DT2119 Lab2: HMM with Gaussian Emissions

Zhanpeng Xie, zxie@kth.se

4 Concatenating HMMs

Transition matrix of 'sil'

	0	1	2	3
0	0.883311	0.116689	0	0
1	0	0.916447	0.0835535	0
2	0	0	0.802877	0.197123
3	0	0	0	1

Transition matrix of 'ow'

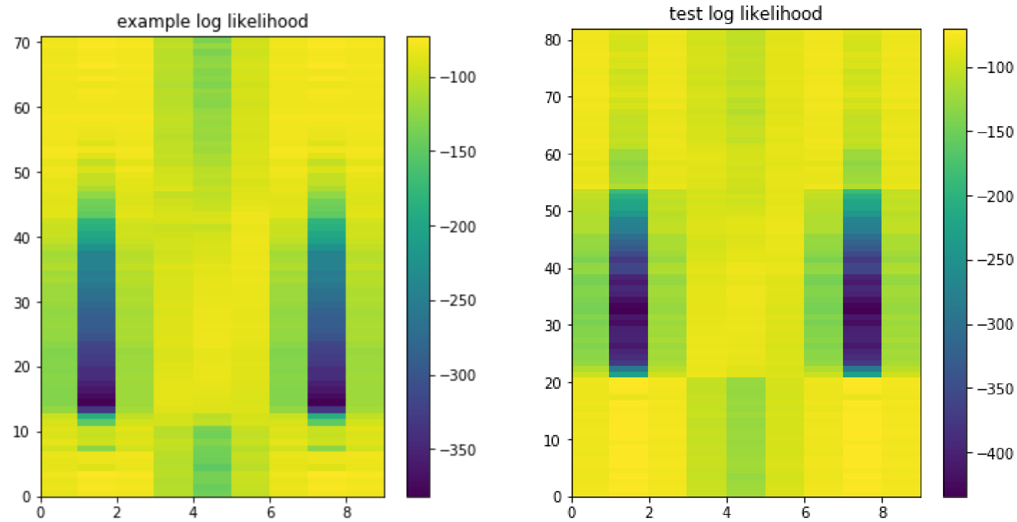
	0	1	2	3
0	0.814828	0.185172	0	0
1	0	0.843619	0.156381	0
2	0	0	0.866247	0.133753
3	0	0	0	1

Transition matrix of word 'o'

	0	1	2	3	4	5	6	7	8	9
0	0.883311	0.116689	0	0	0	0	0	0	0	0
1	0	0.916447	0.0835535	0	0	0	0	0	0	0
2	0	0	0.802877	0.197123	0	0	0	0	0	0
3	0	0	0	0.814828	0.185172	0	0	0	0	0
4	0	0	0	0	0.843619	0.156381	0	0	0	0
5	0	0	0	0	0	0.866247	0.133753	0	0	0
6	0	0	0	0	0	0	0.883311	0.116689	0	0
7	0	0	0	0	0	0	0	0.916447	0.0835535	0
8	0	0	0	0	0	0	0	0	0.802877	0.197123
9	0	0	0	0	0	0	0	0	0	1

