Checkpoint 11.1

Use the matlab script main11a.m to create a random transition matrix. This matrix will be used as a *teacher* to create training and test sequences.

First, we analyze the properties of Markov transition matrices. Let consider a large ensemble of parallel chains starting from the same initial state $y_1 = 1$. The n'th state of the q'th ensemble member is denoted $y_n^{(q)}$. At any given time n we can consider the distribution of states within the ensemble. Let $P_n(j)$ be the probability that ensemble members are in state j at time n. Establish an argument for the temporal evolution of the ensemble distribution, the so-called *Master equation*

$$P_{n+1}(j') = \sum_{j=1}^{K} P_n(j) a_{j,j'}$$

or in matrix notation

$$P_{n+1} = P_n a$$

what are the dimensions of vector P and matrix a?

Under mild conditions (e.g., all elements of a are positive)[5] the ensemble dynamics will converge to a fixed point distribution, called the *stationary distribution* which satisfies,

$$P_*(j') = \sum_{j=1}^{K} P_*(j) a_{j,j'}.$$

This is an eigenvalue problem (explain!)

$$P_* = P_* a.$$

Under the same conditions on a the long term distribution of states attained by an individual chain will also be distributed as $P_*(j)$. Investigate the stationary distribution for the transition matrix a and explain how the program estimates it. Use the *teacher* transition matrix to create increasing length sequences, and observe how the histogram of the observed sequences converge to the stationary distribution. Explain function getint.m.

Now we turn to learning by maximum likelihood (i.e. $\alpha_{j,j'} = 1$). Verify the maximum likelihood estimate of the transition matrix (either analytically or numerically). Use Matlab script (main11a.m) to generate increasing length sequences. Train a *student* transition matrix on these sequences and show that the relative error of the student matrix converges to zero, hence, the student matrix converges to the teacher matrix for large training sets.

The dimensions of the P and a matrixes are shown below:

$$\begin{bmatrix} P_{n-1}(0) \\ P_{n-1}(1) \\ \dots \\ P_{n-1}(K) \end{bmatrix} = \begin{bmatrix} P_n(0) & P_n(1) & \dots & P_n(K) \end{bmatrix} \begin{bmatrix} a_{0\,0'} & a_{0\,1'} & .. & a_{0\,K'} \\ a_{1\,0'} & a_{1\,1'} & .. & a_{1\,K'} \\ .. & .. & .. & .. \\ a_{K\,0'} & a_{K\,1'} & .. & a_{K\,K'} \end{bmatrix}$$

Thus, the dimensions of P are Kx1 or 1xK and the dimensions of a are K x K.

The equation $P_* a = P_*$ look quite similar to the equation of eigenvectors and

eigenvalues $Mv = \lambda v$ with eigenvalues equal to 1. In fact, we can present it the same way by transposing both terms, which give us $a^T {P_*}^T = {P_*}^T$ which means that the transition matrix transpose eigenvector is equal to the transpose of the stationary matrix with eigenvalue equal to 1. When we obtain more than one eigenvector with eigenvalue equal to 1 that means that there is an associated stationary distribution and that the markov chain is reducible.

This is calculated by the program in an iterative fasion, using the power iteration method. First, we need to assume that $A^k q = \lambda^k q$ which is, in general, true for all the cases. In this contest, if we take $v^{(0)}$ as an approximation of an eigenvector of A, with $\|v^{(0)}\| = 1$, we can write this vector as a linear combination of the eigenvectors of A for certain weights c:

$$v^{(0)} = c_1 q_1 + \ldots + c_n q_n,$$

and we will assume for now that $c_1 \neq 0$.

Now

$$Av^{(0)} = c_1 \lambda_1 q_1 + c_2 \lambda_2 q_2 + \ldots + c_n \lambda_n q_n$$

and so

$$\begin{aligned} A^k v^{(0)} &= c_1 \lambda_1^k q_1 + c_2 \lambda_2^k q_2 + \ldots + c_n \lambda_n^k q_n \\ &= \lambda_1^k \left(c_1 q_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k q_2 + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k q_n \right). \end{aligned}$$

Since the eigenvalues are assumed to be real, distinct, and ordered by decreasing magnitude, it follows that for all $i = 2, \ldots, n$,

$$\lim_{k \to \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^k = 0.$$

So, as k increases, $A^k v^{(0)}$ approaches $c_1 \lambda_1^k q_1$, and thus for large values of k,

