Assignment Four

Matthew C. Scicluna
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
Montréal, QC H3T 1J4
matthew.scicluna@umontreal.ca

November 18, 2017

1 Entropy and Mutual Information

Let X be a discrete random variable on a finite space $\mathcal{X}$ with $|\mathcal{X}| = k$.

1. (a) Prove that the entropy $H(X) \geq 0$, with equality only when X is a constant.

   PROOF: WLOG we can assume that $p(x) > 0 \ \forall x \in \mathcal{X}$, (using that $0 \cdot log 0 = 0$). We have that $H(X) = -\sum_x p(x) log p(x) = \sum_x p(x) log p(x)^{-1} \geq 0$, since $p(x) > 0$ and $p(x)^{-1} \geq 1$. If $H(X) = 0$ then $\exists \alpha$ such that $p(\alpha)^{-1} = 1 \Rightarrow p(\alpha) = 1$. Hence X must be a constant, as needed.

   (b) Let $X \sim p$ and q be the Uniform distribution on $\mathcal{X}$. What is the relation between $D(p||q)$ and $H(X)$.

   CLAIM: $D(p||q) = -H(X) + log k$

   PROOF:

$$\begin{aligned}
D(p||q) &= \sum_x p(x) log \frac{p(x)}{q(x)} \\
&= \sum_x p(x) log p(x) - \sum_x p(x) log q(x) \\
&= -H(X) + \sum_x p(x) log k \\
&= -H(x) + log k
\end{aligned}$$

(c) An upper bound for $H(X)$ is $logk$ since $H(X) = logk - D(p||q) \Rightarrow H(X) \leq logk$.

We consider a pair of discrete random variables (X_1, X_2) defined over the finite set $\mathcal{X}_1 \times \mathcal{X}_2$. Let $p_{1,2}$, p_1 and p_2 denote respectively the joint distribution, the marginal distribution of X_1 and the marginal distribution of X_2. Define the mutual information as:

$$I(X_1, X_2) = \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1, x_2) log \frac{p_{1,2}(x_1, x_2)}{p_1(x_1)p_2(x_2)}$$

We again assume WLOG that $p(x_1, x_2) > 0 \; \forall (x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$

2. (a) CLAIM: $I(X_1, X_2) \geq 0$

   PROOF: Notice that $I(X_1, X_2) = D(p_{1,2}||p_1 p_2) \geq 0$ by the positiveness of $D(\cdot||\cdot)$.

   (b) We want to express $I(X_1, X_2)$ as a function of $H(X_1), H(X_2)$ and $H(X_1, X_2)$.

$$\begin{aligned}
I(X_1, X_2) &= \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1, x_2) log \frac{p_{1,2}(x_1, x_2)}{p_1(x_1)p_2(x_2)} \\
&= \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1, x_2) log p_{1,2}(x_1, x_2) - p_{1,2}(x_1, x_2) log p_1(x_1)p_2(x_2) \\
&= -H(X_1, X_2) - \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} \Big(p_1(x_1)p_2(x_2) log p_1(x_1) - p_1(x_1)p_2(x_2) log p_2(x_2) \Big) \\
&= -H(X_1, X_2) - \sum_{j=1}^{2} \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_1(x_1)p_2(x_2) log p_j(x_j) \\
&= -H(X_1, X_2) - \sum_{j=1}^{2} \sum_{x_j\in\mathcal{X}_j} p_j(x_j) log p_j(x_j) \\
&= -H(X_1, X_2) + H(X_1) + H(X_2)
\end{aligned}$$

   And so we can represent $I(X_1, X_2)$ using $H(X_1), H(X_2)$ and $H(X_1, X_2)$, as needed.

   (c) From the previous result we have that $I(X_1, X_2) \geq 0 \Rightarrow H(X_1) + H(X_2) \geq H(X_1, X_2)$, and so the maximal entropy of (X_1, X_2) is $H(X_1)+H(X_2)$. By definition this only occurs when $I(X_1, X_2) = 0$, which only occurs if $p_{1,2}(x_1, x_2) = p_1(x_1)p_2(x_2) \; \forall x_1, x_2 \in \mathcal{X}_1 \times \mathcal{X}_2$. This can be seen directly from the definition of I and using the strict positivity of $p(x_1, x_2)$.