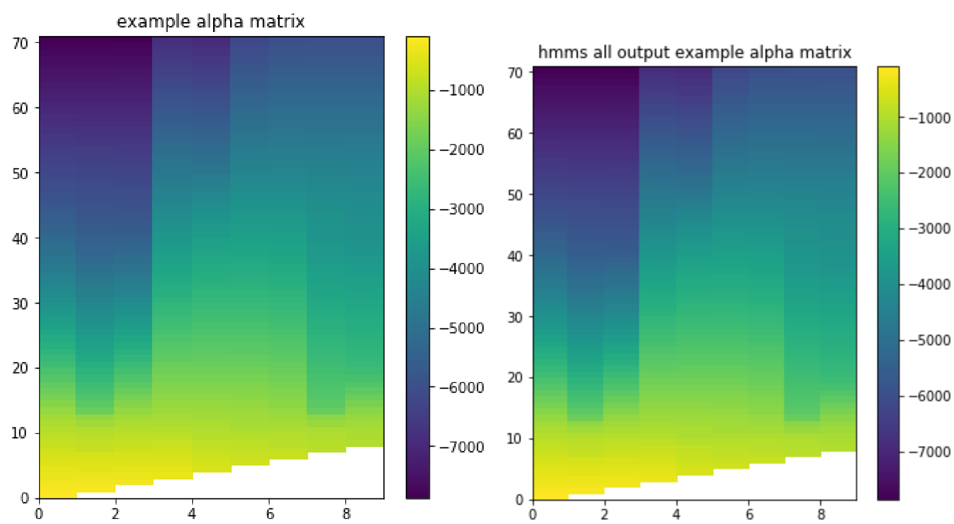
5.1 Gaussian emission probabilities

Log likelihood for example lmfcc and a test utterance of ‘o’ from a male speaker:



Even though the speakers of these two samples have different genders, the figures are very similar to each other. For the example case, components of ‘sil’ have high likelihood for time step= (0, 15) and (50, 70) and component of ‘ow’ has have high likelihood for time step= (15, 50). For the test case, components of ‘sil’ have high likelihood for time step= (0, 20) and (55, 80) and component of ‘ow’ has have high likelihood for time step= (20, 55). The reason for these patterns is the isolated utterances starts and end with silence which corresponds to the component of ‘sil’.

5.2 Forward Algorithm



The result from my implementation is the same as the result in the given example.

Testing 44 utterances:

True Label:

['o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9']

Prediction from lab2_models_all.npz:

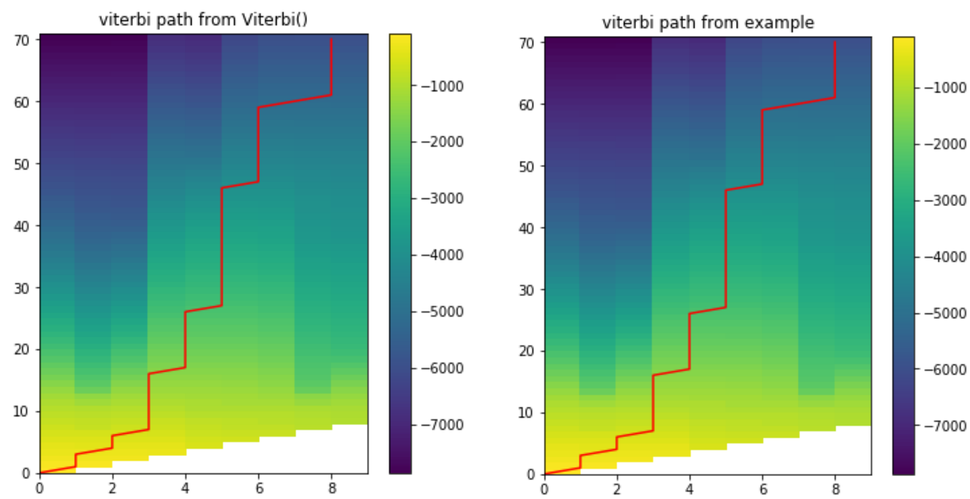
['o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '1', '9']

Prediction from lab2_models_onespkr.npz:

['o', 'o', 'z', 'z', '3', '9', '3', '2', '3', '3', '4', '4', '3', '3', '6', '6', '3', '3', '3', '3', '9', 'z', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9']

From the prediction above, using models all training speakers only made 1 mistake out of 44 data while the models trained on a single speaker made 10 mistakes. I think the single speaker model performed much worse on the data from different speakers.

5.3 Viterbi Approximation



Explanations for the Viterbi path:

State 0-2 and 3-5 stand for 'sil' and state 6-9 stands for 'ow'. The components for word 'o' are ['sil', 'ow', 'sil']. Also the hmm models we have here can't go back to previous state.

