

National Taiwan University

Digital Speech Processing: Report #1

EE2 b03901016

Hao Chen

April 3, 2016

1 Introduction

This program simply simulate the Hidden Markov Model(HMM), the whole process include two parts — training and testing. The training part constructs the models with specified input sequences, while the testing part identify datas to the model it belongs.

2 Environment

- OS : Linux mint 17.2 Rafaela
- Kernel : x86_64 Linux 3.16.0-38-generic
- Shell : zsh 5.0.2

3 Steps

3.1 Training:

Given O and an initial model $\lambda(A, B, \pi)$, adjust λ to maximize $P(\bar{O}|\lambda)$. (Baum-Welch Algorithm)

1. Read the initail model — `model_init.txt`
2. For each model, read its sequence model — `seq_model_0X.txt`
3. For each iteration, calculate and accumulate $\epsilon_t(i, j)$ and $\gamma_t(i)$, and update model parameters.
4. Ouput files — `model_0X.txt`

3.2 Testing:

Given model λ and O , find the best sequences to maximize $P(O|\lambda, q)$. Calculate $P(O|\lambda)$ for each of the five models. (Viterbi Algorithms)

1. Input file — `testing_data0X.txt`
2. For each sequence, output the hypothesis model and it's likelihood to `resultX.txt`
3. Calculate the classification accuracy and ouput to `acc.txt`

4 Usage

- **Makefile** : Simply type "make" to build the programs, "make clean" to remove the program.
- **train** : Follow the form `./train iter model_init.txt seq_model_0X.txt model_0X.txt` to execute. (While "iter" is how many times you update your model.) Else you'd get an assert message.
- **test** : Follow the form `./test modellist.txt testing_dataX.txt resultX.txt` to execute.

5 Result

Since the training data is limit, we can repeat training process again and again. As we iterate the training process more, the model become more well-trained, the characteristic of each model become more obvious, so we have better performance when testing "testing_dataX.txt" using the well-trained models. The following table record the experiment of testing "testing_data01.txt" with "model_01~05.txt"

iter	1	10	20	50	150	300	900	1000	1040	2000
accuracy	0.7660	0.5408	0.7912	0.8228	0.8608	0.8448	0.8704	0.8700	0.8712	0.8692

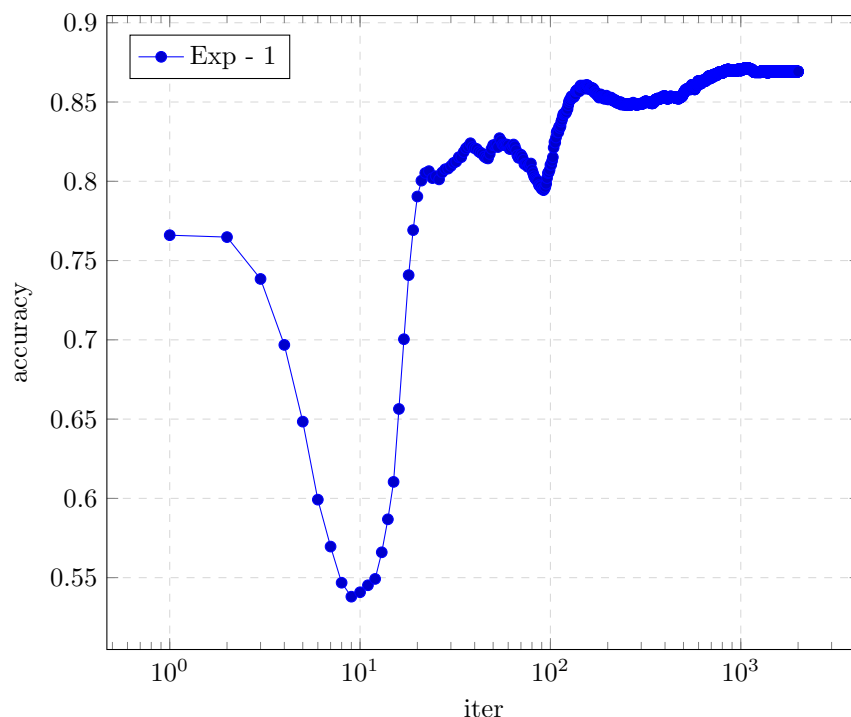


Figure 1: The accuracies of `result1.txt` from `iter = 1` to 2000

We can find out that the accuracy has the lowest value at about $iter = 10$, while has the highest value at about $iter = 869$, this result appear because the training data of each model is limited. But we can observe that the accuracy does not have significant change when $iter$ is large enough.

The limit that computer can calculate might cause the wrong testing answer, because the probability we get after doing "Viterbi Algorithm" is really small, about the magnitude of 10^{-40} to 10^{-60} , some might be even smaller that it cannot be estimate. Using the logarithm is a method to improve accuracy, but it makes the program less efficient, and do not have a lot of changes, so I decided not to use it in my program.