Automatic Speech Recognition 2008-09: Lab-session sheet (4) — Continuous speech recognition (part 2 of 3) —

Hiroshi Shimodaira (*Revision* : 1.1)

In this lab-session you will

- continue part 1
- increase the number of Gaussian mixture components in the pdfs
- see what effect that has on accuracy
- convert the models to tied-mixture models.

1 Updating your environment

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

```
% cd ~/asr/ASR
% ~hshimoda/pub-asr/bin/init-t3
```

2 Increasing the number of mixture components

After you have carried out the embedded training of single-mixture monophone models several times, it's time to increase the number of mixture components of each state. This is done by iterative "mixture splitting" with the HHEd command.

2.1 Copying source models to a new directory

At first, you need to determine the model number to increase the mixture component. If it is "`models/R1/hmm6`" [1], do as follows (here I assume your current directory is `~/asr/ASR`):

```
% mkdir -p models/R2/hmm0
% cp models/R1/hmm6/MODELS models/R2/hmm0
```

As is shown above, we are going to use a new work directory "`models/R2/`" to save new models with multiple mixture components.

2.2 Making an HHEd instruction file

HHEd needs an instruction file[2](save it as `resources/config/mixup.hed`) which contains a single line like:

```
MU 2 {*.state[2-4].mix}
```

where "2" following "MU" is the number of components you want in the Gaussian mixture output densities of the new models.

[1] Further experimental results will depend on which model you've chosen here. It would be interesting to see how different performance you will get depending on the initial model number.

[2] An example is available as "resources/config/mixup2.hed".

2.3 Splitting

The following shows a typical use of HHEd to increase mixtures by splitting.

```
HHEd -C resources/config/config.basic -T 1 \
    -H models/R2/hmm0/MODELS \
    -M models/R2/hmm1 \
    resources/config/mixup.hed \
    model_lists/mono.list
```

Here we assume that `models/R2/hmm0/MODELS` is used as source models, and a new model file is stored in the directory `models/R2/hmm1`. You will need to change those names when you re-run the command to get more number of mixture components.

A sample script file is available as `scripts/mixup`[3]. Copy it to a new file (e.g. `scripts/my-mixup` by typing `cp -p scripts/mixup scripts/my-mixup` on your terminal window), and edit your file to reflect correct model numbers you use.

After you edit the script file, run it once:

```
% ./scripts/my-mixup
```

2.4 Training

Now, retrain the models using `./scripts/train_mm_monophones` script, remembering to give correct model numbers to train. Try 3 or 4 iterations of training.

```
% ./scripts/train_mm_monophones 1 2 3
```

This trains hmm1, followed by hmm2, and hmm3 under `models/R2/`. You can specify as many model numbers as you wish for the command arguments.

It is a good idea that you draw a graph to see how the average log likelihood of the models on training data varies with iterations and the number of mixtures. The average log likelihood is shown as "`average log prob per frame`" in the last part of the output of the script.

2.5 Testing

Test the new models in `models/R2/hmm4` with the following command:

```
% ./scripts/recognise_with_monophone_models -R2 4
```

followed by either of

```
% ./scripts/results -R2 4
  or
% ./scripts/results_summary -R2 4
```

The option `-R2` is essential, as it tells the scripts to use the models under `R2` rather than the default directory `R1`.

Are the new models more accurate than the single-mixture-component models from part 1?

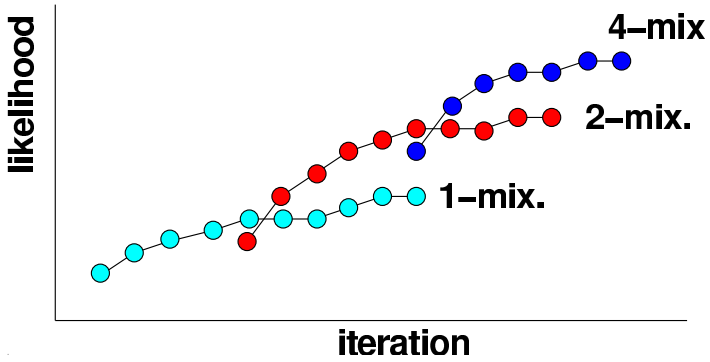
Try increasing the number of mixture components further, e.g. $2, 3, 5, 7, 10, 15, \cdots$, to see how recognition accuracies/speed changes and draw graphs[4]. To this end, follow the previous instructions given in 2.2 and 2.3. Note that you will need to change the `2` in the `resources/config/mixup.hed` file and model numbers each time.

