5. Technique Evaluation - TPSOM

5.1 Introduction

This Chapter deals with the testing and evaluation of the TPSOM described in Chapter 4. The aim of presenting these results is to show the operation of the technique and to show the effects of the parameters on the operation of the algorithms. In the tests that follow, four types of data set were used.

1. An artificial data set was generated to allow comparison between the different techniques. This data set was designed to show the capabilities of the different networks. This set was generated with 7 classes, the overall structure of the data remaining the same.
2. A real-world data set was taken from Thales Research and Technology (UK) and consisted of radar pulse data.
3. A Static Markovian data set was generated from a homogeneous State Transition Matrix (described in Section 5.4.4).
4. A Non-stationary Markovian data set was generated from a discontinuous non-stationary Transition Matrix (described in Section 5.4.5).
5. A real world data set taken from Charles Dickens 'Great Expectations' as a good example of an English text.

The outputs from the TPSOM are:

- The Class Output, this is the arbitrary number assigned to each class within the PSOM. This arbitrary class number can then be matched with known classes during training if required. From this, the number of classes (or states) held by the network can be calculated
- The Recognition Error which is the Euclidean Distance between the input and the focus (best match).
- A class association matrix that compares the labels of the input class to that of the output class.

- A state transition matrix that shows a matrix representation of the network topology.

All of these outputs are dynamic and are studied over time and were created using Raih, as described in Chapter 2. The tests are numbered to give a logical flow although they may not have been performed in the numbered order.

5.2 Artificial Data Sets

Two artificial data sets (Type 1) were created to test the following features of the TPSOM. Each numbered item represents the class number that will be used to refer to it. A stopping pattern, for example, will be referred to as Class 2.

1. **Regular Patterns.** If a pattern is regularly occurring and persistent, then each network should continue to recognise the pattern.
2. **Stopping Pattern.** A pattern that disappears from the data set. Each technique should eventually forget the pattern that has disappeared.
3. **Infrequent Patterns.** A pattern is infrequent if it appears in a regular way but with a large time interval between presentations to the network.
4. **Irregular Patterns.** An irregular pattern appears at random intervals. This type of pattern is used to investigate whether the network's response is not dependent on the frequency of occurrence of the patterns.
5. **Starting Pattern.** A pattern that appears late in the operation. Each technique should be able to accept the new pattern.
6. **Second Regular Pattern.** Another persistent pattern that is added in order to avoid any neurons self-linking.
7. **Third Regular Pattern.** As for class 6.

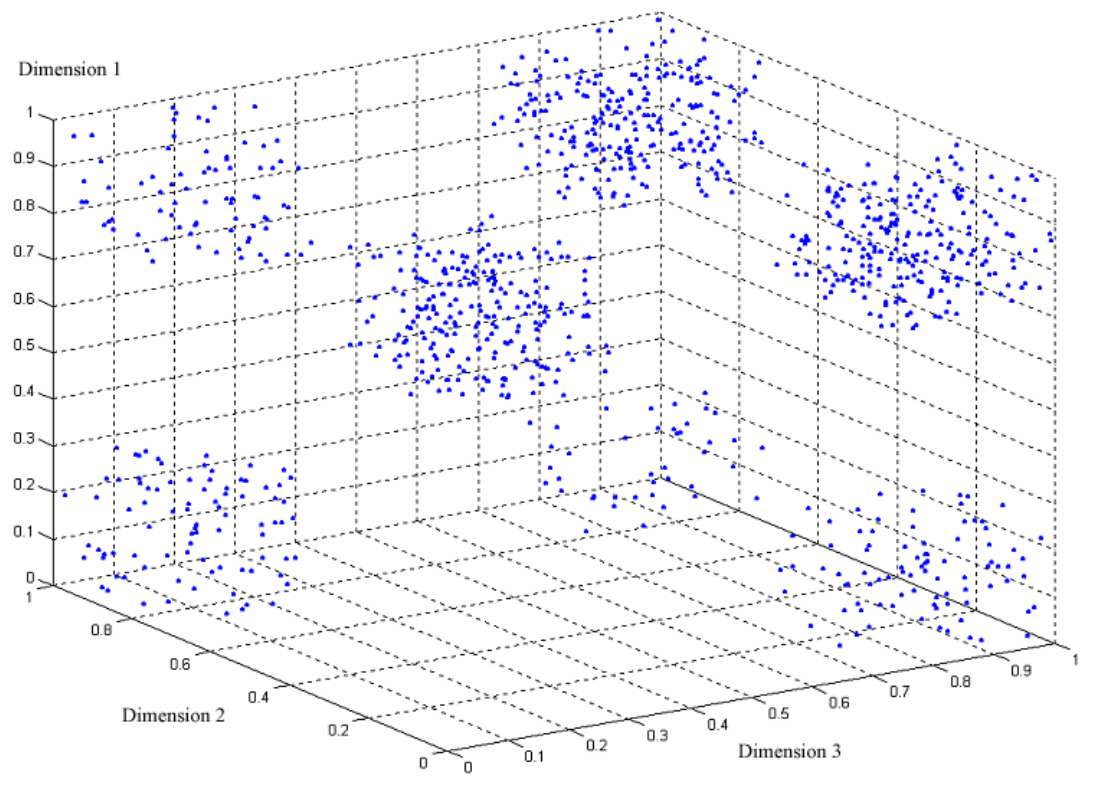

Figure 5.1 The 7 Class Artificial Multivariate Dataset with 20% noise for use with the TPSOM.

It was soon found that for the TPSOM a slightly different data set was required (see TPSOM Section for justification). This second artificial data set has 7 distinct patterns that are corrupted by 20% noise (Figure 5.1). One possible detrimental effect of adding the new patterns is that the density of the random pattern is much less than before.

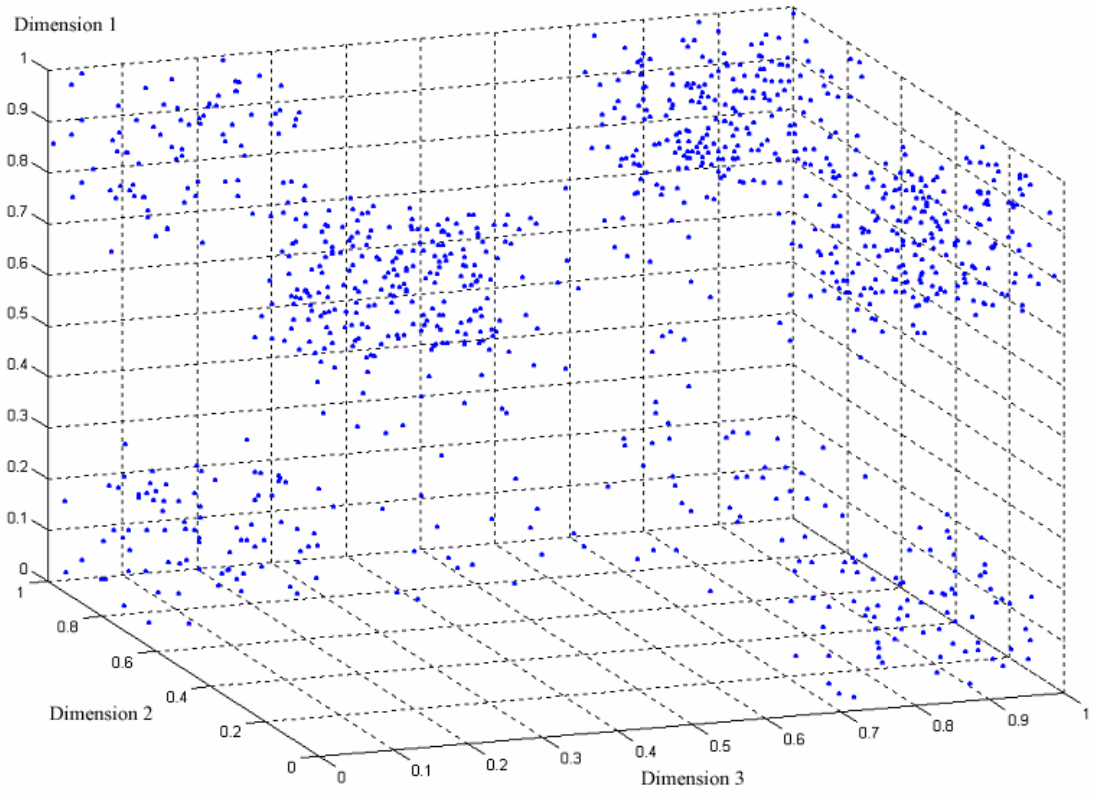

Figure 5.2 The 7 Class Artificial Multivariate Data Set with 20% noise and 100 pure noise patterns. For use with the TPSOM.

To test the TPSOM with a data set corrupted by noise, the 7 class multivariate data set had 100 pure noise patterns added to it (Figure 5.2). Although the binary data set was also corrupted by noise (to give the classes width in the input space), all instances belonged to an input class. The addition of this new set of noise patterns gives patterns that do not belong to any input class. Due to the generation process, some of these new noise patterns fall within the same space as the patterns of an input class, this is unavoidable and the networks will not be able to distinguish between the two types of patterns in this case.

5.3 Great Expectations Data Set

This dataset was used to explore the TPSOM's ability to represent a text in a network diagrammatic form (a state transition matrix). In this case, each node would represent a single letter (a, b, c … z) and the links between nodes would represent the probability that one letter follows another. The problem as a whole is a very large one requiring 26 different nodes. As the size of the network increases and the

probability transitions between states are very similar, then little information can be gained from the network. A quick test confirmed this. Instead, the pattern was reconfigured as the extraction of vowel pairs from the document. As there are only 5 vowels in English, there will only be 5 nodes in the network.

Charles Dickens' Great Expectations was used as it was thought to have reasonably modern test and yet a wide breadth of vocabulary. Only the first 9 chapters were used as it was deemed unnecessary to use the entire book to demonstrate this feature of the network. The first 9 chapters yielded 4663 pairs of vowels.

