# Machine Learning and having it deep and structured HW3

## Group Information

Group Name：<迪普能靈>屎抓蛆鵝能靈

Student：

吳建昇(b01901045)/ 陳奕安(b01901140)/ 方彥鈞(b01502102)/ 楊騏瑄(b01901042)

Member Contribution：

HMM：陳奕安、方彥鈞，Structure SVM：吳建昇、楊騏瑄，Exp：all，Report：all

## Comparisons Between Algorithm

### HMM

實作分為「擷取資料」和使用「Viterbi algorithm」兩個部份。

擷取資料部份是計算 Training data 每個句子裡 Phoneme 出現的機率及相鄰 Phoneme 出現在一起的頻率。對於起始機率向量(1 x 48)、Emission 機率矩陣、及結束機率矩陣(1 x 48)相當簡單，只須從頭到尾對 Training data 數過一次，最後 Normalized 成機率形式即可。對於 Transition 機率矩陣(48 x 48)，我們經過實驗比較後選擇使用結果稍微好一點的 DNN/RNN 的 softmax 結果，而沒有再經過貝氏定理轉換。

對於一個長度為 N 個 Phoneme 的句子，我們先在初始化階段先預留 Dynamic programming 需要的 BackTracking 矩陣(N x 48)及儲存結果的機率矩陣(N x 48)。接著套用 Viterbi Algorithm，計算所有 Phoneme 的排列組合中，機率最大的一組句子。由於我們在 log scale 上相加，不用擔心機率太小的問題。另外在計算機率的公式中，我們加了兩個參數$\alpha$和$\beta$，使得公式變成：

$$Prob = LastProb \times Emission^{\alpha} \times Transition^{\beta}$$

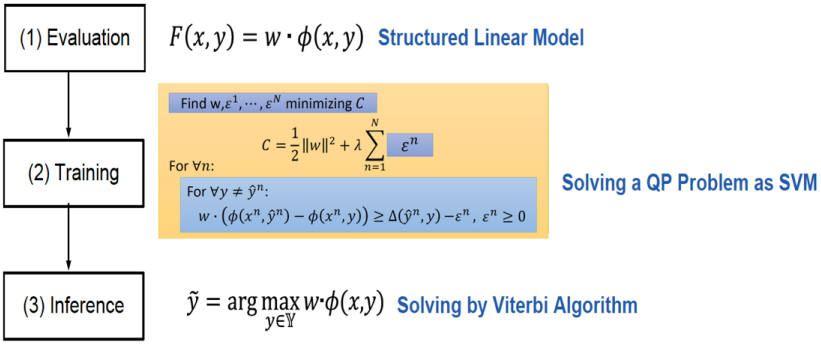
由這兩個參數可以調整 Emission 和 Transition 機率的權重。由於 HMM 擷取資料的部分對同樣的 Training dataset 會有相同的結果，且 Viterbi algorithm 的複雜度僅為 O(N)，所以這是三種方法中最快的一種。

Duration 的概念是讓句子中一個 phoneme 轉換成不同的 Phoneme 時，會持續一定長度才有可能換成下一個 phoneme，如此符合自然發音時不可能在短時間內轉換聲音，但是也有可能因此拉長錯誤的 phoneme 持續的時間，實作上，我們做了兩個 Transition 機率的矩陣，一個是上述從 Training data 得到的，另一個是對角線為 1 的單位矩陣 ，也就是 Transition 維持在同一個 phoneme 的機率是 1，在給定的 duration length 內，使用單位矩陣做為 Transition 機率，

直到 phoneme 維持時間超過 duration length 時。才換成一般的 Transition 機率。我們將 duration length 定為 3，以確保每個出現的 phoneme 重複三次。

另外，開頭與結尾的 sil 是沒有意義的聲音，因此我們的另一個想法是將開頭和結尾連續的 sil 去掉再放入 HMM model。在計算 Transition matrix 的時候，自然也要將開頭和結尾的 sil 去除來計算。

