Speech Recognition

EEEM030 Assignment 2

Faiz Ahmad Khan

University of Surrey

2021/2022

# Contents

[Introduction 1](#_Toc89866609)

[Description 1](#_Toc89866610)

[Implementation 2](#_Toc89866611)

[State Topology and Trellis Diagram 2](#_Toc89866612)

[Forward Procedure 4](#_Toc89866613)

[Backward Procedure 5](#_Toc89866614)

[Occupation Likelihood 6](#_Toc89866615)

[Transition Likelihood 7](#_Toc89866616)

[Baum-Welch 8](#_Toc89866617)

[Comparison 9](#_Toc89866618)

[Conclusion 11](#_Toc89866619)

[Bibliography 12](#_Toc89866620)

[Appendix 13](#_Toc89866621)

[Code 13](#_Toc89866622)

[Main.m 13](#_Toc89866623)

[Forward.m 15](#_Toc89866624)

[Backward.m 16](#_Toc89866625)

[Occupation.m 16](#_Toc89866626)

[Transition.m 16](#_Toc89866627)

[W.m 17](#_Toc89866628)

[Results 18](#_Toc89866629)

# Table of Figures

[Figure 1: State Transition Probabilities (A) 2](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869148)

[Figure 2: Observation Probabilities (B) 2](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869149)

[Figure 3: State Topology 2](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869150)

[Figure 4: Trellis Diagram (1) 3](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869151)

[Figure 5: Forward Likelihood () 4](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869152)

[Figure 6: Backward Likelihood (O1) 5](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869153)

[Figure 7: Occupation Likelihood () 6](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869154)

[Figure 8: Occupation Likelihood vs Naive Bayes () 6](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869155)

[Figure 9: Accumulated Transition Likelihood () 7](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869156)

[Figure 10: Accumulated Occupation Likelihood () 7](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869157)

[Figure 11: Re-estimated State Transition Probabilities () 8](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869158)

[Figure 12: Re-estimated Observation Probabilities () 8](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869159)

[Figure 13: Viterbi State Transition Probabilities () 9](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869160)

[Figure 14: Viterbi Observation Probabilities () 9](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869161)

[Figure 15: vs 9](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869162)

[Figure 16: vs 10](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869163)

[Figure 17: Forward Likelihood (O2-O5) 18](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869164)

[Figure 18: Backward Likelihood (O2-O5) 18](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869165)

[Figure 19: Occupation Likelihood (O2-O5) 18](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869166)

[Figure 20: P(O1-O5|) - Forward & Backward 19](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869167)

[Figure 21: Accumulated Occupation Likelihood (O1-O5) 19](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869168)

[Figure 22: Accumulated Translation Likelihood (O1-O5) 20](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869169)

[Figure 23: Occupation Likelihood vs Naive Bayes (O2-O5) 21](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869170)

[Figure 24: Transition Likelihood (O1) 22](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869171)

[Figure 25: Entry Row Matrix (O1-O5) 22](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869172)

[Figure 26: Exit Column Matrix (O1-O5) 22](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869173)

[Figure 27: Transition Likelihood (O2) 23](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869174)

[Figure 28: Transition Likelihood (O3) 24](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869175)

[Figure 29: Transition Likelihood (O4) 24](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869176)

[Figure 30: Transition Likelihood (O5) 25](file:///C:\Users\faiza\OneDrive\Documents\Speech%20Recognition.docx#_Toc89869177)

# Introduction

Speech recognition is the process of converting human speech into a written/text format. It is different from voice recognition which is just the process of identifying an individual’s voice. Speech recognition has seen a surge in applications over the past decade. It has become common for consumer technologies to use speech recognition to ease accessibility and human-computer interaction. Various algorithms have been developed for speech recognition and one of the oldest and most successfully deployed is the Hidden Markov Model (HMM). This report will assess the effectiveness of HMM as a recursive likelihood algorithm and observe its application for the recognition of given data.

# Description

A Markov chain is defined as,” A stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.” (Lexico.com, 2021). A Hidden Markov Model (HMM) is a statistical model where a sequence of observations is observed and the sequence of states that the model went through to generate the observations is unknown. HMMs are analysed to recover the sequence of states from the observed data. (MathWorks, 2021). To analyse an HMM, the occupation and transition probabilities must be estimated. Then, the Baum–Welch algorithm is used to find the [maximum likelihood](https://en.wikipedia.org/wiki/Maximum_likelihood) estimate of the parameters given a set of observed feature vectors. The algorithm uses the forward-backward algorithm to estimate parameters.

# Implementation

## State Topology and Trellis Diagram

First, given the set of state transition probabilities and discrete observation probabilities, as shown in figures 1 and 2 respectively, the state topology and trellis diagram can be modelled for the first set of observations. The state topology illustrates the probabilities of entry and exit as well as moving from one state to another. The trellis diagram illustrates the possible state changes that can occur for a given set of observations. The state topology and trellis diagram for observation set 1-{6, 4, 2, 1, 3} can be observed in figures 3 and 4 respectively.

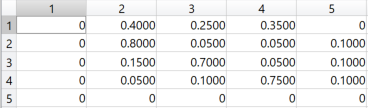


Figure : State Transition Probabilities (A)

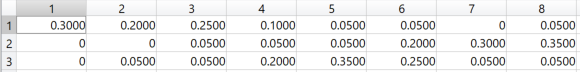


Figure : Observation Probabilities (B)

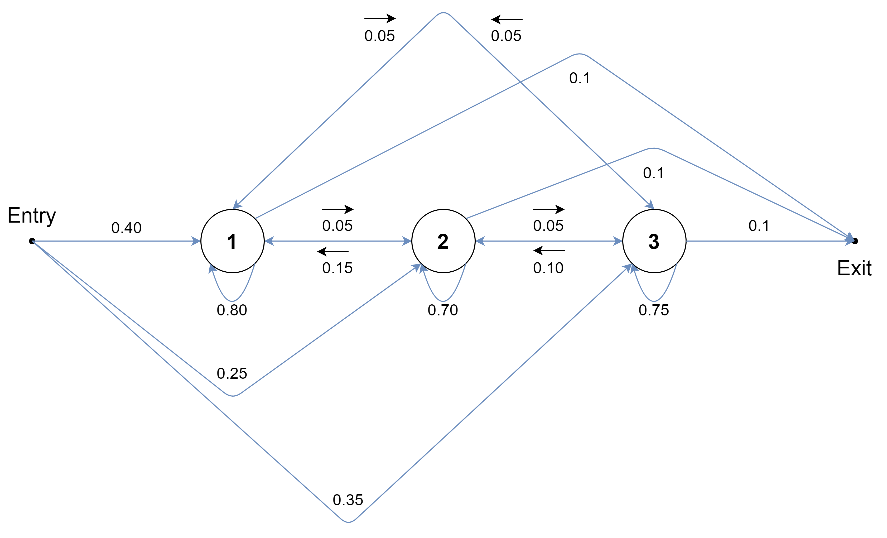


Figure : State Topology

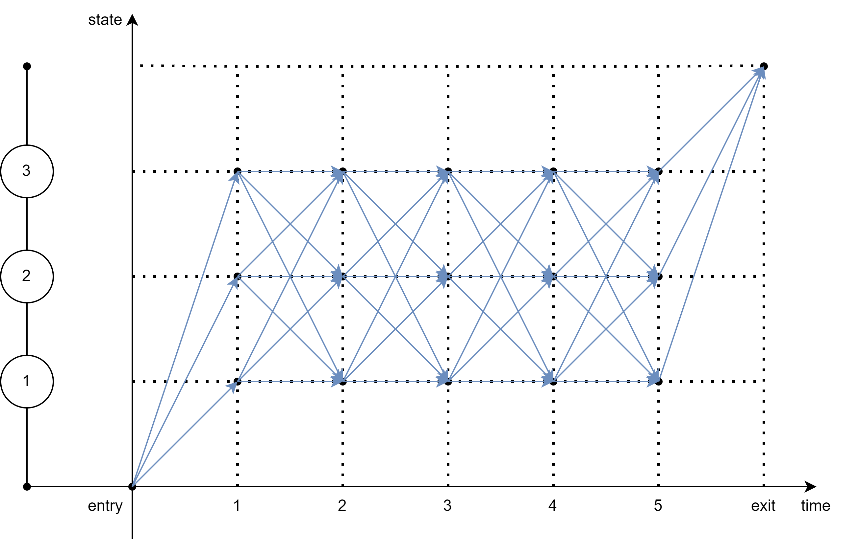


Figure : Trellis Diagram (1)

## Forward Procedure

Next, the forward-backward algorithm is used to calculate forward, and backward likelihoods denoted by and respectively.

The forward likelihood is the probability of observing a sequence of observations from time to and ending up at state at time for a given HMM (). It is given by the equation:

At time = 1, the value of forward likelihood is the probability of starting in state and observing at . This is given by the equation below where is the number of states, is the entry probability for state and is the observation probability of observation for state :

This equation is encoded by line [12](#_Forward.m) of the [Forward.m](#_Forward.m) function which is run once for each of the states.

At time greater than 1, the value of forward likelihood is given by the equation below where is the probability of going from state to :

The equation within the square brackets is encoded by line [17](#_Forward.m) of the [Forward.m](#_Forward.m) function and the final value of is evaluated by line [19](#_Forward.m).

Finally, to find the probability that a given HMM generated observed sequence of observations is given by the equation below where is the probability of exit from state :

This equation is encoded by line [27](#_Forward.m) of the [Forward.m](#_Forward.m) function. For observation set the overall likelihood of the observations is equal to and the forward likelihood matrix can be observed in figure 5.