The novel was converted into vowel pairs by finding each pair of vowels and separating them with a space. For example, the first paragraph is (vowel pairs are in bold):

*"My father's family name b**ei**ng Pirrip, and my Christ**ia**n name Philip, my infant tong**ue** c**ou**ld make of both names nothing longer or more explicit than Pip. So, I called myself Pip, and came to be called Pip."*

The vowel pairs for this paragraph are:

"ei ia ue ou"

It was possible to calculate the state transition matrix for the first 9 Chapters by taking the 25 possible combinations of vowels and calculating the percentage that given a vowel, another would follow. This is shown in Table 5.1. Some properties of the English language can be found in this table: for example, the vowel pair 'AA' is very rare in English, as is 'UU'. The pair 'II' can be found in plurals but is also rare. We can see that in the first 9 chapters of Great Expectations, there are no instances of these three vowel pairs.

Table 5.1 The correct probability that one vowel follows another. Read the rows first and then the columns. For example, A followed by E comes up as 0.002.

	A	**E**	**I**	**O**	**U**

A	0	0.002	0.894	0.002	0.102
E	0.536	0.342	0.101	0.017	0.003
I	0.083	0.412	0	0.495	0.010
O	0.040	0.129	0.041	0.227	0.564
U	0.263	0.333	0.376	0.027	0

5.4 Temporal Plastic Self Organising Map

The Temporal Plastic Self Organising Map's departure from typical network structure led to a departure in the manner in which testing was performed. For these tests, the visual output and the State Transition Matrix (STM) are of interest as well as the error between input and focus. The following tests were performed:

- Test 1: Operation of TPSOM on Artificial Data Set.
- Test 2: Operation of TPSOM on Noisy Artificial Data Set.
- Test 3: Operation of TPSOM on Radar Data.
- Test 4: Operation of TPSOM on Static Markov Chain.
- Test 5: Operation of TPSOM on Dynamic Markov Chain.
- Test 6: Effect of Node Building Threshold, a_n, on outputs.
- Test 7: Effect of Learning Rate, b_r, on outputs.
- Test 8: Operation of TPSOM on plain English vowel pairs.

The artificial multivariate data set used until now is not entirely suitable for this implementation of the TPSOM. The TPSOM creates directed graphs. It is conceivable that a class could be presented twice in a row or two classes could alternate between them. This can be shown graphically in Figure 5.3. Although easily drawn here, a software implementation that resulted in clear graphs was not possible. This is not a limitation of the TPSOM but a limitation of manner in which the software was written.

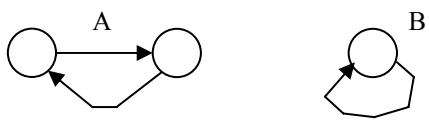

Figure 5.3 TPSOM linking graphical problem illustration.

To overcome this problem, two more patterns were added to the artificial multivariate data set. This addition means that there are never less than three patterns in the graph. The new multivariate data set, known as 7 pattern artificial multivariate data set is shown in Figure 5.1. To facilitate understanding of the state transition matrices, the values that are 0 (where there is no transition between the states) are set to light grey. This problem was later solved for performing tests on the plain English vowel pairs.

The spatial clustering of the TPSOM (such as the locating of focus) is identical to the PSOM. This benefits experimentation as it gives a starting point for some of the network parameters.

5.4.1 Test 1: 7-Pattern Artificial Multivariate Data

The aim of this experiment is to show the TPSOM operating on the 7 pattern artificial multivariate data set (Figure 5.1). The parameters used for this test are shown in Table 5.2.

Table 5.2 TPSOM Test 1 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.29
Link Removal Threshold (a_r)	0.1
Learning Rate (b_r)	0.025

Table 5.3 TPSOM Test 1: State Transition Matrix after 100 iterations.

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	0.00	1.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	1.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	1.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	1.00	0.00	0.00
4	0.23	0.27	0.00	0.00	0.00	0.24	0.25
5	0.00	1.00	0.00	0.00	0.00	0.00	0.00
6	0.00	1.00	0.00	0.00	0.00	0.00	0.00

Table 5.4 TPSOM Test 1: State Transition Matrix after 700 iterations.

	1	**2**	**3**	**5**	**6**
1	0.00	1.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00
3	0.51	0.00	0.00	0.19	0.29
5	1.00	0.00	0.00	0.00	0.00
6	0.50	0.00	0.00	0.49	0.00

Table 5.5 TPSOM Test 1: State Transition Matrix after 850 iterations.

	7	**1**	**2**	**3**	**5**	**6**
7	0.0	1.0	0.0	0.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0	0.0
2	0.0	0.0	0.0	1.0	0.0	0.0
3	0.25	0.37	0.0	0.0	0.25	0.13
5	0.53	0.46	0.0	0.0	0.0	0.0
6	0.34	0.3	0.0	0.0	0.35	0.0

The initial parameter values for the test were taken from the PSOM Test 1. This is possible because the TPSOM and PSOM algorithms and data sets are similar enough to produce acceptable outputs. These values provided a good starting point for a short tuning exercise. It was found that the parameters used in the PSOM Test 1 were adequate to produce an output in this test. A short empirical study was performed to find b_r. This study was achieved by comparing the class labels output of the TPSOM to the actual class labels of the data set. If the outputs are similar, then the network is assumed to be working correctly.

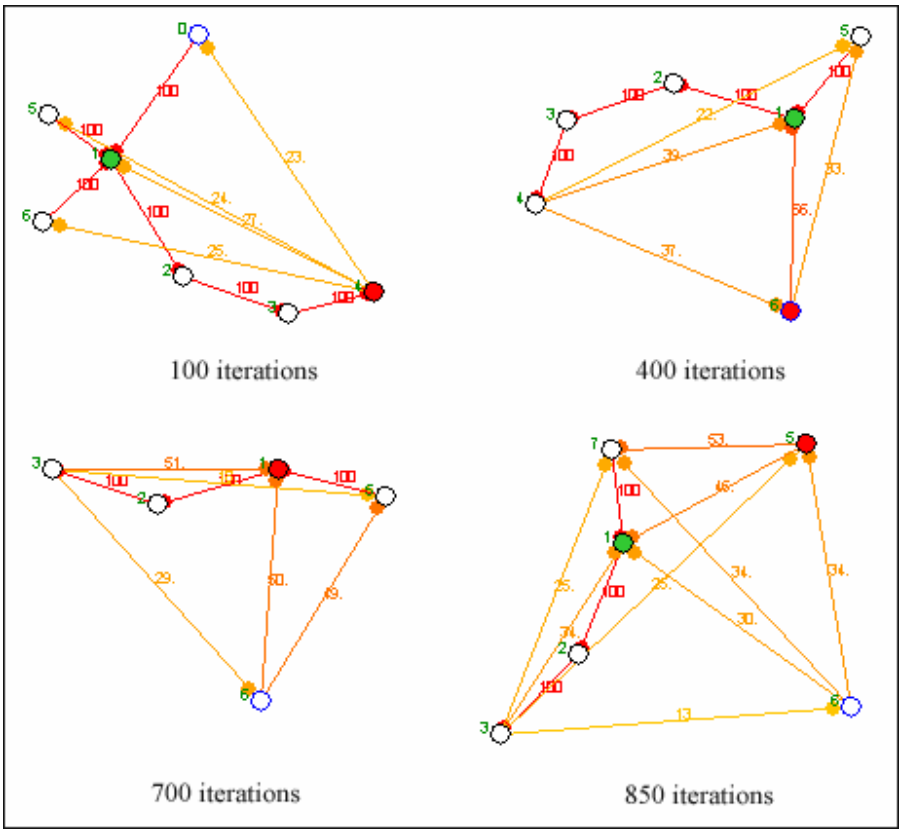

Figure 5.3 TPSOM Test 1: Network states at key points during its operation. Due to the small print of the font, the numeral '0' appears as a rectangle.

Figure 5.3 shows the progression of the network structure at important points during the 1000 iterations. These snap-shots are taken at 100,400, 700 and 850 iterations. To increase implementation speed, circles were drawn instead of arrowheads to represent direction. Link shade denotes the probability of this route being between the two neurons it connects: the higher the probability the shorter and darker the link. This allowed the structure to be quickly analysed (particularly useful when the network evolves in real time on screen). The numbers by the links are the percentage probabilities and the number in the top left hand corner of each neuron is the class number associated with that neuron. The neuron labelled '0' represents the initial neuron.

The 100 iteration picture shows the network after all classes but class 5 have been shown to the network. The structure at this point is quite simple. The network regularly follows the pattern 1→2→3→4 and then from class 4, there are four choices, either back to class 1, to class 0, class 5 or class 6. The most probable is

from 4→1 but marginally with a probability of 0.27. Each state class 0,5 or 6 return to class 1.

Table 5.3 shows the state transition matrix after 100 iterations. Comparing the diagram and Table 5.3, one can see that the probabilities match (the table probabilities add to 1.0, the diagrams are percentages to improve readability). All the links from one neuron (one row in the state transition matrix) sum to 1.0. A small error occurs when the probabilities values are truncated for easy viewing. Therefore, the output values often sum to 0.99 or 99% rather than 1.0 and 100%. This is acceptable as further analysis (not included for the sake of brevity) shows the pre-truncation sum to be 1.0.

After 400 iterations, the initial neuron (neuron 0) has been removed. 400 iterations is the point where class 2 is no longer shown to the network (the class disappears from the data set). At this time, the initial neuron (class 0) has been removed. The strong linkage between patterns 1,2,3 and 4 is still obvious; however, this is unlikely to remain the case as class 2 (represented by the network as class 4) is no longer included in the input set.

