

EN2202 Assignment 1

Sam Lewis
svrlewis@kth.se

Bjoern Fischer

1 Verification of the Markov chain random code

Considering the infinite duration HMM $\lambda = q, A, B$ with

$$q = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}; A = \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix}$$

The q vector defines the initial state probability, that is

$$P_1 = \begin{pmatrix} P(S_1 = 1) \\ P(S_1 = 2) \end{pmatrix} = q = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$$

For the next time step, $t = 2$ the transition probability matrix can be used to find the new state probabilities.

$$P_2 = \begin{pmatrix} P(S_2 = 1) \\ P(S_2 = 2) \end{pmatrix} = (q^T A)^T = ((0.75 \quad 0.25) \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix})^T = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$$

Because the transition matrix for the Markov chain is constant for all time t , this means that the state probability is also constant for all time t , and is equal to q .

This was verified using the `rand` of the the Markov Chain class to generate a sequence of 10000 states using the following MATLAB script.

```
1 mc = MarkovChain([0.75; 0.25], [0.99 0.01 ; 0.03 0.97]);
2 S = rand(mc, 10000);
3
4 state1 = sum(S == 1)
5 state2 = sum(S == 2)
```

Running this script once gave the result of `state1 = 7598; state2 = 2402` which is close to the expected distribution.

2 Verification of the HMM random code

For the system, listed above the expected value and variance can both be theoretically found.

$$E[X_T] = E_S[E_x[X | S]] = \Pr(S = 1)E_x[X | S = 1] + \Pr(S = 2)E_x[X | S = 2] = 0.75 \cdot 0 + 0.25 \cdot 3 = 0.75$$

$$\begin{aligned}\text{Var}[x] &= E_S[\text{Var}_x[X | S]] + \text{Var}_s[E_x[X | S]] \\ &= \Pr(S = 1)\text{Var}_x[X | S = 1] + \Pr(S = 2)\text{Var}_x[X | S = 2] \\ &\quad + \Pr(S = 1)(E_x[X | S = 1] - \mu)^2 + \Pr(S = 2)(E_x[X | S = 2] - \mu)^2 \\ &= 0.75 \cdot 1^2 + 0.25 \cdot 2^2 + 0.75(0 - 0.75)^2 + 0.25(3 - 0.75)^2 \\ &= 3.4375\end{aligned}$$

The following script was then used to verify the theoretical results.

```
1 mc = MarkovChain([0.75; 0.25], [0.99 0.01 ; 0.03 0.97]);
2 b1 = GaussD('Mean', 0, 'StDev', 1);
3 b2 = GaussD('Mean', 3, 'StDev', 2);
4 h = HMM(mc, [b1 ; b2]) ;
5 [X, S] = rand(h, 10000);
6 mean(X)
7 var(X)
```

This script gave the mean to be 0.7315 and the variance 3.3367, again close to the expected results.

3 HMM behaviour

For the HMM listed in section 1, 500 random contiguous output samples were plotted together with the state that the HMM was in.

