DSP Homework 1 資工所碩一 R05922068 彭宇劭

• Experiment Setting

Environment: MacOS Terminal 2.6.1
Programming Language: C++
g++ version: Apple LLVM version 8.0.0 (clang-800.0.42.1)

• Execution

```
make
./train 180 model_init.txt seq_model_01.txt model_01.txt
./train 180 model_init.txt seq_model_02.txt model_02.txt
./train 180 model_init.txt seq_model_03.txt model_03.txt
./train 180 model_init.txt seq_model_04.txt model_04.txt
./train 180 model_init.txt seq_model_05.txt model_05.txt
./test  modellist.txt testing_data2.txt result2.txt
./test  modellist.txt testing_data1.txt result1.txt
```

• Result

Iterate 180 times: 0.786 (Acc)

• Files Description

Code

hmm.h	The code provided by TA Read the parameters from file and write them to HMM structure Read the parameters from HMM structure and write them to file
train.cpp	Implement Baum-Welth algorithm
test.cpp	Implement Viterbi algorithm

Output

result1.txt	The prediction of testing_data1.txt	Format: model0x.txt likehood model0x.txt likehood …
result2.txt	The prediction of testing_data2.txt	
acc.txt	The accuracy of result1.txt	0.xxxx

- ## Data structure

HMM struct：Store parameters of different hmm（5model—> 5structure）

char *model_name	Model name
int state_num	The state number of HMM
int observ_num	The number of observation which can be observed from each state
double initial[]	The vector of Initial state probabilities
double transition[][]	The matrix of state transition probabilities
double observation[][]	The matrix of the observation probability with respect to a each state

HMM_Params：Store parameters that will be use in Baum-Welth algorithm

int sequence_size	The length of a sequence
int state_num	The number of state
double alpha[][]	Forward parameter
double beta[][]	Backward parameter
double gamma[][]	Normalize a*b
double epsilon[t][i][j]	The 3-dim probabilities matrix. At time t, the prob of state i—> state j

HMM_Cumulate：Cumulate each parameter from each sequence. update π, A, B

int counter (N)	The number of sequence
double pi[]	$\sum_{n=1}^{N} \gamma_1^n(i)$
double transition_epsilon[]	$\sum_{n=1}^{N} \sum_{t=1}^{T-1} \varepsilon_t^n(i,j)$
double transition_gamma[][]	$\sum_{n=1}^{N} \sum_{t=1}^{T-1} \gamma_1^n(i)$
double observation_numerator[][]	$\sum_{n=1}^{N} \sum_{t=1, o_t=v_k}^{T} \gamma_t^n(i,j)$
double observation_denominator[][]	$\sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_1^n(i)$

- ## Reference

- DSP2017 spring website
- Viterbi
  https://en.wikipedia.org/wiki/Viterbi_algorithm
- Baum–Welch algorithm
  https://en.wikipedia.org/wiki/Baum–Welch_algorithm
  http://www.indiana.edu/~iulg/moss/hmmcalculations.pdf