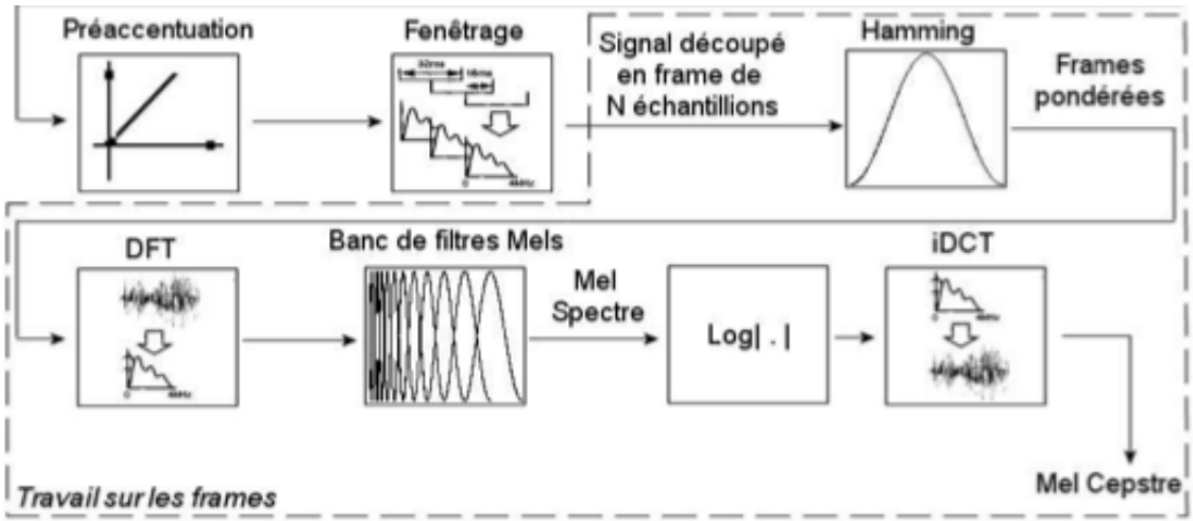

Figure 4: Schéma de MFCC

MFCC

- Préaccentuation du signal
- Découpage du signal en fenêtre
- Application d'une fenêtre de Hamming
- Création du banc de filtres
- Conversion en échelle de mel
- Application d'une DCT (Discrete Cosinus Transform) sur les portions

Décodage acoustique et apprentissage

- Chaînes et modèles de Markov cachés
- Critère du maximum de vraisemblance
- Critère de Viterbi

Adaptation

Méthode MLLR

La méthode MLLR qui signifie Maximum Likelihood Linear Regression permet d'adapter des modèles acoustiques par régression linéaire.

Méthode MAP

La méthode MAP pour estimation du Maximum à postériori est une méthode bayésienne. Elle permet de modifier les paramètres d'un modèle générique pour se rapprocher des données de test.

Création du Modèle acoustique en utilisant Sphinxtrain

Les données d'entrées sont composés, entre autre :

- d'un ensemble de fichiers acoustiques(corpus).
- d'un fichier de transcription qui contient l'ensemble de mots prononcés pour chaque enregistrement(fichier acoustique).
- d'un fichier qui définie la liste des phonèmes utilisées.

Modèle de langage

Un modèle de langage a pour but d'estimer la probabilité a priori de toutes les séquences de mots qu'il est possible de construire à partir du lexique. Pour ce faire, il peut s'appuyer sur différentes sources d'informations, comme par exemple des règles syntaxiques ou sémantiques, ou encore des statistiques issues de gros volumes de données. On se concentre ici sur les modèles de langages statistiques.

Modèle des N-Grammes

Principe : L'idée à la base des modèles de langage n-gramme est que la probabilité d'apparition d'un mot peut être estimée à partir des n-1 mots le précédant. On peut ainsi faire une approximation sur le contexte utilisé pour le calcul des probabilités conditionnelles de la formule suivante :

$$P(W_{1\ldots k}) = P(W_1).P(W_2)|P(W_1). \cdots .P(W_{N-1})|P(W_{1..N-2}). \prod_{l=N}^{k} P(W_l|W_{l-(N-1)..l-1})$$

Création de Modèle de langage en utilisant CMUCLMTK

La création d'un Modèle de langage statistique peut se résumer en trois étapes :

- Collecter des textes
- Transformer les textes en corpus
- Transformer le corpus en une distribution de probabilités

La dictonnaire phonétique

```
abbey aa bb ei
abbot aa bb oo tt
abbott aa bb au tt
abbé aa bb ei
abc aa bb ei ss ei
abchac aa bb ch aa kk
abcès aa bb ss ai
abcès(2) aa bb ss ai zz
abdallah aa bb dd aa ll aa
abdel aa bb dd ai ll
abdelaati aa bb dd ei ll aa aa tt ii
abdelatif aa bb dd ei ll aa tt ii ff
abdelaziz aa bb dd ai ll aa zz ii zz
abdelaziz(2) aa bb dd ei ll aa zz ii zz
abdelazize aa bb dd ei ll aa zz ii zz
abdelghani aa bb dd ei ll gg aa nn ii
abdelhadi aa bb dd ei ll aa dd ii
abdelhafid aa bb dd ei ll aa ff ii dd
abdelhalim aa bb dd ei ll aa ll ii mm
abdelhamid aa bb dd ai ll aa mm ii dd
abdelhamid(2) aa bb dd ei ll aa mm ii dd
abdelkader aa bb dd ai ll kk aa dd ai rr
abdelkebir aa bb dd ei ll kk ei bb ii rr
abdelkrim aa bb dd ai ll kk rr ii mm
abdellah aa bb dd ai ll aa
```

Génération de dictionaire

Il existe plusieurs outils qui nous permettent d'étendre une dictionnaire existante ou de générer une nouvelle dictionnaire :

- Phonetisaurus/sequitur-g2p ;
- espeak for C ;
- FreeTTS/OpenMary Java TTS ;
- etc.

Réalisation du projet

Analyser de résultat

Plusieurs raisons les plus possibles qu'on n'a pas très bien réussit :

- Testeur ;
- Modèle et dictionnaire ;
- Temps de recording ;
- etc.

Amélioration

Amélioration

Les possibilités d'améliorer notre programme :

- Générer notre propre modèle de language ;
- Générer notre propre dictionnaire ;
- Enlever le créneau vide dans le .wav ;
- Utiliser stream directement sans avoir .wav enregistré.

Conclusion