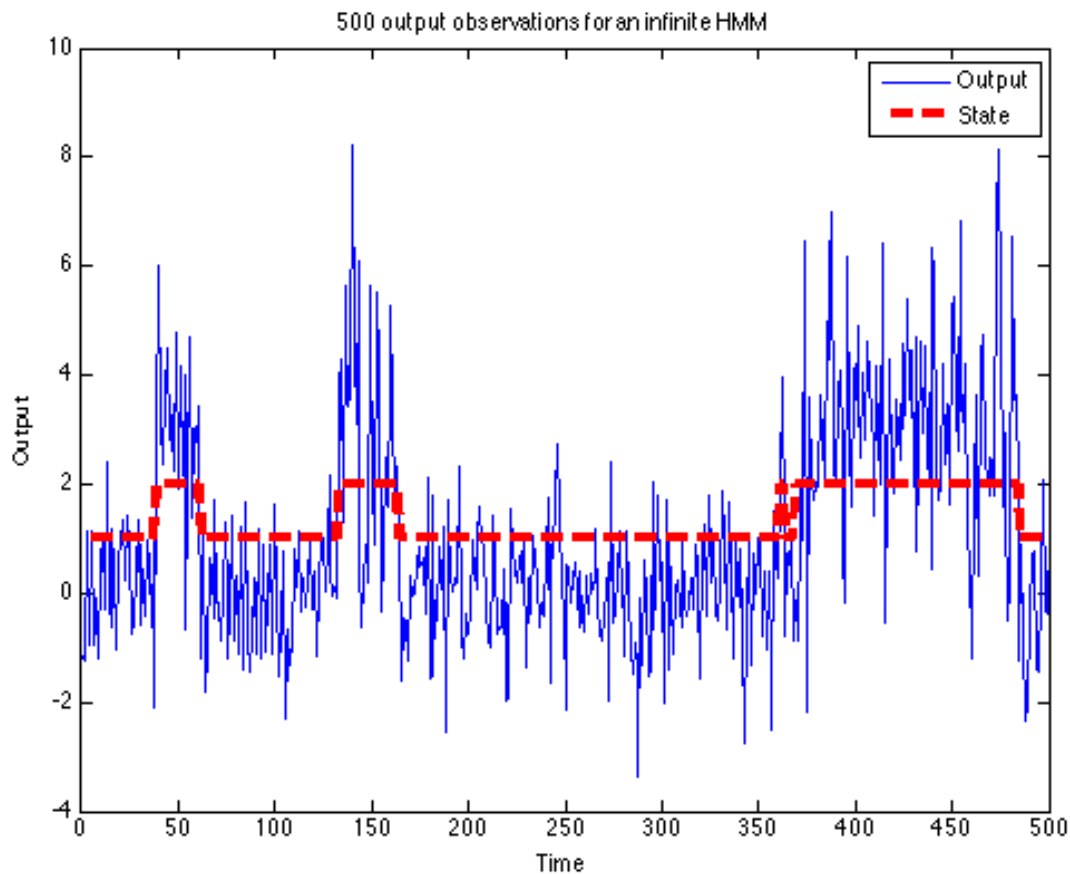


Figure 1: 500 random output observations from the HMM described in section 1. The blue line is the output from the HMM while the red dotted line is the state that the HMM is in.

It is clear from the picture that in the second state, the output is around 3 and in the first state the output is around 0. The second state output has a larger variation, because the standard deviation of the Gaussian for the second state is larger than that of the first state.

The output distributions for the HMM were then altered so that the mean of the output distribution for each state was 0. Again, 500 samples were plotted and the result is shown in figure 2.

