Mini Project: Barcode Decoding with HMMs

Lieqiang GUO; Fubang ZHAO; Xiao PAN

lieqiang.guo@telecom-paristech.fr
fubang.zhao@telecom-paristech.fr
xiao.pan@telecom-paristech.fr

Question 1: draw the directed graphical model for the described model.

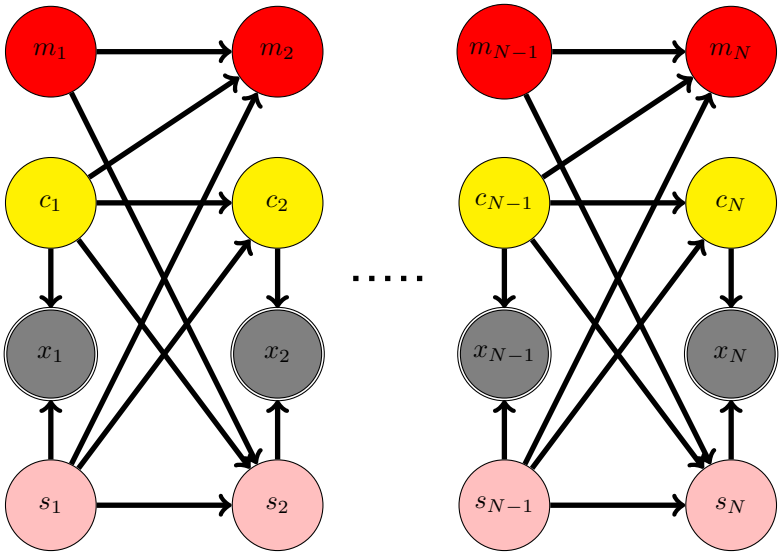

Figure 1: Directed graphical model

Question 2: we are interested in the most likely state trajectory given all the observations. For practical purposes, redefine the described model to an ordinary HMM. How can you construct the transition matrix? What will be the observation model?

We define a new variable $\Psi_n = [s_n, m_n, c_n]$, which encapsulates the state of the model at pixel n. The set of all possible states can be listed in a vector Ω and the state of the system at the pixel n can be represented as $\Psi_n = \Omega(j)$, where $j \in \{1, 2, \ldots, (S \times M \times C)\}$ and $C = max_{s,k}\ell(s)$. The transition matrix will have the following definition.

$$A_{i,j} = p(\Psi_{n+1} = \Omega(i) | \Psi_n = \Omega(j))$$

By using the distributions defined in Section 3.2, we can describe:

$$\begin{aligned} &p(\Psi_n = \Omega(i)|\Psi_{n-1} = \Omega(j)) \\ &\quad = p(c_n|s_{n-1}, c_{n-1})p(s_n|s_{n-1}, m_{n-1}, c_{n-1})p(m_n|s_{n-1}, m_{n-1}, c_{n-1}) \end{aligned}$$

And:

$$p(c_n|s_{n-1}, c_{n-1}) = \begin{cases} \delta(c_n - c_{n-1} - 1), & c_{n-1} \neq \ell(s_{n-1}) \\ \delta(c_{n-1}), & c_{n-1} = \ell(s_{n-1}) \end{cases}$$

$$p(s_n|s_{n-1}, m_{n-1}, c_{n-1}) = \begin{cases} \delta(s_n - s_{n-1}), & c_{n-1} \neq \ell(s_{n-1}) \\ \tau_s(s_n|s_{n-1}, m_{n-1}), & c_{n-1} = \ell(s_{n-1}) \end{cases}$$

$$p(m_n|s_{n-1}, m_{n-1}, c_{n-1}) = \begin{cases} \delta(m_n - m_{n-1}), & c_{n-1} \neq \ell(s_{n-1}) \\ \tau_m(m_n|s_n, m_{n-1}), & c_{n-1} = \ell(s_{n-1}) \end{cases}$$

where:

$$\tau_s(s_n|s_{n-1}, m_{n-1}) = \begin{cases} 1.) & \mathcal{U}(\{1, 3\}), & s_{n-1} = 1 \\ 2.) & \mathcal{U}(\{6, \ldots, 15\}), & s_{n-1} = 3 \\ 3.) & \mathcal{U}(\{6, \ldots, 15\}), & m_{n-1} \neq 6 \bigwedge 6 \leq s_{n-1} \leq 15 \\ 4.) & \delta(s_n - 5), & m_{n-1} = 6 \bigwedge 6 \leq s_{n-1} \leq 15 \\ 5.) & \mathcal{U}(\{16, \ldots, 25\}), & s_{n-1} = 5 \\ 6.) & \mathcal{U}(\{16, \ldots, 25\}), & m_{n-1} \neq 6 \bigwedge 16 \leq s_{n-1} \leq 25 \\ 7.) & \delta(s_n - 4), & m_{n-1} = 6 \bigwedge 16 \leq s_{n-1} \leq 25 \\ 8.) & \delta(s_n - 2), & s_{n-1} = 4 \\ 9.) & \delta(s_n - 2), & s_{n-1} = 2 \end{cases}$$

$$\tau_m(m_n|s_n, m_{n-1}) = \begin{cases} 1.) & \delta(m_{n-1}), & s_n \in \{1, \ldots, 5\} \\ 2.) & \delta(m_n - m_{n-1} - 1), & s_n \in \{6, \ldots, 15\} \bigwedge m_{n-1} \neq 6 \\ 3.) & \delta(m_n - m_{n-1} - 1), & s_n \in \{16, \ldots, 25\} \bigwedge m_{n-1} \neq 6 \end{cases}$$

And the observation model becomes:

$$p(x_n|\Psi_n) = \prod_{i=0}^{1} \mathcal{N}(x_n; \mu_i, \delta_i^2)^{\mathbb{1}(f(s_n, c_n) = i)}$$

Where the definitions of $\mathcal{N}(x; \mu, \delta^2), \mathbb{1}(\cdot), f(\cdot)$ are the same as those in Section 3.3.

Question 3: by using the result of the previous question, simulate the HMM and visualize the simulated data. Use $\mu_0 = 250$, $\mu_1 = 20$, $\delta_0^2 = \delta_1^2 = 5$. Does the simulated x_n look like a real scanline?

