Project 1 in course STAT340 – 2020

H. Rue

September 7, 2020

- **Deadline: Thursday 1st Oct. The report (with code) should be submitted in blackboard, preferably as a compiled R-markdown document.**
- Yes, you have to write your own code all the HMM in all problems. Yes, I know we can find libraries out there doing this for us, but doing every detail yourself is the main purpose of this project.
- You can work in groups of maximum 2, and hand in a joint project [COVID-19 adjustment].

Problem 1

Use the recursions for hidden Markov models (or consider this as a dynamic programming problem), to solve the following.

1. Given a general $m \times n$ matrix A, where $n \gg m$, with random numbers, find a **smooth** path p from $t = 1, \ldots, n$ which maximise $\sum_{t=1}^{n} A_{p_t,t}$. (It is a good idea to use graphics to verify that you get the correct solution, at least for small m and n.)

   (Yes, we did this in class when $n = m$. Make sure you *really* understand it by reimplementing it yourself and extend it to the case where $n > m$, or even $n \gg m$.)

Problem 2

Let x be a two-state Markov chain with transition matrix

$$P = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

where $0 \le p \le 1$. Let the states be 0 and 1. This Markov chain is observed with observations y_t, where

$$y_t | x_t \sim \mathcal{N}(x_t, \sigma^2)$$

or

$$y_t | x_t = \begin{cases} x_t & \text{with probability } q \\ 1 - x_t & \text{with probability } 1 - q \end{cases}$$

For simplicity we assume both σ^2 and q to be known.

Use both observational models and experiment with different scenarios for $\theta = (p, q, \sigma^2)$, where you do the following.

1. Compute the marginal mean and standard deviation and the posterior modal configuration for x (when θ is known).
2. When p is unknown, compute its posterior mode for p, and the posterior distribution for p.
3. Compute the marginal mean and standard deviation for x where the uncertainty for p is taken into account.
4. Compute the global modal configuration for (p, x) jointly.

Problem 3

In this problem the hidden states will index transition matrices; *emission matrix* **E** and *transition matrix* **A**. The transition matrix contains the probabilities of switching from one state to another while the emission matrix holds the probabilities of choosing the objects in each state. For this application we consider a DNA sequence depending on the main four nucleotides A, C, G and T where the nucleotide found at a particular position in a sequence depends on the state at the previous nucleotide position in the sequence according to the HMM structure. The positions are modelled along the sequence as belonging to either one of two states, "AT-state" or "GC-state".

Consider the long DNA sequence given in the file *sequence.txt* which describes a specific bacteriological genome sequence characterised by a precise succession of the main 4 nucleotides for a total of 48502 nucleotides. A snapshot of the sequence is

GGGCGGCGACCTCGCGGGTTTTCGCTCGCT...

You can read this into a vector in `R`, using

```
y = strsplit(readLines("sequence2.txt"),"")
```

We will use a hidden Markov model (HMM) with two different states ("AT-state" and "GC-state") to infer which state of the HMM is most likely to have generated each nucleotide position in the sequence. The emission matrix is given (ie. pre-estimated) as

$$E = \begin{bmatrix} 0.27 & 0.2084 & 0.198 & 0.3236 \\ 0.2462 & 0.2476 & 0.2985 & 0.2077 \end{bmatrix}$$

where the rows define the two states "AT-state"" and "GC-state" while the columns are described by the four nucleotides A,C,G and T respectively. Each value in the matrix above defines the probability of choosing each of the four nucleotides A, C, G and T in both states. For example, the first value in the first row and first column is 0.27 and is the probability of choosing the nucleotide A when in the "AT-state".

Then we define the transition matrix

$$A = \begin{bmatrix} 0.9998 & 0.0002 \\ 0.0002 & 0.9998 \end{bmatrix}$$

which contains the probabilities of switching from one state to another. In this case we have that if the previous nucleotide in the sequence was in the "AT-state"/"GC-state" there may be a probability of 0.0002 that the current nucleotide will be in the "GC-state"/"AT-state".

The task is as follows.

1. Given the model and the DNA sequence you can make use of your version of the algorithm and the matrices above to find the most probable state path, i.e, the most likely state ("AT-state" or "GC-state") that has generated each nucleotide position in the DNA sequence. In practice your algorithm should return the DNA sequence into blocks of nucleotides that were probably generated by the "GC-state" or the "AT-state".

Problem 4

Ooops: This problem is somewhat challenging.

Reconsider the setup in problem 2 when p is known. Let z be an configuration of the Markov chain of length T. If we estimate the Markov chain as z when then truth is x, then we define its loss to be $L(z, x)$. The optimal Bayes estimator (OBE), is defined as

$$z^* = \arg\min_z \mathrm{E}_{x|y} L(z, x)$$

ie that configuration z that minimise the posterior expectation of the loss.

Verify by yourself that

$$z_s^* = \arg\max_{x_s} \pi(x_s|y_1, \ldots, y_T), \qquad s = 1, \ldots, T,$$

is the OBE for this loss-function that simply counts the number of errors

$$L(z, x) = \sum_s 1_{[z_s \neq x_s]}.$$

We will now consider a loss-function that also penalise two neighbour errors, since we do want to avoid clustering of errors:

$$L(z, x) = \sum_s 1_{z_s \neq x_s} + \lambda \sum_{s \sim s'} 1_{[z_s \neq x_s]} 1_{[z_{s'} \neq x_{s'}]}$$

for a fixed $\lambda > 0$. The "$s \sim s'$" is the sum over all nearest neighbour pairs of s and s'.

The task is as follows.

1. Compute the OBE for this new loss function, using a two step procedure: first compute $\mathrm{E}_{x|y} L(z, x)$ and then compute the OBE from a new hidden Markov chain defined from the expected loss. If you do not this do this, please explain how it could be done.