Automatic Speech Recognition 2016-17: Assignment

Hiroshi Shimodaira and Steve Renals (Ver. 1.0)

1 Outline

In this assignment, you will carry out various experiments of continuous word recognition on the TIMIT speech data set using the Kaldi automatic speech recognition toolkit. The purposes of the assignment is to learn basic techniques for HMM-based speech recognition, and familiarise yourself with Kaldi's commands and shell scripts so that you can write scripts of your own to run experiments.

You should submit a report, and make your ASR systems available in you work directory ("*WorkDir*" hereafter) allocated to you in the course so that the marker can check your work (e.g. code and models) if necessary. Marks will be given to the information provided in the report, and not to the systems developed. Your systems will be considered as the evidence of experiments, and therefore the task without corresponding system will not be marked.

Working in pairs

This assignment is intended to be done in pairs: by working with another student, you can discuss ideas and work things out together. Ideally, try to find a partner with a different skill set / degree programme to your own, although this may not be possible in all cases.

You may discuss any aspects of the assignment with your partner and divide up the tasks however you wish; but we encourage you to collaborate on each part rather than doing a strict division of tasks, as this will enable better learning for both of you.

You may also discuss high-level concepts and general programming questions with others in the class; however you may NOT share code, designs and results of experiments, or coursework reports directly with other groups.

For plagiarism/misconduct, please see:
`http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct`

Note that you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (permitting access only to yourself, or your group in the case of group practicals like this one).

Once you have identified your partner, please indicate the information in a file '`partner`' in your *WorkDir*, which can be done by running the following command in your *WorkDir*.

```
echo Partner's_UUN > partner
```

(NB: Replace *Partner's_UUN* above with the actual UUN of your partner.)

2 Coursework submission

The submission deadline is Wednesday, 8th March 2017 at 16:00. Your coursework submission is complete only if (i) you submit your report and (ii) make all your ASR systems available in *WorkDir*. Detailed instructions are shown below.

There is a policy on late coursework:
`http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests`

Submission of report

Only one student in your pair needs to submit. Make sure both student's UUNs (e.g. s1234567) are clearly shown at the top the report. Please do NOT include your names, only UUNs.

- You report should be either a PDF (.pdf) or MS Word (.doc(x)) document, with a double-column format.
- Submit your report file with the "submit" command on a DICE workstation. The following shows an example of submitting a file, "report.pdf".

  ```
  submit asr 1 report.pdf
  ```

- You should receive an email of acknowledgement from the system as soon as your submission has been received successfully. Keep the message as an evidence of your coursework submission.

Submission of ASR systems

The ASR systems (including scripts and models) you used for the assignment should be found in *WorkDir* of the student who submit the report. After the submission deadline, the directory will be locked so that no further changes are possible.

All the scripts you created/modified should be found in *WorkDir*/`my-local`. You should create the directory by yourself. In each task shown in 3.1 below, "[Scripts]" specifies the file name(s) of script you should put in the directory. For example, for Task 1.1, you should put `exp_mono_t1.sh` – a script you used to explore different numbers of Gaussian mixture components, which should include all necessary steps of training and evaluation. You can call other (your) scripts from the script to avoid clutter. Task 1.1 also requires you to put `run_mono_t1_best.sh` – a script to run a recognition evaluation (decoding, scoring, and displaying WER) with the best model you found. It should not include training steps.

Due to a limited disk space, please keep only the models that gave the best performance in the corresponding task, and delete other models and irrelevant files as soon as you finished the task.

3 Assignment specifications

3.1 Tasks

You will carry out continuous word recognition experiments described below on the TIMIT speech corpus using the Kaldi automatic speech recognition toolkit provided in the course. You should use the same training and test sets used in the labs. Recognition performance should be measured with WER on the test set. To get higher marks, you will need to consider not only WER, but also other measures such as log likelihoods on training/test sets and run time.

It should be noted that, strictly speaking, parameter optimisation should be done on a validation set rather than a test set, but this assignment employs the test set instead. Thus, evaluation experiments in this assignment should be considered as informal ones.

Task-1 Monophone models [50 marks]

1.1 Investigate how the number of Gaussian mixture components influences WER, and find the optimal number that gives the lowest WER. You should present a graph which summarises the result of your experiment. [15 marks]

[Scripts] `exp_mono_t1.sh, run_mono_t1_best.sh`

1.2 Investigate how different acoustic features give different WERs, for which try either PLP or Filter bank (FBank) features. Using the optimal number of Gaussian mixture components you obtained in Task 1.1 above, compare the results. [15 marks]

[Scripts] `exp_mono_t2.sh`

1.3 Investigate how the dynamic features (i.e. delta and delta delta features) of MFCCs influence WER. Note that the sample scripts used in the labs employ dynamic features. [10 marks]

[Scripts] `exp_mono_t3.sh`

1.4 Investigate how CMN/CVN influences WER. [10 marks]

[Scripts] `exp_mono_t4.sh`

Task-2 Tied-state triphone models [25 marks]

2.1 Investigate how the number of clusters and the number of Gaussian mixture components influence WER, and seek the optimal configuration of parameters and models that gives the lowest WER. It is acceptable that you just seek local optimal rather than global optimal. [25 marks]

[Scripts] `exp_tri_t1.sh, run_tri_t1_best.sh`

Task-3 Advanced tasks [25 marks]

Further to the tasks described above, define tasks by yourself and carry out investigation. The following are examples. Marks shown below are for reference only, actual marks will vary depending on the contents and quality of experiments and discussions.