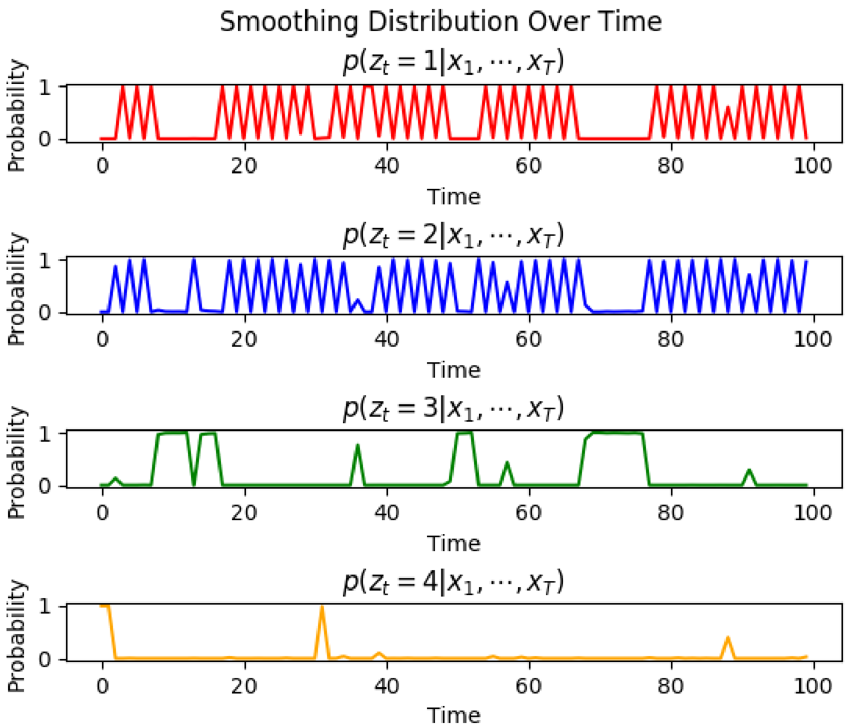

Figure 2.1: The smoothing distribution for the first 100 time points, using the parameters learned in the previous homework.

2 HMM – Implementation

We use an HMM model to account for the possible temporal structure of some data. We consider the following HMM model: the chain $(z_t)_{t=1}^T$ has $K = 4$ possible states, with an initial probability distribution $\pi \in \Delta_4$ and a probability transition matrix $A \in \mathbb{R}^{4\times4}$ where $A_{ij} = p(z_t = i|z_{t-1} = j)$ and conditionally on the current state z_t, we have observations obtained from Gaussian emission probabilities $x_t|(z_t = k) \sim N(x_t|\mu_k, \Sigma_k)$.

2.1 Fake Parameters Inference

We computed the vectors $\alpha(z_t) = p(z_t, x_{1:t})$ and $\beta(z_t) = p(x_{(t+1):T}|z_t)$ on the test set from Assignment 3 using the following parameters: $\pi_k = \frac{1}{4}$, $A_{ii} = \frac{1}{2}$ and $A_{ij} = \frac{1}{6}, i \neq j$, and μ_k, Σ_k as defined in the homework. We used these to compute the posterior of the latent variable over time $p(z_t \mid x_1, \cdots, x_T)$. We plotted the first 100 datapoints in figure 2.1.

2.2 M-Step Derivation

We now derive the M-Step for the Hidden Markov Model.
Let $\theta^{(s)} = (\pi^{(s)}, A^{(s)}, \mu_1^{(s)}, \cdots, \mu_K^{(s)}, \Sigma_1^{(s)}, \cdots, \Sigma_K^{(s)})$ be the ML parameters learned during step s of EM. Let $\gamma_{tk} = P(z_t = k|x_{1:T})$ and $\xi_{tlm} = P(z_t = l, z_{t+1} = m|x_{1:T})$ - which are the quantities that were computed in the E-Step using $\theta^{(s)}$. The Expected Complete Data Log-Likelihood (at step $s+1$) is:

$$Q(\theta, \theta^{(s)}) = \sum_{k=1}^{K} \gamma_{1k} log \pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_{tk} log P(\bar{x}_t|\mu_k, \Sigma_k) + \sum_{t=1}^{T-1}\sum_{l=1}^{K}\sum_{m=1}^{K} \xi_{tlm} log A_{lm} \tag{2.1}$$

To solve for π_k we look for stationary points, subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$

$$\frac{\partial}{\partial \pi_k} Q(\theta, \theta^{(s)}) - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) = \frac{\gamma_{1k}}{\pi_k} - \lambda = 0 \tag{2.2}$$

$$\Rightarrow \pi_k = \frac{\gamma_{1k}}{\lambda} \tag{2.3}$$

And using that $\sum_{k=1}^{K} \pi_k = 1$ we have that:

$$\pi_k^{(s+1)} = \frac{\gamma_{1k}}{\sum_{l=1}^{K} \gamma_{1l}} \tag{2.4}$$

To solve for A_{lm} we look for stationary points, subject to the constraint $\sum_{l=1}^{K} A_{lm} = 1$

$$\frac{\partial}{\partial A_{lm}} Q(\theta, \theta^{(s)}) - \lambda \left(\sum_{l=1}^{K} A_{lm} - 1 \right) = \sum_{t=1}^{T-1} \frac{\xi_{tlm}}{A_{lm}} - \lambda = 0 \tag{2.5}$$

$$\Rightarrow A_{lm} = \frac{\sum_{t=1}^{T-1} \xi_{tlm}}{\lambda} \tag{2.6}$$

And using that $\sum_{l=1}^{K} A_{lm} = 1$ we have that:

$$A_{lm}^{(s+1)} = \frac{\sum_{t=1}^{T-1} \xi_{tlm}}{\sum_{l=1}^{K}\sum_{t=1}^{T-1} \xi_{tlm}} \tag{2.7}$$

To solve for μ_k we look for stationary points

$$\frac{\partial}{\partial \mu_k} Q(\theta, \theta^{(s)}) = \sum_{t=1}^{T} \frac{\partial}{\partial \mu_k} \frac{-\gamma_{tk}}{2} (\bar{x}_t - \mu_k)^T \Sigma_k^{-1} (\bar{x}_t - \mu_k) \tag{2.8}$$

$$= \sum_{t=1}^{T} \frac{-\gamma_{tk}}{2} (\bar{x}_t - \mu_k) \Sigma_k^{-1} = 0 \tag{2.9}$$

$$\Rightarrow \sum_{t=1}^{T-1} \gamma_{tk} (\bar{x}_t - \mu_k) = 0 \tag{2.10}$$

$$\Rightarrow \mu_k^{(s+1)} = \frac{\sum_{t=1}^{T} \gamma_{tk} \bar{x}_t}{\sum_{t=1}^{T-1} \gamma_{tk}} \tag{2.11}$$

To solve for Σ_k we look for stationary points. As in assignment 2, we take the derivative w.r.t Σ_k^{-1}

$$\frac{\partial}{\partial \Sigma_k^{-1}} Q(\theta, \theta^{(s)}) = \sum_{t=1}^{T} \frac{\partial}{\partial \mu_k} \left(\frac{-\gamma_{tk}}{2} log|\Sigma_k| - \frac{\gamma_{tk}}{2} \left(\bar{x}_t - \mu_k^{(s+1)}\right)^T \Sigma_k^{-1} \left(\bar{x}_t - \mu_k^{(s+1)}\right)\right) \tag{2.12}$$