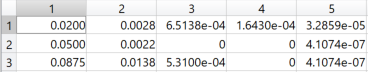


Figure : Forward Likelihood ()

## Backward Procedure

The backward likelihood is the probability of observing a sequence of observations from time to given that at time , the state is for given HMM . It is given by the following equation:

The backward likelihood is calculated starting from the last observation to the first, hence initially the value of time is i.e., the index of the last observation. At time the backward likelihood is the probability of being in state at that time. It is given by the equation below:

This equation is encoded by line [12](#_Backward.m) of the [Backward.m](#_Backward.m) function which is run once for each of the states.

At time lower than , the value of backward likelihood is given by the equation below:

This equation is encoded by line [17](#_Backward.m) of the [Backward.m](#_Backward.m) function and the final value of is evaluated by line [19](#_Backward.m).

The overall likelihood of observations can also be calculated from the backward likelihood values and is given by the equation below:

This equation is encoded by line [27](#_Backward.m) of the [Backward.m](#_Backward.m) function. For observation set the overall likelihood of the observations calculated through backward likelihood is also equal to and the backward likelihood matrix can be observed in figure 6. Comparison of overall likelihood for all observation sets for both backward and forward procedure can be observed in figure 20.

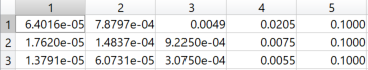


Figure : Backward Likelihood (O1)

## Occupation Likelihood

Once the forward and backward likelihoods have been evaluated, the next step is to train the HMM. This is done by evaluating the occupation and transition likelihoods.

The occupation likelihood is important as it provides the probabilities of being in each state at a given time. It is given by the equation below where is the probability that we are in state at time given set of observations :

Mathematically, it can be evaluated as the product of forward and backward likelihoods normalised by the overall likelihood. This equation is as follows:

This equation is encoded by lines [4](#_Occupation.m) and [5](#_Occupation.m) of the [Occupation.m](#_Occupation.m) function. For observation set , the occupation likelihood can be observed in figure 7. These predictions can be compared to those of a Naïve Bayes classifier and observed in figure 8.

A Naïve Bayes classifier does not capture dependencies between input variables whereas in an HMM the assumption is that the probability of a particular state is dependant on the previous state. Thus, the Naïve Bayes model predicts the state for a set of probabilities associated with an observation whereas the HMM predicts a class sequence for a sequence of observations. In a way, the HMM can be formulated as a product over single Naïve Bayes models. (Batista, 2017). This results in the HMM providing more accurate predictions for state probability for a set of observations.

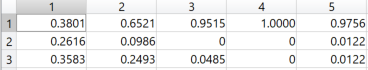


Figure : Occupation Likelihood ()

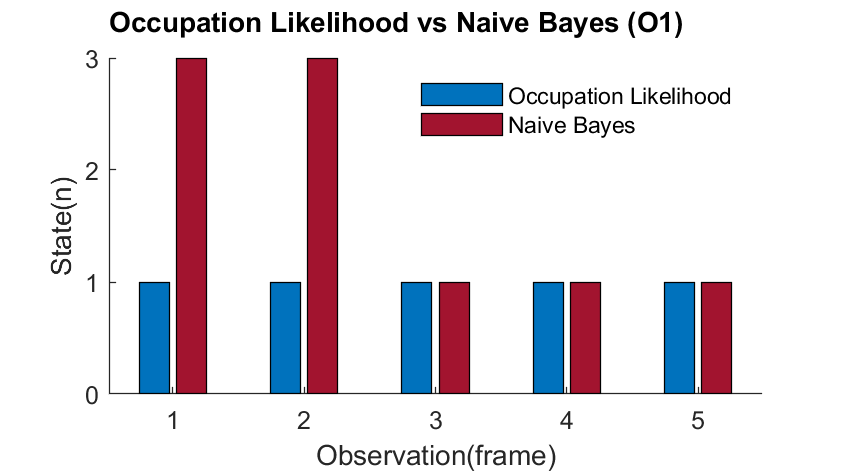


Figure : Occupation Likelihood vs Naive Bayes ()

## Transition Likelihood

The transition likelihood is the probability that we are in state and transition to state next, given set of observations . It is given by the equation below:

Mathematically, the transition likelihood is given by the following equation:

This equation is encoded by line [14](#_Transition.m) of the [Transition.m](#_Transition.m) function. To get transition likelihood over the entire period of observations , the entry and exit occupation likelihoods are concatenated to the calculated transition likelihood values. This is encoded in lines [21](#_Transition.m) and [25](#_Transition.m) of the [Transition.m](#_Transition.m) function. The transition likelihood matrices for along with entry and exit matrices for all observations can be observed in figure 24, 25 and 26 respectively.

The accumulated values for transition and occupation likelihoods give the expected number of transitions from to and the expected number of transitions from respectively. These accumulated values can be observed in figures 9 and 10.

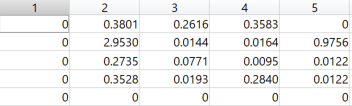


Figure : Accumulated Transition Likelihood ()



Figure : Accumulated Occupation Likelihood ()

## Baum-Welch

Once the accumulated likelihoods have been obtained for the entire set of observations, the Baum- Welch algorithm can be used to re-estimate the HMM and get the new model (). This is done by getting new state transition () and discrete observation () probabilities. The equation for these is given below:

For the equation for new state transition probability, the numerator and denominator values are evaluated for each set of observations and encoded in lines [17](#_Transition.m) and [29](#_Transition.m) of the [Transition.m](#_Transition.m) function respectively.

In the equation for new discrete observation probability, the numerator is the expected number of times we are in state and observe observation out of the given unique observations. This is done by introducing function which returns if or else . This function is encoded by the [W.m](#_W.m) function. For this equation, the numerator and denominator values are evaluated for each set of observations and encoded in lines [44](#_Transition.m) and [36](#_Transition.m) of the [Transition.m](#_Transition.m) function respectively.

Finally, the sum of numerators and denominators for new state transition and discrete observation probabilities over the entire set of observations is evaluated. This is encoded in lines [79](#_Main.m) to [84](#_Main.m) of the [Main.m](#_Main.m) script. Then and are evaluated in lines [87](#_Main.m) to [90](#_Main.m) of the [Main.m](#_Main.m) script. The re-estimated state transition and discrete observation probabilities can be viewed in figures 11 and 12 respectively.

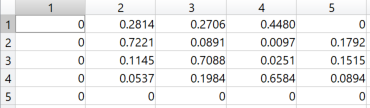


Figure : Re-estimated State Transition Probabilities ()

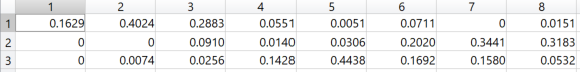


Figure : Re-estimated Observation Probabilities ()

## Comparison

The new state transition and discrete observation probabilities can be compared with the original probabilities. They can also be compared to the probabilities and obtained by re-estimating with Viterbi training. These can be observed in figures 13 and 14 respectively. Additionally, can be plotted against both and . These can be observed in figures 15 and 16.

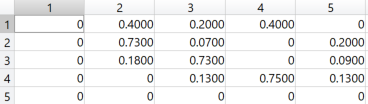


Figure : Viterbi State Transition Probabilities ()

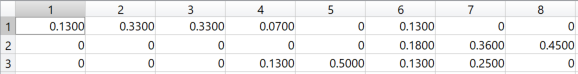


Figure : Viterbi Observation Probabilities ()