- Develop gender dependent acoustic models and carry out recognition experiments. [10 marks]
- Investigate feature transformation and speaker adaptive training to improve WER. [15 marks]
- Investigate how decoding parameters such as word insertion penalty (wip) and language-model weight (lmwt) influence WER, and find the optimal values. [10 marks]

[Scripts] `e.g.  exp_adv_t1.sh`

(NB: Script file names for Task-3 can be arbitrary, please indicate them in the report)

Tips on experiments

- You will find that there is a large number of combinations or a large range of parameters you might need to explore. It will not be feasible to try all the possible combinations / ranges with a fine resolution. Try a coarse resolution first, which would give you some idea as to what/how you should try next.
- Please keep only the models that gave the best performance in the corresponding task, and delete other models and irrelevant files as soon as you finished the task. (See below to see why this is important)
- Due to a limited disk space available, please keep the total disk usage of *WorkDir* less than 3GB, or the total usage of your pair less than 6GB. You can check your disk usage by running the following command in your *WorkDir*.

```
du -hs .
```

Note that in case of having the disk space run out, you will be no longer able to write files – current/further experimental results will be lost.

3.2 Report

Keep the length of your report between 4 and 7 pages (with a double-column format) including figures, tables, and references. Experimental results should be well summarised using figures or tables. You should not only show the results, but also explain your experiments and results.

Please read the following instructions and write a "scientific report". For higher marks, you need to give good discussions based on both theories and experiments.

- Experimental results should be efficiently summarised using figures or tables (but not both if possible) - avoid using a separate graph/table for each experiment.
- Figures/tables should be numbered and captioned.
- Experimental conditions and methods that are different from those in original scripts should be shown clearly and concisely - consider using a table for example. Providing sufficient information is essential in scientific reports so that other people could redo the same experiments.
- Show results even if there was no improvement. It is important to discuss/analyse why there was no improvement.

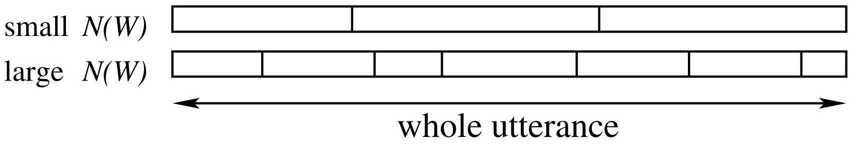
Appendix – Decoding parameters

ASR system uses $P(X|W)$ and $P(W)$ to find the best word sequence W.

- These estimated probabilities are different in reliability and dynamic range.
- More shorter words tend to have lower scores than those with fewer longer words in real implementation (due to $P(W)$).

$$\begin{aligned} W^* &= \arg\max_W P(X|W)\, P(W) \\ &\Rightarrow \arg\max_W P(X|W)\, P(W)^{\mathit{LMS}}\, \mathit{IP}^{N(W)} \qquad \cdots \text{ modified formula} \\ &= \arg\max_W \log P(X|W) + \mathit{LMS} \log P(W) + N(W) \log \mathit{IP} \end{aligned}$$

$$\begin{aligned} \mathit{LMS}: &\quad \text{language model weight (LM scaling factor)} \\ \mathit{IP}: &\quad \text{insertion penalty} \\ N(W): &\quad \text{number of words in } W \end{aligned}$$



- Interpretation of *LMS*
  As $\mathit{LMS} \to 0$, $P(W)^{\mathit{LMS}} \to 1$,
  i.e. the smaller *LMS* becomes, the less important the LM is.

- Interpretation of IP
  Assuming a uniform LM (every word has an equal occurrence probability), $P(W)$ for fewer longer words (i.e. smaller $N(W)$) is greater than $P(W)$ for more shorter words (i.e. larger $N(W)$). IP is used to balance this.

  $0 < IP \leq 1$ (i.e. $\log IP < 0$) the smaller IP becomes, the more shorter words are penalised (i.e. fewer longer words are preferred)

  $1 < IP$ (i.e. $\log IP > 0$) the larger IP becomes, the more shorter words are preferred. (i.e. the more insertion errors.)

- LMS and IP are determined heuristically (on validation data). Larger LMS usually needs larger IP.