

EQ2341 Pattern Recognition and Machine Learning

Assignment 1

Chenting Zhang
chzha@kth.se

Hang Qin
hangq@kth.se

April 21, 2023

Questions

1.1

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 .

Given the initial probability, transition probability for each state as follows:

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

We could derive $P(S_t = j)$ for state $j = 1, 2$ when $t = 1, 2$

$$P(S_1 = 1) = 0.75$$

$$P(S_1 = 2) = 0.25$$

$$\begin{aligned} P(S_2 = 1) &= P(S_1 = 1) \times P(S_2 = 1|S_1 = 1) + P(S_1 = 2) \times P(S_2 = 1|S_1 = 2) \\ &= 0.75 \times 0.99 + 0.25 \times 0.03 = 0.75 \end{aligned}$$

$$\begin{aligned} P(S_2 = 2) &= P(S_1 = 1) \times P(S_2 = 2|S_1 = 1) + P(S_1 = 2) \times P(S_2 = 2|S_1 = 2) \\ &= 0.75 \times 0.01 + 0.25 \times 0.97 = 0.25 \end{aligned}$$

$P(S_t = j)$ Remains constant in the first two time instances. Similarly, we could derive that

$$\begin{aligned} P(S_t = 1) &= P(S_{t-1} = 1) \times P(S_t = 1|S_{t-1} = 1) + P(S_{t-1} = 2) \times P(S_t = 1|S_{t-1} = 2) \\ &= 0.75 \times 0.01 + 0.25 \times 0.97 = 0.75 \end{aligned}$$

$$\begin{aligned} P(S_t = 2) &= P(S_{t-1} = 1) \times P(S_t = 2|S_{t-1} = 1) + P(S_{t-1} = 2) \times P(S_t = 2|S_{t-1} = 2) \\ &= 0.75 \times 0.01 + 0.25 \times 0.97 = 0.25 \end{aligned}$$

Thus, $P(S_t = j)$ is actually constant for all t

1.2

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 $P(S_t)$.

Just as illustrated before, the theoretical results of constant state probability are 0.75 and 0.25 respectively. In the experiment, we could estimate the probability of the occurrence of each state by counting the number of "1" and "2" in the generated state sequence by setting $nData = 10000$. The results for single run and the average of 100 runs are shown in table 1

Table 1: State probability for different run times

state	1 run	100 runs	Relevant frequency
1	7345	7485.28	74.85%
2	2655	2514.72	25.15%

By comparing the empirical result with the theoretical result, we could conclude that our designed Markov Chain rand function works well.

1.3

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 Numpy functions $np.mean()$ and $np.var()$ to calculate the mean and variance of your generated sequence. The result should agree approximately with your theoretical values.

Firstly we derived the theoretical results of the mean and the variance of the observation. We have already known the mean and standard deviation of the conditional expectation of observation sequence X depending on the state sequence S .

$$b_1(x) \sim \mathcal{N}(\mu_1, \theta_1) \quad b_2(x) \sim \mathcal{N}(\mu_2, \theta_2)$$

where $\mu_1 = 0, \mu_2 = 3, \theta_1 = 1, \theta_2 = 2$

Then we can derive the mean and variance of the observation sequence as follows:

$$\begin{aligned} E[X] &= E_Z[E_X[X|S]] \\ &= P[S = 1]E_X[X|S = 1] + P[S = 2]E_X[X|S = 2] \\ &= 0.75 \end{aligned}$$

$$\begin{aligned} Var[x] &= E_S[Var_X[X|S]] + Var_S[E_X[X|S]] \\ E_S[Var_X[X|S]] &= P[S = 1] \times Var_X[X|S = 1] + P[S = 2] \times Var_X[X|S = 2] \\ &= 1.75 \end{aligned}$$

$$\begin{aligned} Var_S[E_X[X|S]] &= P[S = 1] \times Var_S[E_X[X|S = 1]] + P[S = 2] \times Var_S[E_X[X|S = 2]] \\ &= 0.75 \times (0 - 0.75)^2 + 0.25 \times (3 - 0.75)^2 \\ &= 1.6875 \\ Var[X] &= 3.4375 \end{aligned}$$

The theoretical results of mean and variance of observation output sequence are 0.75 and 3.4375 respectively. In the experiment, we could estimate the empirical results by calculating the mean and variance of the output sequence of length 10000 by setting $nData = 10000$. The results for single run and the average of 100 runs are shown in table 2.

