

Assignment 1, Pattern Recognition (EQ2340)

Sebastian Bujwid, bujwid@kth.se

Martin Hwasser, hwasser@kth.se

September 25, 2017

Task A.1.2

Question 1. To verify your Markov chain code, calculate $P(S_t = j), j \in \{1, 2\}$ for $t = 1, 2, 3, \dots$ theoretically, by hand, to verify that $P(S_t = j)$ is actually constant for all t .

The initial probability π and state transition A :

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix} \quad A = \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix} \quad (1)$$

Given π , we see that the probability of being in state 1 or 2 in timestep 1 is:

$$P(S_1 = 1) = 0.75, P(S_1 = 2) = 0.25 \quad (2)$$

Furthermore, we see that the probability of being in state 1 at t_2 is given by:

$$\begin{aligned} P(S_2 = 1) &= \pi_1 * A_{1,1} + \pi_2 * A_{1,2} \\ &= 0.75 * 0.99 + 0.25 * 0.03 \\ &= 0.75 \\ &= \pi_1 \end{aligned} \quad (3)$$

And similarly, the probability of being in state 2 at t_2 is:

$$\begin{aligned} P(S_2 = 2) &= \pi_1 * A_{1,2} + \pi_2 * A_{2,2} \\ &= 0.75 * 0.01 + 0.25 * 0.97 \\ &= 0.25 \\ &= \pi_2 \end{aligned} \quad (4)$$

So since the state probabilities are equal in t_1 and t_2 , we see, by recursion, that the probability transition is constant throughout all timesteps.

Question 2. Use your Markov chain rand function to generate a sequence of $T = 10000$ state integer numbers from the test Markov chain. Calculate the relative frequency of occurrences of $S_t = 1$ and $S_t = 2$. The relative frequencies should of course be approximately equal to $P(S_t)$.

For 100 simulations with $T = 10000$, the average probability of being in $S_t = 1$ approaches very close to 0.75, which is π_1 and what we expected.

Question 3. To verify your HMM rand method, first calculate $E[X_t]$ and $var[X_t]$ theoretically. The conditional expectation formulas $\mu_X = E[X] = E_Z[E_X[X|Z]]$ and $var[X] = E_Z[var_X[X|Z]] + var_Z[E_X[X|Z]]$ apply generally whenever some variable X depends on another variable Z and may be useful for the calculations. Then use your HMM rand function to generate a sequence of $T = 10000$ output scalar random numbers $x = (x_1 \dots x_t \dots x_T)$ from the given HMM test example. Use the standard MatLab functions **mean** and **var** to calculate the mean and variance of your generated sequence. The result should agree approximately with your theoretical values.

$$\begin{aligned}
 E[X_t] &= E_{S_t}[E_{X_t}[X_t|S_t]] \\
 &= \sum_{n=1}^2 E[X_t|S_t = n]P(S_t = n) \\
 &= 0.75 * 0 + 0.25 * 3 \\
 &= 0.75
 \end{aligned} \tag{5}$$

For the variance, we have:

$$var[X_t] = \underbrace{E_{S_t}[var_{X_t}[X_t|S_t]]}_{first} + \underbrace{var_{S_t}[E_{X_t}[X_t|S_t]]}_{second} \tag{6}$$

The first part:

$$\begin{aligned}
 E_{S_t}[var_{X_t}[X_t|S_t]] &= \sum_{n=1}^2 var_{X_t}[X_t|S_t = n]P(S_t = n) \\
 &= 0.75 * 1 + 0.25 * 2^2 = 1.75
 \end{aligned} \tag{7}$$

And the second part:

$$\begin{aligned}
\text{var}_{S_t}[E_{X_t}[X_t|S_t]] &= E_{S_t}[E_{X_t}[X_t|S_t]^2] - E_{S_t}[E_{X_t}[X_t|S_t]]^2 \\
&= (0.75 * 0^2 + 0.25 * 3^2) - 0.75^2 \\
&= 1.6875
\end{aligned} \tag{8}$$

So, using (7) and (8):

$$\text{var}[X_t] = 1.75 + 1.6875 = 3.4375 \tag{9}$$

Question 4. To get an impression of how the HMM behaves, use @HMM/rand to generate a series of 500 contiguous samples X_t from the HMM, and plot them as a function of t . Do this many times until you have a good idea of what characterizes typical output of this HMM, and what structure there is to the randomness. Describe the behavior in one or two sentences in your report. Also include one such plot in the report, labeled using `title`, `xlabel`, and `ylabel` to clearly show which variable is plotted along which axis. You should do this for every plot in the course project.