Differentiating the first term yields

$$\frac{\partial}{\partial \Sigma_k^{-1}} \frac{-\gamma_{tk}}{2} log|\Sigma_k| = \frac{\partial}{\partial \Sigma_k^{-1}} \frac{\gamma_{tk}}{2} log|\Sigma_k^{-1}| \tag{2.13}$$

$$= \frac{\gamma_{tk}}{2} \Sigma_k \tag{2.14}$$

Differentiating the second term yields

$$\frac{\partial}{\partial \Sigma^{-1}} \frac{\gamma_{tk}}{2} \left(\bar{x}_t - \mu_k^{(s+1)}\right)^T \Sigma_k^{-1} \left(\bar{x}_t - \mu_k^{(s+1)}\right) \tag{2.15}$$

$$= \frac{\partial}{\partial \Sigma^{-1}} \frac{\gamma_{tk}}{2} tr \left(\left(\bar{x}_t - \mu_k^{(s+1)}\right) \left(\bar{x}_t - \mu_k^{(s+1)}\right)^T \Sigma_k^{-1}\right) \tag{2.16}$$

$$= \frac{\gamma_{tk}}{2} \left(\bar{x}_t - \mu_k^{(s+1)}\right) \left(\bar{x}_t - \mu_k^{(s+1)}\right)^T \tag{2.17}$$

Where (2.15) and (2.16) come from results proved in class and on Homework 2. After substituting (2.14) and (2.17) into (2.12) and equating to 0 we have that:

$$\sum_{t=1}^{T} \gamma_{tk} \Sigma_k = \sum_{t=1}^{T} \gamma_{tk} \left(\bar{x}_t - \mu_k^{(s+1)}\right) \left(\bar{x}_t - \mu_k^{(s+1)}\right)^T \tag{2.18}$$

$$\Rightarrow \Sigma_k^{(s+1)} = \frac{\sum_{t=1}^{T} \gamma_{tk} (\bar{x}_t - \mu_k^{(s+1)})(\bar{x}_t - \mu_k^{(s+1)})^T}{\sum_{t=1}^{T} \gamma_{tk}} \tag{2.19}$$

2.3 Deriving Parameters using EM Algorithm

We implemented the EM algorithm to learn the parameters of the model, initializing them with the values provided in the homework. We trained the model on the EMGaussians.train dataset. The transition matrix found is as follows:

$$A = \begin{bmatrix} 0.0158 & 0.863 & 0.067 & 0.052 \\ 0.947 & 0.0226 & 0.022 & 0.0266 \\ 0.0289 & 0.027 & 0.874 & 0.0195 \\ 0.0087 & 0.0873 & 0.0371 & 0.902 \end{bmatrix}$$

The rest of the parameters are provided in table 2.1:

Table 2.1: EM Parameter Values

Cluster	π	μ	Σ
1	0	$\begin{bmatrix} -1.94 \\ 4.2 \end{bmatrix}$	$\begin{bmatrix} 3.34 & 0.32 \\ 0.32 & 2.84 \end{bmatrix}$
2	0	$\begin{bmatrix} 4.0 \\ 3.64 \end{bmatrix}$	$\begin{bmatrix} 0.197 & 0.275 \\ 0.275 & 12.4 \end{bmatrix}$
3	0	$\begin{bmatrix} 3.79 \\ -3.97 \end{bmatrix}$	$\begin{bmatrix} 0.95 & 0.077 \\ 0.077 & 1.58 \end{bmatrix}$
4	1	$\begin{bmatrix} -2.96 \\ -3.44 \end{bmatrix}$	$\begin{bmatrix} 6.88 & 6.66 \\ 6.66 & 6.75 \end{bmatrix}$

2.4 Log Likelihood of Train and Test Data

The log likelihood for the training and test data was computed and is displayed in figure 2.2. The log likelihood of the training set is consistently higher then on the test set. This makes sense since the model is being trained to (indirectly) maximize the likelihood of the training set.

