TP 2 : Kaldi, modèles acoustiques et alignement

École thématique CNRS – Big Data Speech

Roscoff, Juillet 2018

Après avoir préparé nos données pour les rendre utilisables par Kaldi, puis avoir construit des coefficients MFCC, considérés comme l'information utile dans le signal audio pour la reconnaissance de la parole, nous allons maintenant :

1. apprendre un modèle acoustique monophone indépendant du locuteur (sans adaptation) ;
2. l'utiliser pour construire un alignement phonème/signal sur des enregistrements pour lesquels nous disposons des transcriptions manuelles.

1 Données lexicales et phonétiques

Afin de pouvoir réaliser ces alignements (qui sont de toute manières nécessaires pour apprendre un modèle acoustique), nous avons besoin de données lexicales (liste de mots) et phonétique (liste de phonèmes, ainsi que l'ensemble des représentations phonétiques de chacun des mot de la liste de mots).

Construire un dictionnaire de prononciation (mots et leur(s) phonétisations(s)) est une tâche qui peut s'avérer complexe. Pour ce TP nous disposons de données lexicales et phonétiques déjà produites, afin de simplifier la tâche.

Récupérez ces données en les copiant dans le répertoire *data/local* que vous allez créer :

```
cd ~/tp_kaldi
mkdir data/local
cp -r db/dict_nosp data/local
```

Comme souvent avec Kaldi, un script permet de vérifier que les données sont valides. Ici il s'agira de *validate_dict_dir.pl*, qui vérifie que toutes les données sont présentes et au bon format. Les fichiers présents contiennent un dictionnaire, la liste des phonèmes, mais aussi d'autres symboles représentant des unités de sons considérés par Kaldi comme des phonèmes particuliers pour traiter les silences et certains sons (inspiration, rire, ...). **Avant de procéder à la validation ci-dessous, parcourez le contenu des différents fichiers du répertoire *data/local/dict_nosp*.**

```
utils/validate_dict_dir.pl data/local/dict_nosp
```

Maintenant que le répertoire *data/local/dict_nosp* a été validé, nous pouvons transformer son contenu en fichiers qui seront directement utilisés par les programmes de Kaldi.

Pour cela, la commande est la suivante :

```
utils/prepare_lang.sh data/local/dict_nosp "<unk>" data/
    local/lang_nosp data/lang_nosp
```

2 Apprentissage d'un modèle acoustique monophone indépendant du locuteur

Le script *train_mono.sh*, qui se trouve dans le répertoire steps permet de réaliser l'apprentissage d'un modèle acoustique. Il nécessite a minima trois arguments :

1. le chemin pour accéder au répertoire contenant les données d'apprentissage (répertoire construit lors du TP1) ;
2. le chemin pour accéder au répertoire contenant le dictionnaire de prononciations et les différentes informations relatives aux phonèmes (voir point précédent) ;
3. le chemin du fichier dans lequel les fichiers générés durant l'exécution seront copiés (nous choisirons *exp/mono0a*).

Exécutez le script *train_mono.sh* avec les bons arguments. 40 itérations sont attendues pour cet apprentissage. **Prenez le temps de consulter le script *train_mono.sh***. Pendant l'apprentissage qui dure une trentaine de minute, vous pouvez travailler sur la section suivante à partir du modèle construit lors de la première itération (note : les modèles se bonifient lorsque le nombre d'itérations augmente).

3 Modèles acoustiques et alignement automatique

Pendant l'apprentissage du modèle acoustique, plusieurs modèles intermédiaires sont créés de manière itérative. Ils seront créés dans le répertoire choisi précédemment, à savoir *exp/mono0a*. Leur nom est de la forme *.mdl, l'étoile étant remplacée par une valeur numérique entière liée au numéro de l'itération durant laquelle le modèle a été construit. Attention, seul le modèle initial et le modèle courant sont conservés. Les autres sont supprimés au fur et à mesure.

Pour comprendre ce qu'est un modèle acoustique, **regardez le contenu du fichier 0.mdl** à l'aide de la commande suivante :

```
gmm-copy --binary=false exp/mono0a/0.mdl  - | less
```

Et la commande suivante permet d'obtenir un résumé des informations importantes qui caractérise un modèle acoustique HMM/GMM :

```
gmm-info  exp/mono0a/0.mdl
```

Pour visualiser le résultat d'un alignement d'un fichier du corpus d'apprentissage, tapez la commande suivante :

```
show-alignments data/lang_nosp/phones.txt exp/mono0a/0.mdl
      "ark:gunzip -c exp/mono0a/ali.1.gz|" | more
```

... et essayez de comprendre ce que représentent les différents arguments de la commande *show-alignments*.

Le résultat de cette commande n'est pas très explicite. Les numéros des états des modèles de Markov qui modélisent les différents phonèmes sont affichés : Kaldi n'a pas besoin d'autre information, Pour faciliter la lecture par un humain les phonèmes trouvés sont explicites.