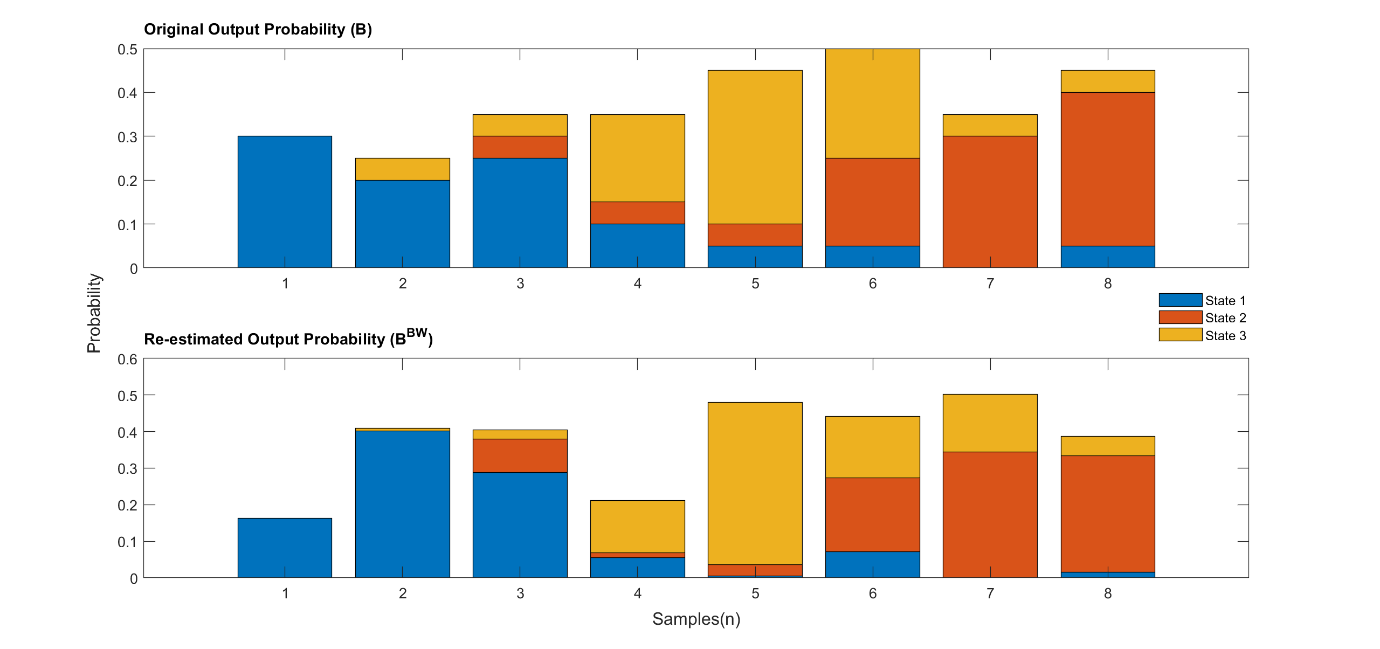


Figure : vs

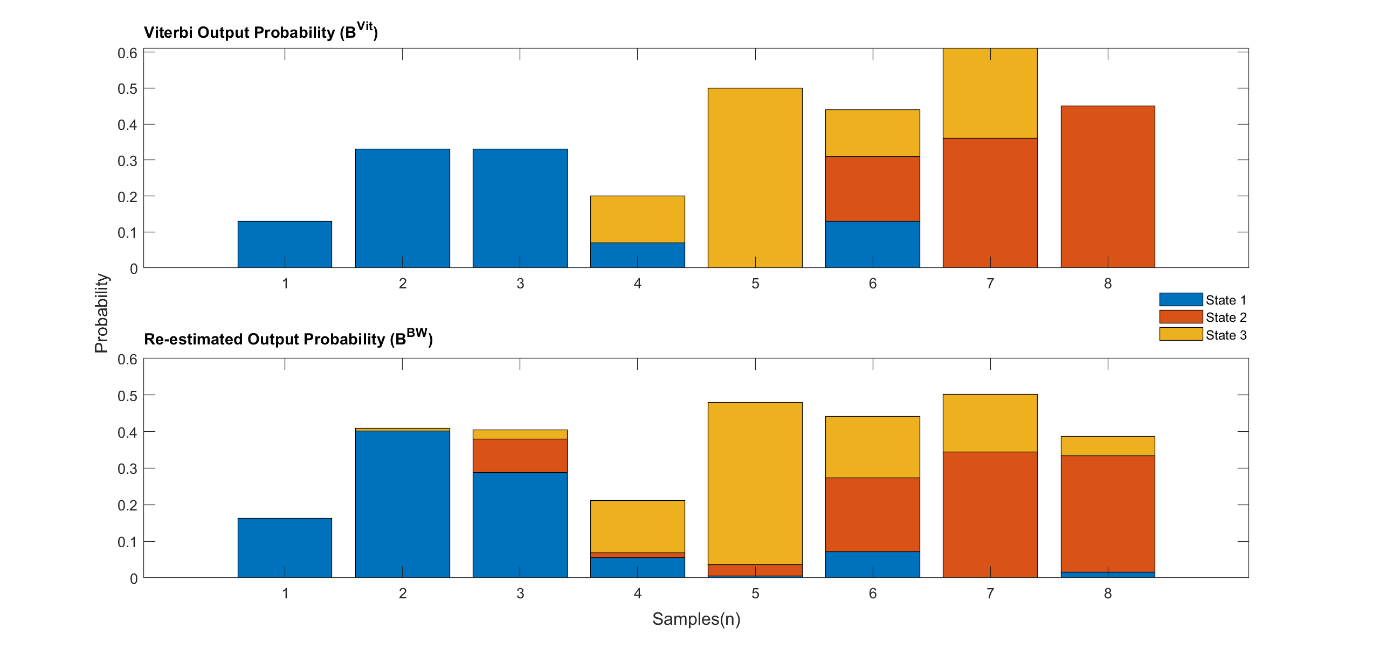


Figure : vs

Comparing the values evaluated using Baum-Welch and Viterbi algorithms, some key observations can be made. Firstly, although the probabilities for state transition and observation differ for both algorithms, they follow an identifiable trend of probability ratios for states, given a set of observations. Thus, any model trained on either of these algorithms would give similar results. Secondly, the Viterbi algorithm finds the single most likely sequence of states and its corresponding probability whereas the Baum-Welch algorithm considers all the possible paths and computes the probability of being at a state at a point in time. Thus, the latter computes exact state occupancies whereas the former only computes an approximation. (King, 2015). This is the reason the Baum-Welch gives more detailed/accurate probability distributions at the cost of computing time and resources as compared to the Viterbi algorithm.

# Conclusion

The various values computed provide an insight into the working, efficiency, and accuracy of Hidden Markov Models for use as a recursive likelihood algorithm for training and recognition. The occupation likelihood values were computed and compared those generated by a Naïve Bayes classifier. It was found that the HMM provides more accurate predictions for state probability for a set of observations as it considers the previous state and probabilities as well. Additionally, the state transition and observation probabilities were re-estimated using the Baum-Welch algorithm and compared to a Viterbi algorithm’s results. It was observed that even though the Baum-Welch provides more accurate results, the probabilities follow a trend and would likely result in similar results for models trained using either of the algorithms.

# Bibliography

* Batista, D. S. (2017, 11 11). *Hidden Markov Model and Naive Bayes relationship*. Retrieved 12 7, 2021, from David S. Batista: https://www.davidsbatista.net/blog/2017/11/11/HHM\_and\_Naive\_Bayes/
* King, S. (2015, 11). *Viterbi vs Forward-Backward (Baum-Welch)*. Retrieved 12 7, 2021, from speech.zone: https://speech.zone/forums/topic/viterbi-vs-backward-forward/
* Lexico.com. (2021). *Markov chain*. Retrieved 12 7, 2021, from Lexico: https://www.lexico.com/en/definition/markov\_chain
* MathWorks. (2021). *Hidden Markov Models (HMM)*. Retrieved 12 7, 2021, from MathWorks Help Center: https://uk.mathworks.com/help/stats/hidden-markov-models-hmm.html#f9074

# Appendix

## Code

### Main.m

1 % Speech Recognition

2 % \_EEEM030 Assignment 2\_

3

4 % Function List:

5

6 % Forward.m

7 % Backward.m

8 % Occupation.m

9 % Transition.m

10 % W.m

11 % -------------------------------------------------------------------------

12

13 % Main.m

14 close all; clear all ;clc;

15

16 A = [ 0 0.40 0.25 0.35 0 ...

17 ; 0 0.80 0.05 0.05 0.1 ...

18 ; 0 0.15 0.70 0.05 0.1 ...

19 ; 0 0.05 0.10 0.75 0.1 ...

20 ; 0 0 0 0 0 ];

21 % -------------------------------------------------------------------------

22

23 B = [ 0.30 0.20 0.25 0.10 0.05 0.05 0.00 0.05 ...

24 ; 0.00 0.00 0.05 0.05 0.05 0.20 0.30 0.35 ...

25 ; 0.00 0.05 0.05 0.20 0.35 0.25 0.05 0.05 ];

26 % -------------------------------------------------------------------------

27

28 O1 = [ 6 4 2 1 3 ];

29 O2 = [ 8 7 6 8 7 3 6 3 ];

30 O3 = [ 5 4 6 7 ];

31 O4 = [ 5 7 5 5 8 8 6 ];

32 O5 = [ 3 2 2 2 2 1 8 7 7 3 ];

33 Observations = {O1 O2 O3 O4 O5};

34 % -------------------------------------------------------------------------

35

36 NB1 = [ 3 3 1 1 1 ];

37 NB2 = [ 2 2 3 2 2 1 3 1 ];

38 NB3 = [ 3 3 3 2 ];

39 NB4 = [ 3 2 3 3 2 2 3 ];

40 NB5 = [ 1 1 1 1 1 1 2 2 2 1 ];

41 Naive\_Bayes = {NB1 NB2 NB3 NB4 NB5};

42 % -------------------------------------------------------------------------

43

44 A\_vit = [ 0 0.40 0.20 0.40 0 ...

45 ; 0 0.73 0.07 0.00 0.20 ...

46 ; 0 0.18 0.73 0.00 0.09 ...

47 ; 0 0.00 0.13 0.75 0.13 ...

48 ; 0 0 0 0 0 ];

49 % -------------------------------------------------------------------------

50

51 B\_vit = [ 0.13 0.33 0.33 0.07 0.00 0.13 0.00 0.00 ...

52 ; 0.00 0.00 0.00 0.00 0.00 0.18 0.36 0.45 ...