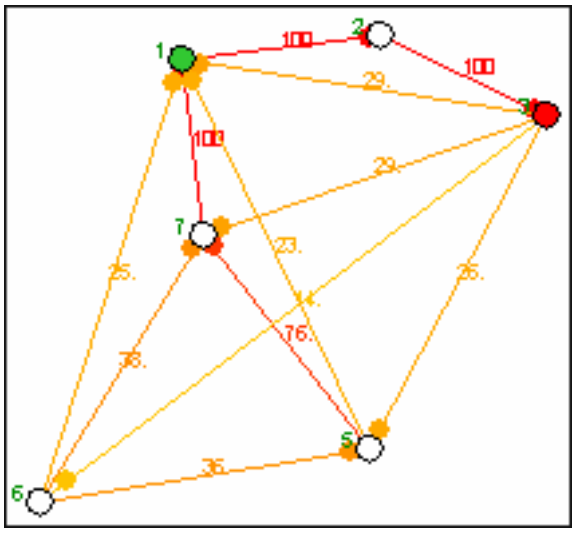

Figure 5.4 TPSOM Test 1: The network state after 1000 iterations.

After 700 iterations, class 2 (shown by the network as class 4) has been removed. This is the smallest network size required to represent this number of classes. At 800 iterations, class 5 is shown to the network for the first time. The plot at 850 iterations

shows this new class assigned class 7 by the TPSOM. Between 850 iterations and 1000 iterations (Figure 5.4) there is no difference in the network structure, only that some of the link lengths have changed to fit the new data. The spatial locations of the neurons are organised to produce the most meaningful network given the changing structure of neurons and links. The spatial locations of the neurons are not the focus here but the ability to analyse the resultant structure.

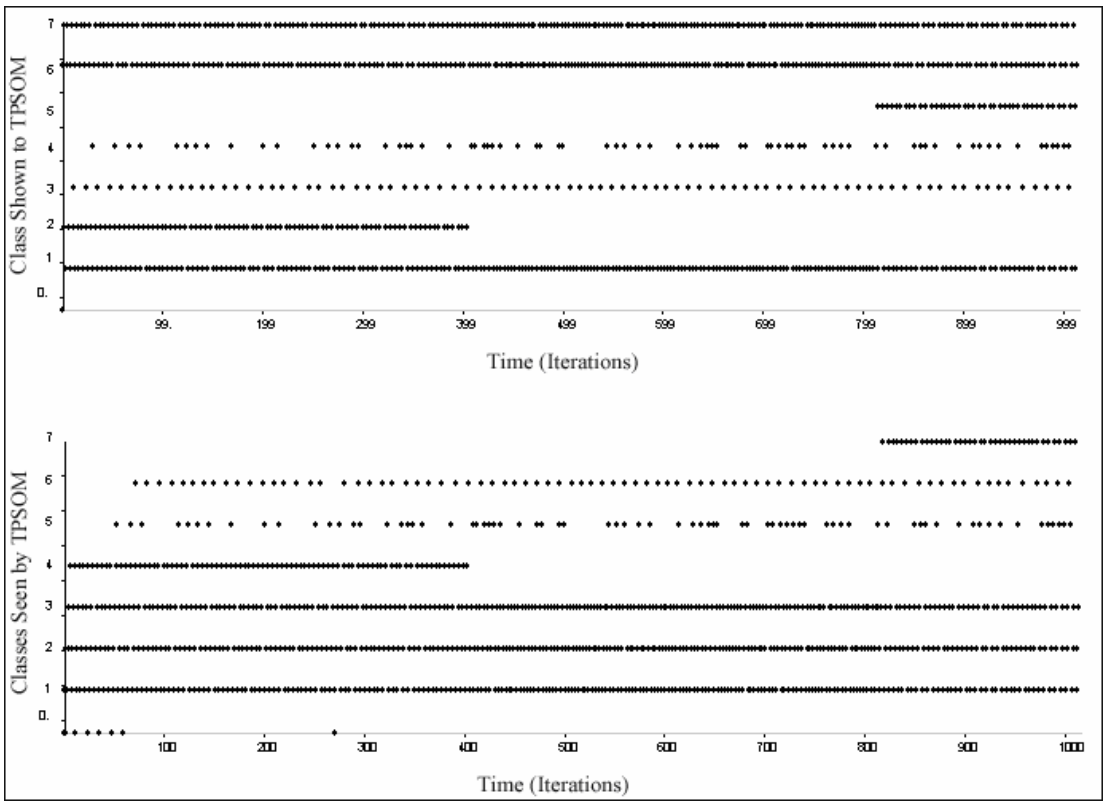

Figure 5.5 TPSOM Test 1: The class output from the actual class labels (top) and those produced by the TPSOM (bottom).

Table 5.6 TPSOM Test 1: Class associations between input class numbers and the TPSOM class numbers.

Input Class Number	TPSOM Class Number
1	1
2	4
3	0,6
4	5
5	7
6	2
7	3

Another output of the test is the Class output. Figure 5.5 shows two class outputs. The top graph is the class given to the input set and the bottom graph is the class given by the network. The class labels given to the input set and by the network are both arbitrary. At each time instance (horizontal axis) there is one instance of a class. This is represented by a dot. For every dot on the top graph (input class number) there is a dot on the lower graph (TPSOM output class). Each line represents a class. Therefore, as the TPSOM is classifying correctly, a line on the top graph will correspond with a line on the lower graph. Table 5.6 shows the associations between the input class number and the TPSOM given class. Initially, the TPSOM assigned input class 3 as class 0, because the initial neuron (labelled class 0) was, by chance, located within the area of input class 3. This only lasted until iteration 300 when the neuron concerned was removed and TPSOM class 6 took over.

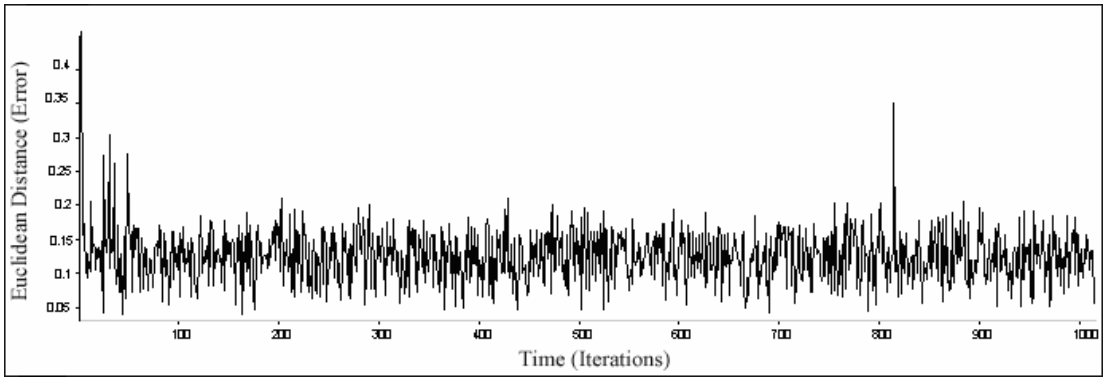

Figure 5.6 TPSOM Test 1: The recognition error over time.

Figure 5.6 shows the recognition error (the Euclidean distance between the input pattern and the focus) from the TPSOM. This is to show that even though there is only one neuron representing each class, the error manages to remain below 0.2. This is a known property of this data set. It is expected that if the classes were of different sizes, then this may no longer be the case. After iteration 50, no new patterns are shown to the network until iteration 800, therefore the recognition error tracks the noise of the input patterns caused by their spread in the input space.

5.4.1.1 Test Summary

In this test, the TPSOM was tested on an artificial dataset that had a known temporal structure. Using parameters based on previous experiments, the TPSOM

satisfactorily classified the 7 classes differently with arbitrary numbering. The final network output is a directed graph that represents the temporal structure of the network. This would not have been visible using the (E)PSOM algorithms.

5.4.2 Test 2: The Operation of TPSOM on Noisy Artificial Data Set.

The aim of this experiment was to see how the TPSOM would react to an input set that contained a pure noise class, patterns that did not belong to any other class and appeared anywhere in the search space at random. This new input set was created by taking the input set from Test 1 (the 7 pattern artificial multivariate input set) and adding 100 random patterns with the class label 0. This expanded the data set from the previous 1000 data points to 1100. Some of these patterns are likely to exist within the clusters of existing patterns but this was considered acceptable. The new data set can be seen in Figure 5.2. Compared with Figure 5.1, it can be seen that there are new 'dots' or patterns that are placed randomly within the search space. These represent the random patterns. As the non-noise classes (those that are to be captured by the network) had not changed from Test 1, the same parameters were used (Table 5.2). The detrimental effect of adding this pure noise pattern is that the density of the non-frequent patterns is reduced.

The TPSOM was tested over the 1100 data points with network state samples (graphical and state transition matrices) taken at 100, 400, 700, 960 and 1100 iterations. The network states can be seen in Figure 5.7 and Figure 5.8. After 100 iterations, the network does not look dissimilar from the 100 iterations of the original data set (Figure 5.3). The largest difference is the number of high probability links, which is less in this test than before. The State Transition Matrix after 100 iterations (Table 5.7) reflects this, as there are no entries of probability 1.0 or close to it. The difference really occurs after 400 iterations when the additional links caused by the noise patterns increase the connectivity between the neurons and thus make the structure less coherent. By comparing Figure 5.3 and Figure 5.7 for 400 iterations, it can be seen that the complex structure of the network no longer clearly shows the temporal nature of the input set as shown in Figure 5.3. The noise input patterns have increased the connectivity such that the main characteristics in the data are no longer clearly visible. Also, the number of neurons in the network in this test is far greater

as the noise patterns exist in all areas of the input space. The State Transition Matrix at 400 iterations (Table 5.8) has very few large values and has many entries.

Between 700 and 950 iterations, the situation worsens; Class 2 disappears after 400 iterations but the neurons pertaining to this class remain in the network even after 950 iterations. As there are so many small links within the network, neurons very rarely become unlinked and thus the class 2 neuron is retained. From an efficiency standpoint, this retention of information is undesirable although the addition of neurons will stop when there are enough to cover the whole input space.

Table 5.7 TPSOM Test 2: State Transition Matrix after 100 iterations.

