Pattern Recognition - Report for Assignment 1

Navneet Agrawal, navneet@kth.se
Lars Kuger, kuger@kth.se

I. Abstract

II. Background and Problem Formulation

In order to build a pattern recognition system that is able to recognize speech, melodies or written characters, the *hidden Markov model* (HMM) can be used. A HMM is defined by a set of parameters $\lambda = \{q, A, B\}$ where q is a vector with the probabilities for the initial state, A is the so-called transition matrix which describes the probabilities of going from state S_{i-1} to state S_i and B is a vector containing the output probability density functions of each state $f_{\mathbf{X}|S_i}(\mathbf{X} = \mathbf{x}|S_i = k)$. A more detailed description of HMMs can be found in [1]. The problem that will be dealt with in the course of this report is the implementation of an HMM signal source in MATLAB based on the provided code as well as the verification of its correct functionality.

III. Methodology

The implementation of the code was guided by the instructions and the function definitions which were already given. Since HMMs can also be used in further contexts it is useful to build a toolbox containing all relevant parts of a pattern recognition system. This toolbox is and will be developed using object oriented programming in MATLAB to keep the code well-arranged and similar to the model. For a better understanding the code of the used classes that were already given was available.

The verification of the code can be done by comparison of the values obtained from the MATLAB model to the theoretically expected values. In this particular case, stationary probabilities, expected values and variances will be used. In addition, hypothesis regarding the shape of output curves in a graph will be stated and compared to the plots obtained from the model. This is done to confirm that the model is working as expected.

IV. Results

A. Code

The code of the @DiscreteD/rand, @MarkovChain/rand, and @HMM/rand is enclosed with this report as MATLAB scripts. The code for @DiscreteD/rand uses *Acceptance-Rejection*[2] algorithm for generating random values from a probability distribution function.

B. Theoretical values

In the following, a stationary state distribution as well as the expected value of the output $E[X_t]$ and its variance $var[X_t]$ will be calculated. An HMM with the following parameters λ is used

$$q = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix} \qquad A = \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix} \qquad B = \begin{pmatrix} b_1(x) \\ b_2(x) \end{pmatrix} \tag{1}$$

with $b_1(x)$, $b_2(x)$ scalar Gaussian density functions with $\mu_1 = 0$, $\sigma_1 = 1$ and $\mu_2 = 3$, $\sigma_2 = 2$, respectively.

According to Definition 5.2 in [1], a stationary state distribution requires that $\mathbf{p} = A^T \mathbf{p}$ holds where the elements are $p_j = P[S_t = j]$. In order to find this particular $\mathbf{p}$, the eigenvalues and, in case there is an eigenvalue equal to one, the corresponding eigenvector is needed. The eigenvalues θ can be obtained from the characteristic equation $det(A - \theta I) = \theta^2 - 1.96\theta + 0.96 \overset{!}{=} 0$ which gives $\theta_1 = 1$. The corresponding eigenvector $\mathbf{p}_1$ then is $\mathbf{p}_1 = (0.75\ 0.25)^T$. It is clear that the stationary state distribution is equal to the initial state distribution $\mathbf{p}_1 = \mathbf{q}$. Therefore, it is safe to conclude that $P(S_t = j),\ j \in \{1, 2\}\ \forall\ t$ is constant.

The calculation of the expected value can be done as shown below

$$\begin{aligned}
E_X[X] &= E_S[E_X[X|S = i]] && (2)\\
&= \sum_i E_X[X|S = i] P_S[S = i] && (3)\\
&= \sum_i E_X[B_i] P_S[S = i] && (4)\\
&= 0 \cdot 0.75 + 3 \cdot 0.25 && (5)\\
&= 0.75\ . && (6)
\end{aligned}$$

Note that the random variable B_i corresponds to output of the probability density function $b_i(x)$ and the dependency of X on S was used in eq. (2).