For this question, we have found an ambiguity about the length of quiet white zone, we referred to the statement of your question(page 3) and Wikipedia, so we defined that the length equals to 9(but in fact, it does not make a large difference for the questions below). See Figure 2,Figure 3,Figure 4 and Figure 5 for the simulating results. And by comparing Figure 5 with Figure 7, we can see that the simulated x_n looks like a real scanline. What's more, from Figure 6, we can find that it looks like a real scanline.

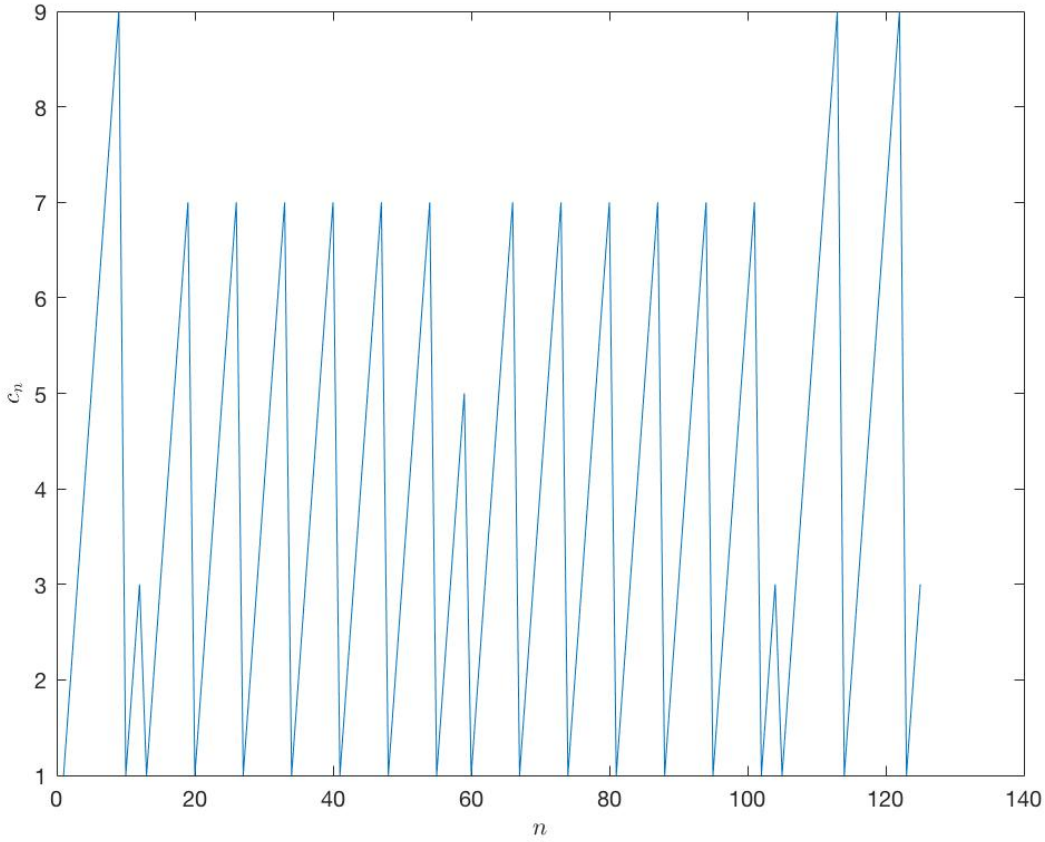

Figure 2: simulated c_n

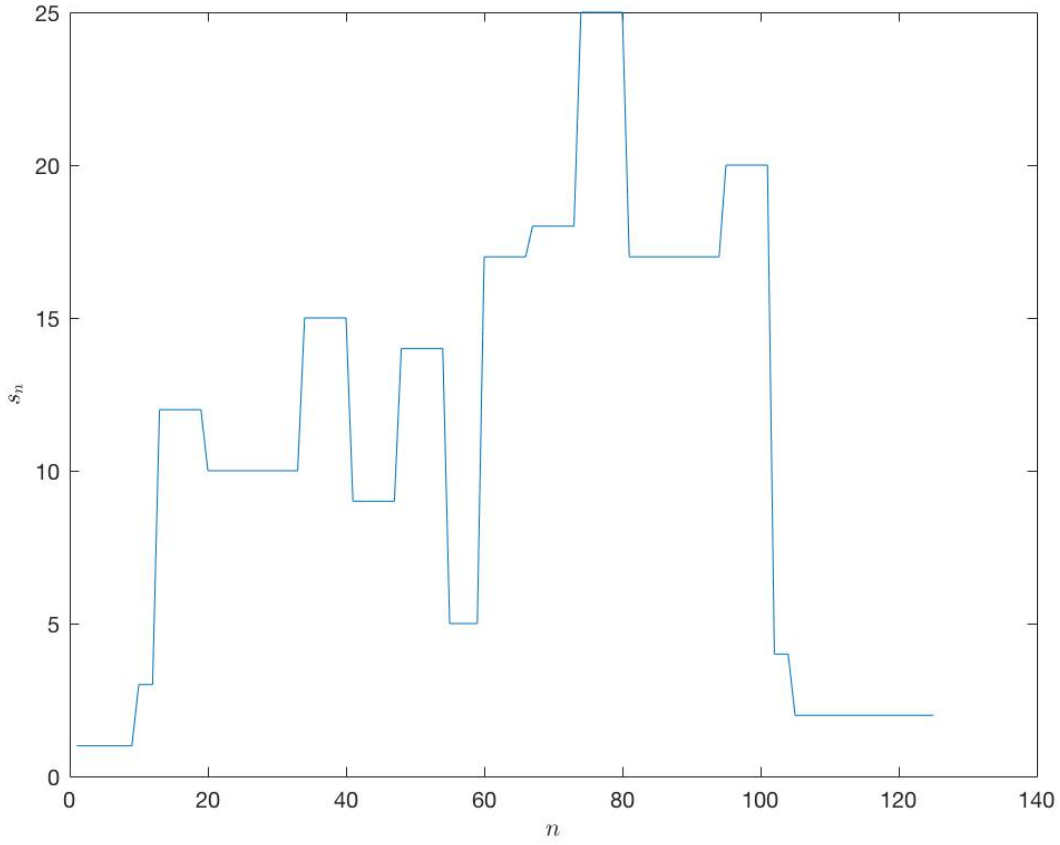

Figure 3: simulated s_n

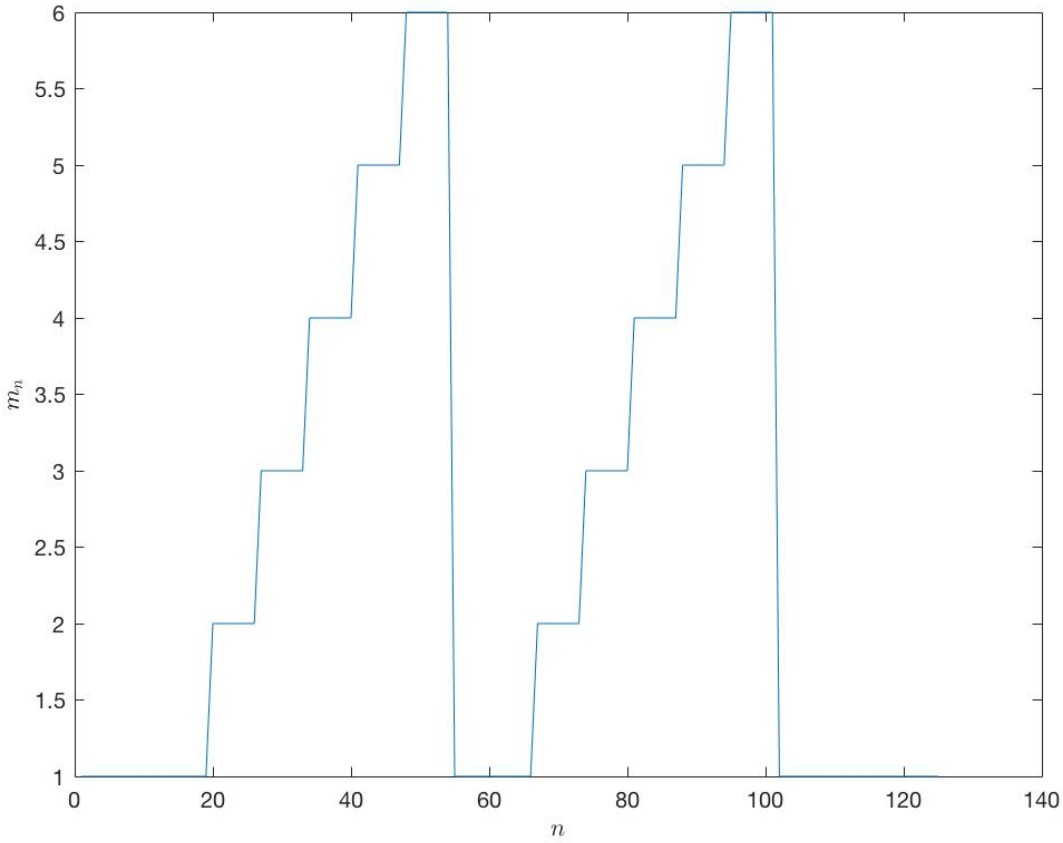

Figure 4: simulated m_n

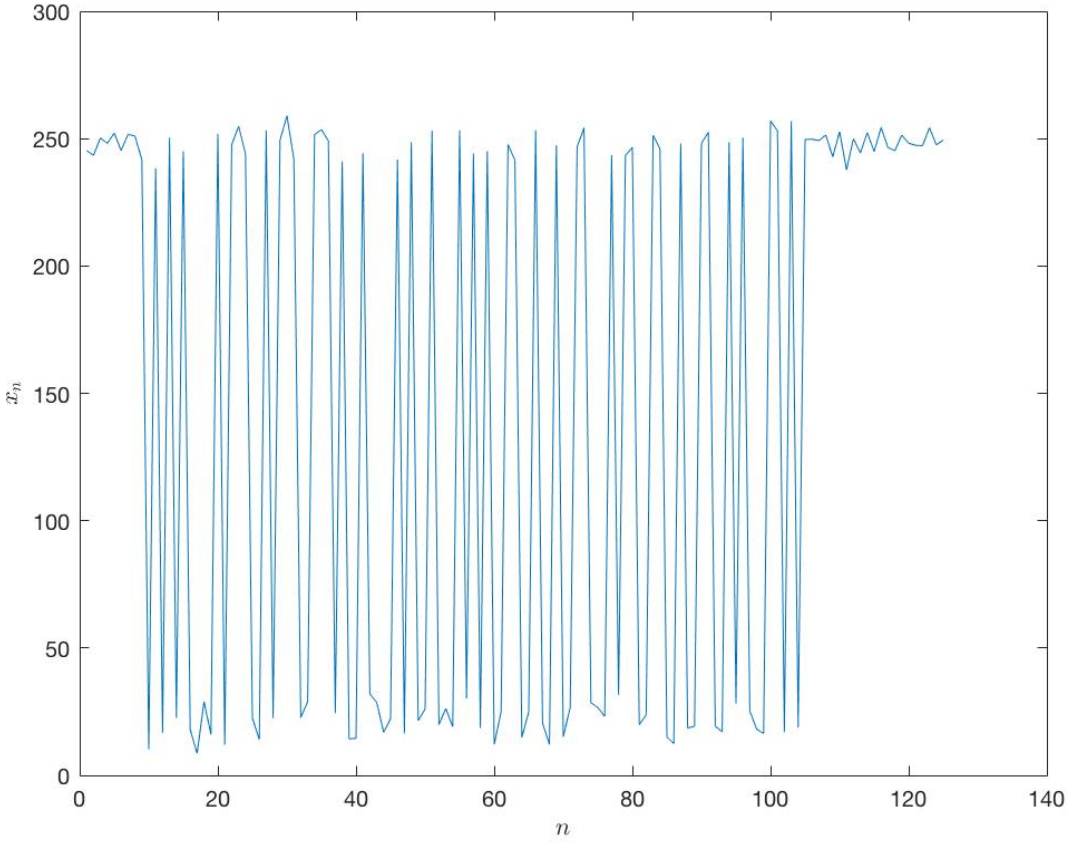

Figure 5: simulated x_n(0 represents black, 255 represents white)