After each increase in the number of components, retrain the models for a few iterations. Always save new models (created either by one iteration of training, or increasing the number of mixture components) in a new `hmm?` directory.

[3] You will find another script, `scripts/mixups`, which is a more sophisticated version of the script, and does not require you to prepare the `mixup.hed` instruction file.

[4] It is a good idea that you compare the graph of likelihood on training data and the graph of recognition accuracies on test data.

Go over your notes and make sure you understand what you just did, what mixtures of Gaussians are, and why increasing the number of mixture components improves accuracy.



Note

You will see that there are enormous number of combinations of parameters which will affect training speed, recognition speed and recognition accuracies. Although it does not take much time for carrying out a single session of experiment with the default parameter set, it can take much longer for certain sets of parameters. In addition, it's not feasible to try all of the possible combinations of the parameters. Plan what sort of experiments are needed and how.

You should start experiments as soon as possible in order not to miss the deadline.

* You may discuss with your colleagues, but you are not allowed to share any programs or outcomes of experiments.

3 Tied Mixture Monophone Models

After you have trained multiple-mixture monophone models, you will find that recognition performance of your models reaches a peak and does not improve even if you further increase the number of mixture components[5]. You can now convert your current HMMs to those of *tied mixture models*, in which similar mixture components are shared (tied) across all the states of all the models. See HTKBook[§11.3] for details.

3.1 Single-stream tied-mixture monophones

To convert the current models to tied-mixture ones, we will use the following HHEd commands (which can be found in `./resources/config/tied-single-stream.hed`.

```
JO 128 2.0
TI MIX_ {*.state[2-4].stream[1].mix}
HK TIEDHS
```

The `./scripts/mk_tm_monophones` script calls HHEd with the edititing commands above.

Before you use the script, you need to decide what model number in the `models/R2/` directory to use for this conversion. Assuming the model number you've chosen is K, run the script with K as its first argument

```
% ./scripts/mk_tm_monophones -i  K
```

This will create an initial set of tied-mixture models in the new directory `models/R3/hmm0`.

Run the following script to train the initial models.

[5]It is worth discussing why this happened. What about the log-likelihood on training data?

```
% ./scripts/train_tm_monophones 0
```

This will take much longer time, e.g. 10-20 minutes. Then repeat the training script two or three times with changing the model number[6]

Recognition experiments can be done in the following manner.

```
% ./scripts/recognise_with_monophone_models -R3 1
% ./scripts/results_summary -R3 1
```

Again, do not forget the `-R3` option, otherwise the default directory `R1` will be used and existing files under `R1` might be overwritten[7] .

3.2 Multiple-stream tied-mixture monophones

Tied-mixture approach works best if the information relating to different sources such as delta coefficients and energy are separated into distinct data streams (see HTKBook[§5.13]), and mixture components are shared in each stream separately.

The following HHEd commands (which can be found in `./resource/config/tied-multi-streams.hed`) will convert given models into an initial set of four-stream tied-mixture monophones

```
SS 4
JO 128 2.0
TI st1 {*.state[2-4].stream[1].mix}
JO 64 2.0
TI st2 {*.state[2-4].stream[2].mix}
JO 64 2.0
TI st3 {*.state[2-4].stream[3].mix}
JO 64 2.0
TI st4 {*.state[2-4].stream[4].mix}
HK TIEDHS
```

Now you need to edit your own `./scripts/mk_tm_monophones` script so that the lines 8–14 look like this:

```
#----- TYPE-A: single-stream tied-mixture models ---
# edcmdfile="resources/config/tied-single-stream.hed"
# HMMTYPE=R3
#
#----- TYPE-B: multi-stream tied-mixture models ---
edcmdfile="resources/config/tied-multi-streams.hed"
HMMTYPE=R3m
```

Then run the script.

```
% ./scripts/mk_tm_monophones -i  K
```

This will create an initial set of tied-mixture models in the new directory `models/R3m/hmm0`.

Run the following script to train the initial models.

```
% ./scripts/train_tm_monophones -R3m 0
```

Repeat the training script two or three times with changing the model number.

Recognition experiments can be done by

```
% ./scripts/recognise_with_monophone_models -R3m  N
% ./scripts/results_summary -R3m  N
```

where N denotes the model number to use for recognition.

[6] You can specify more than one model number like: `% ./scripts/train_tm_monophones 1 2`.

[7] It might be a good idea that you change file permissions of the existing files you created to avoid the overwritten accident.