	0	**1**	**2**	**3**	**4**	**5**
0	0.00	0.34	0.00	0.00	0.33	0.33
1	0.00	0.00	0.48	0.30	0.22	0.0
2	0.00	0.42	0.00	0.58	0.00	0.00
3	0.00	0.29	0.00	0.00	0.71	0.00
4	0.35	0.33	0.00	0.00	0.00	0.32
5	0.00	0.59	0.00	0.41	0.00	0.00

Table 5.8 TPSOM Test 2: State Transition Matrix after 400 iterations.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0.00	0.23	0.15	0.15	0.14	0.18	0.00	0.16
1	0.00	0.00	0.43	0.00	0.19	0.00	0.18	0.21
2	0.00	0.00	0.00	0.84	0.16	0.00	0.00	0.00
3	0.00	0.00	0.19	0.00	0.42	0.21	0.19	0.00
4	0.27	0.27	0.13	0.00	0.00	0.14	0.18	0.00
5	0.00	0.59	0.00	0.15	0.26	0.00	0.00	0.00
6	0.25	0.25	0.25	0.00	0.26	0.00	0.00	0.00
7	0.00	0.50	0.50	0.00	0.00	0.00	0.00	0.00

Table 5.9 TPSOM Test 2: State Transition Matrix after 700 iterations.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0.00	0.25	0.11	0.12	0.10	0.18	0.13	0.12
1	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
2	0.16	0.19	0.00	0.51	0.14	0.00	0.00	0.00
3	0.16	0.17	0.12	0.00	0.21	0.12	0.11	0.11
4	0.23	0.21	0.10	0.16	0.00	0.00	0.14	0.15
5	0.00	0.45	0.21	0.19	0.00	0.00	0.00	0.15
6	0.16	0.17	0.17	0.17	0.16	0.17	0.00	0.00
7	0.18	0.22	0.19	0.20	0.00	0.20	0.00	0.00

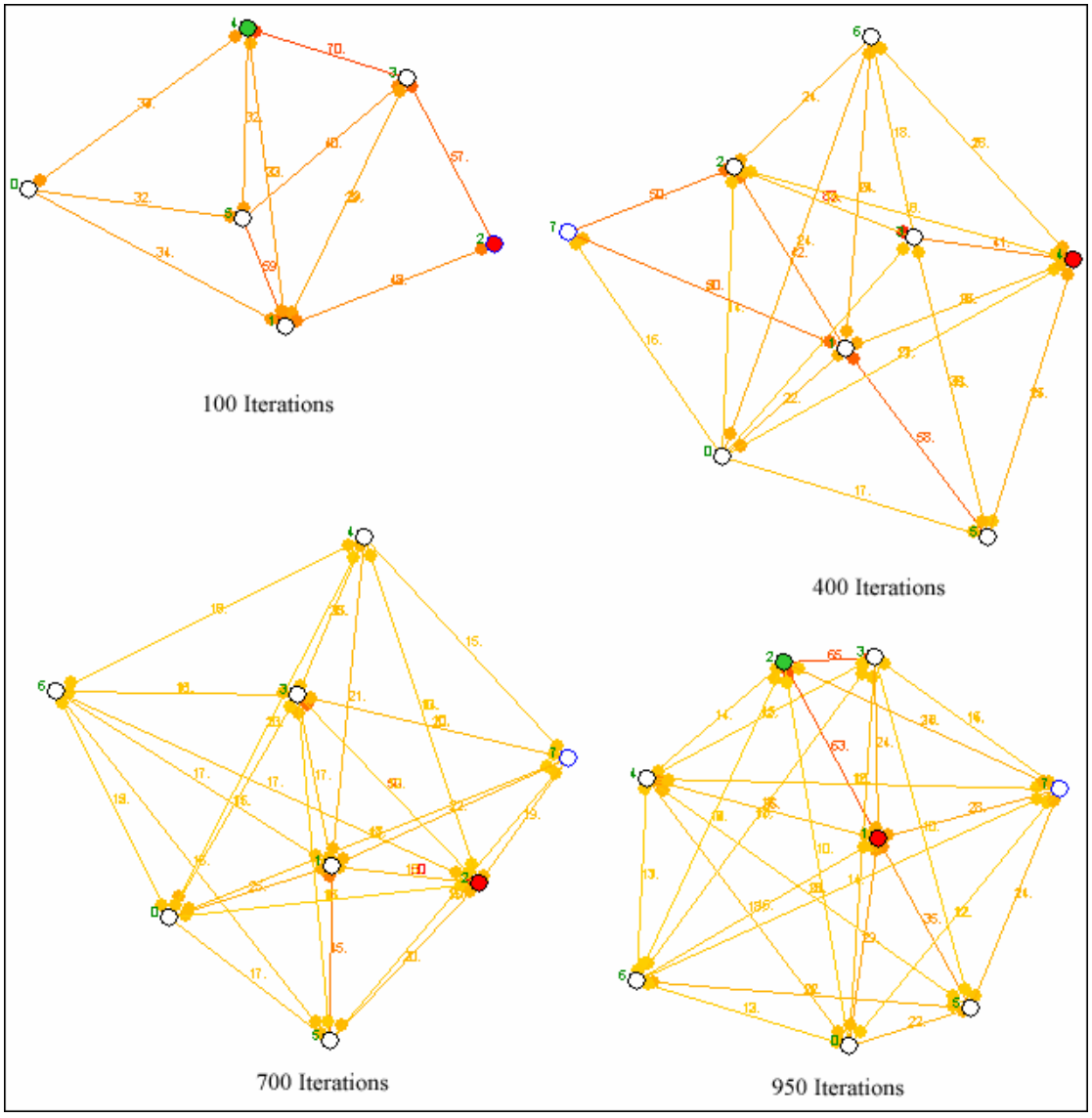

Figure 5.7 TPSOM Test 2: Network states through time at key point during operation.

Table 5.10 TPSOM Test 2: State Transition Matrix after 950 iterations.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0.00	0.30	0.10	0.13	0.00	0.23	0.12	0.12
1	0.00	0.00	0.80	0.00	0.00	0.00	0.20	0.00
2	0.00	0.00	0.00	0.58	0.00	0.00	0.14	0.28
3	0.19	0.24	0.00	0.00	0.16	0.11	0.14	0.14
4	0.21	0.21	0.00	0.15	0.00	0.16	0.13	0.14
5	0.00	0.36	0.00	0.00	0.20	0.00	0.22	0.22
6	0.13	0.15	0.15	0.14	0.13	0.15	0.00	0.14
7	0.16	0.30	0.17	0.19	0.00	0.19	0.00	0.00

Table 5.11 TPSOM Test 2: State Transition Matrix after 1100 iterations, the entries in grey show that persistent, regular ordering still remains in the network amongst noise.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0.00	0.29	0.10	0.12	0.00	0.24	0.12	0.13
1	0.21	0.00	0.79	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.89	0.00	0.00	0.11	0.00
3	0.21	0.22	0.00	0.00	0.16	0.11	0.12	0.18
4	0.17	0.24	0.13	0.12	0.0	0.12	0.10	0.11
5	0.19	0.27	0.00	0.00	0.17	0.00	0.17	0.19
6	0.13	0.16	0.15	0.14	0.14	0.14	0.00	0.14
7	0.00	0.33	0.11	0.13	0.15	0.13	0.15	0.00

The State Transition Matrix for 950 iterations is completed approximately 50 iterations after Class 5 is shown to the network. No new neurons are added to the network, and a neuron previously associated with noise is now likely to be used to represent input Class 2. The connectivity is still high, with many of the matrix entries filled with small values, indicating transitions that are not regularly occurring. This is reflected in Figure 5.7 (950 iterations) where the graph is so highly connected that no meaningful structure is captured in the network representation.

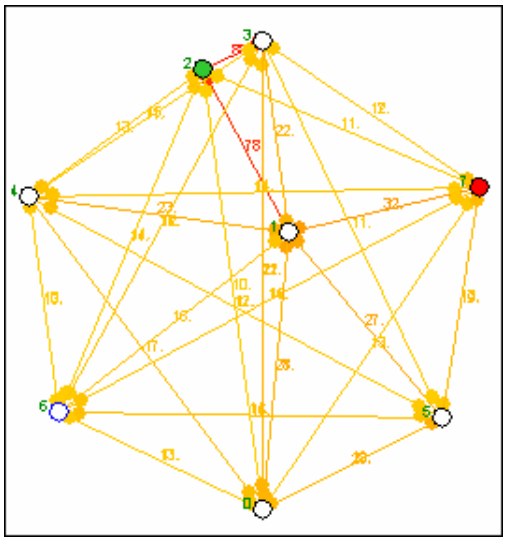

Figure 5.8 TPSOM Test 2: The network state after 1100 iterations.

Figure 5.8 shows the network structure after 1100 iterations. This is the most stable state of the network and yet there are too many links to extract any meaningful information. Although there are many links with low probability (as can be seen in Table 5.11) there are two links of high probability (highlighted in grey). These links, 1→2 and 2→3 are the only structure amongst the large number of links. These two links are represented by the darkest links in Figure 5.8. This presence of links with high probability shows that although the spatial noise levels are high, some structure can still be represented by the network, even when the signal to noise ratio is very low.

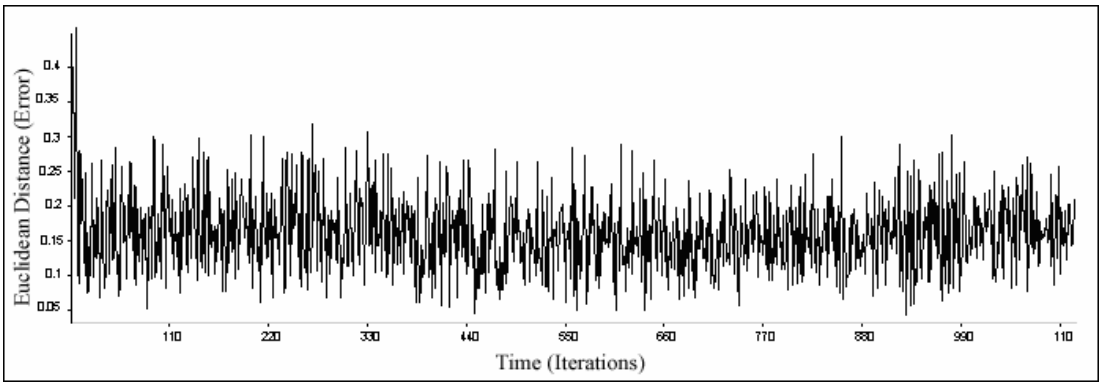

Figure 5.9 TPSOM Test 2: The recognition error through time.