The same dependency will be used in the calculation of the variance as given below

$$\begin{aligned}
var[X] &= E_S[var_X[X|S]] + var_S[E_X[X|S]] && (7)\\
&= \sum_i P_S[i] \cdot (var_X[X|i] + && (8)\\
&\quad +(E_X[X|i] - E_S[E_X[X|i]])^2) && (9)\\
&= \sum_i P_S[i] \cdot (var_X[X|i] + && (10)\\
&\quad +(E_X[X|i] - E_X[X])^2) && (11)\\
&= 0.75(\sigma_1^2 + \mu_1 - E_X[X]) + && (12)\\
&\quad +0.25(\sigma_2^2 + \mu_2 - E_X[X]) && (13)\\
&\approx 3.4375\ . && (14)
\end{aligned}$$

TABLE I: In the table the relative frequency of the states $S = 1$ and $S = 2$ as $\hat{p}_1$ and $\hat{p}_2$ respectively is shown for every time N that a sequence of $T = 10000$ states was generated.

N	1	2	3	4	5	6	7	8	9	10
$\hat{p}_1$	0.76	0.74	0.77	0.71	0.80	0.69	0.74	0.74	0.70	0.68
$\hat{p}_2$	0.24	0.26	0.23	0.29	0.20	0.31	0.26	0.26	0.30	0.32

TABLE II: In the table the expected value $E[X]$ and the variance $var[X]$ of the output random variable X is shown for every time N that a sequence of $T = 10000$ output values was generated.

N	1	2	3	4	5	6	7	8	9	10
$E[X]$	0.80	0.79	0.79	0.62	0.72	0.69	0.71	0.65	0.85	0.61
$var[X]$	3.50	3.52	3.69	3.01	3.37	3.38	3.38	3.11	3.66	3.06

C. Comparison to model

In the previous section several theoretical values were calculated. In the following these will be compared to the values that we obtain from the implemented model.

To begin with, the stationary state distribution was calculated to be $\mathbf{p}_1 = (0.75\ 0.25)^T$ and constant for all t. This can be compared to the relative frequency of the states in a sequence of $T = 10000$ states. Such sequences were generated ten times as can be seen from table I. The mean value of the relative frequencies are $\hat{\mathbf{p}} = (0.7335\ 0.2665)$ which is very similar to the value $\mathbf{p}$ that we were expecting.

Furthermore, the expected value $E[X] = 0.75$ and the variance $var[x] = 3.4375$ were calculated in the previous section. These theoretical values can be compared to the mean of the expected values $\hat{\mu}_X = 0.7228$ and variances $v\hat{a}r[X] = 3.3693$ found from the model after generating 10 sequences of length $T = 10000$ as shown in table II. Again, we can conclude that the theoretical values and the values obtained from the model are approximately equal.

D. Infinite duration HMM

Using HMM parameter λ given in 1, we plots a sequence of 500 random values generated using @HMM/rand as shown in figure 1. There are two states in the system: state S_1 and S_2. The random values with mean around 0 (lower variance) correspond to state S_1 and those with mean around 3 (higher variance) correspond to state S_2. Another observation from the graph is that the state transitions are rare because transition probabilities from one state to another are very low i.e. $A[1, 2] = 0.01$ or $A[2, 1] = 0.03$.
In case of zero mean distributions (as shown in the lower plot in figure 1), the only distinguishing factor is the variance. The patches of low and high variance in the plot may possibly belong to S_1 and S_2 respectively. However it is very difficult to estimate the state sequence of underlying Markov chain from just the observed output sequence X.

