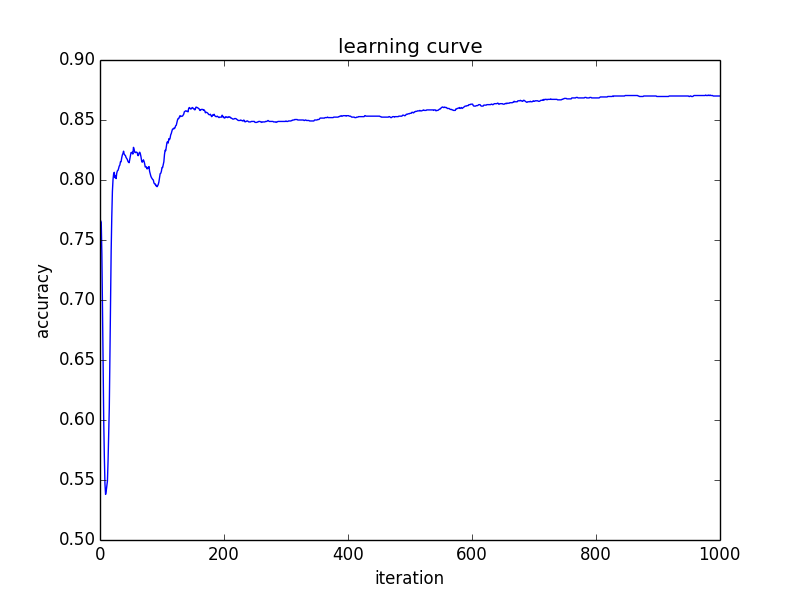
Fundamentals of Speech Signal Processing 2015 Spring

**Homework 1**

***Discrete Hidden Markov Model Implementation***

資訊三 B01902040 鍾毓安

**Summary of my results**



* Above is the learning curve where: coordinate x is the ***training iteration*** and coordinate y is the corresponding ***accuracy***.
* Highest accuracy: 0.8708
* My tips:

===== Array Access Method 1 =====

double array[N][T];

for(int i = 0 ; i < N ; i++){

for(int j = 0 ; j < T ; j++){

array[i][j] = … // do the task

}

}

===== Array Access Method 2 =====

double array[N][T];

for(int i = 0 ; i < T ; i++){

for(int j = 0 ; j < N ; j++){

array[j][i] = … // do the task

}

}

The former method is much more efficient than the latter one due to the architecture of computer memory alignment: sequential accessing the memory outperforms jumping over to access memory. Keep this in mind, the code will run faster.

**Program Execution**

* Source code

1. hmm\_train.cpp
2. hmm\_test.cpp
3. compute\_acc.cpp

* Makefile

all:

g++ hmm\_train.cpp –o train

g++ hmm\_test.cpp –o test

g++ compute\_acc.cpp –o compute\_acc

run:

./train 800 model\_init.txt seq\_model\_01 model\_01.txt

./train 800 model\_init.txt seq\_model\_02 model\_02.txt

./train 800 model\_init.txt seq\_model\_03 model\_03.txt

./train 800 model\_init.txt seq\_model\_04 model\_04.txt

./train 800 model\_init.txt seq\_model\_05 model\_05.txt

./test modellist.txt testing\_data1.txt result1.txt

./test modellist.txt testing\_data2.txt result2.txt

./compute\_acc

clean:

rm ./train

rm ./test

**Experiment Environment**

* linux4.csie.ntu.edu.tw