

TP 1 et 2 - Reconnaissance de commandes audio par programmation dynamique

M1 informatique UE MCS 2016-2017

Résumé

Ces deux séances de TP vont vous permettre de tester l'algorithme de programmation dynamique vu en TD2 puis de réaliser la mise en œuvre d'un système de reconnaissance audio de mots isolés (constituant des commandes pour les drones).

1 Implémentation de l'algorithme de programmation dynamique sur les exemples vus en TD

1.1 Matrice de coûts

Écrivez un programme sous Matlab (ou octave) `CalculDistanceDTW.m` qui implémente le calcul et l'affichage de la matrice des coûts. Afin de rendre facilement modifiable le calcul des distances, créez un sous-programme par fonction de coût à implémenter. Utiliser par exemple `feval(distance,sequence1,sequence2,i,j)` pour calculer la distance.

```
function [D,g]=CalculDistanceDTW(sequence1,sequence2,distance)
(...)
```

```
function d=distance_num(sequence1,sequence2,i,j)
d=abs(sequence2(j)-sequence1(i));
end
```

```
function d=distance_adn(sequence1,sequence2,i,j)
if sequence1(i)==sequence2(j)
d=0;
else
d=1;
end
end
```

```
function d=distance_sons(sequence1,sequence2,i,j)
(...)
end
```

```
function d=distance_vect(sequence1,sequence2,i,j)
(...)
end
```

1.2 Contraintes globales

Ajouter la prise en compte de contraintes globales (ie : non calcul lorsque les cases sont trop éloignées de la diagonale). Permettez la modification de ce paramètre.

1.3 Conservation des chemins

Ajoutez à `CalculDistanceDTW.m` la possibilité de stocker les chemins possibles, afin de pouvoir déterminer le chemin optimal. Créez une fonction `AfficheChemins.m` pour générer une sortie graphique pour pouvoir représenter ce chemin optimal (vous pouvez sur-imprimer le meilleur chemin sur la l’affichage de la matrice `g`).

1.4 Application aux exercices

Testez vos programmes sur les exercices vu en TD. Que se passe-t-il si les contraintes locales sont changées ? A partir de quelle position les contraintes globales ne changent pas les résultats ?

2 Système de reconnaissance audio de mots de commande

2.1 Système de base

Vous trouverez sur l’espace partagé des enregistrements audio de mots de commandes. Utilisez les mots du locuteur M01 comme référence et testez la reconnaissance par programmation dynamique des fichiers du locuteur M04.

Le programme `parametrisation.m` vous permettra de transformer le fichier audio en matrice de paramètres et d’effectuer un débruitage. La matrice en sortie est composé d’autant de vecteurs colonnnes que de trames d’analyses. Le nombre de lignes correspond à la dimension du vecteur représentatif : ici 12.

Calculer le score de reconnaissance : nombre de fichiers bien reconnus sur nombre de fichiers testés. Afficher la matrice de confusion du système (en ligne les références, en colonne les sorties du système).

2.2 Système multi-locuteurs

Vous disposez maintenant de plusieurs locuteurs de référence (M01, M02, M03, F01, F02, F03). Combinez ces références afin d’améliorer le taux de reconnaissance.

2.3 Voix personnelle

Enregistrez votre voix avec l’application Android et testez-là.

3 Implémentation C++

Vous allez devoir implémenter les algorithmes en C++ en utilisant l’embryon de programme permettant de faire de la commande vocale sur smartphone.