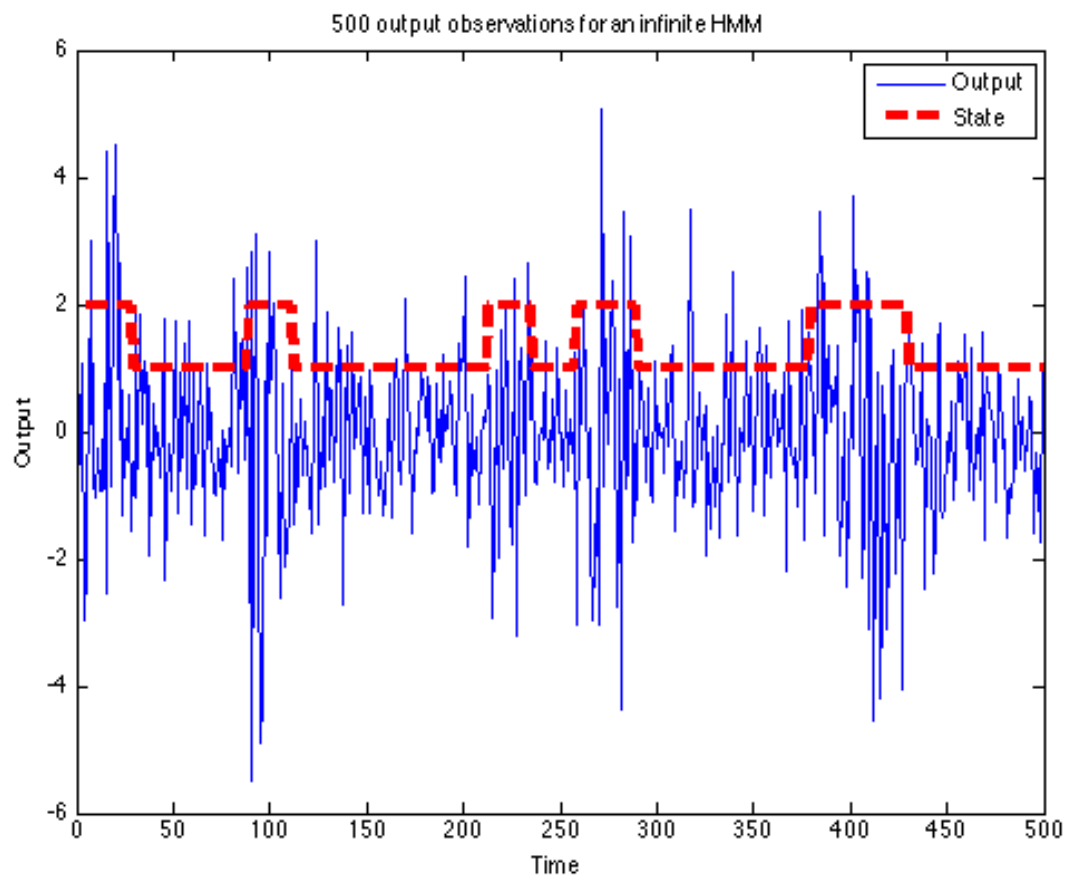


Figure 2: 500 random output observations from the HMM described in section 1 with $\mu_1 = \mu_2 = 0$

This time it is harder to distinguish the state sequence purely from looking at the output from the HMM. However, when the state is state 2, the standard deviation is higher so the output range is larger. This helps in identifying the state sequence but it is still more difficult than the previous model.

4 Finite HMM verification

To test finite duration HMMs the HMM from section 1 was modified so that..

$$A = \begin{pmatrix} 0.89 & 0.01 & 0.1 \\ 0.03 & 0.87 & 0.1 \end{pmatrix}$$

This means that the probability of the state sequence terminating is always 0.1, so on average a state sequence should terminate after 10 steps. The following script was written to test this practically.

```
1 mc = MarkovChain([0.75; 0.25], [0.89 0.01 0.1; 0.03 0.87 0.1]);
2
3 b1 = GaussD('Mean', 0, 'StDev', 1);
4 b2 = GaussD('Mean', 3, 'StDev', 2);
5
6 h = HMM(mc, [b1 ; b2]) ;
7
8 tests = 1000;
9 average = 0;
10
11 for i=1:tests
12     [X, S] = rand(h, 500);
13     average = average + length(X);
14 end
15
16 average/tests
```

This resulted in an average sequence length of 9.94, very close to what was expected.

5 Vector valued HMM verification

To verify that the HMM code worked for when the output distributions generate random vectors the following code was used.

```
1
2 mean1 = [1 2];
3 mean2 = [5 7];
4
5 cov1 = [1 0; 0 1];
6 cov2 = [2 1; 1 4];
7
8 b1 = GaussD('Mean', mean1, 'Covariance', cov1);
9 b2 = GaussD('Mean', mean2, 'Covariance', cov2);
10
11 h = HMM(mc, [b1 ; b2]) ;
12
13 [X, S] = rand(h, 500);
```

This worked as expected, with the X output vector consisting of feature vectors of length 2.