Introduction to Graphical Models: Mini Project

Due on December 3, 2017 at 11:55 pm

Taught by Assisitant Professor Umut Simsekli

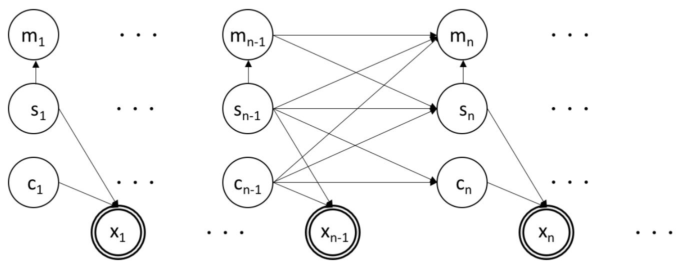
Pengfei MI, Rui SONG, Yanting LI

This document is the report for the Mini-Project of the course "Introduction to Graphical Models" of Master 2 Data Science. This project is done in group of three, with group members: Pengfei MI, Rui SONG, and Yanting LI.

Question 1

Directed graphical model

Solutions:
The directed graphical model for our model can be drawn like below



Question 2

Transition Matrix

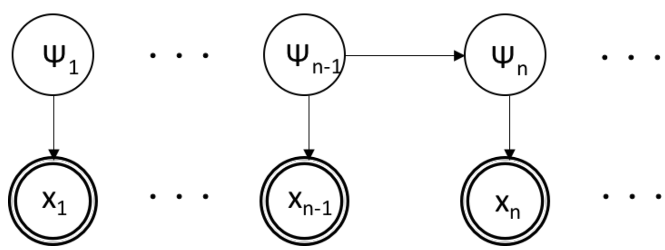
Solutions:
Define a variable $\Psi_n \equiv [s_n, m_n, c_n]$. The set of all states can be listed in a vector Ω then the state of the system at pixel n can be represented as $\Psi_n = \Omega(j)$ where $j \in \{1, 2, ..., (S \times M \times C)\}$ and $C = max_{s,k} l(s)$ ($= 7$ in our model). Then the transition matrix is:

$$\begin{aligned} A(i,j) &= p(\Psi_{n+1} = \Omega(i) | \Psi_n = \Omega(j)) \\ &= p(c_{n+1} = c_i, s_{n+1} = s_i, m_{n+1} = m_i | c_n = c_j, s_n = s_j, m_n = m_j) \\ &= p(c_{n+1} = c_i | c_n = c_j, s_n = s_j) p(s_{n+1} = s_i, m_{n+1} = m_i | c_n = c_j, s_n = s_j, m_n = m_j) \\ &= p(c_{n+1} = c_i | c_n = c_j, s_n = s_j) p(s_{n+1} = s_i | c_n = c_j, s_n = s_j, m_n = m_j) p(m_{n+1} = m_i | s_{n+1} = s_i, c_n = c_j, s_n = s_j) \\ &= \begin{cases} \delta(c_{n+1} - c_n - 1)\delta(s_{n+1} - s_n)\delta(m_{n+1} - m_n) & \text{if } c_n \neq l(s_n) \\ \delta(c_{n+1} - 1)\tau(s_{n+1}|s_n, m_n)\tau(m_{n+1}|s_{n+1}, m_n) & \text{if } c_n = l(s_n) \end{cases} \end{aligned} \tag{1}$$

where $\tau(s_{n+1}|s_n, m_n)$ and $\tau(m_{n+1}|s_{n+1}, m_n)$ are defined in the description.
Then the observation model can be written like:

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

Then the graphical model becomes:



Question 3

Simulation of HMM

Solutions:
The general idea to simulate the HMM model is: for each states $j \in \{1, 2, ..., (S \times M \times C)\}$, we generate a random number r according to a uniform distribution in the interval $[0, 1]$. Then we find the state k such that $\sum_{i=1}^{k} A(i, j) < r$ and $\sum_{i=1}^{k+1} A(i, j) >= r$. Then we will transfer from state j to state i.
For the code, please look at file $template_code.ipynb$. The barcode simulated is : As we can see this looks

[1 8 4 5 2 5 9 5 0 0 7 9]

very much like a real scanline. And the barcode string is ”184525950079”.

Question 4

Fill code part 1, 2, and 3

Solutions:
Please refer to file ”template_code.ipynb”.

Question 5

Compute the filtering distribution and marginals

Solutions:
Please refer to file ”Part4.py”.

Question 6

Compute the smoothing distribution and marginals

Solutions:
Please refer to file ”Part5.py”.

Question 7

Viterbi algorithm

Solutions:

Question 8

Ran algorithms on different random barcodes

Solutions: