姓名：陳品君 學號：R06944025 系級：網媒一

Fundamentals of Speech Signal Processing __ HW1

- Environment

  在此次作業之中，是在mac OS上執行程式，並以C++語言進行程式的撰寫與編譯。下圖為此電腦環境下的規格與其餘環境：

	規格
機型	MacBook Pro (13-inch, 2016, Two Thunderbolt 3 ports)
處理器	2 GHz Intel Core i5
記憶卡	8 GB 1867 MHz LPDDR3
執行環境	Apple LLVM version 9.0.0 (clang-900.0.38) Target: x86_64-apple-darwin16.7.0 Thread model: posix

- How to execute

  1.Make

  編輯makefile檔案，在裡面輸入指令後，便可執行編譯train.cpp與test.cpp兩個檔案，並生出所需檔案。

  make的功能為查閱每一步驟所需要的檔案，如果沒有便執行指令以產生該檔案，而若已有檔案，則跳過該步驟繼續往下。因此若寫程式者了解原始碼的撰寫步驟，就可以依序寫好指令，所以只要有Makefile檔案，只要鍵入make就可以成功編譯出檔案。

  2../train iteration model_init.txt seq_model_0x.txt model_0x.txt

  此指令用來根據現有的model，放入相同結果（相同model）的資料來改善λ。

  此處的參數有四項：

  (1)iteration：迭代的次數，重複訓練模型的次數。

  (2)model_init.txt：放入一開始需要的initial model，作為實現Baum-Welch algorithm的所需要件。

  (3)seq_model_0x.txt：此處x為1至5的數字，為model1至model5的data，用此來計算出Baum-Welch algorithm內所需要的參數，並用這些算出來的參數來改善λ。

  (4)model_0x.txt：此處x為1至5的數字，為model1至model5各自訓練完的結果，也就是新的model1至model5，接下來便會用此λ去進行test。

  3../test modellist.txt testing_data1.txt result.txt

  在進行testing時，是以Viterbi Algorithm來完成的。Viterbi Algorithm可以用來算出在此model中，能有最大機率的路徑，因此可以用來把所需預測的O丟入五種模型內，看在哪個model中的P*最大，此O便最有可能屬於該model。因此，需要有三個參數：

(1)modellist.txt：此處輸入進此list，內容包含所有model的檔名，再用程式來抽取出list中的model檔案，並得到各個model的λ。用來告知程式有哪些model要用來參與預測，並提供參數。

(2)testing_data1.txt：為輸入的observations，作業的答案便是要求出這些observations各自屬於哪個model。

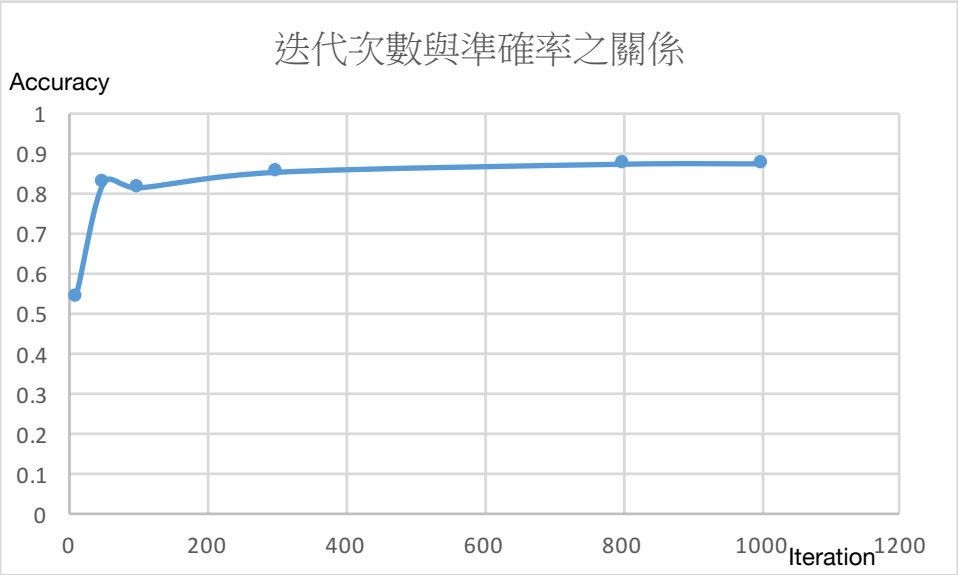
(3) result.txt：為輸出的檔案，內容為兩行，第一行為預測此列之observation屬於哪個model，而第二行為機率。

- 討論：迭代對於training之影響

  根據EM Theory，迭代會慢慢把機率加大，直至收斂。方式如下流程：

  λ = (A, B, π) —> $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ —> λ = (A, B, π) —> $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ —> …

  不斷輪迴下去，透過實作可以觀察出迭代對於準確率的影響



由上圖可看出，迭代次數在50次內急速上升，並在之後緩慢地持續成長，越多的迭代次數，則準確率會慢慢提升並趨於平衡，最終選擇了1000次迭代次數作為結果。

下表為不同迭代次數的準確率比較表格。

Iteration	10	50	100	300	800	1000
Accuracy	0.541016	0.823129	0.810324	0.84914	0.869148	0.869948