53 ; 0.00 0.00 0.00 0.13 0.50 0.13 0.25 0.00 ];

54 % -------------------------------------------------------------------------

55

56 % Evaluate alpha, beta, gamma, xi and estimated A and B values for set of observations

57 for i = 1:(size(Observations,2))

58 O = Observations{i};

59 [alpha,PO1] = Forward(A,B,O);

60 alpha\_matrix(i) = {alpha};

61 [beta,PO2] = Backward(A,B,O);

62 beta\_matrix(i) = {beta};

63 PO1\_matrix(i) = {PO1};

64 PO2\_matrix(i) = {PO2};

65 gamma = Occupation(alpha,beta,PO1);

66 gamma\_matrix(i) = {gamma};

67 [A\_underscore(:,:,i),A\_bar(:,:,i),B\_underscore(:,:,i),B\_bar(:,:,i),xi,entry,exit] = Transition(68 xi\_matrix(i) = {xi};

69 entry\_matrix(i) = {entry};

70 exit\_matrix(i) = {exit};

71 end

72

73 % Evaluate numerator and denominator for estimated A & B

74 A\_numerator = zeros(size(A));

75 A\_denominator = zeros(size(A\_bar(:,:,1)));

76 B\_numerator = zeros(size(B));

77 B\_denominator = zeros(size(B\_bar(:,:,1)));

78

79 for i = 1:(size(Observations,2))

80 A\_numerator = A\_underscore(:,:,i) + A\_numerator;

81 A\_denominator = A\_bar(:,:,i) + A\_denominator;

82 B\_numerator = B\_underscore(:,:,i) + B\_numerator;

83 B\_denominator = B\_bar(:,:,i) + B\_denominator;

84 end

85

86 % Evaluate A\_BW and B\_BW

87 A\_BW = A\_numerator./A\_denominator;

88 A\_BW(isnan(A\_BW)) = 0;

89 B\_BW = B\_numerator./B\_denominator;

90 B\_BW(isnan(B\_BW)) = 0;

91 % -------------------------------------------------------------------------

92

93 %Plots

94 %{

95 figure(1)

96 hold on

97 [M,I] = max(gamma\_matrix{1});

98 y = [I;NB1];

99 x = 1:1:(size(O1,2));

100 b = bar(x,y);

101 b(1).FaceColor = [0 0.4470 0.7410];

102 b(2).FaceColor = [0.6350 0.0780 0.1840];

103 t = title('Occupation Likelihood vs Naive Bayes (O1)');

104 ax = gca;

105 ax.TitleHorizontalAlignment = 'left';

106 xlabel('Observation(frame)')

107 ylabel('State(n)')

108 lgd = legend('Occupation Likelihood','Naive Bayes');

109 legend('boxoff');

110 hold off

111

112 figure(2)

113 hold on

114 t = tiledlayout(2,1);

115 xlabel(t,'Samples(n)')

116 ylabel(t,'Probability')

117

118 ax1 = nexttile;

119 bar(ax1,B\_vit','stacked')

120 t1 = title(ax1,'Viterbi Output Probability (B^{Vit})');

121 ax1.TitleHorizontalAlignment = 'left';

122

123

124 ax2 = nexttile;

125 bar(ax2,B\_BW','stacked')

126 t2 = title(ax2,'Re-estimated Output Probability (B^{BW})');

127 ax2.TitleHorizontalAlignment = 'left';

128 lgd = legend('State 1','State 2','State 3');

129 legend('boxoff');

130 hold off

131 %}

### Forward.m

1 % Forward.m

2 function [alpha,PO] = Forward(A,B,O)

3

4 T = size(O,2); % No. of observations

5 N = size(B,1); % No. of states

6 sum = 0;

7

8 alpha = zeros(N,T);

9 for t = 1:T

10 if t == 1

11 for i = 1:N

12 alpha(i,t) = A(t,(i+1))\*B(i,O(t));

13 end

14 else

15 for j = 1:N

16 for i = 1:N

17 sum = sum + alpha(i,(t-1)) \* A((i+1),(j+1));

18 end

19 alpha(j,t) = sum \* B(j,O(t));

20 sum = 0;

21 end

22 end

23 end

24

25 PO = 0;

26 for i = 1:N

27 PO = PO + alpha(i,T) \* A((i+1),size(A,2));

28 end

29

30 end

### Backward.m

1 % Backward.m

2 function [beta,PO] = Backward(A,B,O)

3

4 T = size(O,2);

5 N = size(B,1);

6 sum = 0;

7

8 beta = zeros(N,T);

9 for t = T: -1: 1

10 if t == T

11 for i = 1:N

12 beta(i,t) = A((i+1),size(A,2));

13 end

14 else

15 for i = 1:N

16 for j = 1:N

17 sum = sum + A((i+1),(j+1)) \* B(j,O(t+1)) \* beta(j,(t+1));

18 end

19 beta(i,t) = sum;

20 sum = 0;

21 end

22 end

23 end

24

25 PO = 0;

26 for i = 1:N

27 PO = PO + A(1,(i+1)) \* B(i,O(1)) \* beta(i,1);

28 end

29

30 end

### Occupation.m

1 % Occupation.m

2 function gamma = Occupation(alpha,beta,PO1)

3

4 gamma = alpha.\*beta;

5 gamma = gamma / PO1;

6

7 end

### Transition.m

1 % Transition.m

2 function [A\_underscore,A\_bar,B\_underscore,B\_bar,xi,entry,exit] = Transition(alpha,beta,PO1,3

4 xi = zeros(size(A));

5 A\_underscore = zeros(size(A));

6 T = size(O,2);

7 N = size(B,1);

8 %--------------------------------------------------------------------------

9 % Calculate Transition likelihood and a\_underscore and a\_bar

10

11 for t = 2:T

12 for i = 1:N

13 for j = 1:N

14 xi((i+1),(j+1),(t-1)) = (alpha(i,(t-1)) \* beta(j,t) \* A((i+1),(j+1)) \* B(j,O(t))) 15 end

16 end

17 A\_underscore = xi(:,:,(t-1)) + A\_underscore;

18 end

19

20 for i = 1:N

21 A\_underscore(1,(i+1)) = gamma(i,1);

22 entry(1,(i+1)) = gamma(i,1);

23 end

24 for i = 1:N

25 A\_underscore((i+1),size(A,2)) = gamma(i,T);

26 exit((i+1),1) = gamma(i,T);

27 end

28

29 A\_bar = sum(A\_underscore,2);

30 %--------------------------------------------------------------------------

31 % Calculate b\_underscore and b\_bar

32

33 denom = zeros(N,1);

34 for j = 1:N

35 for t = 1:T

36 denom(j,1) = denom(j,1) + gamma(j,t);

37 end

38 end

39

40 numer = zeros(size(B));

41 for k = 1:size(B,2)

42 for j = 1:N

43 for t = 1:T

44 numer(j,k) = (gamma(j,t) \* W(k,t,O)) + numer(j,k);

45 end

46 end

47 end

48

49 B\_underscore = numer;

50 B\_bar = denom;

51

52 end

### W.m

1 % W.m

2 % Function to check if current observation matches k

3 function out = W(k,t,O)

4

5 if k == O(t)

6 out = 1;

7 else

8 out = 0;

9 end

10 end

## Results

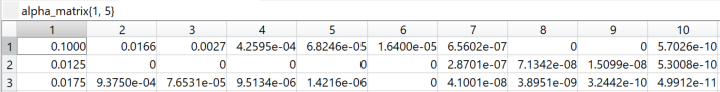
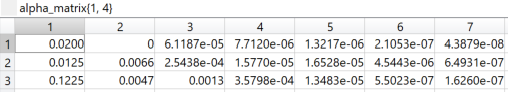
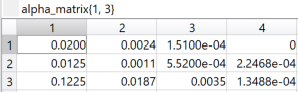
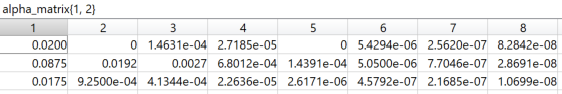


Figure : Forward Likelihood (O2-O5)

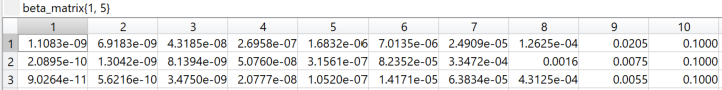
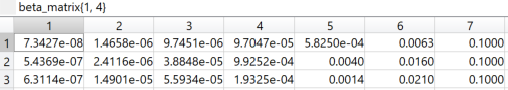
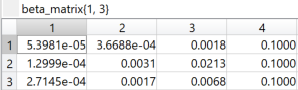
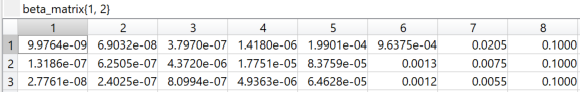


Figure : Backward Likelihood (O2-O5)

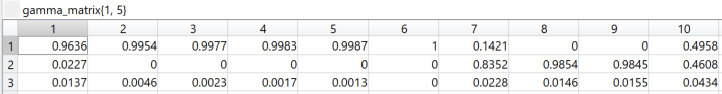
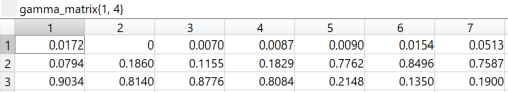
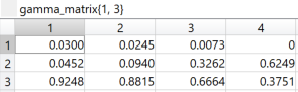
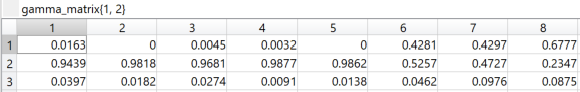


Figure : Occupation Likelihood (O2-O5)