Mais on peut obtenir quelque chose de plus explicite avec la commande suivante, qui fournit les résultats de l'alignement au format CTM, très utilisé en reconnaissance de la parole. Pour chaque phonème, son instant de départ et sa durée est fournie :

```
ali-to-phones --ctm-output exp/mono0a/0.mdl "ark:gunzip -c
      exp/mono0a/ali.1.gz|" -  | utils/int2sym.pl -f 5 exp
     /mono0a/phones.txt - | more
```

Pour l'instant, nous n'avons examiné que les fichiers du corpus d'apprentissage alignés à l'aide d'un modèle appris sur ce corpus.

Lorsque l'apprentissage de votre modèle monophone est terminé (si c'est vraiment trop long, contactez l'un des intervenants pour utiliser un modèle pré-calculé), **procédez à un alignement phonème/signal de données absentes du corpus d'apprentissage**, c'est-à-dire les données préparées lors du TP1 qui se trouvent dans le répertoire *data/test.orig*.

Pour cela, exécutez la commande suivante :

```
steps/align_si.sh --nj 4 data/test.orig data/lang_nosp exp
     /mono0a exp/mono0a_ali
```

Le répertoire *exp/mono0a_ali* accueillera les fichiers produits par cette commande, qui contiendront les alignements.

Adaptez la commande *ali-to-phones* utilisée plus haut afin d'afficher sous la forme d'un fichier au format CTM le résultat de l'alignement des fichiers de test qui viennent d'être alignés.

Commentaire

Une dernière remarque : dans ce TP nous sommes contenté d'un apprentissage sur un petit nombre de données, avec un jeu de phonèmes réduit à 33 phonèmes. De plus, nous nous sommes arrêtés à des modèles acoustiques monophones qui ne tiennent pas compte du contexte d'un phonème. Nous n'avons

pas non plus appliqué d'adaptation au locuteur et n'avons pas utilisé de DNN (réseau de neurones profonds) à la place des GMM (Modèles de mélanges Gaussiens). Tout cela est possible avec Kaldi qui est à la pointe de la modélisation acoustique.

En utilisant ce que vous avez vu lors de ces deux dernières séances de TP, et en prenant le temps d'examiner le fichier *run.sh* qui constitue une recette, vous serez tout à fait capable d'améliorer vos modèles acoustiques pour obtenir de meilleurs alignements. N'hésitez pas à nous (=Yannick Estève et Laurent Besacier) contacter si vous aviez des questions ou rencontriez un problème avec Kaldi dans vos propres projets.

Annexe

Voici différentes commandes utilisées pour faire ce TP (certains noms de répertoire peuvent être modifiés).

```
#Fin du TP precedent
steps/compute_cmvn_stats.sh data/train.orig/ data/train.orig/log data/train.orig/data
steps/compute_cmvn_stats.sh data/test.orig/ data/test.orig/log data/test.orig/data

#Preparation des donnees lexicales et phonetiques
cp -r db/dict_nosp data/local/
utils/validate_dict_dir.pl data/local/dict_nosp
utils/prepare_lang.sh data/local/dict_nosp "<unk>" data/local/lang_nosp data/lang_nosp

#Apprentissage d'un modele monophone
steps/train_mono.sh  data/train.orig data/lang_nosp/ exp/mono0a

#Quelques extractions d'information
gmm-info  exp/mono0a/0.mdl
gmm-copy --binary=false exp/mono0a/2.mdl  - | less


#Acceder aux alignements sur le corpus d'apprentissage
show-alignments data/lang_nosp/phones.txt exp/mono0a.bak-yannick/final.mdl "ark:gunzip -c exp/mono0a.bak-yannick/ali.1.gz|" | more

ali-to-phones --ctm-output exp/mono0a.bak-yannick/final.mdl "ark:gunzip -c exp/mono0a.bak-yannick/ali.1.gz|" -  | utils/int2sym.pl -f 5 exp/mono0a/phones.txt - | more

ali-to-phones --ctm-output exp/mono0a.bak-yannick/final.mdl "ark:gunzip -c exp/mono0a.bak-yannick/ali.1.gz|" -  | utils/int2sym.pl -f 5 exp/mono0a/phones.txt - | more


#Acceder aux alignements sur le corpus de test
steps/align_si.sh --nj 4 data/test.orig data/lang_nosp exp/mono0a.bak-yannick exp/mono0a.bak-yannick_ali

```

```
29 ali-to-phones --ctm-output exp/mono0a.bak-yannick/final.
       mdl "ark:gunzip -c exp/mono0a.bak-yannick_ali/ali.1.
       gz|" -  | utils/int2sym.pl -f 5 exp/mono0a/phones.txt
        > exp/mono0a.bak-yannick_ali/ali.phon.1.ctm
30
31 for i in exp/mono0a.bak-yannick_ali/ali.*.gz;
32 do ali-to-phones --ctm-output exp/mono0a.bak-yannick/final
       .mdl "ark:gunzip -c $i|" -  | utils/int2sym.pl -f 5
       exp/mono0a.bak-yannick/phones.txt - > ${i%.gz}.ctm
33 done
```