Digital Speech Processing HW1

National Taiwan University
CSIE4611, 2017 Fall
Written By: B04902077 Wei-Hsuan Chiang
Discussed with: B04902028 Hao-Hsian Hong

Environment

- Equipment: Macbook Pro with macOS Sierra 10.12.6
- Compiler: g++

  ```
  Configured with: --prefix=/Library/Developer/CommandLineTools/usr --with-gxx-include-dir=/usr/include/c++/4.2.1
  Apple LLVM version 8.1.0 (clang-802.0.42)
  Target: x86_64-apple-darwin16.7.0
  Thread model: posix
  InstalledDir: /Library/Developer/CommandLineTools/usr/bin
  ```

- CPU: 2.8 GHz Intel Core i7
- RAM: 16 GB 1600 MHz DDR3

Model Building

I used the HMM basic problem 3 mentioned in the slides as a reference to implement my model-building algorithm. It is so-called Baum–Welch Algorithm. After several discussion and tests, I decided to set the iteration number to 1500, since with a larger number of iteration, the model might be more precise comparing to the result of a small number of iteration. And among several large numbers, 1500 seems to have the best result. On average, one model takes 8 minutes to build, so in total, it takes about 40 minutes to finish building all the 5 models. Such models provide a good prediction basis for the testing part. It can achieve up to 85% accuracy. However, the time spent on the training seems to be too much, there are still some way for me to improve the model-building algorithm. The following is the time measurement result based on *seq_model_02.txt*:

```
$ time ./train 1500 model_init.txt seq_model_02.txt model_02.txt
real    7m31.864s
user    7m29.751s
sys     0m0.901s
```

Testing

Viberti Algorithm is applied in the test program, as suggested in the homework slides and lecture notes. With this algorithm and the model built in the previous part, the test program shows a relatively precise prediction outcome. In addition, this does not consume much time and does not have high time complexity. The following is the time measurement result:

```
$ time ./test modellist.txt testing_data1.txt result1.txt
real    0m0.133s
user    0m0.122s
sys     0m0.003s
$ time ./test modellist.txt testing_data2.txt result2.txt
real    0m0.130s
user    0m0.120s
sys     0m0.003s
```

How to Execute

By Makefile. Makefile is included in the submission folder and the execution commands and examples are as follows:

```
$ make all
$ ./train 1500 model_init.txt seq_model_01.txt model_01.txt #Parameter as mentioned in the homework slides.
$ ./test modellist.txt testing_data1.txt result1.txt
$ ./acc_calculate testing_answer1.txt result1.txt acc.txt
$ make clean
```

Result

- Accuracy: 87.00%, under the public dataset (testing_data1.txt / testing_answer.txt)

  ```
  $ ./acc_calculate testing_answer1.txt result1.txt acc.txt
  $ cat acc.txt
  0.8700
  ```