Figure 1 shows the output of the HMM, usually starting at $S = 1$, with emissions near 0. Occasionally, it will transition to state 2 and emit output with a mean of 3 but with higher variance. It usually stays in the same state.

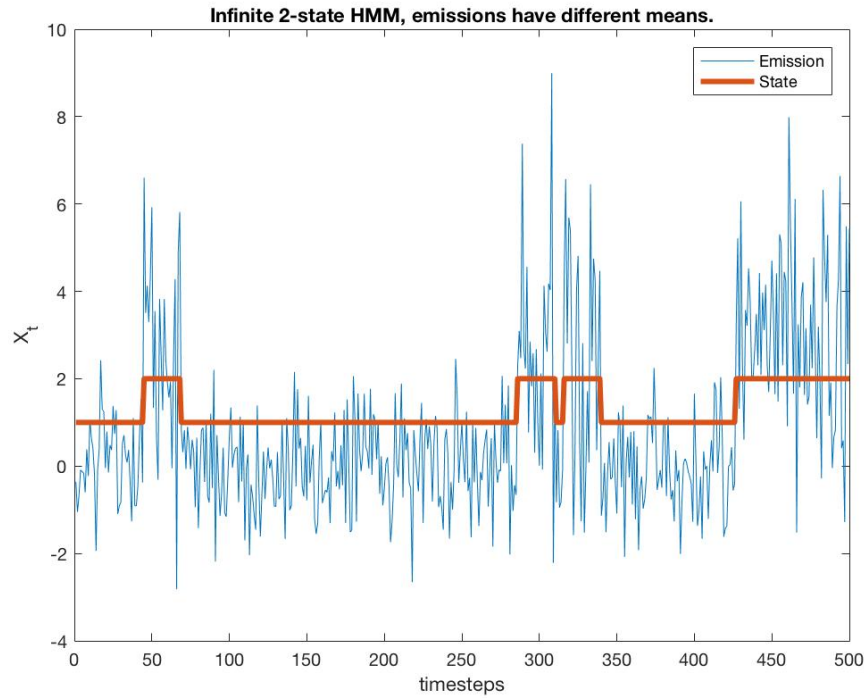


Figure 1: Infinite-duration HMM with 2 states where the emissions have different means.

Sampling from the HMM shows that the theoretical results are very close to the approximations.

Question 5. Create a new HMM, identical to the previous one except that it has $\mu_2 = \mu_1 = 0$. Generate and plot 500 contiguous values several times using @HMM/rand for this HMM. What is similar about how the two HMMs behave? What is different with this new HMM? Is it possible to estimate the state sequence S of the underlying Markov chain from the observed output variables x in this case?

The HMMs both transition between states in the same way, except now it becomes much more difficult to discern when the state changes. Occasionally, the state transitions into another state, and we see less or more variance depending on that state, but the variances are so close that it would be more difficult to estimate states.

It's still possible to estimate a state sequence of the underlying Markov chain, however the estimation will be less certain.

Figure 2 shows the emissions and the underlying state. As we see for state 2, the values tend to be more distant from the mean.