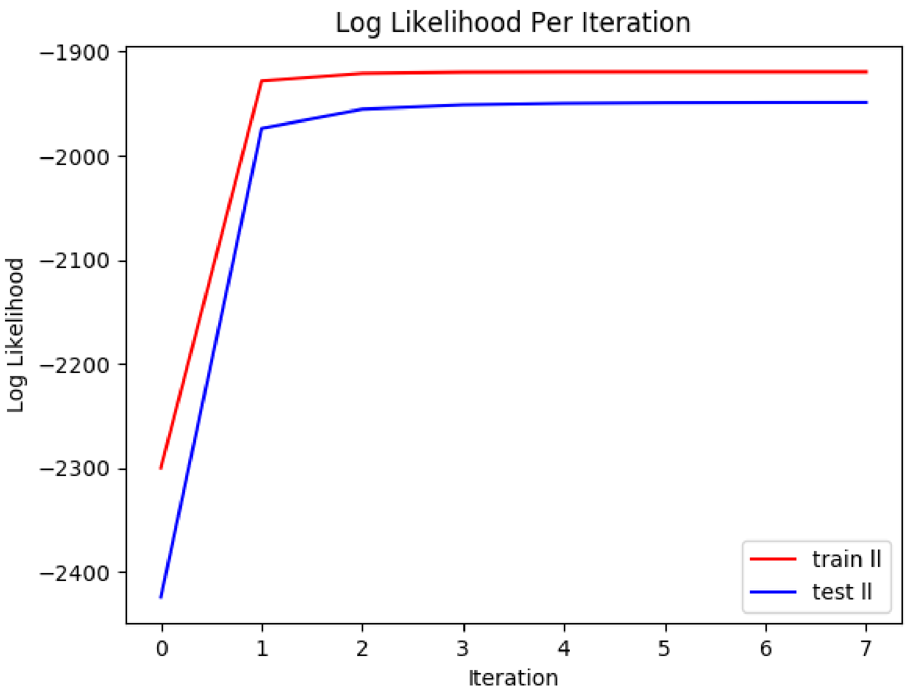

Figure 2.2: The log likelihood computed for each iteration of the EM algorithm.

Table 2.2: Training and Test Log Likelihood for Various Models

Model	Train LL	Test LL
HMM	-1919	-1949
GMM	0	0

2.5 Comparing Previous Models

We compared the values of log-likelihoods of the Gaussian mixture model and of the HMM on the train and test data. The results are presented in table 2.2.
Compare these values. Does it make sense to make this comparison? Conclude. Compare these log-likelihoods as well with the log-likelihoods obtained for the different models in the previous homework.

2.6 Viterbi Decoding Pseudocode

Let $x_{1:T}$ be observations and $z_{1:T}$ be hidden states, where $z_t \in \{1, \cdots, K\}$. Define $V, P \in \mathbb{R}^{T \times K}$ where V_{ij} is the probability of the most likely path $z_1, \cdots z_i$ with $z_i = j$ and P_{ij} is the value of the corresponding z_{i-1}. Let $O_{ij} = P(\bar{x}_i | z_i = j)$. The algorithm is presented below.

Algorithm 1 Calculate $\arg\max_{z_{1:T}} P(z_{1:T} | \bar{x}_{1:T})$

for state $j \in \{1, \cdots, K\}$ **do**
  $V_{1j} \leftarrow \pi_j O_{1j}$ ▷ $P(\bar{x}_1, z_1 = j)$
  $P_{1j} \leftarrow -1$ ▷ initialize with anything
for time $i \in \{2, \cdots, T\}$ **do**
  for state $j \in \{1, \cdots, K\}$ **do**
    $V_{ij} \leftarrow \max_k O_{ij} A_{kj} V_{i-1,k}$
    $P_{ij} \leftarrow \arg\max_k O_{ij} A_{kj} V_{i-1,k}$ ▷ $P(\bar{x}_i | z_i = k) P(z_i = k | z_{i-1} = j) V_{i-1,k}$
$z_T \leftarrow \arg\max_k V_{Tk}$
for $i \in \{T, T-1, \cdots 2\}$ **do**
  $z_{i-1} \leftarrow V_{i, z_i}$ ▷ backtracking
return $z_1, \cdots, z_T$

