Brique PAMU, module PAROL / Projet

Olivier Cappé

Juin 2010

Table des matières

1 Données

Le fichier sig.mat contient plusieurs matrices de cellules (*cell array*) à charger dans MATLAB (version 5 ou supérieure).

Les matrices de cellules permettent de rassembler dans une même structure des données de types différents, en l'occurrence dans l'exemple qui nous intéresse des vecteurs de taille variable. Contrairement aux matrices usuelles l'indexation d'une matrice de cellules se fait à l'aides des accolades. Par exemple

```
A{1,1} = [1 2; 3 4];
A{1,2} = 7
```

créée une matrice de cellules `A` contenant deux cellules, la première contenant une matrice 2×2 et la seconde un scalaire.

Le fichier `sig.mat` contient une première matrice de cellules 10×12 nommée `SIG` qui correspond à 12 répétitions des chiffres de 0 à 9 par le même locuteur, échantillonnées à 8 kHz. Ainsi `SIG{1,3}` est le signal correspondant à la troisième répétition du mot "0" et `SIG{9,1}` celui correspondant à la première répétition du mot "8". Chaque signal est un vecteur colonne (dont la longueur est variable) qui peut être écouté grâce à la commande `soundsc`. Les quatre premières répétitions de chaque mot sont représentées sur la figure 1.

Une seconde matrice de cellules, nommée `SIG2`, contient 6 répétitions des chiffres de 0 à 9 par un second locuteur. Enfin, `SIG_MIXED` contient 6 répétitions des chiffres par le premier locuteur (dans `SIG_MIXED{:,1:6}`) puis par le second (dans `SIG_MIXED{:,7:12}`).

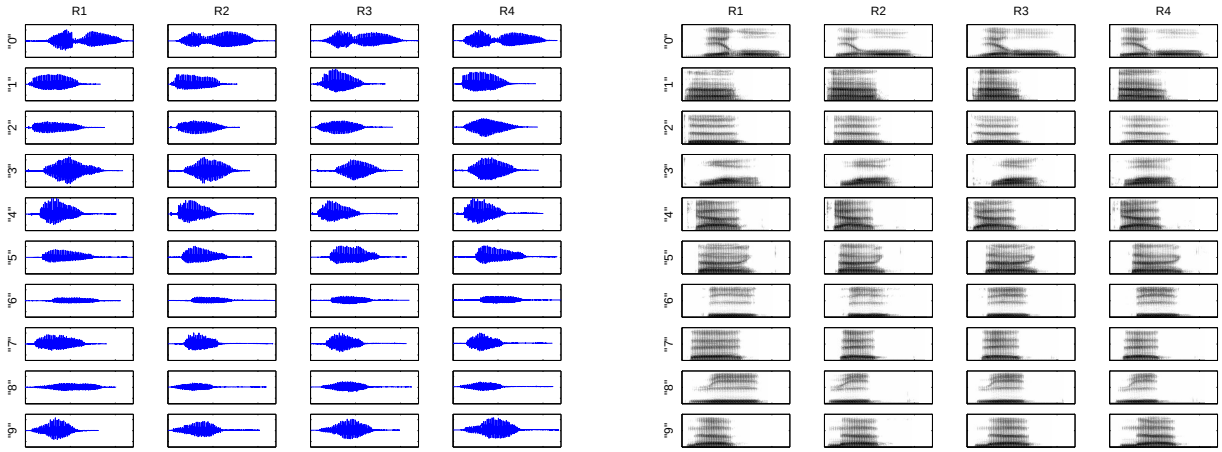


FIG. 1 – Quatre premières répétitions (R1 à R4) de chaque mot. A gauche, les formes d'onde (0.6 secondes de signal) ; A droite, les spectrogrammes.

2 Paramétrisation

Écrire un script MATLAB permettant de calculer la paramétrisation cepstrale associée à chacune des prononciations. En ce qui concerne les paramètres de l'analyse, on se fixera de préférence sur les choix suivants :

Taille des trames 256 échantillons (soit 32 ms)

Décalage des trames 128 échantillons

Type de paramétrisation Cepstre (réel) standard (en échelle fréquentielle linéaire) calculé par FFT

Fenêtre de pondération Hanning

Ordre du cepstre $p = 10$

Pour chaque signal, le résultat de l'analyse est une séquence de paramètres que l'on stockera dans une matrice en respectant la convention suivante

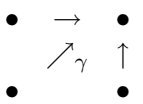
$$\text{nombre de vecteurs cepstraux} \left\{ \underbrace{\begin{bmatrix} X_1 \text{ (1er vecteur, en ligne)} \\ X_2 \text{ (second vecteur)} \\ \vdots \\ X_{l_X} \text{ (dernier vecteur)} \end{bmatrix}}_{p} \right. \tag{1}$$

Notons que le nombre de vecteurs cepstraux obtenus (l_X) dépend de la longueur du signal, il est donc variable. Étant donné que l'on obtient une matrice de ce type par signal, on aura tout intérêt à les stocker dans une matrice de cellules `PARAM` (de telle façon que `PARAM{i,j}` contienne la $j$ème répétition du $i$ème mot).

3 Alignement temporel par programmation dynamique

Écrire une fonction MATLAB réalisant l'alignement temporel par programmation dynamique entre deux séquences de vecteurs stockées avec la convention (1). On utilisera la norme Euclidienne entre les

vecteurs et les contraintes temporelles suivantes



avec une pondération γ que l'on pourra éventuellement choisir différente de 1. Il est indispensable de réaliser cette fonction de manière paramétrée afin de pouvoir la valider sur l'exemple mono-dimensionnel étudié en cours.

