

Automatic Speech Recognition 2011-12: Lab-session sheet (5)

— Continuous speech recognition (part 3 of 3) —

Hiroshi Shimodaira

(Revision : 1.1)

In this lab-session we will

- adapt speaker-independent HMMs that we have trained in the previous sessions to target speaker's speech using MLLR
- investigate the performance of speaker adaptation with varying the number of regression classes.

1 Updating your environment

To make your experimental environment up-to-date, re-run the initialisation command as we did last week:

```
% cd WorkDir1
% /afs/inf.ed.ac.uk/group/teaching/asr/bin/init-t3
```

2 MLLR speaker adaptation

In this session, we focus on adaptation of mean vectors of Gaussian components only, and we will not consider adaptation of variance parameters. We assume supervised adaptation here, meaning that we know the utterances (i.e. phone sequences) of adaptation data for the target speaker.

As you remember, adapting a mean vector, μ , of a Gaussian distribution linearly can be formulated as

$$\begin{aligned}\hat{\mu} &= A\mu + \mathbf{b} = W\xi \\ \xi &= (1, \mu^T)^T \\ W &= (\mathbf{b}, A)\end{aligned}$$

where W is a transformation matrix.

In HTK, the adaptation procedure normally consists of the following steps:

Step 0 Prepare a set of speaker-independent HMMs and its “stats” file that contains state occupation statistics.

Step 1 Generate a regression-class tree.

Step 2 Given a set of adaptation data of a target speaker, estimate speaker's transformation matrix W for each regression class.

Step 3 Given a set of test (evaluation) data of the target speaker, adapt the HMMs to the speaker and carry out recognition.

Speaker adaptation experiments can be done using either monophone models (R1, R2) or tied-state triphone models (R9).

The following sections show examples of adapting the monophone models stored in `models/R2/hmm4`. Thus, you will need to modify script files according to the models you actually use. For the experiments with tied-state triphone models, you need to use different script files from those shown below, i.e. use the versions whose file names *do not* contain “`_monophones`”.

¹/afs/inf.ed.ac.uk/group/teaching/asr/Work/YourLoginName

2.1 Step 0: preparation

Here, we assume that we are going to adapt `models/R2/hmm4`. First of all, make sure that you have those files:

```
models/R2/hmm4/MODELS  ... HMM parameter file
models/R2/hmm4/stats   ... state occupation statistics file
```

If you do not have the “stats” file above², run those commands to create it:

```
% mv models/R2/hmm4/MODELS models/R2/hmm4/MODELS.bak
% ./scripts/train_monophones_realigned_stats -R2 3
```

Make sure you have got the stats file now.

2.2 Step 1: generating regression class tree

Run the following command to create a script file of your own.

```
% cp -p scripts/mlr_mk_reg_classes_monophones scripts/my_ml_r_mk_reg_classes_monophones
```

Edit your script, `scripts/my_ml_r_mk_reg_classes_monophones`, so that the variables, `IN_HMMTYPE` and `in_modelnum`, have the correct model type and model number for adaptation. You could also change the value of `NRegClasses` to specify the number of regression classes (default is 2).

Run the script:

```
% ./scripts/my_ml_r_mk_reg_classes_monophones
```

which will create the following files in `models/R2/hmm4/classes`.

```
global          ... global transform
regtree_2.tree  ... regression class tree
regtree_2.base  ... list of base classes
```

(The digit “2” above represents the number of regression classes.) For details, see HTKBook[Section 3.6.2, 9.1.3, 9.1.4].

2.3 Step 2: estimate transformations

Run the following command to create your own script file.

```
% cp -p scripts/mlr_estimate_xforms_monophones scripts/my_ml_r_estimate_xforms_monophones
```

Edit your script, `scripts/my_ml_r_estimate_xforms_monophones`, so that the variables `IN_HMMTYPE`, `in_modelnum`, and `NRegClasses` are properly defined.

As you will find in the script, the script calls `HERest` twice to estimate transformations using adaptation data set, “dev-01.scp”.

Pass	transformation class	xform-file extension
1st-pass	global class	mlr1
2nd-pass	regression classes	mlr2

The adaptation set, “dev-01.scp”, contains the same speakers as those of the test set, “si_dt5a-div3.scp”, but utterances are mutually exclusive between the two sets.

Now, run your script:

```
% ./scripts/my_ml_r_estimate_xforms_monophones
```

²You will have a stats file already for triphone models (R9), because it is needed for tree clustering. A stats file can be created with the “-s” option for `HERest`.

which will create a directory

```
models/R2/hmm4/xforms_2
```

containing those transformation files for each speaker.

```
c31.mllr1,  c31.mllr2,  
c34.mllr1,  c34.mllr2,  
      :      :  
c49.mllr1,  c49.mllr2
```

2.4 Step 3: adaptation and recognition

Do the following to create your own script.

```
% cp -p scripts/mllr_recognise_monophones scripts/my_mllr_recognise_monophones
```

Edit your script, `scripts/my_mllr_recognise_monophones`, so that the variables `IN_HMMTYPE`, `in_modelnum`, and `NRegClasses` are properly defined.

As you will see in the script, it calls `HVite`, which adapts the HMMs and carries out recognition with the adapted models. Transformation file for the speaker is automatically chosen according to the speech file name, which contains speaker ID (this is done with the “-h” option).

`HVite` is called with a lot of options. Try to understand the meaning of each option with consulting the `HTKbook`.

Now, run the script:

```
% ./scripts/my_mllr_recognise_monophones
```

which might take a while to finish. Make sure if the script (`HVite`) has not produced a warning message of “No tokens survived”. If there was more than one such a message, you would need to change the pruning threshold (`-t`) of `HVite`.

To get recognition accuracy, run the following script:

```
% ./scripts/show_results_small -R2mllr 4
```

3 Further study

There are a lot of issues that were not covered in the lab sessions. For example, those were missing:

- Parameterising speech data (into MFCCs)
- Creating lists of training/testing sets
- Converting phone/word labels into HTK readable form
- Creating word (pronunciation) dictionaries
- Creating language models
- Recognition with a trigram word language model

4 DICE computing environment

4.1 Disk quota

There is a limitation on how much amount of disk space you can use in your home directory. This is called “disk quota”. You would reach the limit of disk quota very easily. The current status can be shown by running the following command³

```
% fs listquota
```

(you need to run the command in a directory of the disk volume that you’d like to check)

The *WorkDir* provided for you in the ASR course is located in a separate disk space from your home directory, meaning it has a different disk quota from that of your home directory.

Although there is no quota limit for each student in the work space, there is a quota limitation on the whole area for the ASR course, which is 25GB in total. Therefore, please try to keep your disk usage less than 1GB if possible. You can check how much disk space you are currently using by this way:

```
% cd WorkDir
% du -hs
```

4.2 AFS

Not only your home directory but also the ASR course directories are under the distributed networked file system called AFS. The AFS is reasonably fast as long as you are in the university network, but it can be very slow if you (i.e. your computer programs) are accessing files from another network outside the campus network. Thus, if you want to work from home, it would be advisable that you use a secure remote login command (i.e. ssh) to log-on a DICE machine and carry out your experiments there so that every computation and file access are done on the machine inside the campus network. (NB: wireless network is slow even if it is inside the campus)

If you find that your programs run much slower than usual even if you are using a DICE machine which nobody except you is using (i.e. no other heavy process are running), please let me know with detailed information about what you were doing on which machine so that I could ask the computer support to look into the situation.

³This assumes you are using a AFS disk volume. For non-AFS disk volumes, you should use the “quota” command instead.