The algorithm works since

$$\arg\max_{z_{1:T}} P(z_{1:T}|\bar{x}_{1:T}) \tag{2.20}$$

$$= \arg\max_{z_{1:T}} P(\bar{x}_{1:T}, z_{1:T}) \tag{2.21}$$

$$= \arg\max_{z_T} \left\{ \arg\max_{z_{T-1}} P(\bar{x}_T|z_T)P(z_T|z_{T-1}) \left\{ \arg\max_{z_{T-2}} P(\bar{x}_{T-1}|z_{T-1}) \cdots \right\}\right\} \tag{2.22}$$

$$= \arg\max_{z_T} \arg\max_{z_{T-1}} O_{T,Z_T} A_{z_T,z_{T-1}} V_{T-1,z_{T-1}} \tag{2.23}$$

Which is essentially a forward pass with max operation instead of sum. Steps $4-7$ computes the needed V and P. Step 8 uses these to compute (2.20), returning z_T. Steps $9-10$ use P and z_T to recover $z_{1:T-1}$.

2.7 Viterbi Decoding Implementation

We implemented Viterbi decoding. We used the parameters learned with the EM algorithm to compute the most likely sequence of states with the Viterbi algorithm. The results are presented graphically in Figure 2.3.

2.8 Marginal Probability Computations

For the datapoints in EMGaussian.test, we computed the marginal probability $p(z_t|x_1, \cdots, x_T)$ for each point to be in state $\{1, 2, 3, 4\}$ using the parameters learned on the training set. For each state, we plotted the probability of being in that state as a function of time. The results for the first 100 datapoints are presented in figure 2.4. We then computed their most likely state according to $p(z_t|x_1, \cdots, x_T)$ and using Viterbi decoding. The results are presented in Figure 2.5. Note that the most likely sequence of states coincided for the first 100 datapoints (although the did differ at t=448).

2.9 How to Train the Model With Unknown Number of Clusters

If the number of states were unknown, there are several ways to be able to choose the number of clusters. One such way would be to run the above with different numbers of clusters.

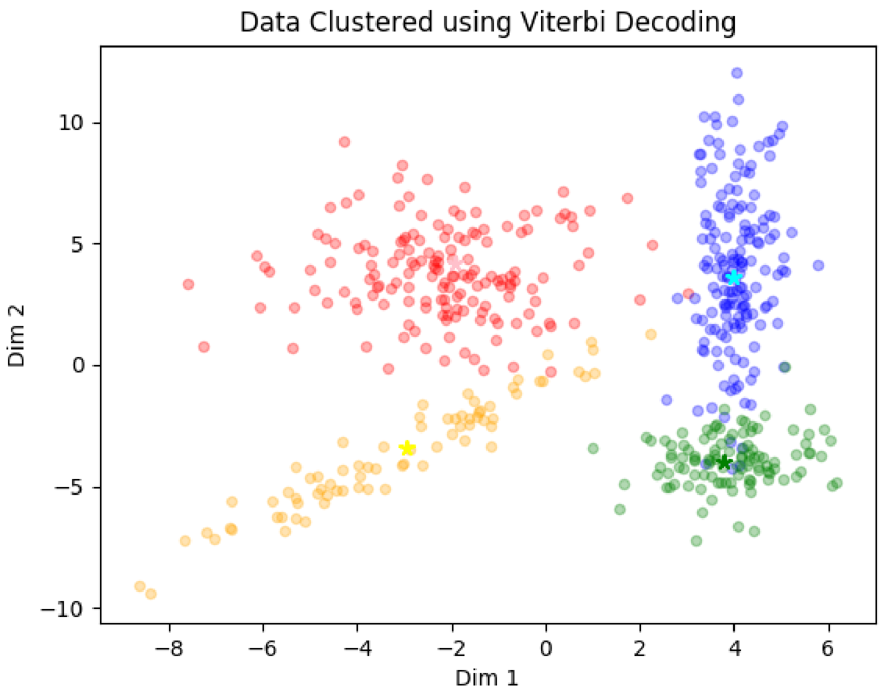

Figure 2.3: The training data is presented along with the cluster means, inferred from EM. The datapoints are colored based on each datapoints' Viterbi decoded cluster assignment.

