數位影音報告 By r01922164 資工所 李揚

Environment and how to execute it:

我有在工作站和windows7環境下跑過基本上都OK

然後操作方法就跟助教規定的一模一樣:

./test modellist.txt testing\_data.txt result.txt

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

有一點值得特別注意的是，我的test有順便去算testing\_data1的accuracy，所以在這支程式裡會去跟testing\_answer.txt去做比較然後output出acc.txt。所以一定要放testing\_answer.txt。

我的accuracy:

0.8696 in 2000 iterations’ models

心得:

我是這一學期進入台大資工，所以這次我進台大的第一個程式作業，所以還算是蠻興奮的，也很高興最後能達到accuracy的要求。

我共花了十天做這個作業，前面三天架構，後面七天debug。我覺得這次作業整體來說不會太難。

但有以下幾點需要注意:

1. Debug時，如果是一些只有在accuracy看得出來的微小的錯誤，比較難以看出是錯在test還是train，所以最好一開始寫好可以獨立測試這兩個部分的測資。
2. 這之作業裡有用到很多二維甚至三維的index，要非常小心處理標號間的命名，以免發生錯亂。(我個人是I for time, j for state, k other needs)
3. 測試時間會有點久，我寫了一支autotest.c，可以在一支程式裡train五個model，測驗時會比較方便。

程式架構:

我是用c語言去寫這支程式，考量蠻簡單的，因為matlab是付費軟體，而且用C可以藉此機會用一下系上的工作站。

程式大致上可分為三部分:1.讀檔.計算3.寫檔

我的命名方式如下:

gamma: transition分子 sumga: sum of gamma(for all training sequence)

ep : transition分母 sumep: sum of ep(for all training sequence)

gammaOK: observation分子 sumgaOK:sum of gammaOK(for all training sequence)

test方面，則用viterbi陣列去計算。

註解部分是我的debug的測試

希望這部分方便助教讀我的code。

我的一些發現:

1. forward, backward variable的數值分析:

我覺得這點會是我在實做之前很少考量到的一點，forward和backward variable在計算時會一列一列變小。

我覺得原因如下

Forward(t+1,j)

=sigma(all state i)(transition(I,j))\*forward(t,i))observation(seq(t+1),j)

transition(i,j)，跟observation(seq(t+1),j)都是對所有state累加會為一的值。

也就是說，但我們累加時卻只有累加number of states次，所以我每進入下一個時間的forward variable，總值會變成(1/number of states)。

同理，backward variable和viterbi都會有這樣的現象

這時候就要注意精準度問題了!不過這次作業只有六個state和50個time frame，所以最後的值還在double可以計算的範圍之中。

(1/6)^50=1.2371931e-39

所以還OK，但如果要做更大規模的實做時(state和time frame增加)，數值上勢必要做一些特殊處理。

1. Some tests on the training set

我把兩千次train出來的model，餵進seq\_model\_01, seq\_model\_02,…, seq\_model\_05的sequence。

會得到以下的結果:

model\_01: 0.999300

model\_02: 0.999100

model\_03: 0.999600

model\_04: 0.673400

model\_05: 0.678600

可以很明顯的發現，model\_04、model\_05是這整套系統accuarcy的bottle neck。

而我去把testing\_data1的答案一個一個output出來，發現大部分的錯誤都是發生在答案為model\_04.txt和model\_05.txt。

所以，我們如果要提高這個辨識系統的accuracy很明顯要從model\_04,model\_05，這兩個model著手。然後我去看model\_04和model\_05的learning curve(其實就是把各個iteration train出來的精準度拿出來看)，發現這兩個model也快要converge了。我可以判斷，model\_04、和model\_05，有可能收斂到一個不好的local maximum，所以如果要提高這個辨識系統的精準度，可以嘗試改變model\_04和model\_05的初始。

不過我們隨便亂挑的一個initialization居然可以在五個model裡fit好三個model的training set，表示這演算法其實有很高的機率收斂到global maximum。