Figure 5.9 shows the recognition error during the network's operation. Due to the noise, the error level does not settle to a minimum value and it is impossible to see

any prominent spikes associated with the presentation of a new pattern (see Figure 5.6 at iteration 800). More neurons can be created by reducing the size of a_n (as shown in EPSOM Test 2) which would have the effect of reducing the overall error.

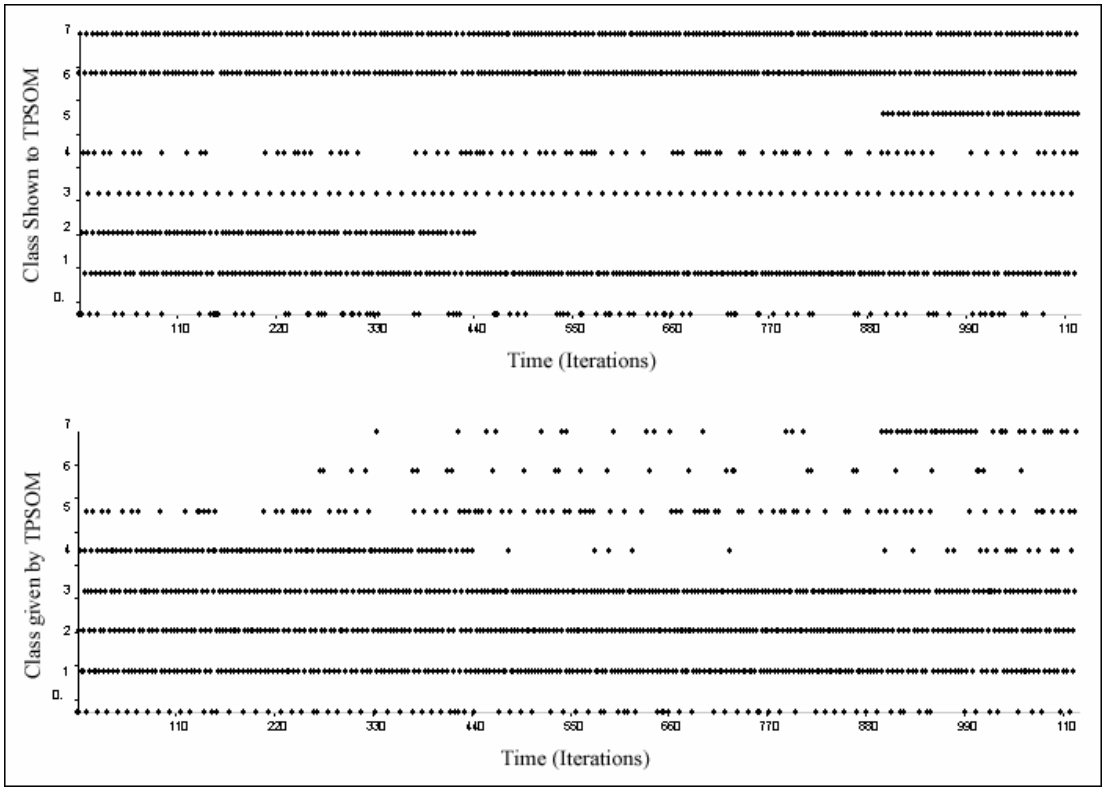

Figure 5.10 TPSOM Test 2: The class output of the network.

Figure 5.10 shows the class outputs from the arbitrary labels of both input set (upper) and the TPSOM (lower). Generally the ability to classify is still good; however this is secondary to the graphical output. If classification was the most important output here then the (E)PSOM should be used instead.

5.4.2.1 Test Summary

In this test, the TPSOM was used to find the temporal structure in a data set where not all the input patterns belonged to a desirable class. The TPSOM does not give a good graphical representation of the input space when 9% of the patterns in the input set (100 in a total of 1100 patterns) are noise patterns distributed randomly around the input space. The TPSOM, like the (E)PSOM, can still separate input classes from noise but this is secondary to the ability to produce a meaningful graph. This result is expected because the addition of noise patterns for this seven-pattern set has reduced

the relative importance of the desirable classes to the point that noise patterns are as common as input patterns.

5.4.3 Test 3: Radar Data Set.

The aim of this test was to see if there was any temporal structure to the radar data set. The parameters (Table 5.12) were taken initially from PSOM Test 9 and then empirically tuned such that the class output of the TSOM was similar to the PSOM of Test 9. For this data set, there is no known state transition matrix. Instead this test was to explore whether the data set did have any structure. The learning rate, b_r was found empirically.

Table 5.12 TPSOM Test 3 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.21
Link Removal Threshold (a_r)	0.1
Learning Rate (b_r)	0.01

Network states were recorded at 5000, 15000, 20000 and 21484 iterations. The first state was to show the network before it had settled down, the second iteration was halfway through the data set and the final state was the final state of the network. These were chosen empirically as the best points to show the operation of the network.

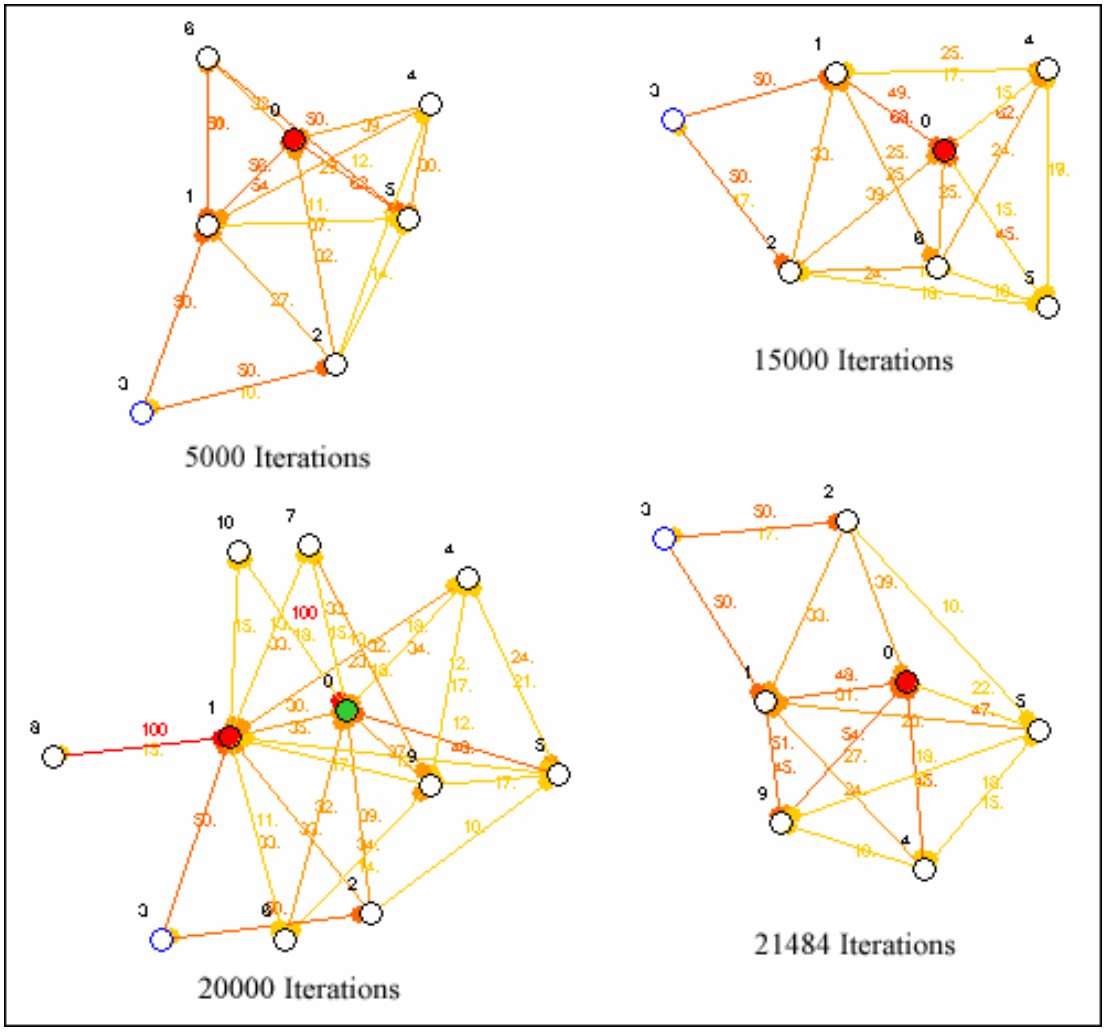

Figure 5.11 TPSOM Test 3: Network states every 5000 iterations and then finally at 21484 iterations.

Figure 5.11 shows network states at 5000 iteration intervals and the final state at 21484. Throughout the life of the network the TPSOM fails to show any meaningful structure within the data. None of the network outputs have any recurring features that consistently reappear between network snapshots. This is reflected in the state transition matrices below.

Table 5.13 TPSOM Test 3: State transition matrix after 5000 iterations.