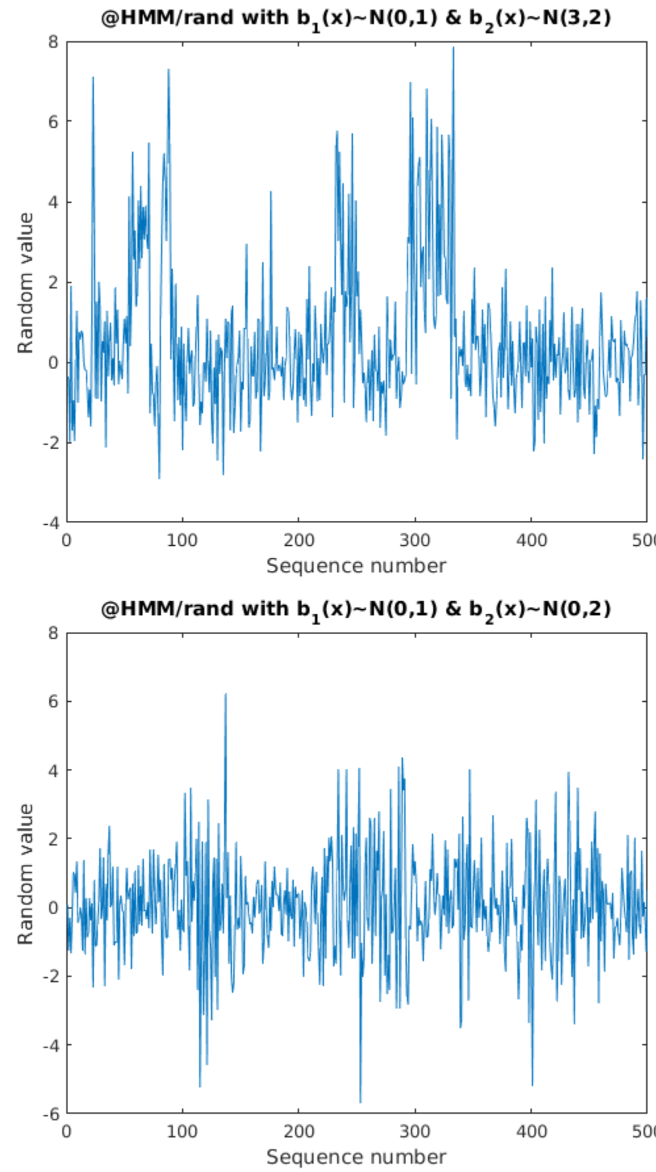

Fig. 1: Infinite duration HMM

E. Finite duration HMM

A *finite* duration HMM has an additional exit state in its state transition matrix. When the sequence reach the exit state, the Markov chain stops. In order to test our functions for generating finite duration HMM we changed the transition matrix to have an exit state. New transition matrix is:

$$A = \begin{bmatrix} 0.88 & 0.02 & 0.1 \\ 0.04 & 0.86 & 0.1 \end{bmatrix}$$

We run the finite HMM using this new parameter A for a sequence of 500 random numbers. For three such runs, the sequence number (duration) after which the HMM reaches the exit state are 8, 17, 6. In order to verify our implementation of finite HMM, we can compare the cumulative probability density (cdf) of number of states (N) before the exit-state, to a theoretical value given by 15.

$$P(X \le N) = \Sigma_{t=1}^{N}(1 - P(S_{exit}))^{t-1} * P(S_{exit}) \qquad (15)$$

For N = 3, this value comes out to be 0.2710. Figure 2 shows the cdf of N before exit state generated using HMM function. We can see that the generated finite HMM's N is approximately equal to the theoretical value.

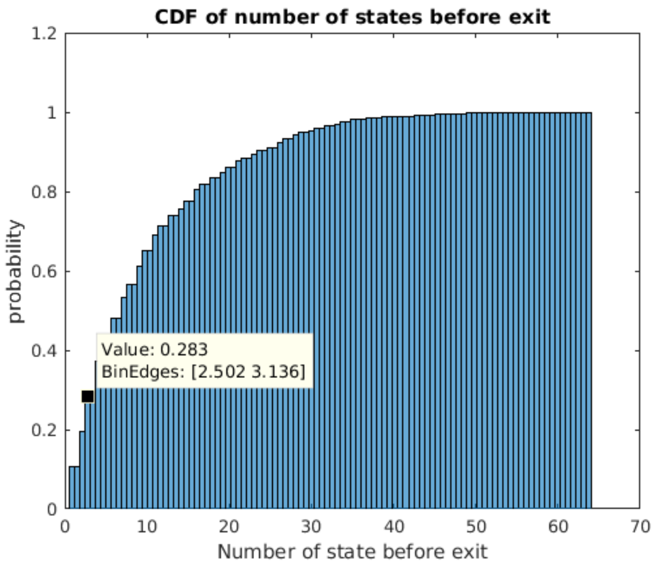

Fig. 2: Finite Duration HMM (run 500)