La syntaxe de la fonction est la suivante :
`[distance, M, C] = DTW(X, Y, gamma)`

En entrée Les deux séquences de vecteurs $(X_1, \ldots, X_{l_X})$ et $(Y_1, \ldots, Y_{l_Y})$ stockées dans deux matrices, ainsi que le paramètre γ.

Calcul de la matrice de similarité Étant donné les deux séquences de vecteurs, calculer la matrice $D(l_X \times l_Y)$ telle que $D(i,j) = \|X_i - Y_j\|$.

Initialisation de la matrice des distances cumulées Calculer le contenu de la première ligne et de la première colonne de la matrice des distance cumulées $C(l_X \times l_Y)$ telle que $C(i,j)$ contiennent la distance cumulée le long du chemin de distance cumulée minimale rejoignant le noeud (i,j) au noeud initial $(1,1)$.

Calcul de la matrice des distances cumulées Pour i allant de 2 à $\min(l_X, l_Y)$, calcul de $C(i,i)$ puis $C(i+1,i), \ldots C(lx,i)$ et $C(i,i+1), C(i,i+2), \ldots C(i,ly)$ (on rappellera ce qui justifie cette manière de procéder). Pour stocker le prédécesseur le long du chemin optimal menant en (i,j), on pourra utiliser un tableau tridimensionnel $B(l_X \times l_Y \times 2)$ puisque que le prédécesseur est un noeud du réseau défini par ses deux coordonnées (on pourra alternativement utiliser un "code" représentant les trois prédécesseurs possibles).

Bactracking A partir du noeud (l_X, l_Y), reconstitution du chemin optimal (en partant de la fin).

En sortie La distance cumulée le long du chemin de moindre coût (`distance`) et le chemin d'alignement (`M`), par exemple sous la forme d'une matrice M à deux colonnes telles que $X_{M(i,1)}$ soit le vecteur mis en correspondance avec $Y_{M(i,2)}$ (attention le nombre de lignes de cette matrice qui correspond à la longueur du chemin optimal est variable). La fonction pourra également renvoyer la matrice `C` des distances cumulées.

4 Application aux données de parole

On réalisera dans un premier temps l'alignement d'une séquence de référence (par exemple celle correspondant à la première répétition du premier mot) par rapport à toutes les autres, avec les données de `SIG`.

1. Les performances de discrimination sont elles satisfaisantes ?
2. Quelle est l'influence du choix de γ (pondération du trajet diagonale) sur les chemins d'alignements (on choisira des cas caractéristiques pour juger de la pertinence des chemins d'alignement obtenus) ? Ce paramètre a-t-il une influence importante sur les scores de similarité ?
3. Quelle hypothèse implicite fait on lorsque l'on utilise la distance Euclidienne pour mesurer la proximité des vecteurs cepstraux ? Cette hypothèse vous paraît elle acceptable au vu des différences entre séquences alignées correspondant à différentes répétitions d'un même mot ? Proposer une méthode permettant d'estimer à partir des données alignées, une pondération adéquate pour la comparaison des vecteurs cepstraux. Cette pondération améliore-t-elle les performances de discrimination ?

5 Evaluation de la reconnaissance

La fonction `evaluation_recognition` donnée permet d'évaluer de façon systématique les performances de discrimination en utilisant une validation croisée. La syntaxe de cette fonction est la suivante :

```
[confusion, accuracy, D] = evaluation_recognition(x, DTW_cback, gamma, protocol);
```

Le rôle des différents paramètres et valeurs de sortie est donné ici :

`x` est la matrice de cellules des paramètrisations calculée par la fonction réalisée dans la partie 2.

`DTW_cback` est un pointeur vers la fonction réalisant l'alignement. Par exemple, `@DTW` si votre fonction se nomme `DTW`.

`gamma` est le paramètre γ de l'alignement. Ce paramètre sera passé en troisième argument de `DTW_cback`.

`protocol` est une valeur entière comprise entre 1 et 3 indiquant le protocole d'évaluation à utiliser.

`confusion` est la matrice de confusion, de taille 10×10.

`accuracy` est le taux de reconnaissance.

`D` est la matrice des distances entre chacun des éléments de la base de données, de taille $10N \times 10N$, où N est le nombre de répétitions.

Les trois protocoles d'évaluation proposés sont :

Protocole 1 La base de données est divisée aléatoirement en 3 sous-ensembles. Deux de ces sous-ensembles (2/3 des répétitions) sont utilisés comme base d'apprentissage, et le sous-ensemble restant (1/3 des répétitions) est utilisé comme base de test. La procédure est répétée par rotation des ensembles de test et d'apprentissage.

Protocole 2 Une base contenant une prononciation de chaque chiffre est utilisée comme base d'apprentissage, les répétitions restantes constituant une base de test. Cette procédure vise à évaluer les capacités de généralisation du système de reconnaissance.

Protocole 3 La première moitié de la base est utilisée comme base d'apprentissage, et la seconde comme base de test. La procédure est répétée en inversant le rôle des deux sous-ensembles. Cette procédure vise à évaluer les capacités de généralisation sur une base contenant des données issues de deux locuteurs.

1. Evaluez la reconnaissance à l'aide du protocole 1 sur les données `SIG` et `SIG2`. Quelles sont les confusions ? Quelle valeur du paramètres γ offre les meilleurs résultats ?
2. Evaluez la capacité de généralisation à l'aide du protocole 2 sur les données `SIG` et `SIG2`. Que remarquez-vous ?
3. Evaluez la capacité de généralisation à un autre locuteur à l'aide du protocole 3 sur les données `SIG_MIXED`. Expliquez les résultats.
4. Quelles limitations des approches de la reconnaissance vocale par alignement ont été mises en évidence ? Proposez quelques solutions possibles à ces problèmes.