Figure : P(O1-O5|) - Forward & Backward

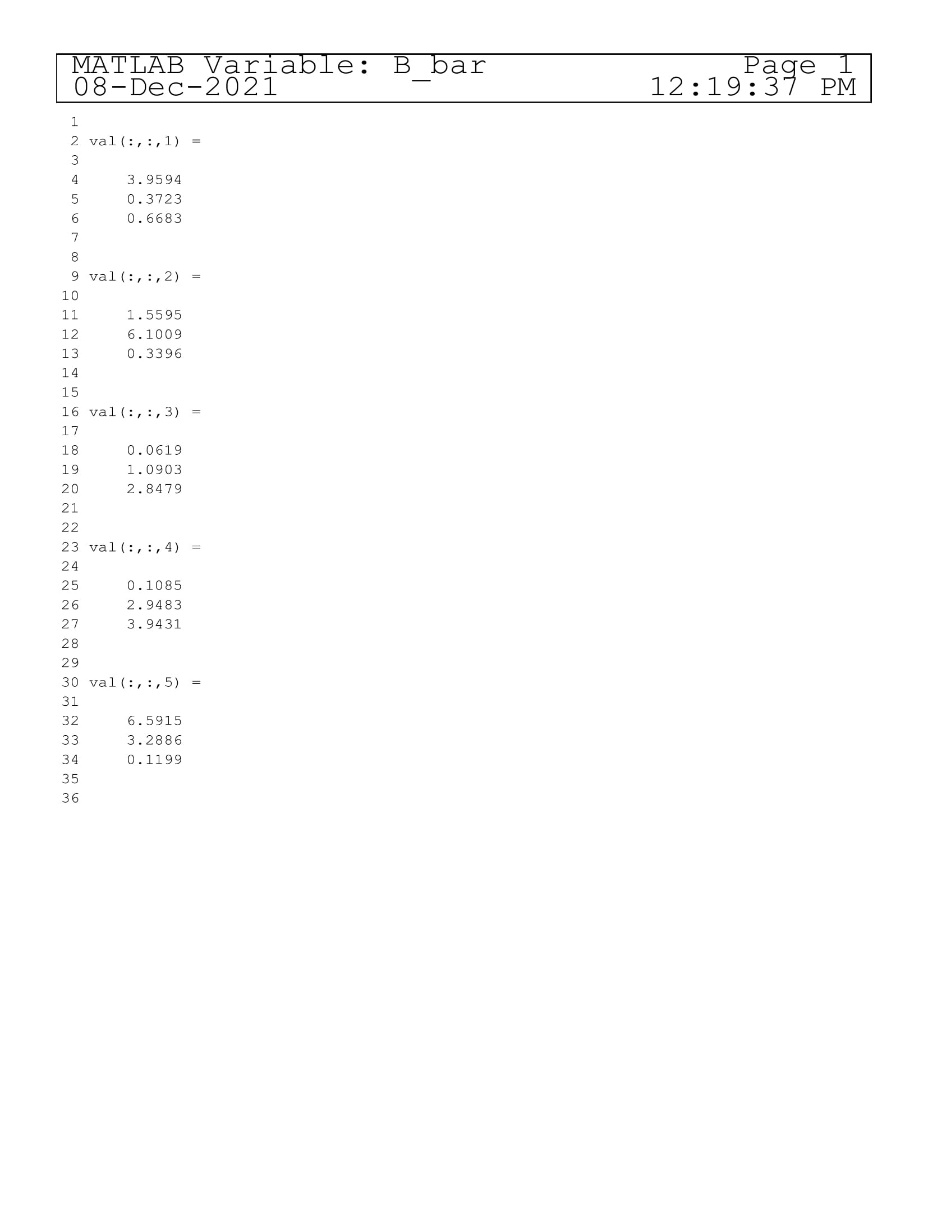


Figure : Accumulated Occupation Likelihood (O1-O5)

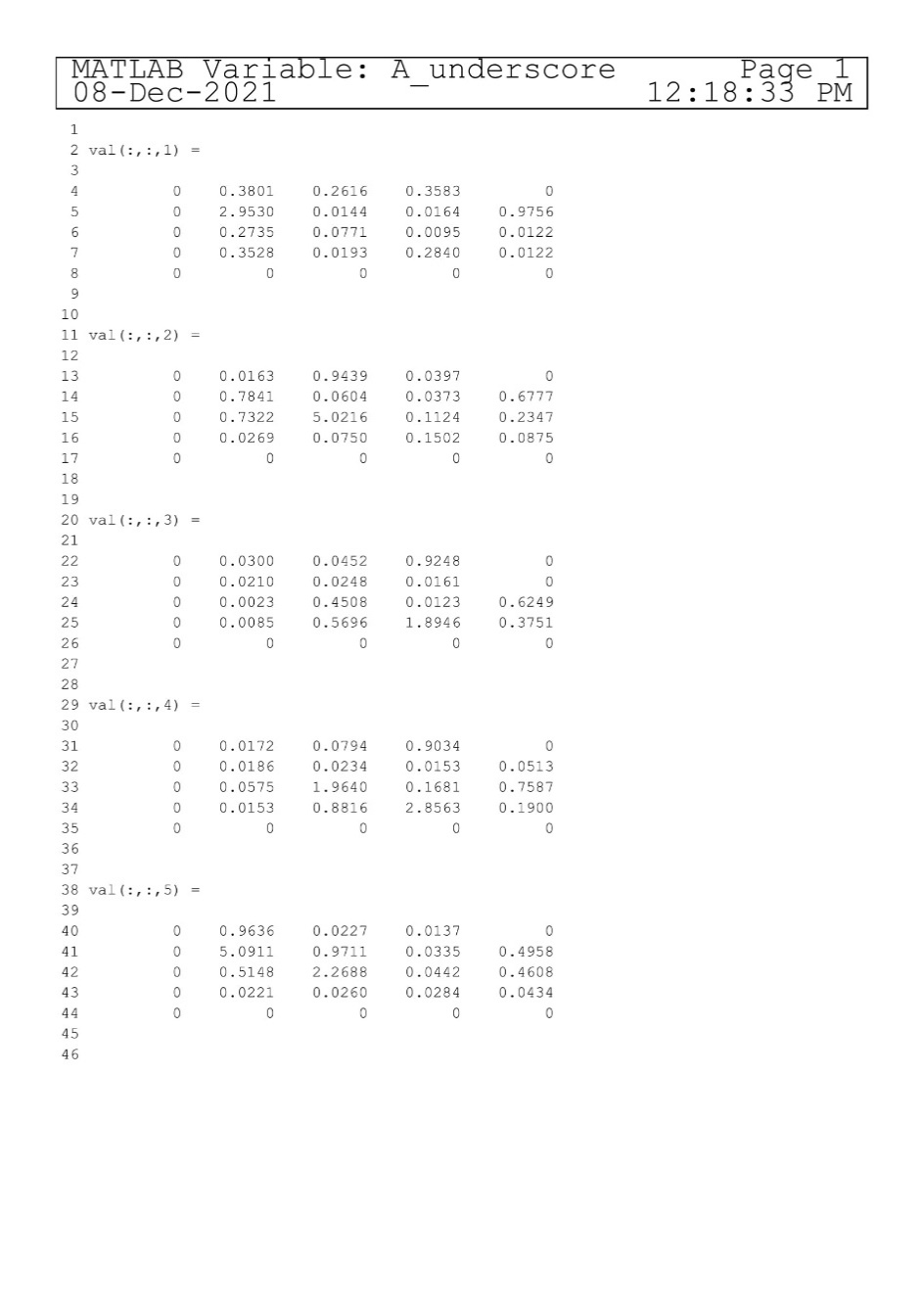


Figure : Accumulated Translation Likelihood (O1-O5)