	2	3	4	5	6	7	8
2	0.00	0.55	0.00	0.00	0.00	0.13	0.32
3	0.57	0.00	0.00	0.00	0.00	0.12	0.32
4	0.33	0.28	0.00	0.11	0.14	0.15	0.00
5	0.00	0.50	0.50	0.00	0.00	0.00	0.00
6	0.39	0.30	0.00	0.00	0.00	0.31	0.00
7	0.62	0.38	0.00	0.00	0.00	0.00	0.00
8	0.00	0.50	0.00	0.00	0.00	0.50	0.00

Table 5.14 TPSOM Test 3: State transition matrix after 21484 iterations (final state).

	2	3	4	9	6	7	14
2	0.00	0.32	0.00	0.00	0.19	0.22	0.27
3	0.49	0.00	0.00	0.00	0.00	0.00	0.51
4	0.40	0.40	0.00	0.10	0.00	0.10	0.00
9	0.00	0.50	0.50	0.00	0.00	0.00	0.00
6	0.46	0.25	0.00	0.00	0.00	0.18	0.11
7	0.48	0.24	0.00	0.00	0.16	0.00	0.13
14	0.54	0.46	0.00	0.00	0.00	0.00	0.00

Table 5.13 and Table 5.14 show state transition matrices for two time snap shots of the TPSOM. The time index of 5000 (Table 5.13) was chosen for the first snapshot because it would give the TPSOM enough time to settle. None of the values in Table 5.13 are very large (>0.6) which would suggest that there is little or no order to the patterns. Table 5.15 shows the final state of the TPSOM's state transition matrix. Columns 2 and 3 show a list of very high values, which would suggest that classes 2 and 3 are the most popular. However, this does not show temporal structure as such, just the statistical density of the classes concerned.

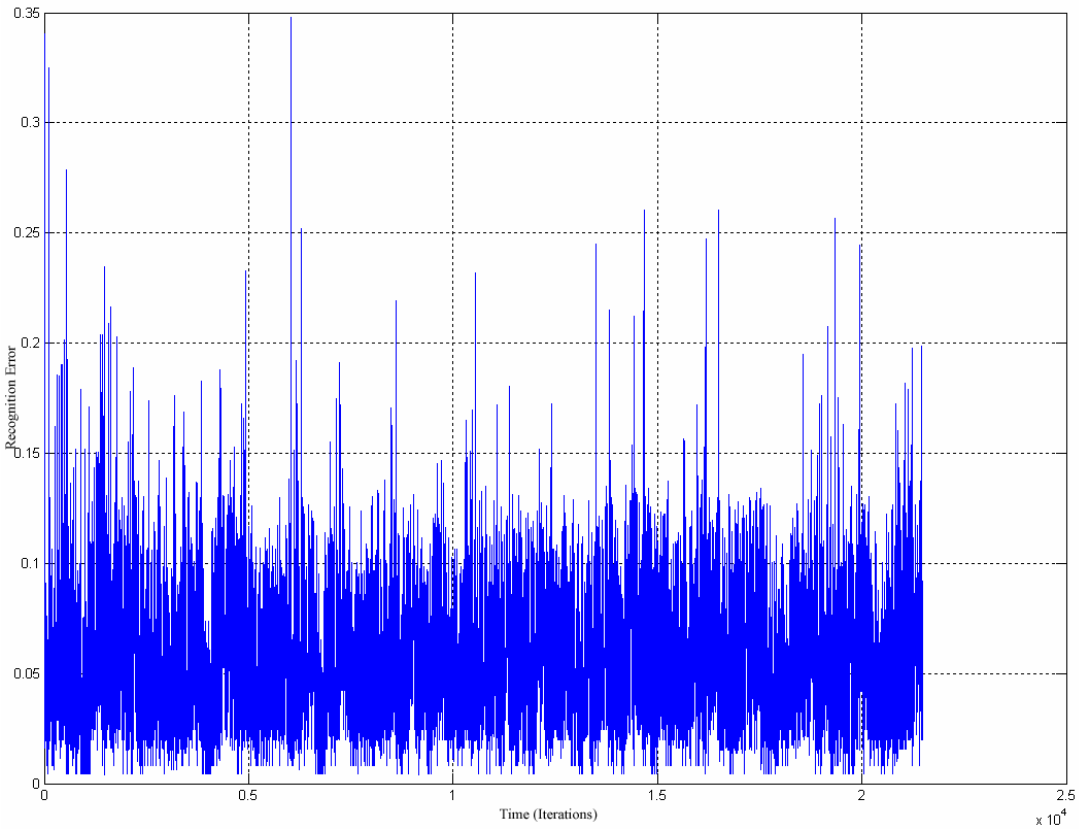

Figure 5.12 TPSOM Test 3: Recognition Error over time.

One might be led to conclude that the TPSOM was not functioning correctly during its operation. Figure 5.12 shows the recognition error over time for the operation of the network. The recognition error given here is greater than that produced by the PSOM. This is expected as only one neuron is used to represent a class or state rather than having 3 or more as with the PSOM. However, beyond the general larger trend, the error really is not that oscillatory or as large as would be expected if the network was not learning and continually building new neurons.

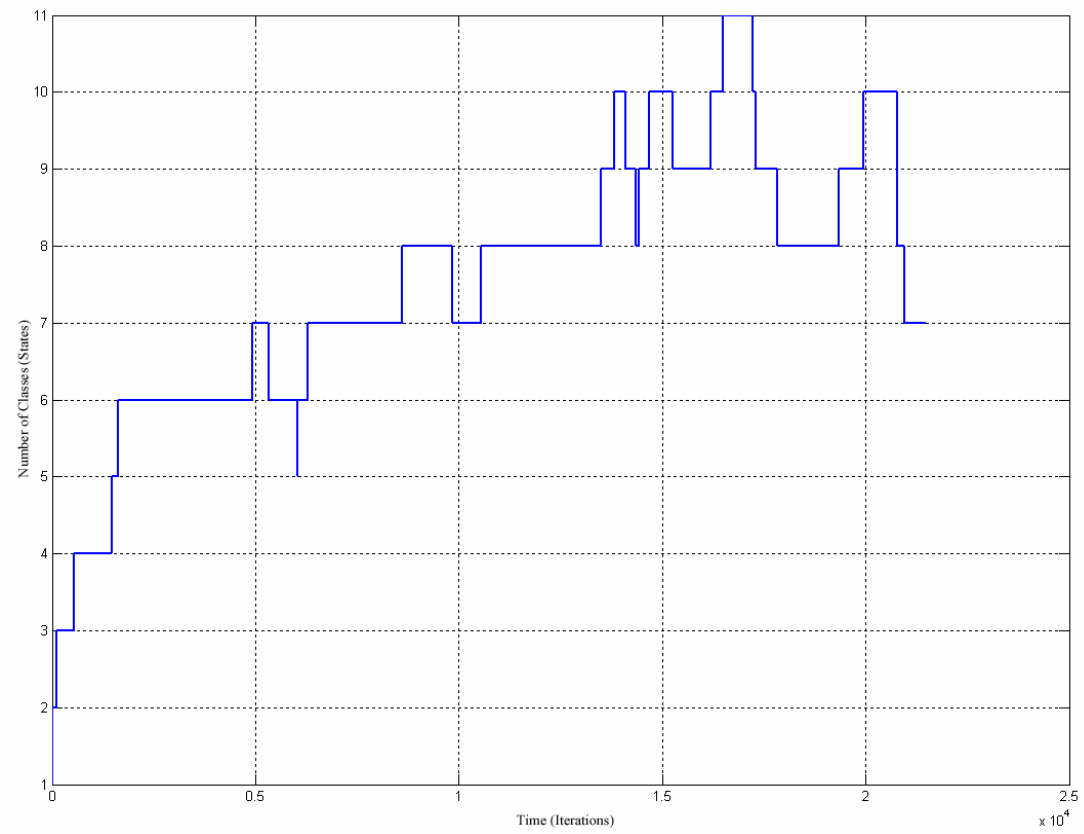

Figure 5.13 TPSOM Test 3: The number of classes (states or neurons) over time.

Figure 5.13 shows the number of classes during the operation of the network. As each class is represented by one neuron and each state is one class, neuron, class and state are effectively identical for the purposes of this test. Given that there are 4 classes within the input set, a maximum of 11 for the number of classes could be an indicator of the network failing to classify. However, these additional classes may not have lasted in the network or become states that are held in the network but never accessed, and so no such conclusion can be drawn.

Table 5.15 TPSOM Test 3: The relationship between the input class number and the TPSOM class number. Only significant rows shown.

		Input Set Class Number			
		1	2	3	4
TPSOM Class Number	0	7	0	1	0
	2	12859	0	2	1
	3	0	0	6172	0
	4	6	0	0	621
	6	472	1	0	0
	7	278	0	0	0
	10	0	0	26	0
	12	0	0	21	0
	14	0	938	0	0
	15	0	53	1	0

Table 5.15 shows the relationship between the input class number and the TPSOM class number. Only significant rows are shown for the sake of brevity. As can be clearly seen, the TPSOM has classified the classes well. Of input classes 1 and 2, only 5% of the patterns were wrongly classified and a very small proportion of classes 3 and 4 were consistently mis-classified. This suggests that the classes were classified properly on the whole.

5.4.3.1 Test Summary

In this test the TPSOM was used to investigate the temporal properties of the radar data set. The network was found to classify well, with a low recognition error and a consistency of input pattern classification. However, no temporal pattern was found in the data. This was not entirely surprising as radar data does not have any temporal structure where one class (radar source) follows another. For this to occur, radar sources would need to be synchronised with each other such that they arrived in an ordered pattern at the point where the pulses are collected. Such an occurrence would be a coincidence. During the test, it was noticed that the same neuron would be called as the focus many times consecutively.

5.4.4 Test 4: Static Markovian Chain

A static Markov Chain is one where the probabilities connecting the states do not change during the operation of the system. The input state for the network is generated from a State Transition Matrix. The data is generated in the following manner:

1. Choose a state starting point from all the states available. This is the state that will represent the first input to the network – chosen arbitrarily.
2. For each state, there is a corresponding vector which is the same size as the input to the TPSOM. This vector represents a point in the input space and whose elements are either 1 or 0.
3. Before being applied to the TPSOM, the input is perturbed by 20% random noise such that the input to the TPSOM remains between 0 and 1.
4. Apply the input to the network.
5. Generate a random number between 0 and 1.
6. Using the state transition matrix, use the random number to generate the next state.
7. Use this state and go to step 2.