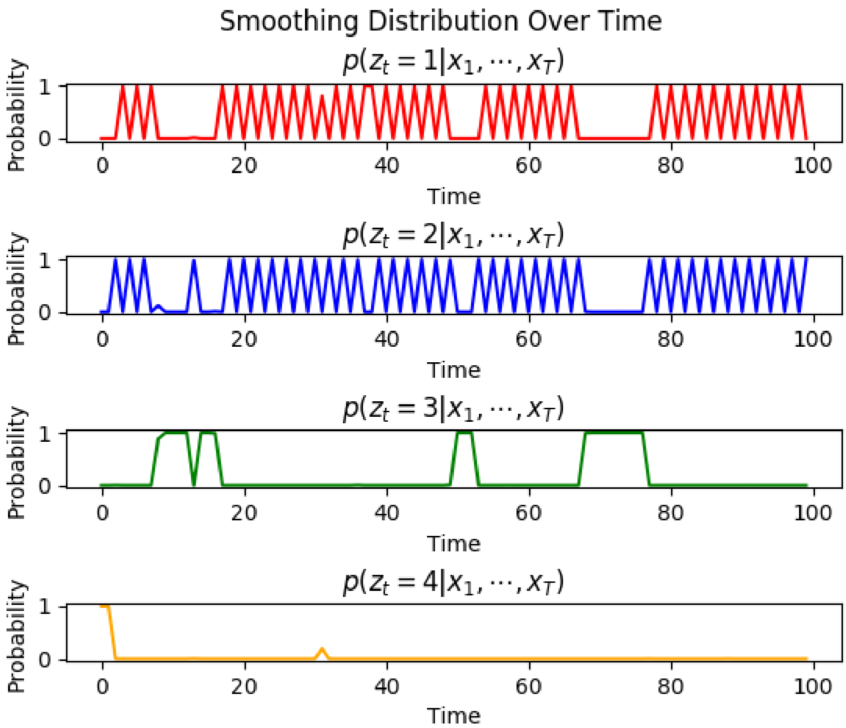

Figure 2.4: The smoothing distribution for the first 100 time points of the test set, using the parameters learned with EM.

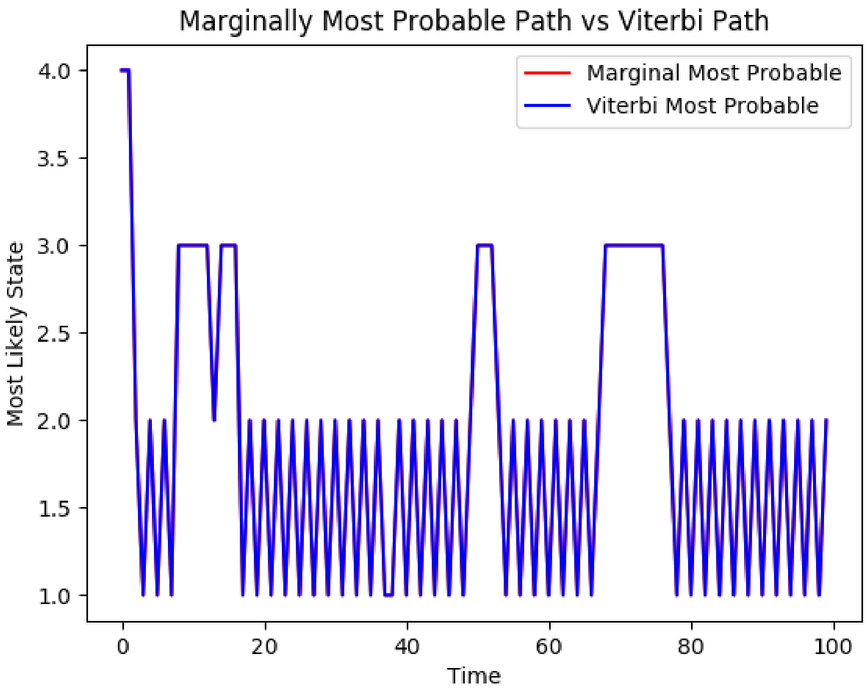

Figure 2.5: The The most likely state for the first 100 time points of the test data, using Viterbi decoding and the marginal probability.