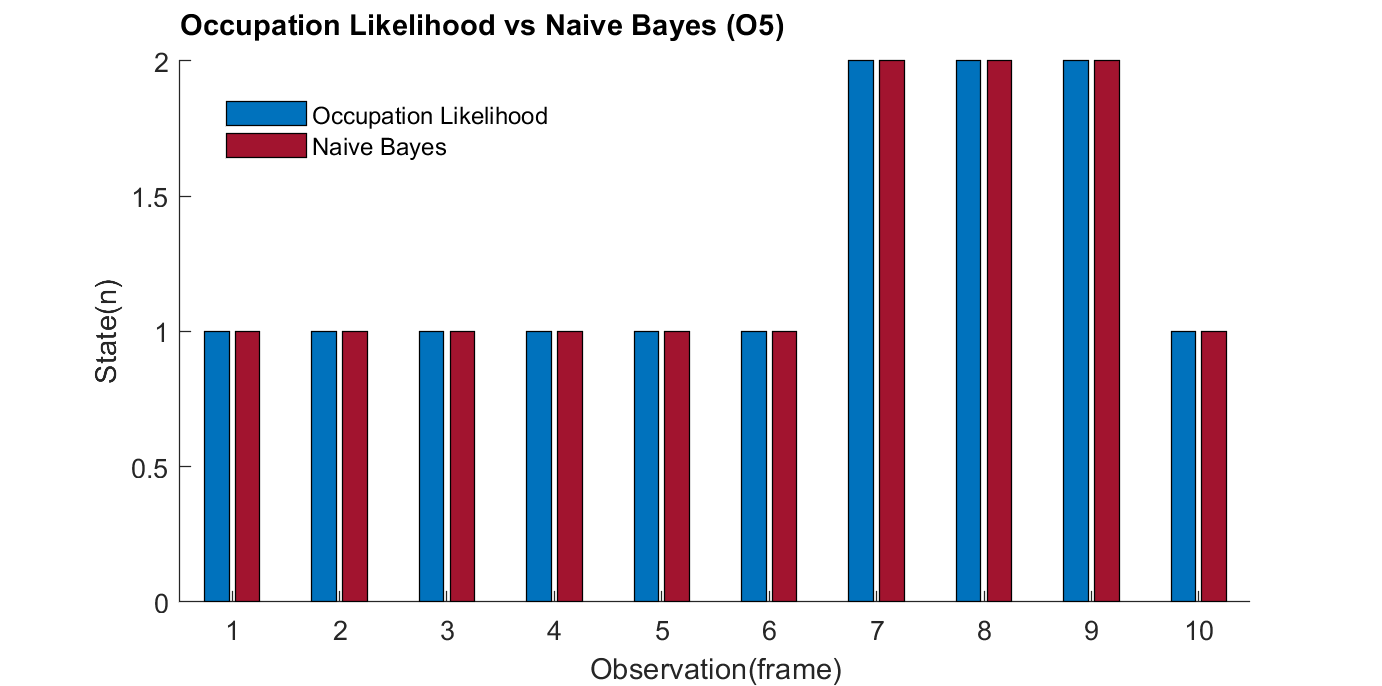
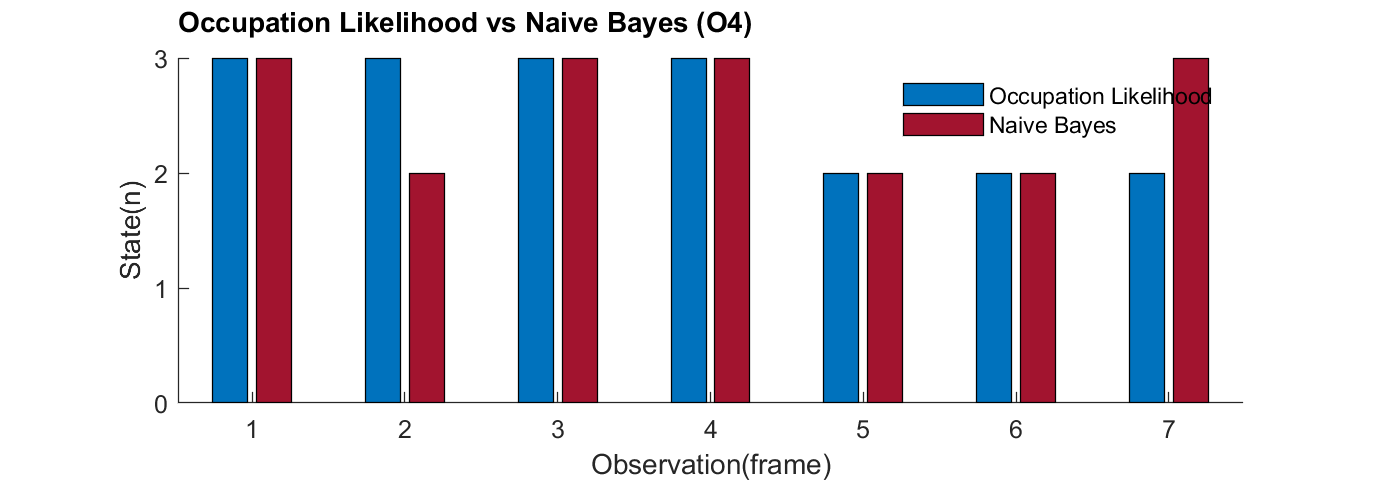
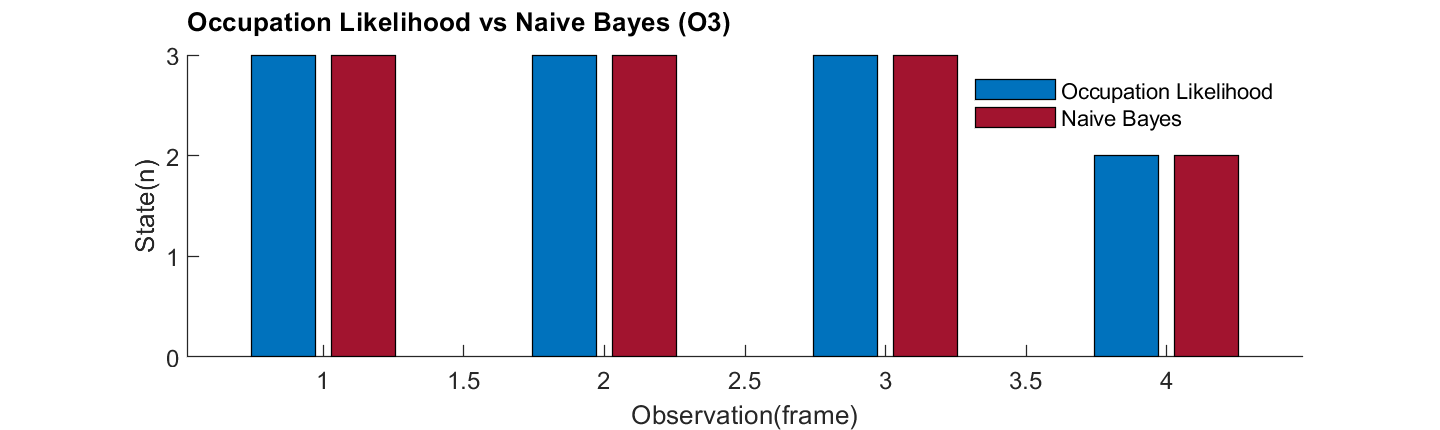
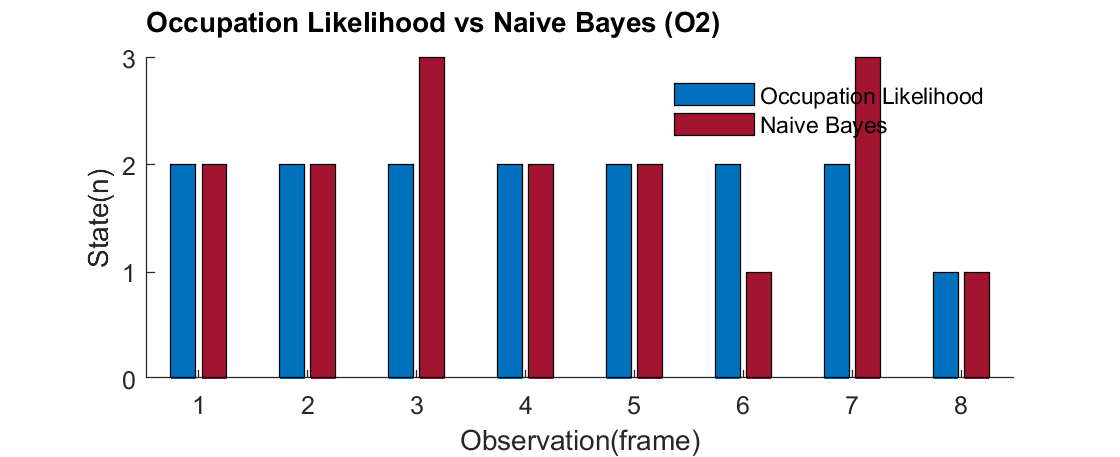


Figure : Occupation Likelihood vs Naive Bayes (O2-O5)

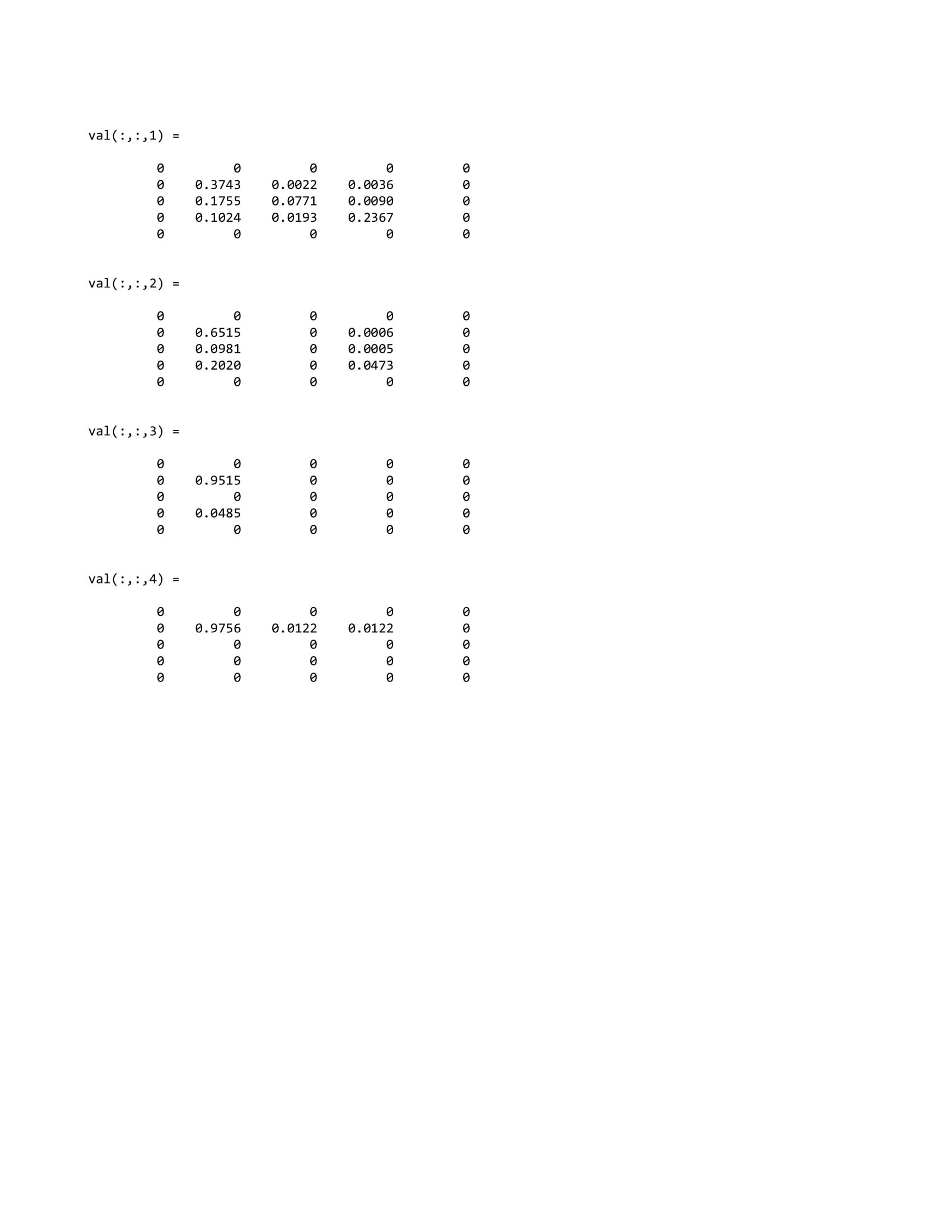


Figure : Transition Likelihood (O1)