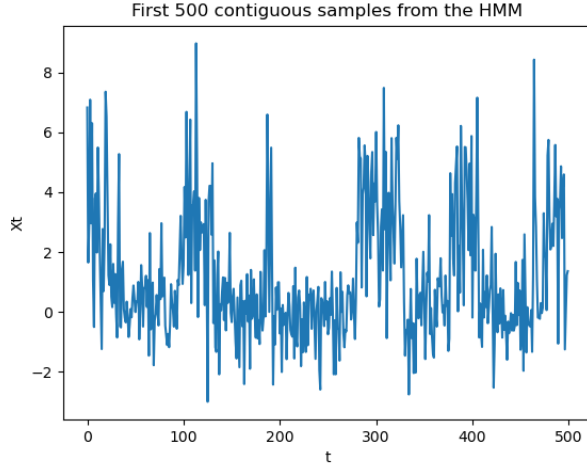


Figure 1: One example of the first 500 samples generated by HMM

Table 2: Empirical Mean and variance of X

	1 run	100 runs	Theoretical
Mean	0.7478	0.7481	0.75
Variance	3.3581	3.4139	3.4375

By comparing the empirical result with the theoretical result, we could conclude that our designed HMM rand function works well.

1.4

Figure 1 illustrates a series of 500 continuous samples generated from HMM. The distributions in B , the transition matrix A and the initial probability q characterize typical output of this HMM. From the figure 1, the output sequence X_t has two pattern clearly, because the transition probabilities make the next state is more likely to stay at the current state, and the two Gaussian distribution has clear difference in second order statistics. For example, when t is around 300, X_t has larger mean and variation, which indicates that these samples are most likely generated from distribution b_2 .

1.5

Figure 2 illustrates a series of 500 continuous samples generated from HMM after μ_1 and μ_2 are set to 0. The most obvious difference of this HMM is that the generated samples are likely evenly distributed around zero because both two Gaussian distribution have zero mean. Estimating the state sequence of the underlying Markov chain from the observed output variables in this case is very difficult, because the small difference in variance does not affect the characteristic of the generated samples to much.

1.6

The function *testfinitehmm* is used to check if our hmm can handle the finite situation. The new test hmm is constructed with the following initial and transition probabilities.

$$q = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix} \quad A = \begin{pmatrix} 0.4 & 0.4 & 0.1 \\ 0.4 & 0.4 & 0.1 \end{pmatrix}.$$

At each state, the probability of transition to the end state is 0.1. The *testfinitehmm* will set the maximum length as a very large number, and check the real length of the sample sequence

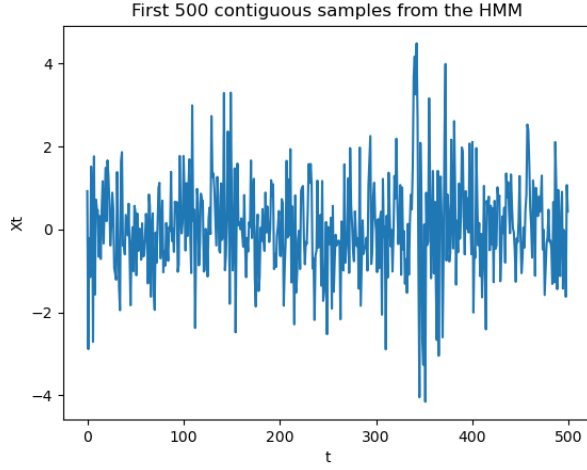


Figure 2: One example of the first 500 samples generated by HMM when $\mu_1 = \mu_2 = 0$

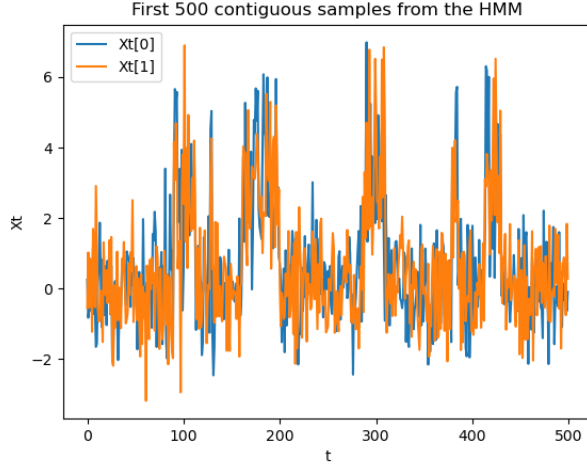


Figure 3: One example of the first 500 samples generated by HMM when $\mu_1 = \mu_2 = 0$

generated by hmm. Most likely the real length should smaller than the maximum length if the maximum length is very large, and this is evidence that our hmm can handle the finite situation.

1.7

In this case we define a new test hmm with new distribution to make it generate a random vector as a sample. The two Gaussian distribution is defined with following covariance matrices respectively.

$$\Sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

The Markov chain is the same one as the first test. Figure 3 shows that the test hmm can generate random vector with two correlated elements.