### Structure SVM




read_struct_examples：將 rnn with softmax 檔案讀入成 pattern, label 兩個類別，其中 pattern 包含二維矩陣 observe_object[frame 編號][48 維 feature 編號]，記錄 feature、frame_number 記錄該 sentence 含幾個 frame、speaker，label 中包含 state_object[frame 編號]記錄答案。

Viterbi：演算法的設計同 HMM，且習慣上把機率在 log domain 進行計算，所以直接使用 wx$\varphi$(x,y)代表取過 log 的機率，而非用$e^{\mathrm{wx}\varphi(\mathrm{x,y})}$代表機率，以加快速度。我們設計一個參數以控制 loss，在計算 most violated constraint 時會考慮 loss，因此在每個 frame 會將除了最佳值之外每個不是正確答案的 state++

find_most_violated_constraint：使用含 loss 的 viterbi 找出 y_bar

classify_struct_example：使用不含 loss 的 viterbi 找出 y_max

write&read_struct_model：使用 svm_common.c 中提供的 write&read_model

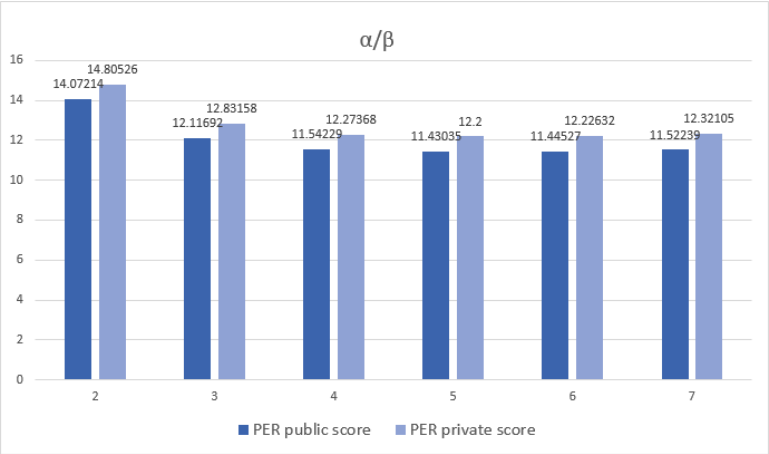
loss ：解 QP 時使用，傳回一個 sentence 中 y_bar 與答案不同的 frame 的個數

## Experimental results

### HMM

在實驗中，我們的 Observation matrix 使用 ACC 為 70%的 RNN softmax output，但觀察後發現每一個維度的值都在 0.01 到 0.05 中間，使得 Emission probability 的影響過小，因此在使用 Viterbi algorithm 的時候，我們需要調高 predict phoneme 跟其他

phoneme 之間的差距，利用前述 Prob 的公式，分別測試$\alpha/\beta$=2,3,4,5,6,7 並丟入 hw2 的 kaggle 做測試，得到下圖表：




可得到$\alpha/\beta$約為 5~6 有較好的結果。

選擇$\alpha/\beta$=5 的參數，以下是我們的兩個 feature 加入 HMM 後的結果。我們的對照組是在 HMM 結束後，將預測出的 sequence 中，不到連續三個的 phoneme 去掉。三樣結果如下表格：($\alpha/\beta$=5)

| | PER |
|---|---|
| Basic | 11.30068 |
| With duration | 11.05405 |
| with sil-trimming | 12.127 |

對於 Sil-trimming，由結果推測，去掉頭尾 sil 對結果影響不大。對於 HMM 有沒有 duration 的差別，我們發現將 Duration 加入 HMM 的效果，可以比對照組在最後輸出時才去掉有誤差的 Phoneme 更好，表示這是個有用的 feature。

# SVM

./svm_empty_learn -c 100 大約 2100 個 iteration，總 CPU run time 為 22935(s)

