[M2, MVA]

Probabilistic graphical models

Maha ELBAYAD
maha.elbayad@student.ecp.fr

Homework 3

December 29, 2015

HMM implementation

1. After evaluating the forward/backward messages $\alpha, \beta \in \mathbb{R}^{T\times K}$ we compute the probabilities denoted P_q and P_{qq} defined as:

$$P_q(t,i) = \mathbb{P}(q_t = i|\bar{u}_1....\bar{u}_T) = \frac{\alpha(t,i)\beta(t,i)}{\sum_{j=1}^{K}\alpha(t,j)\beta(t,j)}$$

$$P_{qq}(i,j,t) = \mathbb{P}(q_t = i, q_{t+1} = j|\bar{u}_1....\bar{u}_T) = \frac{\alpha(t,i)\beta(t+1,j)A_{i,j}f_j(\bar{u}_{t+1})}{\mathbb{P}(\bar{u}_1,...,\bar{u}_T)}$$

with A the probability transition matrix $\in \mathbb{R}^{K\times K}$.
And $f_i(x)$ is the density of the gaussian distribution corresponding to the i^{th} state:

$$\forall x \in \mathbb{R}^p,\ f_i(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}}\exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]$$

2. Using the parameters learned in the previous homework with uniform initial probability π
And:

$$A = \frac{1}{6}\begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}$$

We compute the messages and get the following P_q for the first 100 samples.

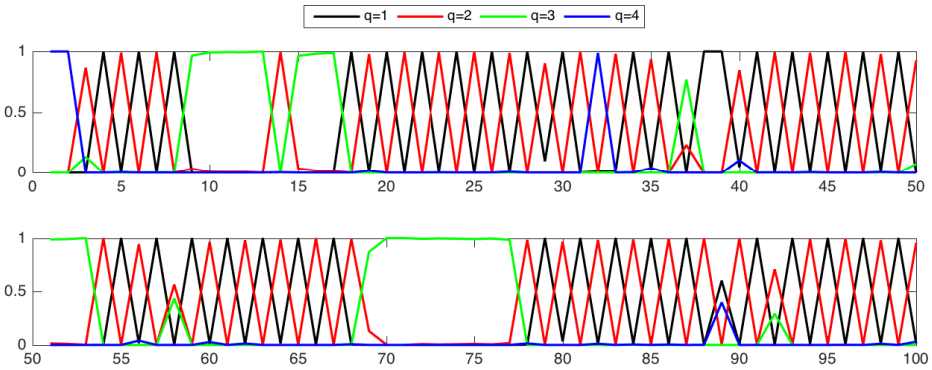


Figure 1: Hidden variables probability t=1:100 on the test data

3. The log-likelihood of the model with parameters $\theta = (\pi, A, (\mu_i)_i, (\Sigma_i)_i)$ takes the value:

$$\begin{aligned} l(\theta) &= \log\left(\mathbb{P}(q_1)\prod_{t=1}^{T-1}\mathbb{P}(q_{t+1}|q_t)\prod_{t=1}^{T}\mathbb{P}(\bar{u}_t|q_t)\right) \\ &= \sum_{i=1}^{K}\delta(q_1=i)\log(\pi_i) + \sum_{t=1}^{T-1}\sum_{i,j=1}^{K}\delta(q_t=i,q_{t+1}=j)\log(A_{ij}) \\ &+ \sum_{t=1}^{T}\sum_{i=1}^{K}\delta(q_t=i)\log(f_i(\bar{u}_t)) \end{aligned}$$

At the (E) step of the expectation-maximization algorithm, we substitute the terms δ with the probabilities evaluated with the messages α, β. At the (M) step, the optimizaton problem is similar to the one seen in homework 2, the updates are as follows:

$$\pi_i = P_q(1,i)$$

$$A_{i,j} = \frac{\sum_{t=1}^{T} P_{qq}(i,j,t)}{\sum_{t=1}^{T}\sum_{j=1}^{K} P_{qq}(i,j,t)}$$

$$\mu = P_q^T U \in \mathbb{R}^{K\times p}, \quad U \text{ being the matrix of rows } (u_t)_t$$

$$\Sigma_i = \frac{\sum_{t=1}^{T} P_q(t,i)(\bar{u}_t-\mu_i)^T(\bar{u}_t-\mu_i)}{\sum_{t=1}^{T} P_q(t,i)}$$

5. While training the HMM we keep track of the log-likelihoods on the training set as well as on the test set. We compare in the figure below two initializations: `-GM` initialized with the Gaussian mixture parameters of homework 2 and `-RI` a random initialization of $(\Sigma_i)_i, (\mu_i)_i$.