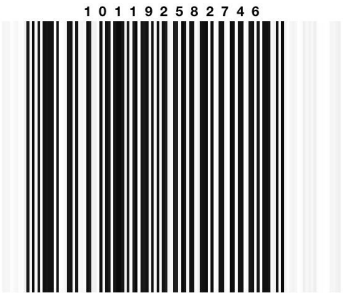

Figure 6: simulated Barcode

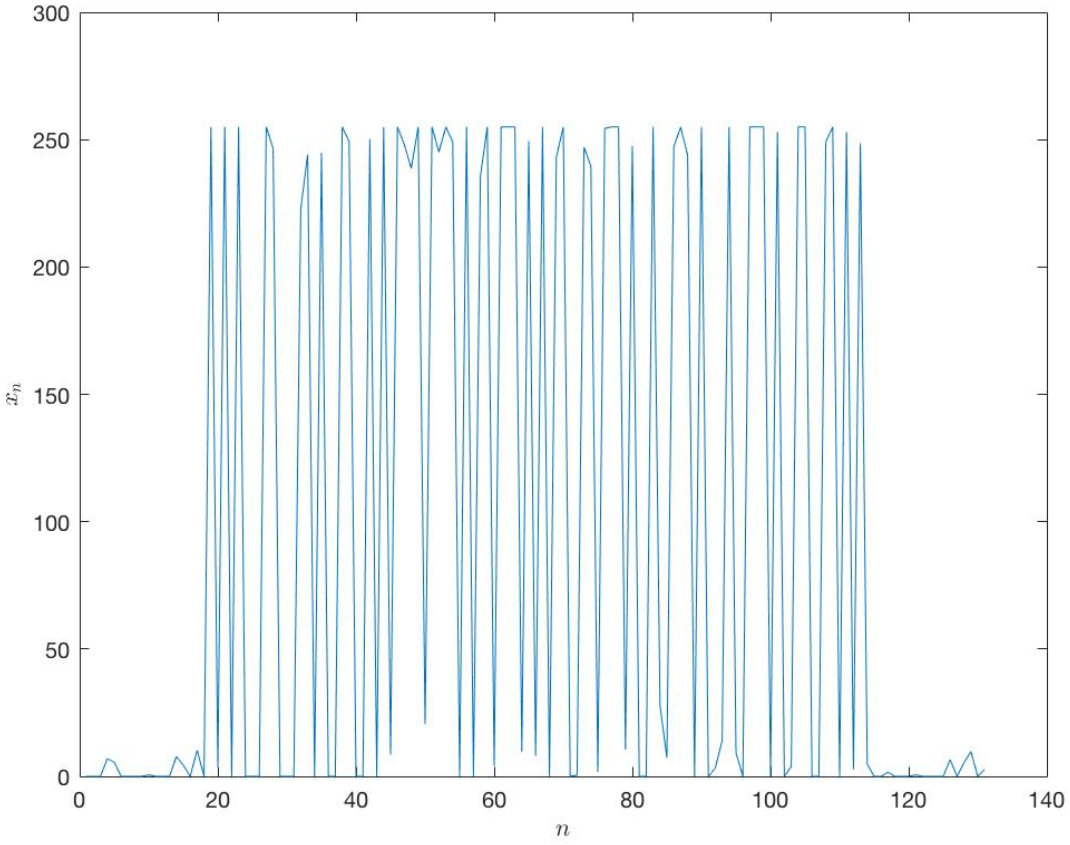

Figure 7: generated real scanline

Question 4: use the 'template_code.m'. Fill the parts 1, 2, and 3.

See the code.

Question 5: compute the filtering distribution $p(\Psi_n|x_{1:n})$ by using the forward recursion (fill part 4). How can you compute the marginals $p(s_n|x_{1:n}), p(c_n|x_{1:n}), p(m_n|x_{1:n})$?

Define $\alpha_{k|k-1}(\Psi_k) = p(\Psi_k, x_{1:k-1})$, and $\alpha_{k|k}(\Psi_k) = p(\Psi_k, x_{1:k})$. Then we have:

$$\begin{aligned}
\alpha_{k|k-1}(\Psi_k) &= p(\Psi_k, x_{1:k-1}) \\
&= \sum_{\Psi_{k-1}} p(\Psi_k, \Psi_{k-1}, x_{1:k-1}) \\
&= \sum_{\Psi_{k-1}} p(\Psi_k|\Psi_{k-1}, x_{1:k-1}) p(\Psi_{k-1}, x_{1:k-1}) \\
&= \sum_{\Psi_{k-1}} p(\Psi_k|\Psi_{k-1}) p(\Psi_{k-1}, x_{1:k-1}) \\
&= \sum_{\Psi_{k-1}} p(\Psi_k|\Psi_{k-1}) \alpha_{k-1|k-1}(\Psi_{k-1}) \\
\alpha_{k|k}(\Psi_k) &= p(\Psi_k, x_{1:k}) \\
&= p(x_k|\Psi_k, x_{1:k-1}) p(\Psi_k, x_{1:k-1}) \\
&= p(x_k|\Psi_k) \alpha_{k|k-1}(\Psi_k)
\end{aligned}$$

So, we can use α message to compute $p(\Psi_n|x_{1:n}), p(s_n|x_{1:n}), p(c_n|x_{1:n})$ and $p(m_n|x_{1:n})$.

$$\begin{aligned}
p(\Psi_n|x_{1:n}) &= \frac{p(\Psi_n, x_{1:n})}{p(x_{1:n})} \\
&= \frac{p(\Psi_n, x_{1:n})}{\sum_{\Psi_n} p(\Psi_n, x_{1:n})} \\
&= \frac{\alpha_{n|n}(\Psi_n)}{\sum_{\Psi_n} \alpha_{n|n}(\Psi_n)} \\
p(s_n|x_{1:n}) &= \frac{\sum_{\{\Psi_n|s_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{p(x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|s_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{\sum_{\Psi_n} p(\Psi_n, x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|s_n \in \Psi_n\}} \alpha_{n|n}(\Psi_n)}{\sum_{\Psi_n} \alpha_{n|n}(\Psi_n)} \\
p(c_n|x_{1:n}) &= \frac{\sum_{\{\Psi_n|c_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{p(x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|c_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{\sum_{\Psi_n} p(\Psi_n, x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|c_n \in \Psi_n\}} \alpha_{n|n}(\Psi_n)}{\sum_{\Psi_n} \alpha_{n|n}(\Psi_n)} \\
p(m_n|x_{1:n}) &= \frac{\sum_{\{\Psi_n|m_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{p(x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|m_n \in \Psi_n\}} p(\Psi_n, x_{1:n})}{\sum_{\Psi_n} p(\Psi_n, x_{1:n})} \\
&= \frac{\sum_{\{\Psi_n|m_n \in \Psi_n\}} \alpha_{n|n}(\Psi_n)}{\sum_{\Psi_n} \alpha_{n|n}(\Psi_n)}
\end{aligned}$$