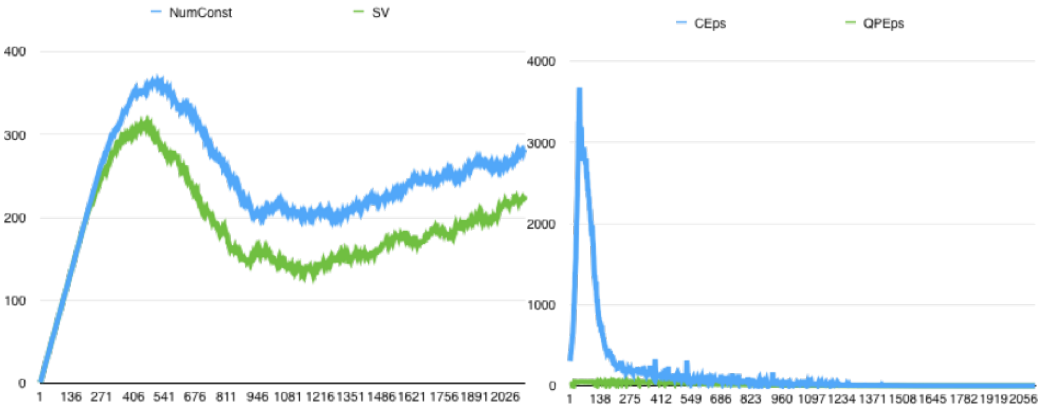
（input data 為 RNN softmax 48 維）

./svm_empty_learn -c 1000 大約 3200 個 iteration，總 CPU run time 為 35612(s)

（input data 為 RNN 沒有做 softmax 的 48 維）

```
Final epsilon on KKT-Conditions: 0.09245
Upper bound on duality gap: 9.02927
Dual objective value: dval=25761.34570
Primal objective value: pval=25770.37497
Total number of constraints in final working set: 282 (of 2107)
Number of iterations: 2108
Number of calls to 'find_most_violated_constraint': 7791168
Number of SV: 222
Norm of weight vector: |w|=70.98320
Value of slack variable (on working set): xi=232.44608
Value of slack variable (global): xi=232.51068
Norm of longest difference vector: ||Psi(x,y)-Psi(x,ybar)||=138.04261
Runtime in cpu-seconds: 22935.26
Writing learned model...done
```

```
Final epsilon on KKT-Conditions: 0.09547
Upper bound on duality gap: 92.79884
Dual objective value: dval=160950.85916
Primal objective value: pval=161043.65800
Total number of constraints in final working set: 420 (of 3186)
Number of iterations: 3187
Number of calls to 'find_most_violated_constraint': 11779152
Number of SV: 360
Norm of weight vector: |w|=32.25270
Value of slack variable (on working set): xi=160.45120
Value of slack variable (global): xi=160.52354
Norm of longest difference vector: ||Psi(x,y)-Psi(x,ybar)||=172.75126
Runtime in cpu-seconds: 35612.85
Writing learned model...done
```




圖表解析：NumConst 及 SV 與 constrains 有關且趨勢相同，CEps 推測是代表 the amount by which the most violated constraint found in the current iteration was violated，所以在前面的 iteration 可以找到違反量很大的也就是求解的區間可以縮小的很快，但隨著 train 的時間拉長之後求解區間已經很難被縮小了，而使得 CEps 變小，QPEps 在註解裡提到是 percision, up to which this modelis accurate，雖然不太明白意思，但這兩個數值都接近 0 時 train 就會結束

## Comparison with HW2

### HMM v.s. RNN:

使用 HW1 DNN 的 softmax output，並用$\alpha/\beta$=5 加上有 duration 的 HMM，得到 PER=13.3000 的結果，因為與 RNN 使用相同的 DNN output，可知我們實作的 HMM 的效果不如 RNN。推測原因出於 softmax 出來的 output 在 predict phoneme 跟其他 phoneme 之間的差距太小，即使在做出$\alpha/\beta$的權重調整後，仍然無法得到比 RNN 更好的結果。

HW2:

| | | | |
|---|---|---|---|
| 迪普能靈 | 12.82338 | 12 | Wed, 18 Nov 2015 01:28:54 |

HW3:

| | | | | | |
|---|---|---|---|---|---|
| 23 | ↓5 | <迪普能靈>屎抓蛆鵝能靈 | 10.11486 | 11 | Mon, 07 Dec 2015 04:00:21 |