For this test the Markov Chain Source State Transition Matrix is given by Table 5.16.

Table 5.16 TPSOM Test 4: The Markov Chain Source State Transition Matrix.

	0	**1**	**2**	**3**	**4**
0	0.00	1.00	0.00	0.00	0.00
1	0.00	0.00	0.40	0.60	0.00
2	0.00	0.00	0.00	1.00	0.00
3	0.00	0.00	0.00	0.00	1.00
4	0.20	0.20	0.60	0.00	0.00

This State Transition Matrix can also be visualised as a state graph as in Figure 5.14.

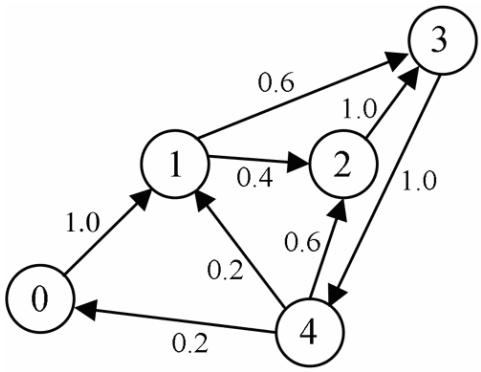

Figure 5.14 TPSOM Test 4: The Markov Chain Model pictorally.

Although the model displayed in Figure 5.14 is accurate, it does not best show the structure of the network as the link lengths do not correspond to the values displayed by them. Ideally large numbers should be short links and small numbers would be long links. This is so that states that often follow each other are kept close together. This is almost impossible to achieve by drawing, especially as the number of states increases. The TPSOM will organise this network using its layout algorithm to produce a satisfactory arrangement for the network.

The data generated by the Markovian Source is stochastic and bounded between 0 and 1. This is achieved by first checking the value of the output vector that, before noise, is either 0 or 1. A random value is generated and biased depending on the amount of noise required (in this test, 20% noise was used, thus the random value was between 0 and 0.2). If a value in the output vector is 1, then the random value is subtracted from the output vector. If the value in the output vector is 0, then the random value is added. The upshot of this method is that the output is always bounded between 0 and 1. The pre-noise classes used for this test were taken from the standard classes above.

The TPSOM Parameter a_n was taken from EPSOM Test 1, the learning rate being determined empirically. The aim of the test is to see if the TPSOM can create a state transition matrix similar to that which is held in the Markov Source. As the Markov Source is static, there is no requirement for states to be removed and thus the threshold for link removal, a_r, is set to 0. By doing so, links of value less than 0 can

only be removed. As links are bounded between 0 and 1, this is impossible. This allows the network to settle to stable values without removing any links. Before testing dynamicity, it is necessary to confirm that the static system can be captured. The test parameters are summarised in Table 5.17.

Table 5.17 TPSOM Test 4 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.29
Link Removal Threshold (a_r)	0.01
Learning Rate (b_r)	0.0

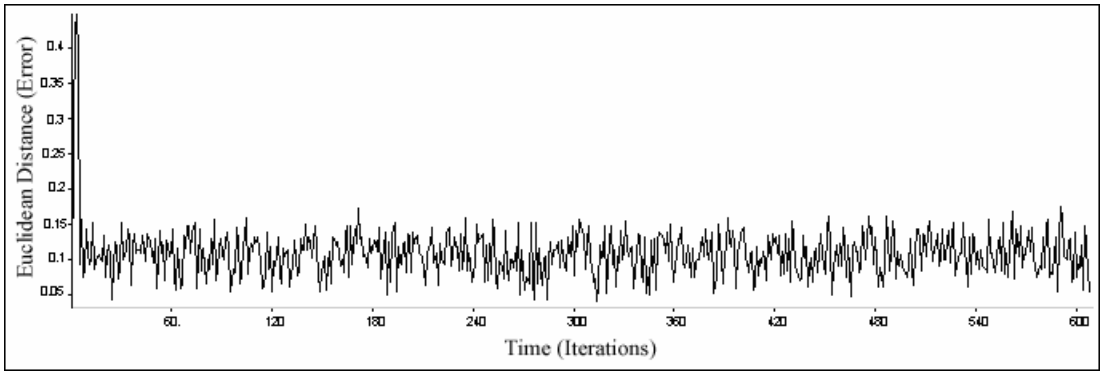

Figure 5.15 TPSOM Test 4: The recognition error over time.

As shown previously, the speed at which the TPSOM's recognition error (Figure 5.15) becomes stable (models only the noise) is small compared to the time that is required to organised the probabilistic links. This speed of convergence is mostly to do with the nature of the PSOM rather than the TPSOM. Altering those parameters that the TPSOM shares with the PSOM will have similar effects as for the PSOM. Further comparison is provided later.

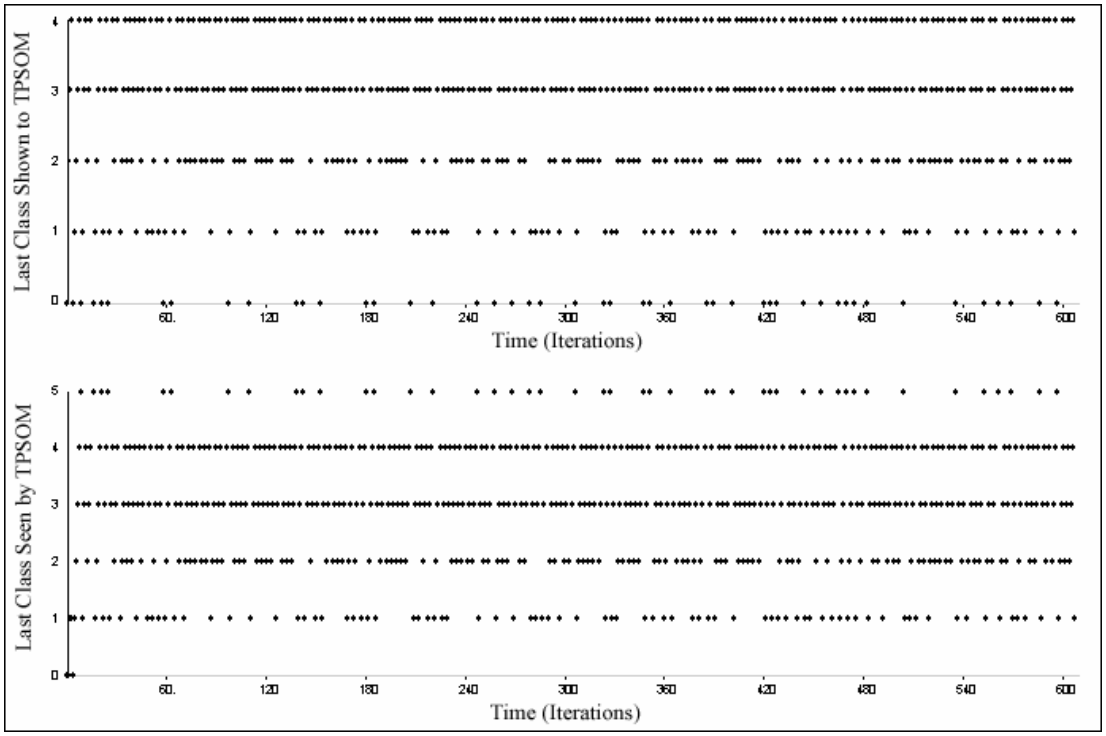

Figure 5.16 TPSOM Test 4: The class output from the actual class labels (top) and those produced by the TPSOM (bottom).

As the class numbers for both the TPSOM and the Markovian Source are arbitrary, the numbers given on the Class Label graphs (Figure 5.16) are purely for reference. It can be seen that the TPSOM has classified correctly by comparing the two graphs in Figure 5.16. If the pattern of dots in one line of the TPSOM output (bottom graph) is the same as the pattern in a line from the Markovian Source (top graph) then it can be said that the TPSOM is classifying correctly. In this manner, it can be seen that TPSOM Class 6 (top line of bottom graph) is equivalent to Markovian Source Class 0 (bottom line, top graph). The classes can be matched by monitoring the TPSOM output and comparing the class number shown to the network and the class number assigned by the network. By doing so, one can see that the classes match as follows:

Table 5.18 TPSOM Test 4 class associations

Markovian Class	TPSOM Class
0	5
1	1
2	2
3	3
4	4

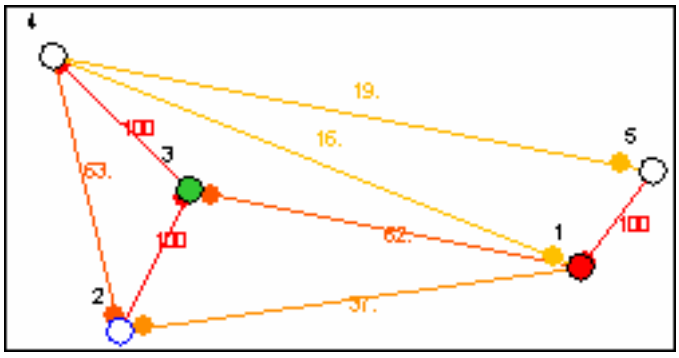

Figure 5.17 TPSOM Test 4: The final network state after 600 iterations.

Table 5.19 TPSOM Test 4: State transition matrix.

	1	2	3	4	5
1	0.00	0.37	0.63	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00
4	0.17	0.63	0.00	0.00	0.20
5	1.00	0.00	0.00	0.00	0.00

The TPSOM consistently matched one class to another after approximately 40 iterations. Figure 5.17 shows the final state of the graph. As the graph structure did not change after the first 40 iterations, no other network states were recorded. The structure is similar to that shown in Figure 5.14. By replacing the class numbers with those given in Table 5.18, one can see that the structures are identical. Clearly visible in Figure 5.17 but not visible in Figure 5.14 is the cyclic substructure of states 4→2→3 which occurs more often than the rare states 1 and 5. The layout of the TPSOM makes this structure visible on inspection, not easily achieved diagrammatically.