Question 6: compute the smoothing distribution $p(\Psi_k|x_{1:N})$ by using the forward-backward recursions (fill part 5). How can you compute the marginals $p(s_k|x_{1:N})$,$p(c_k|x_{1:N})$, $p(m_k|x_{1:N})$?

Define $\beta_{k|k+1}(\Psi_k) = p(x_{k+1:N}|\Psi_k)$, and $\beta_{k|k}(\Psi_k) = p(x_{k:N}|\Psi_k)$. Then we have:

$$\begin{aligned}
\beta_{k|k+1}(\Psi_k) &= p(x_{k+1:N}|\Psi_k) \\
&= \sum_{\Psi_{k+1}} p(x_{k+1:N}|\Psi_{k+1}, \Psi_k) p(\Psi_{k+1}|\Psi_k) \\
&= \sum_{\Psi_{k+1}} p(x_{k+1:N}|\Psi_{k+1}) p(\Psi_{k+1}|\Psi_k) \\
&= \sum_{\Psi_{k+1}} p(\Psi_{k+1}|\Psi_k) \beta_{k+1|k+1}(\Psi_{k+1}) \\
\beta_{k|k}(\Psi_k) &= p(x_{k:N}|\Psi_k) \\
&= p(x_k, x_{k+1:N}|\Psi_k) \\
&= p(x_k|x_{k+1:N}, \Psi_k) p(x_{k+1:N}|\Psi_k) \\
&= p(x_k|\Psi_k) \beta_{k|k+1}(\Psi_k)
\end{aligned}$$

And:

$$\begin{aligned}
p(\Psi_k|x_{1:N}) &= p(x_{k+1:N}|\Psi_k, x_{1:k}) p(\Psi_k, x_{1:k}) \\
&= p(x_{k+1:N}|\Psi_k) p(\Psi_k, x_{1:k}) \\
&= \alpha_{k|k}(\Psi_k) \beta_{k|k+1}(\Psi_k)
\end{aligned}$$

We then define $\gamma_k(\Psi_k) = \alpha_{k|k}(\Psi_k)\beta_{k|k+1}(\Psi_k)$, so $p(\Psi_k|x_{1:N}) = \gamma_k(\Psi_k)$.

$$\begin{aligned}
p(s_k|x_{1:N}) &= \frac{\sum_{\{\Psi_k|s_k \in \Psi_k\}} p(\Psi_k, x_{1:N})}{p(x_{1:N})} \\
&= \sum_{\{\Psi_k|s_k \in \Psi_k\}} \gamma_k(\Psi_k) \\
p(c_k|x_{1:N}) &= \frac{\sum_{\{\Psi_k|c_k \in \Psi_k\}} p(\Psi_k, x_{1:N})}{p(x_{1:N})} \\
&= \sum_{\{\Psi_k|c_k \in \Psi_k\}} \gamma_k(\Psi_k) \\
p(m_k|x_{1:N}) &= \frac{\sum_{\{\Psi_k|m_k \in \Psi_k\}} p(\Psi_k, x_{1:N})}{p(x_{1:N})} \\
&= \sum_{\{\Psi_k|m_k \in \Psi_k\}} \gamma_k(\Psi_k)
\end{aligned}$$

Question 7: compute the most-likely path by using the Viterbi algorithm (fill part 6):

$$\Psi^*_{1:N} = (s^*_{1:N}, m^*_{1:N}, c^*_{1:N}) = \underset{\Psi_{1:N}}{argmax}\ p(\Psi_{1:N}|x_{1:N})$$

By postprocessing this optimal path, decode the barcode string (fill part 7).

$$\begin{aligned}\Psi^*_{1:N} &= \underset{\Psi_{1:N}}{argmax}\, p(\Psi_{1:N}|x_{1:N}) \\ &= \underset{\Psi_{1:N}}{argmax}\, \frac{p(\Psi_{1:N})p(x_{1:N}|\Psi_{1:N})}{p(x_{1:N})} \\ &= \underset{\Psi_N}{max}\, p(x_N|\Psi_N)\underset{\Psi_{N-1}}{max}\, p(\Psi_N|\Psi_{N-1})\cdots\underset{\Psi_2}{max}\, p(\Psi_3|\Psi_2)p(x_2|\Psi_2)\underset{\Psi_1}{max}\, p(\Psi_2|\Psi_1)p(x_1|\Psi_1)p(\Psi_1)\end{aligned}$$

Question 8: from now on, generate random barcodes by using 'generate_barcode.m'. Run your algorithms for different random barcodes. Set $\mu_0 = 255$, $\mu_1 = 0$, $\delta_0^2 = \delta_1^2 = 1$.

(i) Visualize the (marginal) filtering distributions.
(ii) Visualize the (marginal) smoothing distributions.
(iii) Visualize the most-likely path.
(iv) Try different values for the variable 'obs_noise' in 'generate_barcode.m' and repeat the previous steps. How does the algorithm behave when you increase the value of 'obs_noise'?

(i) See Figure 8 for the filtering distribution.
(ii) See Figure 9 for the smoothing distribution.
(iii) See Figure 10 for the most-likely path.
(iv) When we increase the value of 'obs_noise', we can see that

(i) The filtering and smoothing distributions become more dispersal(the distributions are not concentrated). Because the information the algorithm get becomes more uncertain.
(ii) We can not decode correctly, but we are correct for the most part of decoding we usually make mistake on 1 or 2 digits when we decode the Barcode.

