TP6. Dynamic Time Warping (DTW)
(Alignement dynamique dans le temps)

Une des difficultés rencontrées dans la reconnaissance de la parole est que, malgré l'inclusion de (plus ou moins) les mêmes sons dans le même ordre dans les différentes énonciations des mêmes mots, il n'y aura pas synchronisations/correspondances exactes en termes de temps chaque partie dans le mot. Et donc, les efforts pour reconnaître les mots en les jumelant avec des motifs conduiront à des résultats inexacts s'il n'y a aucun alignement dans le temps. Bien qu'il ait été en grande partie remplacé par des modèles de Markov cachés (HMM) la technique de programmation dynamique DTW est utilisée à la reconnaissance vocale pour améliorer les différences temporaires entre les énoncés de tests et modèles (templates). Le principe de cette technique est l'existence de «étapes» dans l'espace (les délais dans les énoncés de test ou des délais dans templates / modèles) et de constater que l'espace dans le chemin qui maximise la correspondance locale entre les trames de temps qui sont soumis à des contraintes par défaut aligné étapes autorisés. "Le coût de similitude 'total trouvés par cet algorithme est une indication de la qualité du match entre énoncés d'essai et des modèles (templates), ces informations peuvent ensuite être utilisées pour choisir le modèle (template) que vous convient le mieux.
L'exemple de code est un simple alignement de DTW de mise en œuvre pour les fichiers sonores pour évaluer la similarité entre les sons et d'aligner le tempo d'un fichier audio pour correspondre à celui d'une référence, par exemple pour synchroniser deux prononciations (paroles) du même mot.
Les sous-programmes sont prévus:

simmx.m - facilite le calcul de la matrice totale correspondante locale respectivement à calculer la distance entre toute paire de cadres (vecteurs acoustiques) du signal de test et le signal modèle.
dp.m - implémente l'algorithme de programmation dynamique permettant simplement trois étapes (contraintes locales) -(1,1), (0,1) et (1,0)- avec des poids égaux.
dp2.m - variante alternative expérimentale qui permet à cinq étapes (contraintes locales) - (1,1), (0,1), (1,0), (1,2) et (2,1) - avec des poids différents pour indiquer «chemins escarpés», mais sans limitation rigide en termes de régions dans lesquelles trouver des correspondances. Syntaxe comme dans dp.m
dpfast.m - version rapide en utilisant une version MEX (dpcore) pour exécuter la boucle interne non guidé, jusqu'à 200 fois plus vite!
dpcore.c - code source C pour routine MEX qui améliore la routine dpfast.m comme vitesse (help MEX).

MISE EN ŒUVRE

1. Etudier le principe de la méthode DTW du cours 9.
2. Analyser l'application dtw.m et les routines utilisées.
3. Changez l'application de manière à introduire les fichiers (* .wav) a la mise en œuvre de l'application.
4. Quels sont les paramètres utilisés pour les vecteurs acoustiques?
5. Testez les trois variantes de l'algorithme de programmation dynamique (dp.m, dp2.m, dpfast.m) pour différents fichiers correspondant aux mêmes énoncés (ex. 929 _ *. wav). Comparez les distances selon les temps des énoncés. Comparer les temps d'exécution des applications (dp.m, dp2.m, dpfast.m) en fonction des énoncés d'entrée alignés.
6. Répétez les expériences pour différentes mots (ex Hall.wav; Girl.wav, ...)
7. Prenez des captures d'application pour des situations différentes et inclurez-les en relation avec résultats précédents et des commentaires correspondants.
8. Changez l'application de manière à extraire des coefficients de prédiction (LPC 12) comme des vecteurs acoustiques. (Optionnel).

9. Envoyez le rapport.