Testing all 44 utterances in the data:

True Label:

['o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9']

Prediction from lab2_models_all.npz:

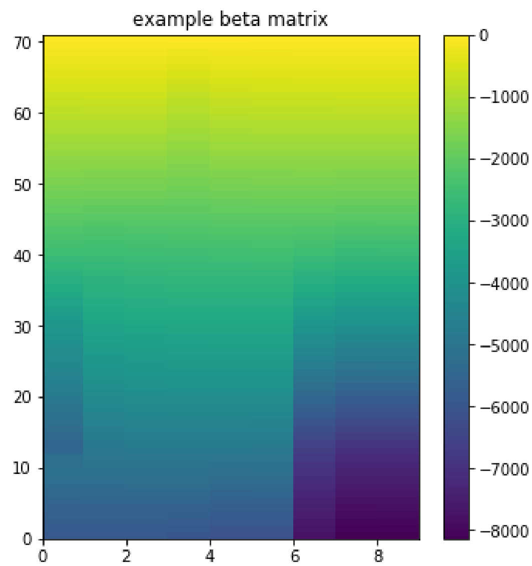
['o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9']

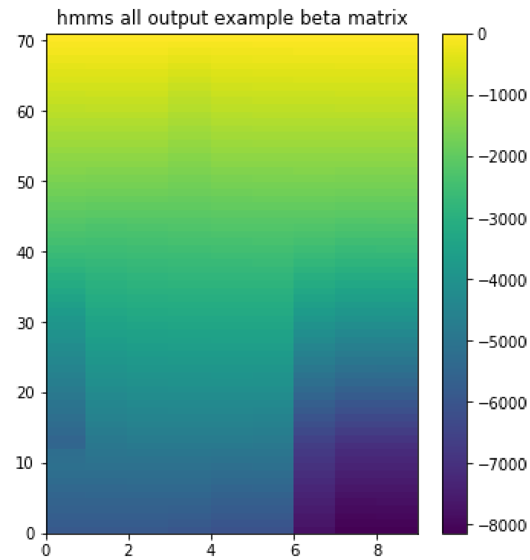
Prediction from lab2_models_onespkr.npz:

['o', 'o', 'z', 'z', '3', '9', '3', '2', '3', '3', '4', '4', '3', '3', '6', '6', '3', '3', '3', '3', '9', 'z', 'o', 'o', 'z', 'z', '1', '1', '2', '2', '3', '3', '4', '4', '5', '5', '6', '6', '7', '7', '8', '8', '9', '9']

The mistakes for prediction from lab2_models_onespkr.npz are the same. But mistakes for prediction from lab2_models_all.npz was corrected. Viterbi is much faster since logsumexp is much slower than np.max.