(i) When noise = 10, the decoded barcode is the same as the real code.
Real code: 092645973108
Decoded code: 092645973108

(ii) When noise = 120, the decoded barcode has some mistakes.
Real code: 092645973108
Decoded code: 041625070108

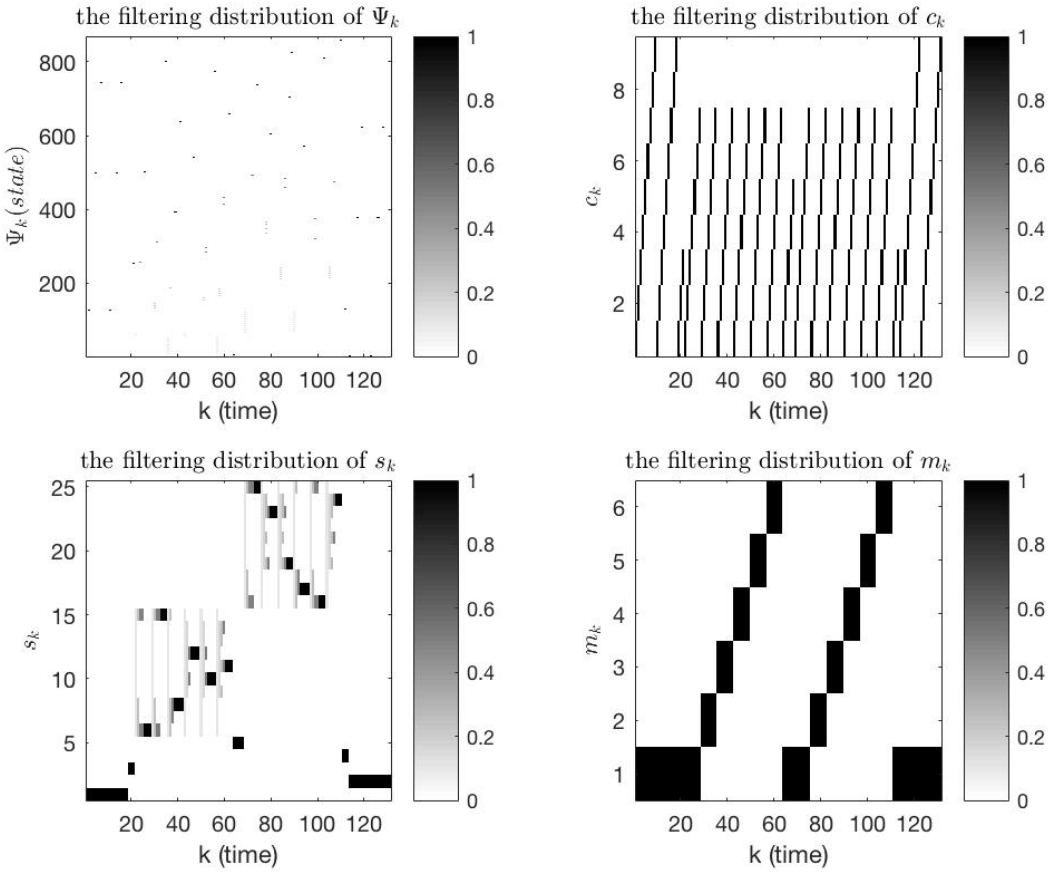

Figure 8: the filtering distribution with noise=10

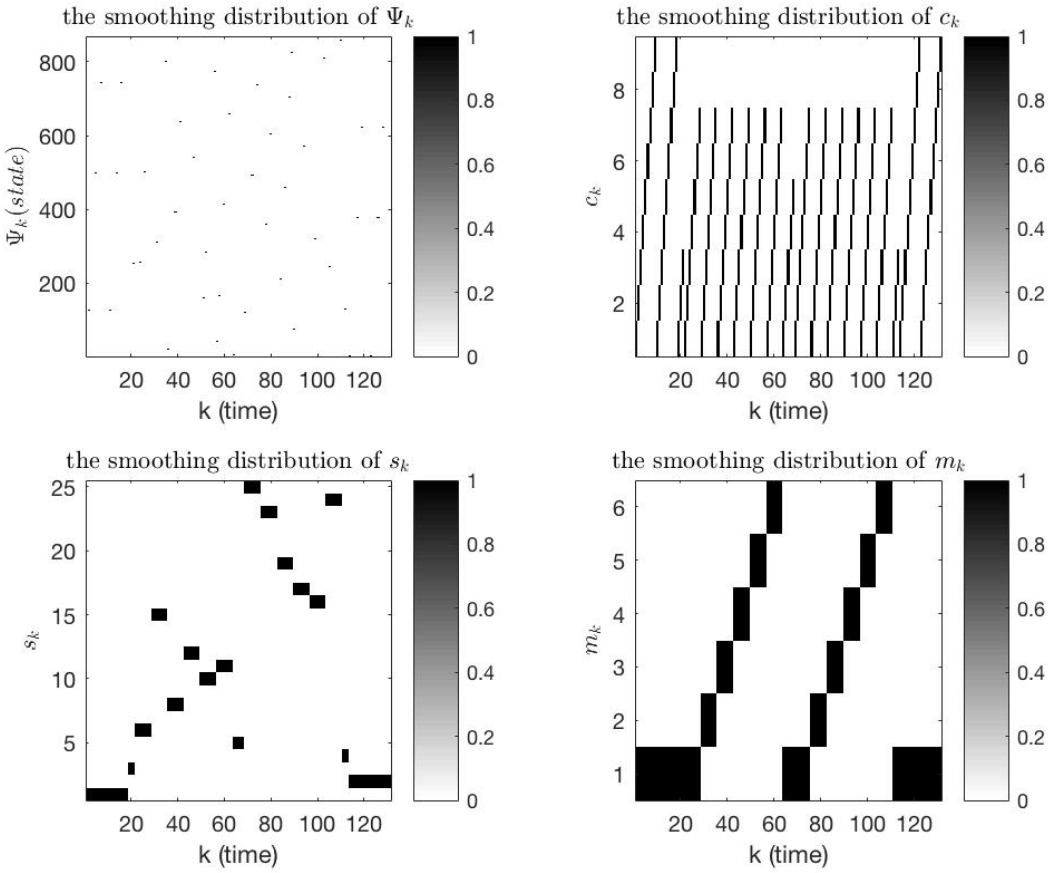

Figure 9: the smoothing distribution with noise=10

Visualize the most-likely path:

