TD – Système simple de reconnaissance de la parole

L'objectif de ce TD est de développer un système simple de reconnaissance de mots isolés en se basant sur une paramétrisation du signal de parole à l'aide des coefficients LPC, et d'une reconnaissance par l'algorithme de classification des k plus proches voisins se basant sur la distance élastique entre signaux paramétrisés. Ce système sera développé en langage Python.

1. Présentation des données fournies

Les données disponibles (que vous pourrez compléter avec vos propres enregistrements) consistent en des enregistrements audio de personnes prononçant des chiffres (de 0 à 9) :
- 4 personnes
- 2000 enregistrements au total (50 enregistrement de chaque chiffre par personne)
- prononciation anglaise

Les noms des fichiers sont sous la forme 'chiffre_nom_index.wav', par exemple '1_theo_47.wav'.

Pour charger en mémoire les données audio sous forme d'un vecteur, à partir d'un fichier *wav*, il est possible d'utiliser la fonction *read* du module Python *scipy.io.wavfile:*
(https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.read.html)

```
import scipy.io.wavfile as wav
Fe, s = wav.read('1_theo_47.wav')
```

Dans ce cas, *s* est un vecteur numpy contenant les valeurs des échantillons audio et *Fe* est la fréquence d'échantillonnage.

2. Paramétrisation des données audio à partir des coefficients LPC

Afin de représenter les signaux audio, une paramétrisation sera réalisée, consistant à associer à chaque trame successive du signal, un vecteur des coefficients LPC (cf cours).

La méthode considérée ici pour le calcul des coefficients LPC repose sur l'équation de Yule-Walker exprimant la relation entre les coefficients LPC et les covariances du signal considéré :

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(N) \\ R(-1) & R(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & R(1) \\ R(-N) & \cdots & R(-1) & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Dans cette équation, les a_i sont les coefficients LPC d'un modèle d'ordre N, σ^2 est la puissance de l'erreur de prédiction et $R(i)$ est la fonction d'autocovariance définie par $R(i) = E(s(t+i)s(t))$.
Comme le signal *s* est supposé stationnaire sur la durée de trame considérée (20 à 30 ms), $R(i) = R(-i)$ et donc la matrice de covariance est symétrique.

3. Calcul de la distance entre deux signaux paramétrisés

Afin de permettre la reconnaissance d'un mot, une fois le signal audio correspondant paramétrisé, il est nécessaire de procéder à une phase de classification.

La technique de classification considérée dans le cadre de ce TD, est une classification par recherche des formes les plus similaires (algorithme des k plus proches voisins). Pour cela, la définition d'une distance appropriée doit être réalisée. Afin de prendre en compte la variabilité de prononciation des mots, nous allons considérer une distance élastique, fondée sur la programmation dynamique. Le principe a été présenté en cours, et consiste à mettre en correspondance les échelles temporelles des formes à reconnaître à l'aide de transformations non linéaires.

4. Classification par l'algorithme des k plus proches voisins

La classification se fera par l'algorithme des k plus proches voisins dont le principe est le suivant :
- Considérer une base de référence (ensemble de données dont on connait la classe)
- Pour une nouvelle donnée à classer, déterminer les k exemplaires de la base de référence les plus proches au sens d'une certaine distance (distance élastique dans notre cas) et affecter à cette donnée la classe majoritaire parmi ses k voisins.

5. Travail à réaliser

Le travail à réaliser consiste à écrire un programme Python permettant la réalisation d'un système simple de reconnaissance de mots isolés en se basant sur une paramétrisation du signal de parole à l'aide des coefficients LPC, et d'une reconnaissance par l'algorithme des k plus proches voisins, basé sur la distance élastique entre signaux paramétrisés.

Le programme doit donc :
- lire les fichiers audio
- afficher leur forme d'onde
- calculer leur matrice des coefficients LPC
- calculer et afficher la matrice des distances entre les signaux audio
- réaliser la classification par l'algorithme des k plus proches voisins

6. Compte-rendu

Ce travail peut être réalisé par monôme ou par binôme, et un compte-rendu numérique devra être produit, soit sous forme d'un notebook Python, soit sous forme d'une archive « .zip » contenant le code Python et le rapport au format pdf. Le compte-rendu devra contenir notamment les explications concernant le principe des fonctions que vous avez programmées ainsi que leur code commenté. Les expérimentations que vous aurez réalisées devront être décrites et les résultats également commentés. Ce travail devra être déposée sur le site « moodle » dans la rubrique « Comptes Rendus » -> « Devoir : reconnaissance de la parole » pour le mercredi 4 janvier 2023.