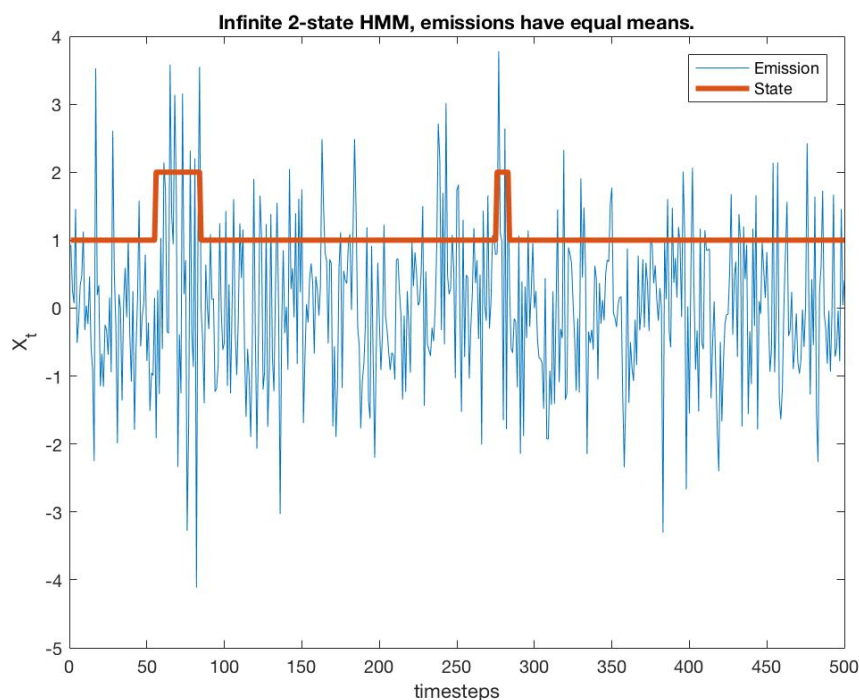


Figure 2: Infinite-duration HMM with 2 states, same mean.

Question 6. Another aspect you must check is that your rand-function works for *finite-duration* HMMs. Define a new test HMM of your own and verify that your function returns reasonable results.

We initialized an HMM with the following properties:

$$\pi = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad A = \begin{pmatrix} 0.90 & 0.1 & 0 \\ 0.00 & 0.90 & 0.1 \end{pmatrix} \quad B = \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.6 & 0.3 & 0.1 \end{pmatrix} \quad (10)$$

Since it's a left-right HMM, and the states can only transition to themselves or the state to the right, we can compute the expected sequence length by realizing that the number of trails before we transition into the next state is a geometric distribution, and has an expected value of $1/p$. We define p as the probability that the state transitions into the next state. As such, and the expected number of timesteps we spend in state 1 and 2 is $1/A_{1,2}$ and $1/A_{2,3}$ respectively. In total the expected sequence length is thus: $1/0.1 + 1/0.1 = 20$.

Continuously sampling from the HMM a thousand times and taking the average length of all sequences, we find that it's very close to 20, which is what we would expect.

Figure 3 shows a reasonable outcome when sampling a sequence from the HMM. The following code was used to approximate the sequence length:

```
mc = MarkovChain([1 0],[0.9 0.1 0;0 0.9 0.1]);
pD(1) = DiscreteD([0.1 0.3 0.6]);
pD(2) = DiscreteD([0.6 0.3 0.1]);
h = HMM(mc, pD);

s = 0
for i = 1:10000
    [X, S] = rand(h, 1000);
    s = s+length(S);
end
disp(s/10000) % close to 20
```

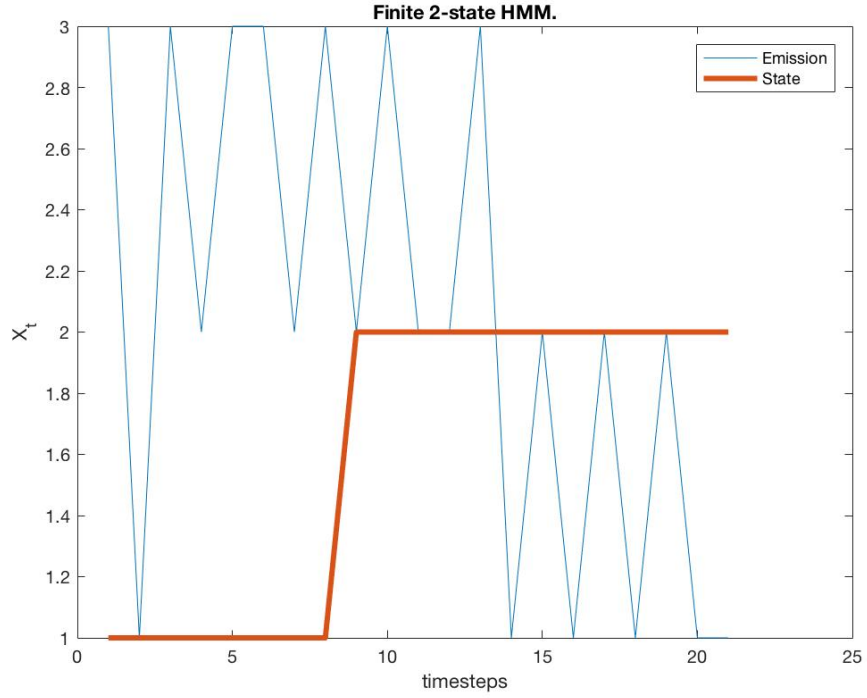


Figure 3: Finite-duration HMM with 2 states and an end state.

