Report MLSP Assignment 2 –GMM, and HMMGMM

Sonu Dixit

|  |  |  |  |
| --- | --- | --- | --- |
|  | Error Rate | Code file | Max EM Iteration |
| GMM 2 Diagonal Covariance | 25 % | GMM\_implementation.py | 60 |
| GMM 2 Full Covariance | 21% | gmm\_2\_mixture\_full\_cov.py | 20 |
| GMM 5 Diagonal Covariance | 30% | gmm\_5\_diag.py | 15 |
| GMM 5 Full Covariance | 25% | gmm\_5\_full.py | 15 |

Q3:

Plots (log likelihood versus em iteration steps )

Q3\_e\_ Full covariance option is giving better test performance.

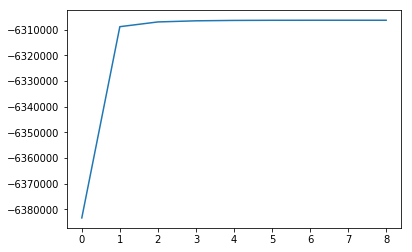


Figure 1= speech\_data\_2mixture\_diagonal, iteration stopped after 8 since no significant improvement

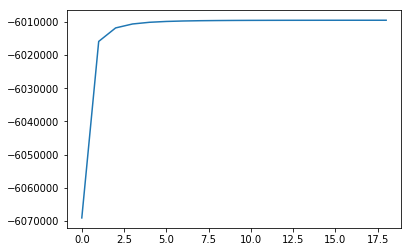


Figure 2 = music\_data\_2\_mixture\_diagonal\_cov\_ignore decimals on x axis, it should be integers only

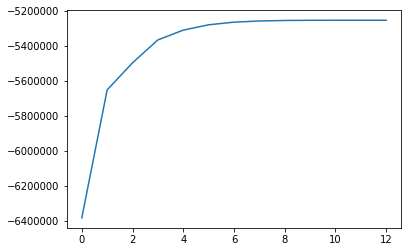


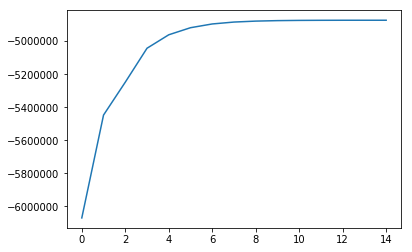
Figure 3= speech\_data\_2\_mixture\_full\_cov

Figure 4music\_data\_2\_mixture\_full\_cov

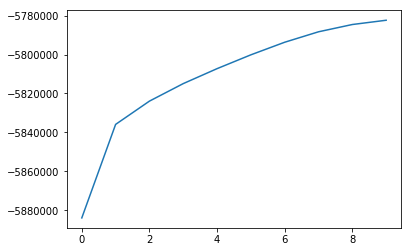


Figure 5= speech\_data\_5\_mixture\_full\_cov

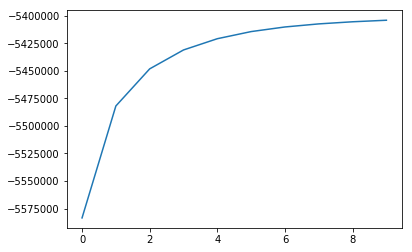


Figure 6=music\_data\_5\_mixture\_full\_cov\_

Q4:

1. Likelihood computation using Forward and backward computation:

Code file : HMM\_v2.py

Function for forward computation:

forward\_calculation(A,B,pie,Observation):

* observation is a 1d array assumed it contains 0,1,2,3,4....starting with 0
* A is transition prob matrix num\_states X num\_states
* B is emission prob matrix num\_states X num\_observation\_symbols
* pie is initial distribution 1d array
* it returns alpha and **scale factor**. Scale factor has been used to deal with limited capacity to represent floating points in programs.

Function for backward computation:

backward\_calculation(A,B,pie,Observation,scale\_factor):

Scale factor is the scales returned by forward calculation. It returns betas.

Flat start initialization is being done in following functions:

A\_initialise(num\_states,data\_shapes)

Q4\_b\_Viterbi Algorithm:

Function for this is:

viterbiAlgorithm(observation,A,B,pie)

Q4\_c\_Baum\_Welch:

I have written all the functions. But while beta calculation, probabilities becomes NAN because of very low scale factor being returned while alpha calculation. So I could not move ahead. The functions are:

update\_pi (gamma\_all)

update\_A(gamma\_all,alpha\_All,beta\_All,observation\_all)

update\_C(gamma\_all,previous\_mix\_coeff,B\_all,mu\_list,cov\_list,observation\_all)

update\_mu(gamma\_all,previous\_mix\_coeff,B\_all,mu\_list,cov\_list,observation\_all)

update\_covs(gamma\_all,previous\_mix\_coeff,B\_all,mu\_list,cov\_list,observation\_all)