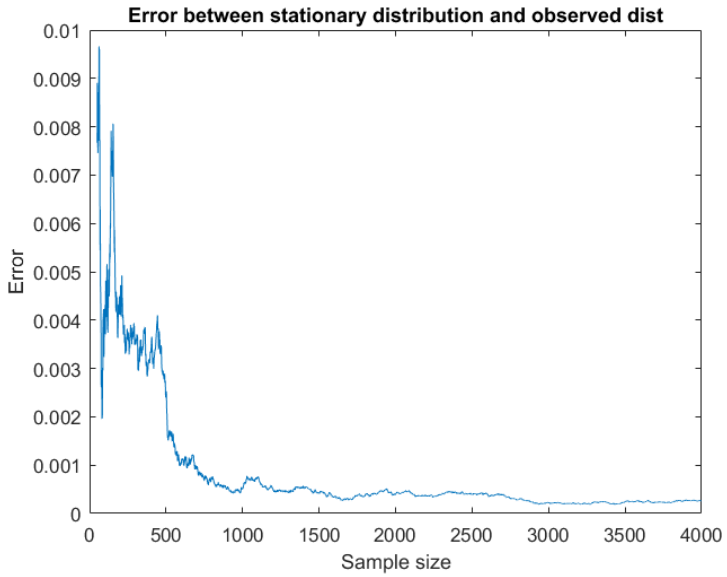
$$q_1 \approx \frac{A^k v^{(0)}}{\left\| A^k v^{(0)} \right\|}.$$

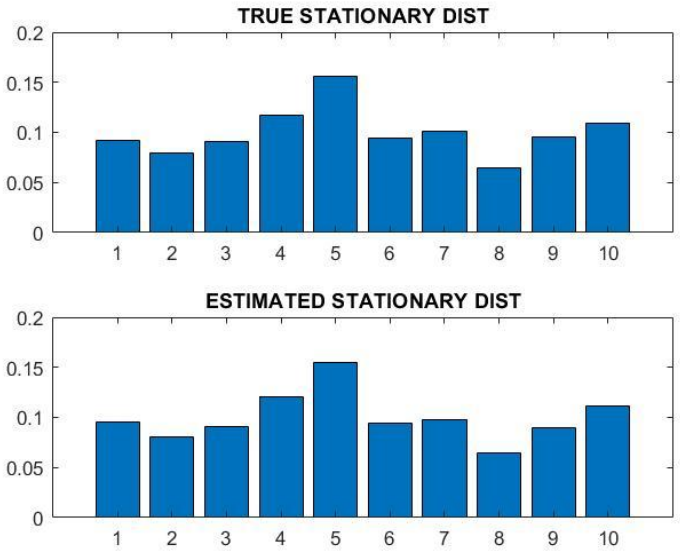
Thus, we can iterate to find the value of q1 (p in the code) by iterating with the normalized transition matrix and an initialized random vector p with norm equal to 1.

Once the value of the eigenvector p is calculated the next step is to calculate the teacher transition matrix. This is calculated by taking the cumulative sum of each of the raws in a_jj' normalized. Afterwards we are going to use this cumulative matrix to estimate a large sequence of states. In order to do so, we are going to use the function getint, which algorithm can be explained as follows:

Firstly, we take a random number between 0 and 1.Then, given a row of the ac matrix (the cumulative probabilities to other states), we are going to select the estate number of the next value after the random number in the row. This will give us a path to continue with our sequence, in a fashion given by the probabilities of the teacher matrix.

In this iterative process we can see how the value of the observed sequences converges to the stationary distribution. In order to visualize this we are going to plot the error between the stationary distribution out of the eigenvector property and the distribution that we are creating.





We see that even when the sample size is really small the error is still really small. However, we can consider that it converges at around sample size of 3000.

We can also see the final result in a histogram form in the following graph:

Checkpoint 11.2

We design a simple "switching" non-stationarity with two Markov models tak
Model 1 generates the states in the intervals $[1, N_1]$ and $[N_1 + N_2 + 1, 2N_1 +$
Model 2 generates the states in the interval $[N_1 + 1, N_1 + N_2]$.

We estimate Markov models using the MAP estimator for overlapping wind
a set of different window sizes. Compare the likelihood of the transition matrix
on the training data and the likelihoods of the two "true" models.

We evaluate the estimated Markov models by their L_1 distances to the tr
and using the log-likelihood on test data. Discuss the impact of the window siz
prior α on the estimated Markov model and these generalization performance
Which window size would you recommend?