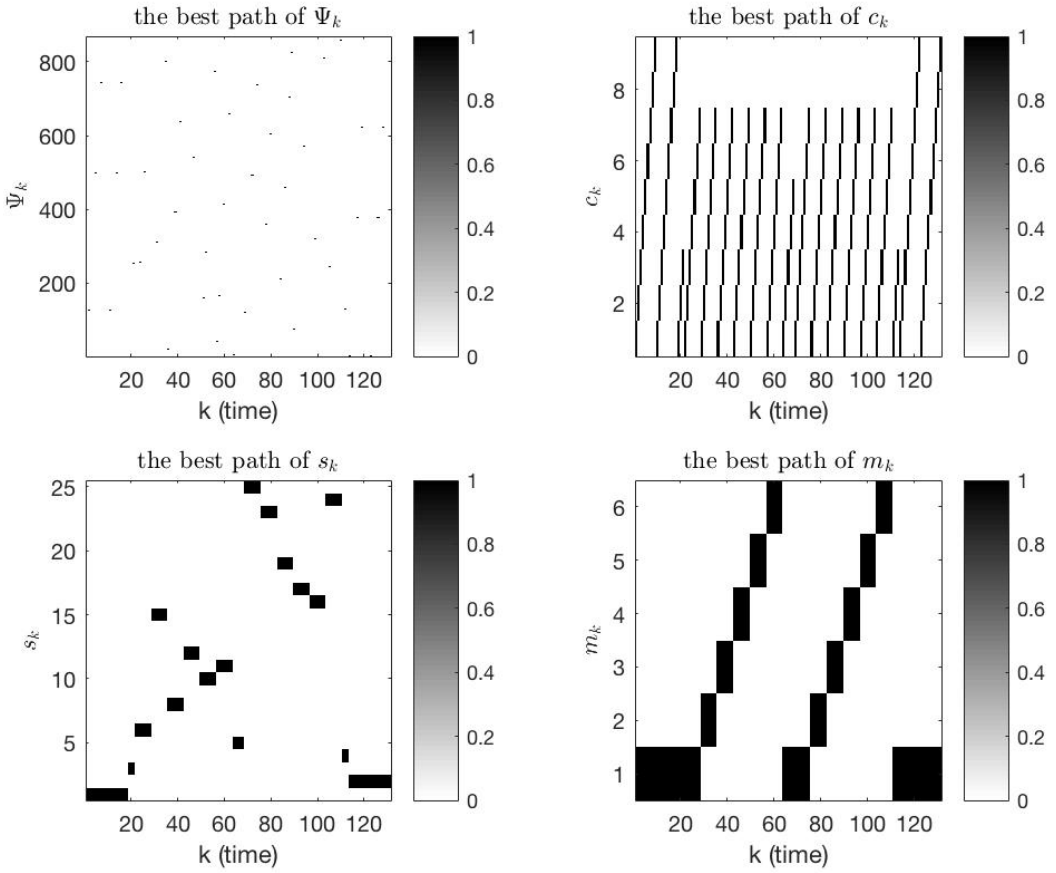

Figure 10: the best path with noise=10

The figures below show the situation when the noise is equal to 120. We can see that the image of distribution is not that clear now.