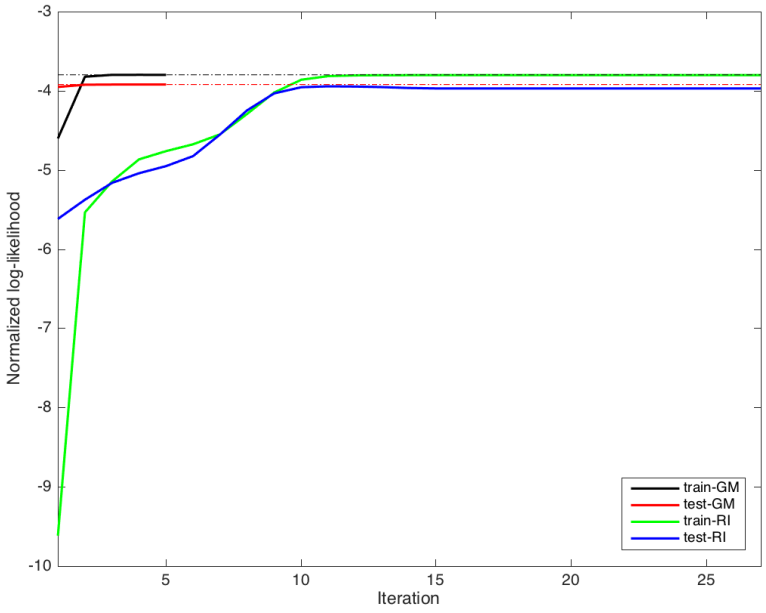


Figure 2: log-likelihood evolutions on the train/test sets

6. Comparison:
We note that with the GM initialization we're tuning the HMM model in only 5 iterations, a random initialization might take a longer time $\sim$ 25 iterations to converge, but still considerably few iterations compared to $\sim$ 75 for the gaussian mixture. Comparing the HMM model loglikelihood to the gaussian mixture likelihood doesn't make much sense since the assumptions are different as the data is considered i.i.d for the GM but with temporal structure in HMM.

	Train	Test
Isotropic GM	-5.31	-5.29
General GM	-4.66	-4.82
HMM from GM	-3.80	-3.92
HMM from rand	-3.80	-3.97

Table 1: Normalized log-likelihoods

7. Given the observations $\{u_1, ..., u_T\}$, the states $\{1, ..K\}$, the transition matrix A and the emission matrix N (in this case gaussian probabilities).
The outputs are: $V, P \in \mathbb{R}^{T \times K}$ the probabilities of the most likely path and the path itself MAP.

Viterbi pseudocode:

Initialization:

for $k \in \{1...K\}$ **do**

  $V_{1,k} = \mathbb{P}(u_1|q_1 = k)\pi_k$

  $P_{1,k} = 0$

end

Main loop:

for $t \in \{1...T\}$ **do**

  for $k \in \{1...K\}$ **do**

    $V_{t,k} = \max_{i\in\{1..K\}} \mathbb{P}(u_t|q_t = k)A_{i,k}V_{t-1,i}$

    $P_{t,k} = \arg\max_{i\in\{1..K\}} \mathbb{P}(u_t|q_t = k)A_{i,k}V_{t-1,i}$

  end

end

Retrieve the path:

$MAP_T = \arg\max_k V_{T,k}$

for $t \in \{T...2\}$ **do**

  $MAP_{t-1} = P_{t,MAP_t}$

end

8. We compute the most likely path of states with the Viterbi algorithm and classify the train data accordingly.

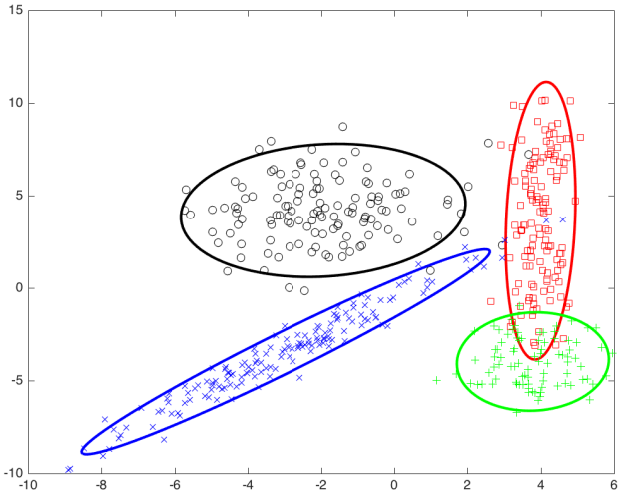


Figure 3: Viterbi Classification of the train set

9. We compute the marginal probability $\mathbb{P}(q_t|u_1, ..., u_T)$ for each $t \in \{1..100\}$ based on the parameters learnt from the train set.

