Team 得意得意的佑佑

MLDS HW3 Report

Data Format

DNN softmax hw1 71%正確率的結果。

Algorithms

HMM

- 原型
  transition probability 使用 train data label ground truth 算出來，對同一個sequence中各個連續的label pair i, j, count(label i to label j) += 1，transition probability P(state i to state j) = count(i to j) / (∑count(i to s) for each s)。而emission probability P(Observation | State) = P(State | Observation) * P(Observation) / P(State) ，其中P(Observation)定為一，因為在比較過程中任何一種可能的sequence P(Observation)乘積相同，可以忽略。而P(State)可以從training data label ground truth 中統計出來。然後使用Viterbi algorithm算出機率最高的label sequence。並省略start和end的機率。
- 3-Duration Model
  為了模擬聲音連續的特性，強迫每一個label要連續三次或以上，作法是將一個label代表的state變成3個，例如state_x -> [state_x_1, state_x_2, state_x_3]。P(state_x_2 | state_x_1) = 1, P(state_x_3 | state_x_2) = 1, P(state_x_3 | state_x_3) = 原本的transition probability(x to x), 對於除了x以外的label y，P(state_y_1 | state_x_3) = 原本的transition probability(y to x)。
- Emission probability 修正
  對於原本從DNN來的P(State | Observation)做修改，對於一個Observation，所有的state中 Max(P(State | Observation)) 如果夠高則維持原樣，如果不夠高，則將他的機率下修，並把下修的機率分配給其他的state的機率，以達成smooth的效果。會有這樣的想法是因為如果Max不夠大的話，則他很有可能是predict錯誤的，所以把機率攤平，在Viterbi的時候更依靠其他的transition probability和emission probability。
- 省略P(State)
  在算emission probability 的時候，理應是P(Observation | State) = P(State | Observation) * P(Observation) / P(State) ，這邊我們同時省略P(Observation) 和 P(State)。

Structural SVM

- 原型
  Toolkit: Struct SVM, python version.
  定義factor: 長度為48 * 48(emission) + 48 * 48(transition) 並省略start和end的部分。
  定義loss: y 與 y-bar 不同的label的個數，並除以y的長度作為normalization。
  Psi: emission的部份為 P(state_i) = ∑(seq內所有的Observation加總), P(state_i)是一個48維的vector. transition的部份同HMM.
- 加速

Team 得意得意的佑佑

在python版本中，即使用矩陣加速還是太慢，所以我們又實作了Struct SVM, C version. 在C的api中，我們實作了會自動由大到小調整e的版本，透過循序的增加準確度，來自動找出最適當的e，並同時找到loss最小的模型。

Smoothing

第一個pass的時候對所有的label做trimming並算出每個label的count，如果小於3，並且兩側的label是一樣的，則猜測他應該跟兩側的label一樣，把它變成兩側的label。第二個pass把所有count小於3的label刪除。

Experiments

- **HMM**

Emission Probability的修正，如果Max(P(State | Observation)) < alpha，則 Max *= 0.8，並把減少的部份平均分配給其他state的probability。

HMM results

Model	Testing Data Average Edit distance
原型	12.07
原型 + smoothing	15.90
3-Duration	11.75
3-Duration + Emission probability修正(alpha = 0.7)	11.56
3-Duration + Emission probability修正(alpha = 0.8)	11.57
3-Duration + Emission probability修正(alpha = 0.7) + 省略P(state)	11.21

- **Structural SVM**

Struct SVM results (run with python version)

Model	Testing Data Average Edit distance
c100 e0.0001 smoothing	16.64
c100 e0.0001 none-smoothing	14.44
c50000 e0.1 none-smoothing	15.88
c5 e0.001 none-smoothing	15.38
c50000 e0.1 smoothing	17.08
c0.1 e0.0001 none-smoothing	23.98
c5 e0.0001 none-smoothing	15.41

Struct SVM results (run with C version)

Model	iterations	AED
c1 none-smoothing with auto-optimized e	380	16.66

Team 得意得意的佑佑

Model	iterations	AED
c10 none-smoothing with auto-optimized e	890	14.59
c100 none-smoothing with auto-optimized e	2992	15.41

Analysis

- Duration HMM
  從原先的HMM Model到改進過後的3-Duration利用的是，聲音的連續性的性質，強迫將出現的label產生有連續的特性，借此我們可以做出類似smoothing的效果，將連續出現的label中所產生可能是雜訊的label移除，並從實驗的結果中可以看出跟原先的HMM相比有些許的進步。
- Emission probability Modification HMM
  HMM Model中出現機率較小的機率的state轉換，透過參數的調整(alpha) ，將其機率平均分散到其他的轉換上，此作法的目的是將可能是雜訊的state轉換的機率變小，以達到去除可能的雜訊，但其效果需要透過參數的實驗，來提升performance，從上面實驗可以看出不同的alpha對於實驗結果的影響。
- Structural SVM
  不同於HMM的機率模型，Structural SVM透過尋找邊界來區分不同的方式，透過training data來區分出邊界，學習時將有violation的點給予適當的懲罰，並更新參數，目的在求出能算出最小的loss參數，將不同的label分的最開，而不同於HMM，Structural SVM更注重於初始參數的調整，從實驗的結果可以看出不同的參數對於Structural SVM的成效有顯著的影響，而相較於HMM Model，Structural SVM的結果較不理想，可能的原因是沒有找出最理想的參數，也有可能是對於這筆taining data，他的結構分布無法透過Structural SVM找出正確的邊界來，以致於training data的結果無法完整的適用於testing data。此外我們發現在C version中，參數C的值等於100時反而稍大於等於10的結果，推測有可能是這筆taining data不足以表現原始data的樣貌，而在學習的過程中發生了overfiting。

Team Contribution