F. State-conditional random distribution (B) as vector

We use the following state-conditional Gaussian vector distributions: $b_1(x)\ \mathcal{N}(\mu_1, \Sigma_1)$ and $b_2(x)\ \mathcal{N}(\mu_2, \Sigma_2)$ where,

$$\mu_1 = [-6, 6] \quad \& \quad \Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$$

$$\mu_2 = [2, -2] \quad \& \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The output generated from these parameters is a sequence of random vector X with size $(2 \times N)$. We have plotted both random sequences separately in figure 3. The transitions in X_1 and X_2 are occurring at same instants. The state-conditional distributions used to generate X_1 and X_2 can be deduced by the mean of the random sequence in the plot. As you can see, in figure 3 (black), the random sequence starts at mean value -6 which corresponds to $b_1(x)$ - $\mu_1(1)$ and transition to mean value 2 which corresponds to $b_2(x)$ - $\mu_2(1)$. Similar behavior can be observed in plot below (red line) where means are changing from same state transitions from $\mu_1(2)$ to $\mu_2(2)$. Slight different in variance can also be seen in these two plots for means around $(6, -6)$, due to Σ_1.

V. Conclusions

The MATLAB functions to generate HMM model are complete and working as expected. These functions can be used to generate a sequence of HMM random vectors to fit relevant models. The tests performed during this assignment provided an insight into behavior and characteristics of HMM models.

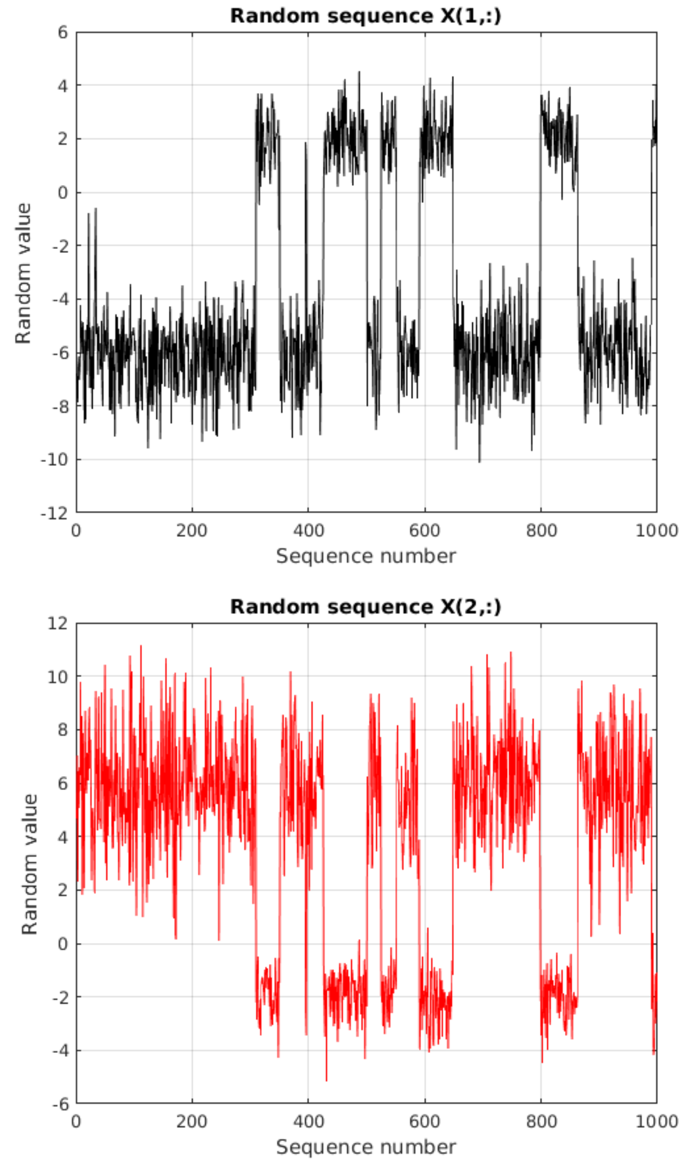

Fig. 3: HMM random sequence using vectorized B

References

[1] Arne Leijon and Gustav Eje Henter, *Pattern Recognition - Fundamental Theory and Exercise Problems*, KTH School of Electrical Engineering, 2015

[2] Luca Martino, Joaquin Miguez *Generalized rejection sampling schemes and applications in signal processing*, Signal Processing, 2010