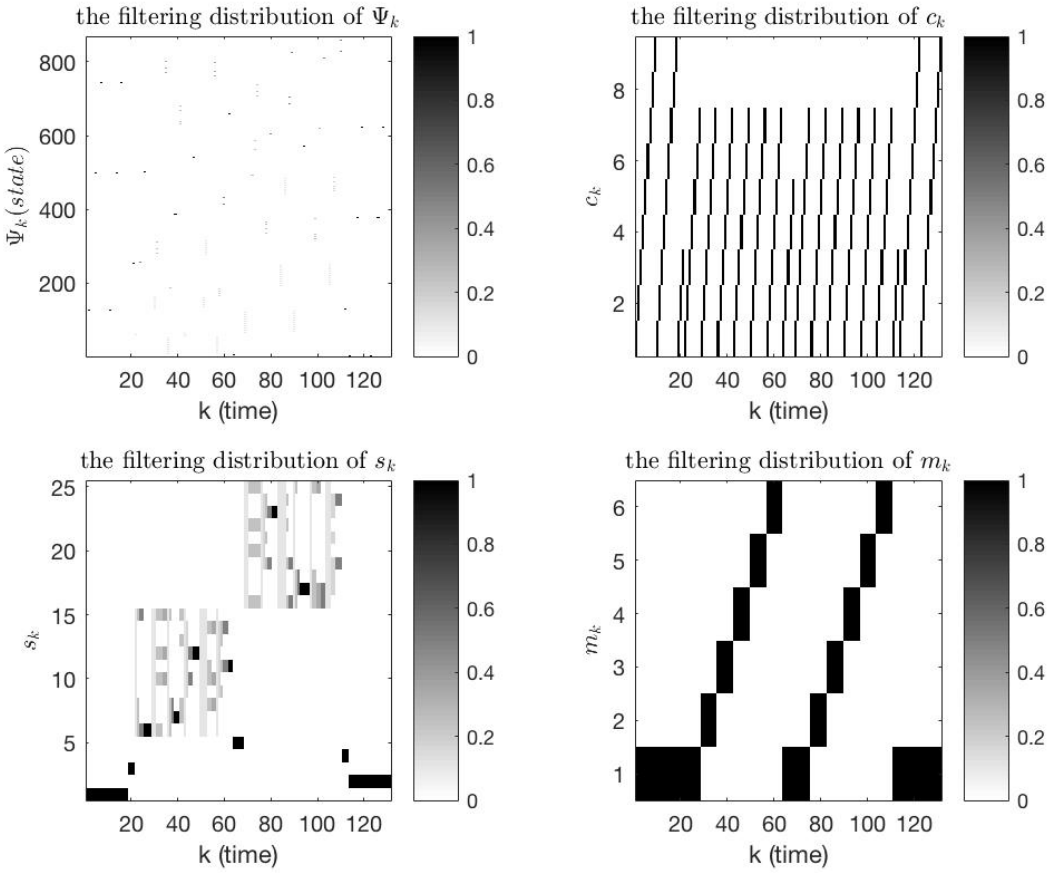

Figure 11: the filtering distribution with noise=120

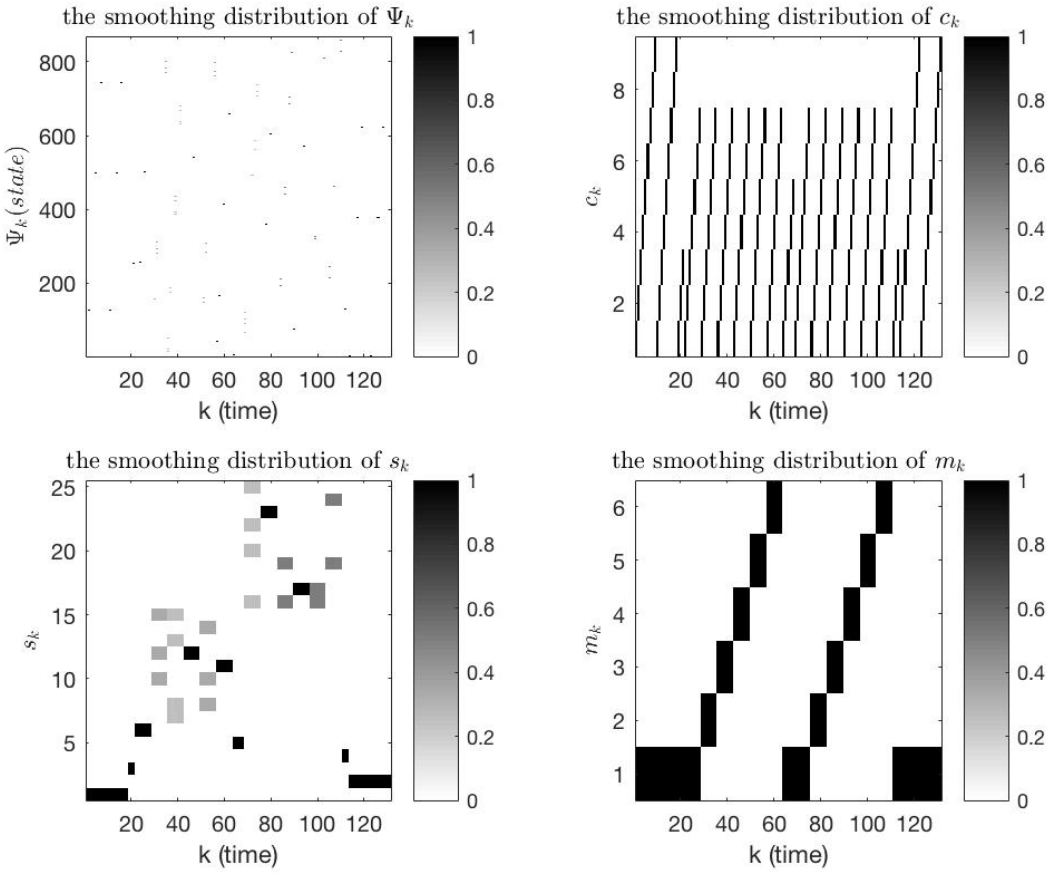

Figure 12: the smoothing distribution with noise=120

Visualize the most-likely path:

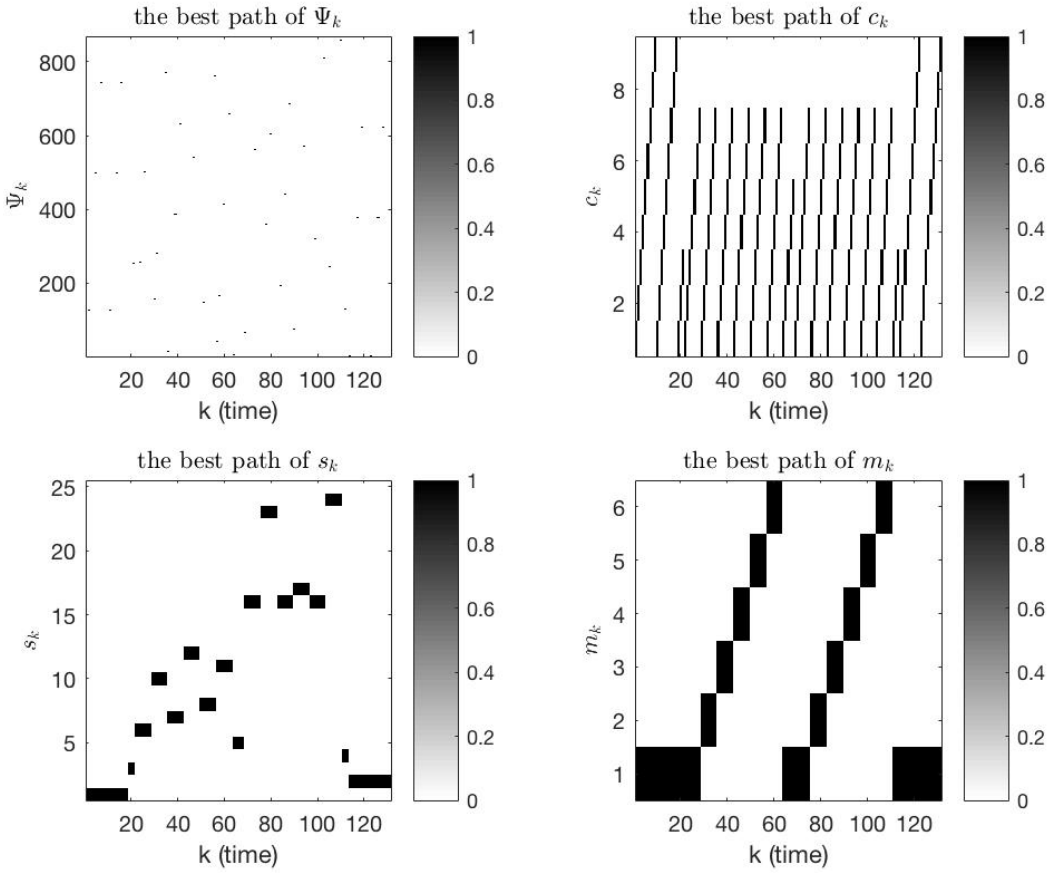

Figure 13: the best path with noise=120