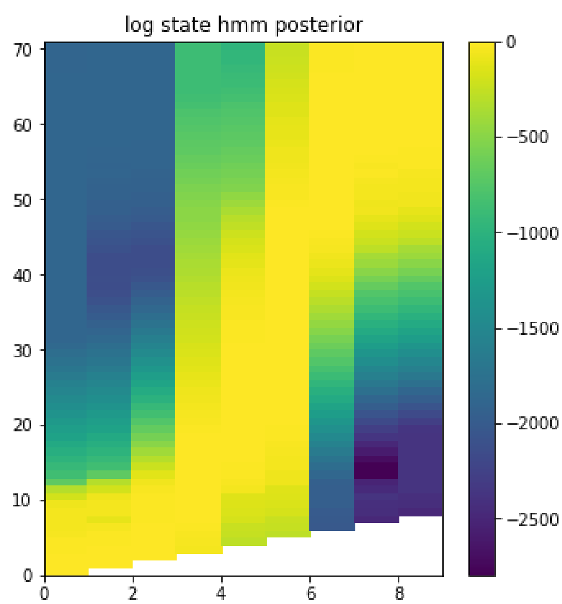
5.4 Backward algorithm

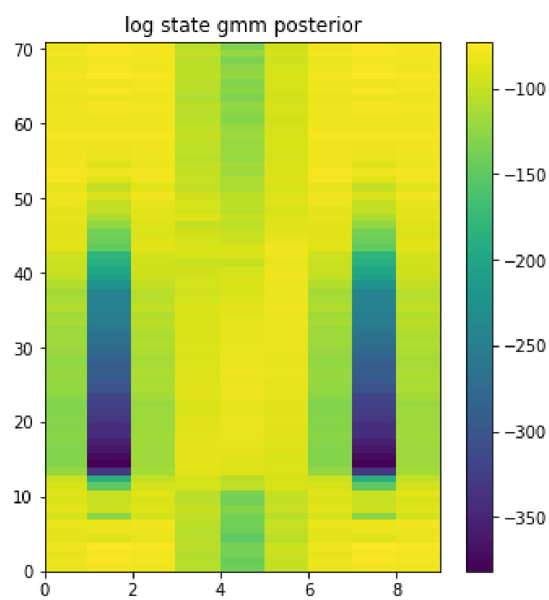




The figures above shown I have finished the backward() correctly.

6.1 State posterior probabilities





Difference of HMM and GMM posteriors:

HMM posteriors figure has zero probability mass at early time step for latter states. HMM has start probabilities as prior probabilities which ensures the model won't start from the some intermidiate states.

Summing over time steps:

[1.34597458 2.09555994 3.55849998 9.73752276 10.12040999 20.5329264

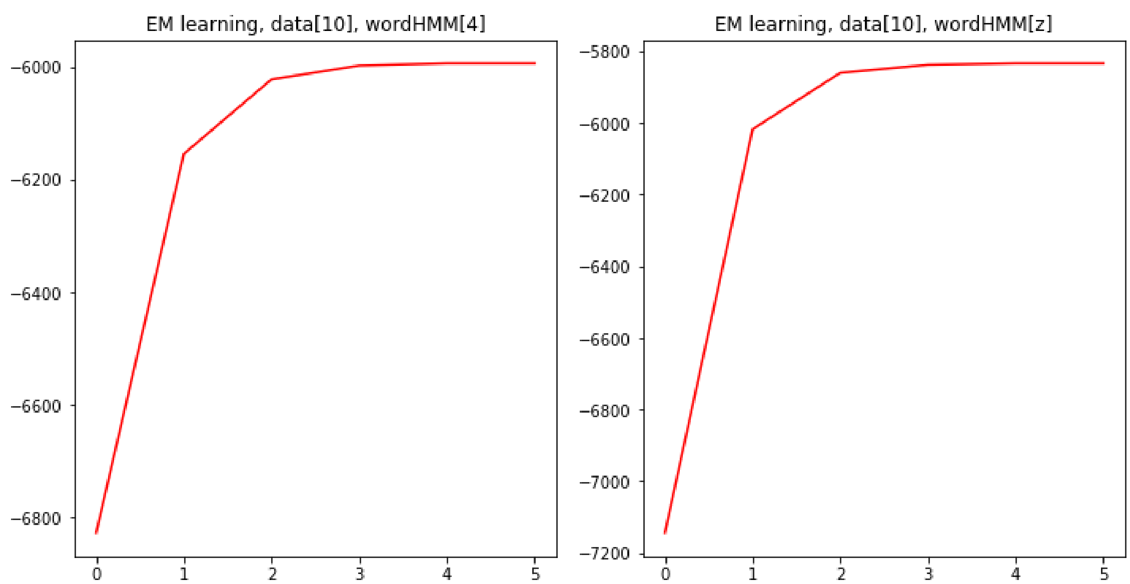
 12.9968938   1.20869929 9.40351326]

Meaning: unnormalized probability of the chance of a randomly selected frame belongs to the corresponding state.

Summing over both states and time steps:

The value equals to the length of the number of frames.

6.2 retraining the emission probability distributions



The log likelihood increased monotonously which is one of the properties of EM algorithm. The model converged very quickly, they took 5 iterations to converge in both cases.