Figure : Entry Row Matrix (O1-O5)



Figure : Exit Column Matrix (O1-O5)

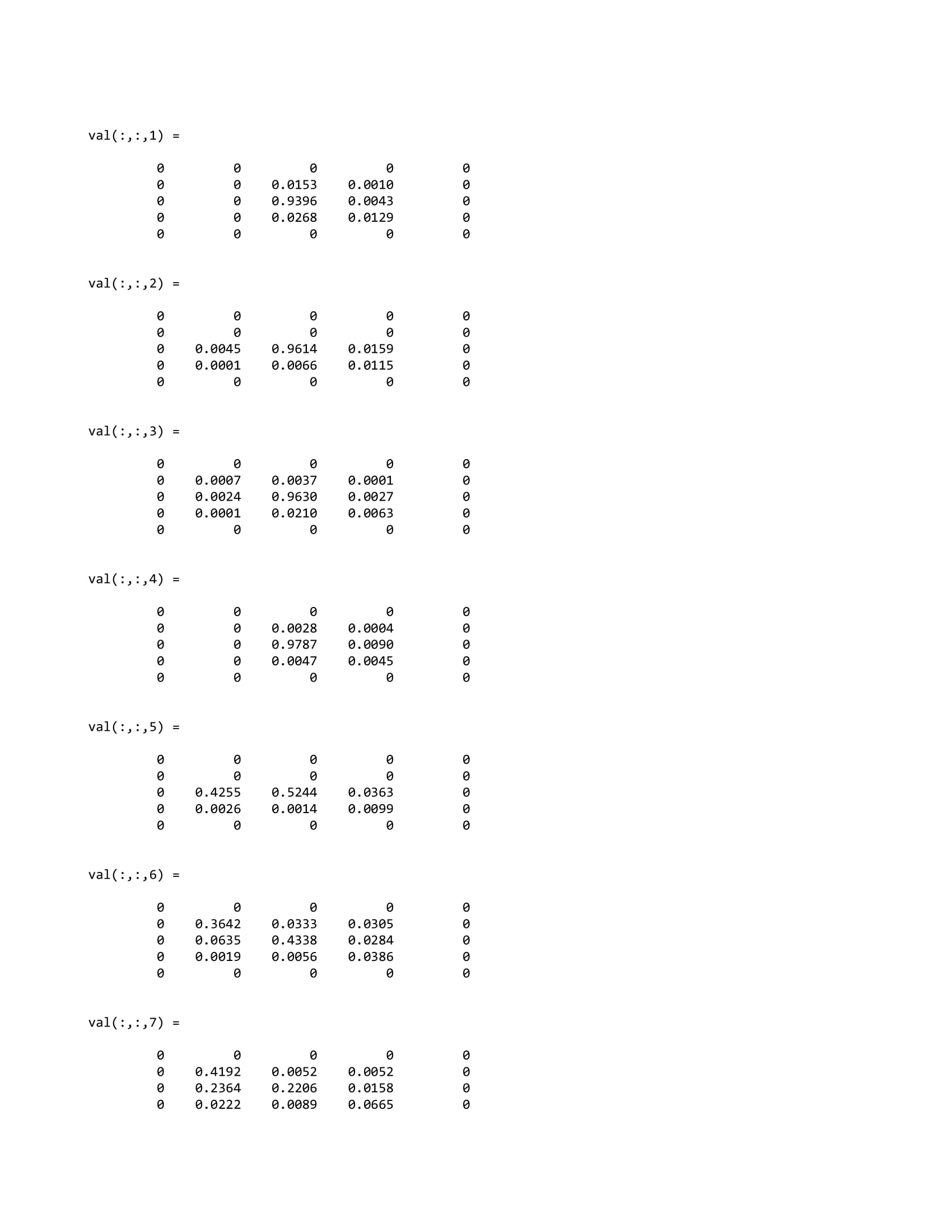


Figure : Transition Likelihood (O2)

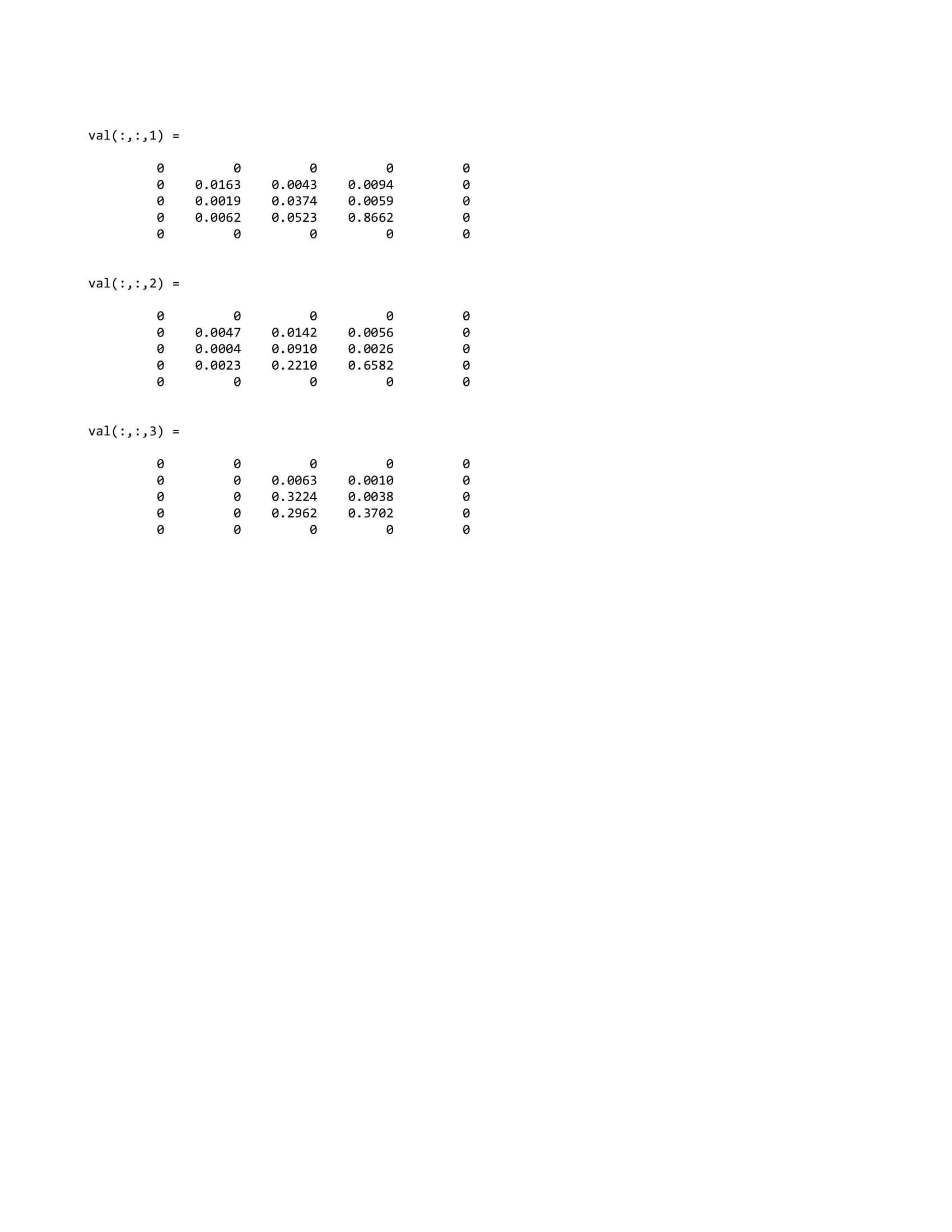


Figure : Transition Likelihood (O3)