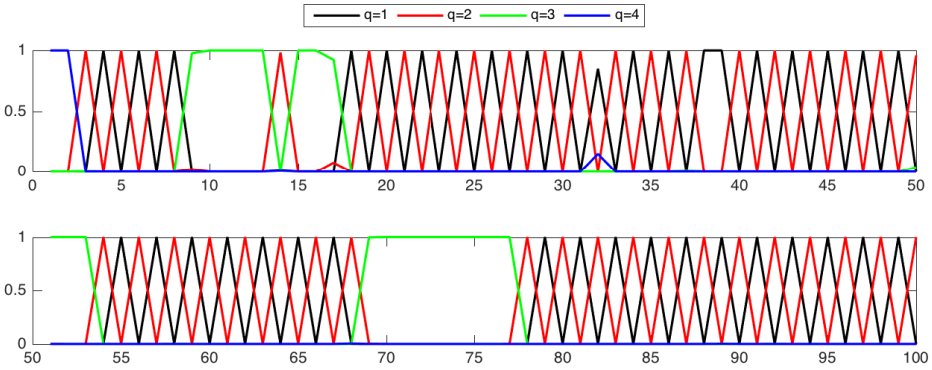
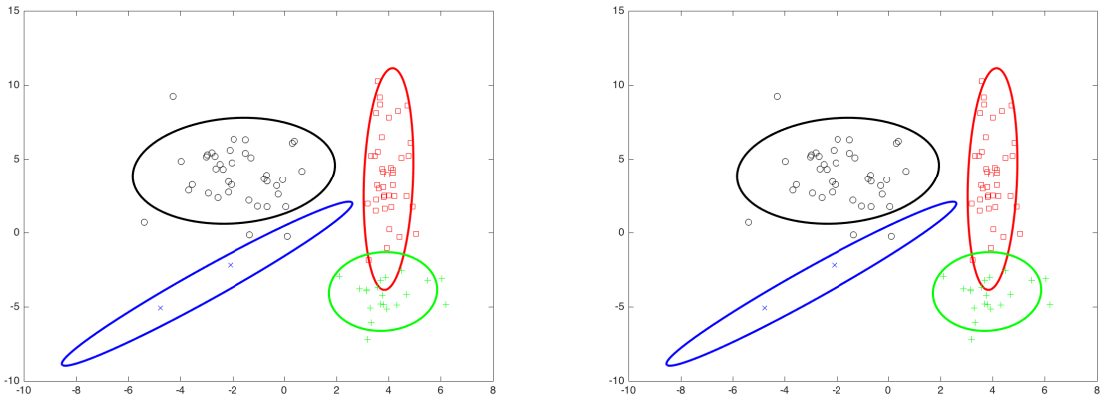


Figure 4: Hidden variables probability t=1:100 on the test data

10. We classify the test data using the previously computed marginal probabilities (figure 5a).



(a) Marginal probabilities

(b) Viterbi

11. We compute the most likely path of states with the Viterbi algorithm using the paramaters learned from the training set and classify test data accordingly (figure 5b). We note that the classification results with Viterbi's algorithm are exactly similar to the classification issued with marginal probabilities.

12. For choosing the number of hidden states K we can train different models and compare their likelihoods on a validation set. For K ranging from 2 to 7 we get the likelihoods shown in figure 5. As we can see the appropriate choice is K=4 as it doesn't overfit the training set. Moreover, for $K > 4$ the model gets more complex, with more iterations needed to converge, yet the training set likelihood has slightly plateaued.

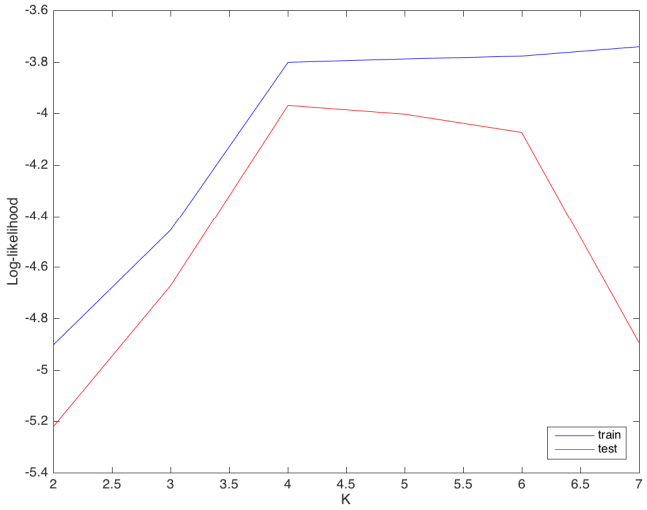


Figure 5: Cross-validation of K

K	2	3	4	5	6	7
Iterations	13	30	27	36	47	65

Table 2: Required iterations for tolerance = 10^{-10}