Question 7. Finally, your rand-function should work also when the state-conditional output distributions generate random vectors. Define a new testHMM of your own where the outputs are Gaussian vector distributions, and verify that this also works with your code. (Note that a single instance of the GaussD class is capable of generating vector output; stacking several GaussD-objects is not correct.) At least one of the output distributions should have a non-diagonal covariance matrix such as:

$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix} \quad (11)$$

Figure 4 shows a 3-state HMM with the following attributes:

$$\pi = \begin{pmatrix} 0.75 \\ 0.25 \\ 0 \end{pmatrix} \quad A = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.1 & 0.80 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{pmatrix} \quad B = \begin{pmatrix} b_1(x) \\ b_2(x) \\ b_3(x) \end{pmatrix} \quad (12)$$

where $b_1(x)$, $b_2(x)$ and $b_3(x)$ are 2d Gaussian density functions with the following means and covariances:

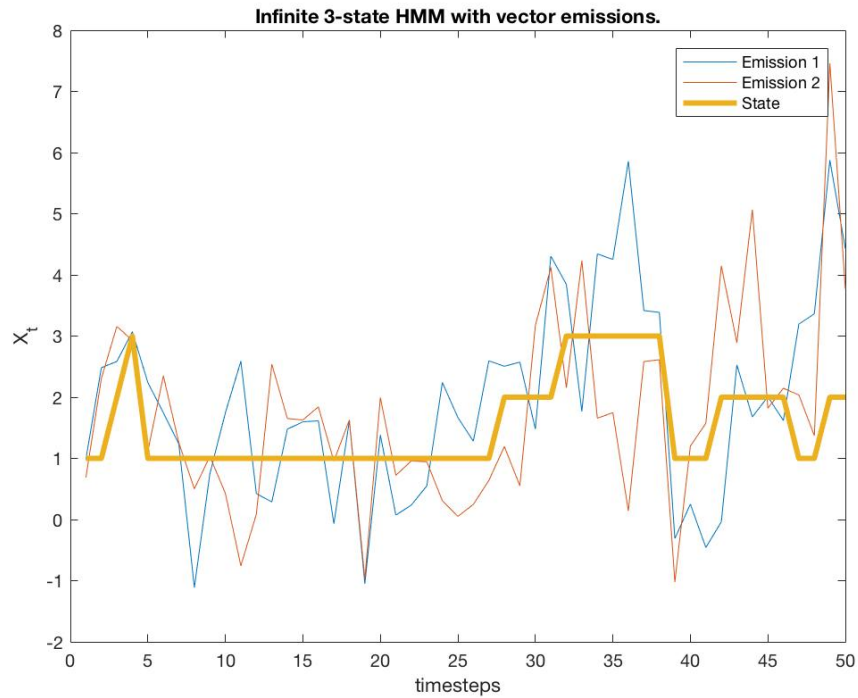


Figure 4: 3-state HMM with 2d emissions.

$$\mu_1 = (0, 0), \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mu_2 = (2, 2), \Sigma_2 = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}, \mu_3 = (3, 3), \Sigma_3 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

We see that in state 1, there's no correlation between the emissions, whereas in state 2, the emissions seem to follow a similar pattern, and in state 3 the emissions are negatively correlated, such that when one emission increases, the other decreases.

```
% the following generates a 3-state HMM with 2d output emissions.
pDgen(1)=GaussD('Mean',[1 1], 'Covariance', [1 0; 0 1]);
pDgen(2)=GaussD('Mean',[2 2], 'Covariance', [2 1; 1 4]);
pDgen(3)=GaussD('Mean',[3 3], 'Covariance', [1 -1; -1 1]);
mc=MarkovChain([0.75 0.25 0],[0.9 0.1 0;0.1 0.8 0.1; 0.1 0.1 0.9]);
h=HMM(mc, pDgen)
[X, S] = rand(h, 50);
```