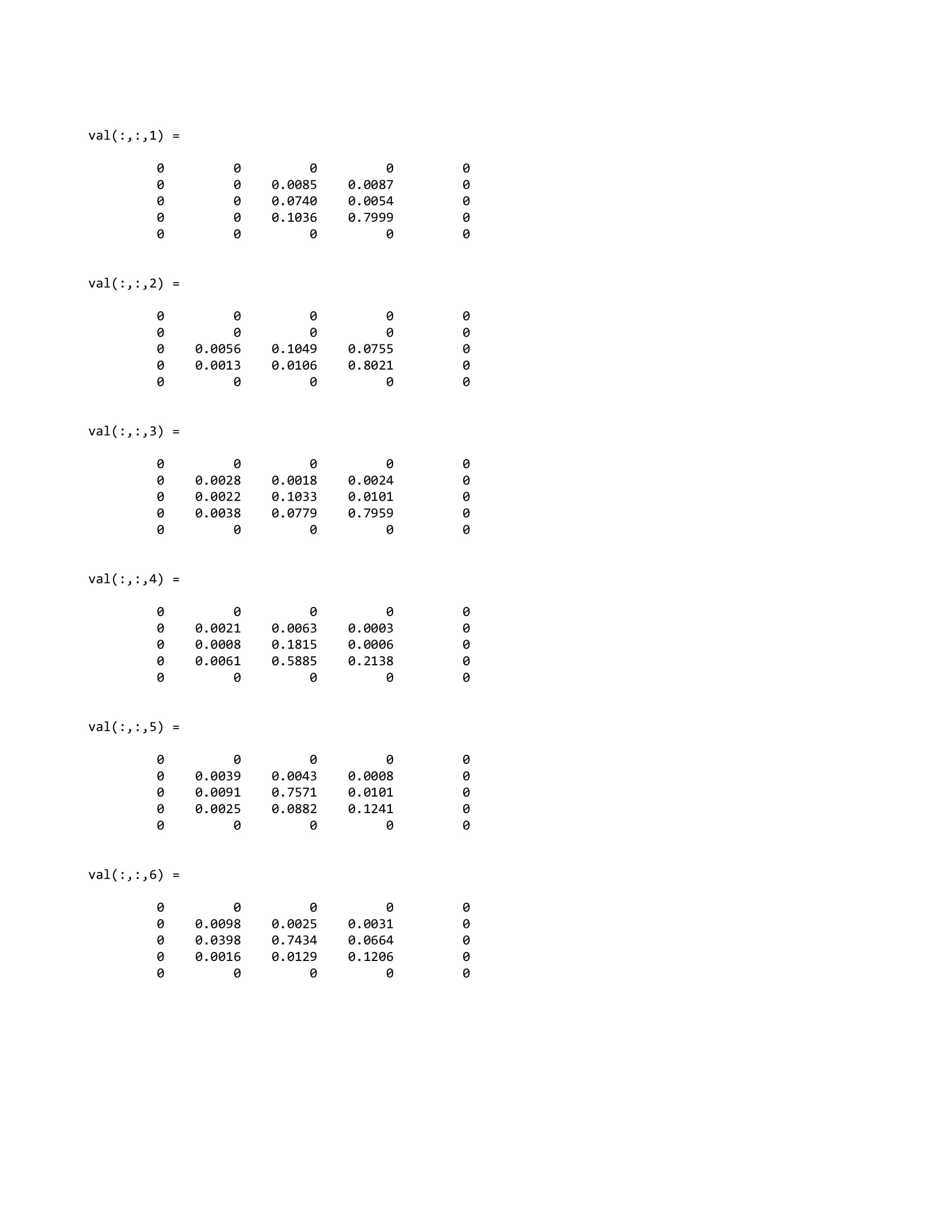


Figure : Transition Likelihood (O4)

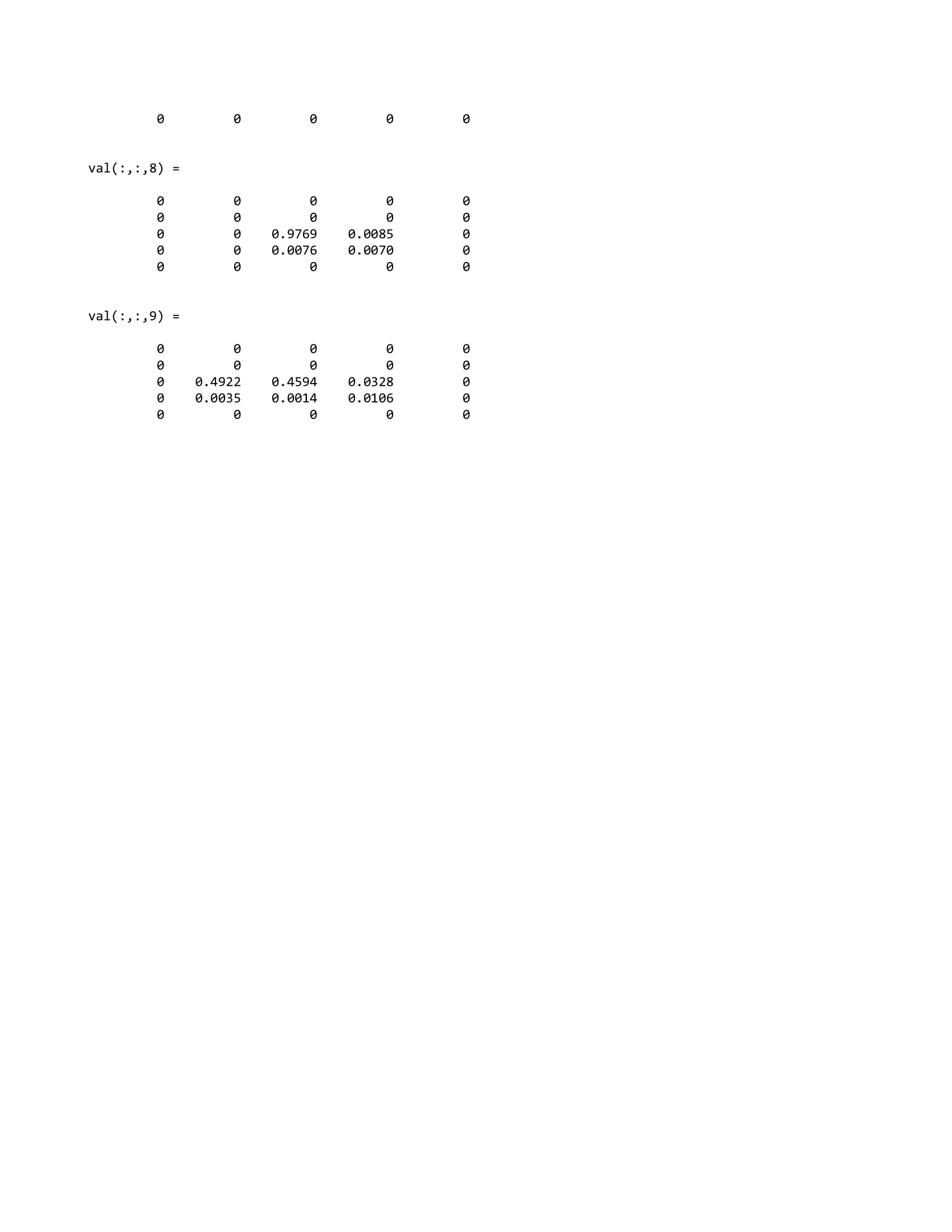
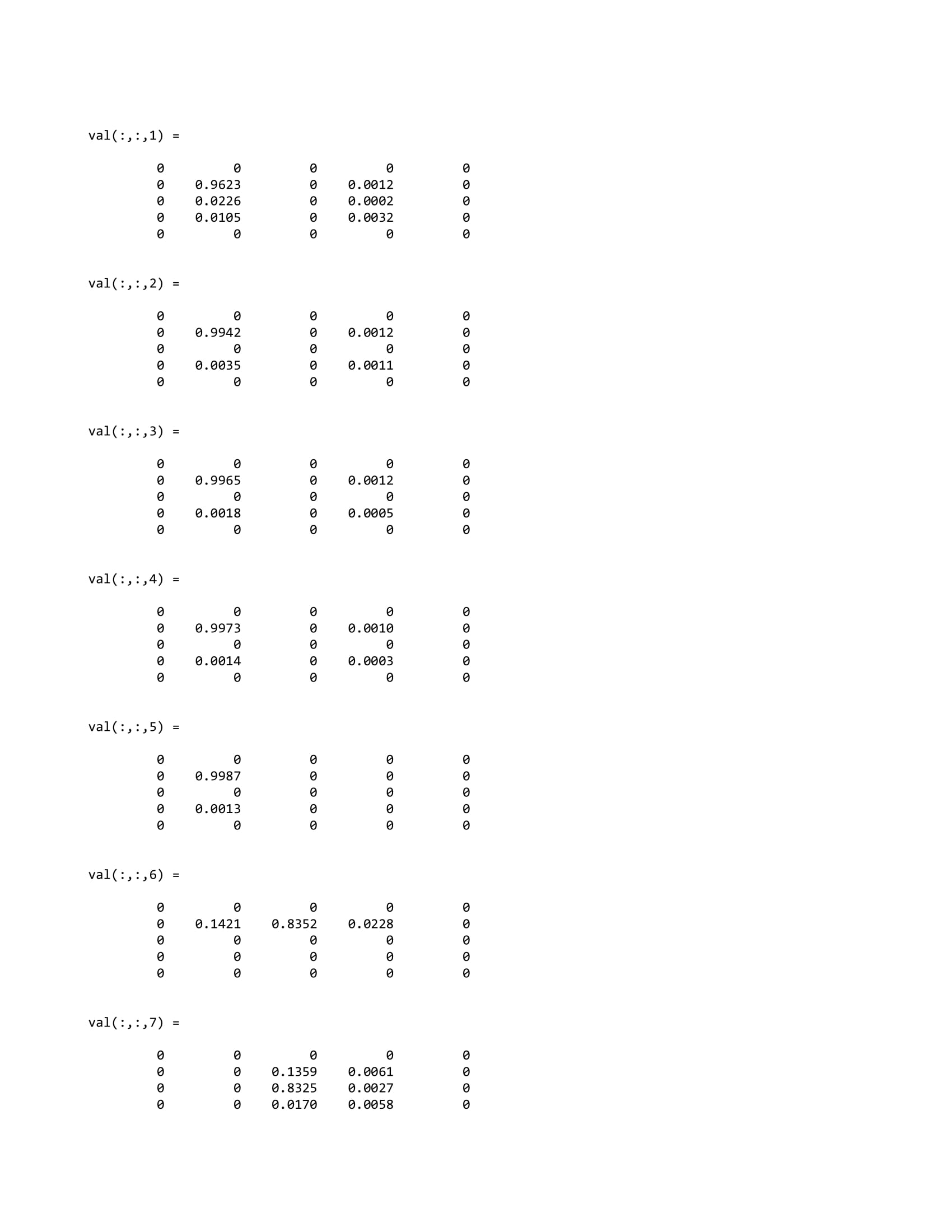


Figure : Transition Likelihood (O5)