As the source of the input patterns is a State Transition Matrix, it is possible to create a measure of error between the TPSOM State Transition Matrix and the input equivalent. This is achieved by first calculating the difference between corresponding entries in the State Transition Matrices of input and the TPSOM and then summing the absolute values in this matrix. This sum is a measure of error between the matrices. This is only possible after the matrices are rearranged (the arbitration process that matches up the class names) and will not possible for matrices of

differing sizes. This case could arise if the TPSOM used more neurons (states) to represent the input space than was necessary. Snapshots for the TPSOM State Transition Matrix were taken at 100, 200, 300, 400 and 600 iterations. The State Transition Matrix Error is given below (Table 5.20).

Table 5.20 TPSOM Test 4: State Transition Matrix Error over time.

Time	STM Error
100	0.69
200	0.54
300	0.34
400	0.18
600	0.12

As Table 5.20 shows, the error between the input State Transition Matrix begins large and then reduces to a small and acceptable value, given that the network has only been used for 600 iterations. This process of error extraction cannot be automated as the matrix transformation required to compare the two matrices requires an observer. It is expected that this error tends to 0 at time infinity.

5.4.4.1 Test Summary

In this test, the output from a static Markovian Chain source was used to test the TPSOM ability to find temporal structure from a Markovian chain. The TPSOM was used to generate a state transition matrix which could be compared to the Static Markov Chain. It was found that the TPSOM could generate a satisfactory graph without any prior knowledge of the temporal structure of the data. The only a-priori data that the TPSOM was given was clustering information concerned with splitting up the input classes sensibly (as specified at the start of this section). The error between the TPSOM state transition matrix and the Markov Chain source state transition matrix was found to decrease as more examples were shown and it is hypothesised that the error would tend to 0 at time infinity.

5.4.5 Test 5: Dynamic Markov Chain

The dynamic Markov chain used here is simulated by two static Markov Chains with equal number of states between which the model switches. It is assumed that if the TPSOM can deal with such a discontinuity, than a more gradual change would also be possible. In this case, only the state transition matrix of the input is changed, the classes remaining the same. This is not a requirement of the network, but improves the clarity of the test. The test was conducted by having two State Transition Matrices and an integer switch time index. Before this time index (number of iterations), State Transition Matrix 1 is used as the input to the TPSOM, after this State Transition Matrix 2 is used. The State Transition Matrices (Table 5.21 and Table 5.22) have different elements; non-zero entries in the first matrix do not exist in the second matrix and vice versa. This should show the TPSOM changing its structure completely, without having to relearn any classes.

Table 5.21 TPSOM Test 5: State Transition Matrix 1.

	0	**1**	**2**	**3**	**4**
0	0.00	1.00	0.00	0.00	0.00
1	0.00	0.00	1.00	0.00	0.00
2	0.00	0.00	0.00	1.00	0.00
3	0.00	0.00	0.00	0.00	1.00
4	1.00	0.00	0.00	0.00	0.00

Table 5.22 TPSOM Test 5: State Transition Matrix 2.

	0	**1**	**2**	**3**	**4**
0	0.00	0.00	0.00	0.00	1.00
1	0.00	0.00	0.00	1.00	0.00
2	1.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1.00	0.00	0.00
4	0.00	1.00	0.00	0.00	0.00

The parameters for this test are given in Table 5.23. The crossover point for the change from State Transition Matrix 1 to 2 was arbitrarily set to 500. The output classes were not perturbed by noise because the aim of the test was to show the manner in which the TPSOM operates on this type of dynamic Markov chain, not an investigation of its noise rejection properties in such a scenario.

Table 5.23 TPSOM Test 5 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.29
Link Removal Threshold (a_r)	0.01
Learning Rate (b_r)	0.0

The TPSOM responded well to the sudden change in temporal structure. New links were forged and old links were slowly phased out and then removed. Snapshots of the output graph and TPSOM state transition matrix were taken at intervals chosen before and after the switching times. The Recognition Error for the TPSOM was understandably small after the initial expansion period. This is because the input patterns were not corrupted by noise. As the Recognition Error was close to 0 for all of the network's operation, graphical displays of the results show very little.

Table 5.24 TPSOM Test 5: The relationship between the input class number and the TPSOM class number for the dynamic Markov Chain data set.

		Input Set Class Number				
		0	1	2	3	4
TPSOM Class Number	0	1	0	1	0	1
	1	0	1	0	1	0
	2	1	0	1	0	0
	3	0	0	0	125	0
	4	0	0	0	0	125
	5	125	0	0	0	0
	6	0	124	0	0	0
	7	0	0	124	0	0

Table 5.24 shows the relationship between the input class number and the dynamic Markov chain input. This is summarised in Table 5.25. This summary is possible as there were only a few misclassifications. These misclassifications occur when the network is initially organising its structure.

Table 5.25 TPSOM Test 5: The overall relationship between Markov Class and TPSOM class

Markovian Class	TPSOM Class
0	5
1	6
2	7
3	3
4	4

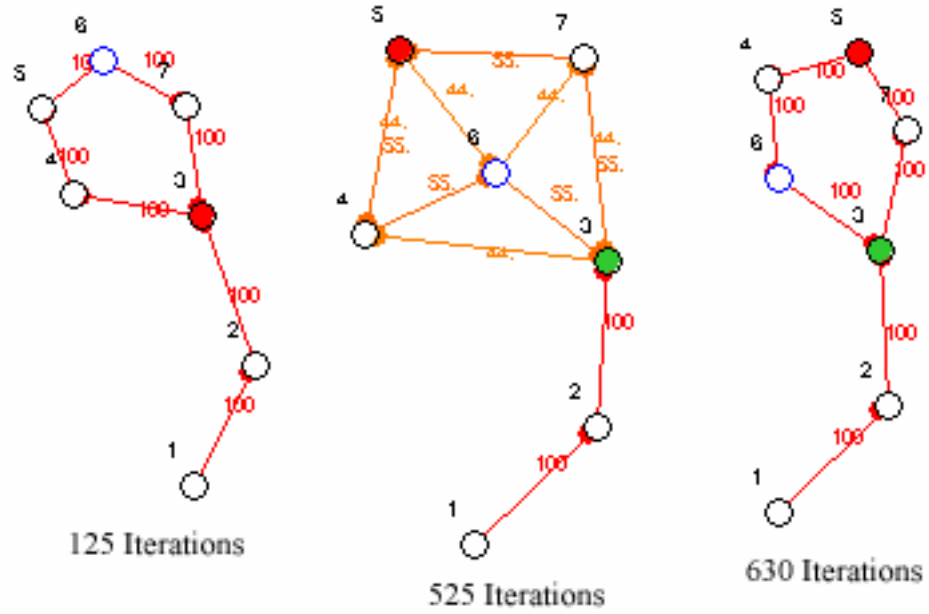

Figure 5.18 TPSOM Test 5: The network graphs at 125, 525 and 630 iterations.

Table 5.26 TPSOM Test 5: State Transition Matrix after 125 iterations.

	1	2	3	4	5	6	7
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00
7	0.00	0.00	1.00	0.00	0.00	0.00	0.00

Table 5.27 TPSOM Test 5: State Transition Matrix after 630 iterations.

	1	2	3	4	5	6	7
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	1.00
4	0.00	0.00	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	1.00	0.00	0.00	0.00
6	0.00	0.00	1.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	1.00	0.00	0.00

Figure 5.18 shows the visual output of the network at 125, 525 and 630 iterations. At 125 iterations, the network has settled down with the first Markov Chain. Neurons 1 and 2 are associated with initial classes that are only seen once in the operation of the network. Long chains like this are a known feature of the algorithm as there is no way of deleting neurons without inbound links that are not directly connected to the focus. The probabilities of these links are 1.0, or 100%. The State Transition Matrix is not included here as it is identical to the input State Transition Matrix. The Markov Chain matrix changes after 500 iterations.

At 525 iterations, the network has seen the new structure for 25 iterations and has created the new links to account for this change in structure. For the next 100 iterations, the network will strengthen the new links while decreasing the importance (probability) of the old links.

After 630 iterations, the network has taken on a new structure. Although similar in appearance (a ring with a short chain off one side), the structure of the network is different. Note that the order of the neurons within the ring has changed. This new structure corresponds to the State Transition Matrix of the second Markov Chain source. The error between the input and TPSOM State Transition Matrices at iteration 125 and 630 is zero. This shows that the network is representing the two different structures correctly.

5.4.5.1 Test Summary

This test has demonstrated the TPSOM's ability to represent a Markov Chain model with a discontinuity during its operation. The TPSOM copes with this change by first

creating new links required for the new structure before slowly reducing the strength of the original structural links. Approximately 130 iterations after the Markov Chain input's State Transition Matrix was changed, there was no difference between the TPSOM's structure and the new input system structure. This signifies that the network is able to change its own topology to meet the change in topology of the input set.

5.4.6 Test 6: The Effect of the Node Building Parameter

This test is to show the effect of the Node Building Parameter, a_n, on the operation of the network. It is expected that for low values of a_n, the network will grow very quickly and many classes will be formed. Long chains will be created as each input pattern will not be classified as an existing pattern. This will lead also to very large class association as a single class will be represented by many classes in the TPSOM. Large values of a_n will create small networks where a few neurons represent many different classes. This will lead to small class association matrices but poor classification. The data set used for this test is the 7 pattern Multivariate Artificial data set (Figure 5.1) in order to allow comparison with earlier tests. The parameters used in this test are shown in Table 5.28 and are based on the TPSOM Test 1's parameters.

Table 5.28 TPSOM Test 6 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.01-0.6
Link Removal Threshold (a_r)	0.1
Learning Rate (b_r)	0.025

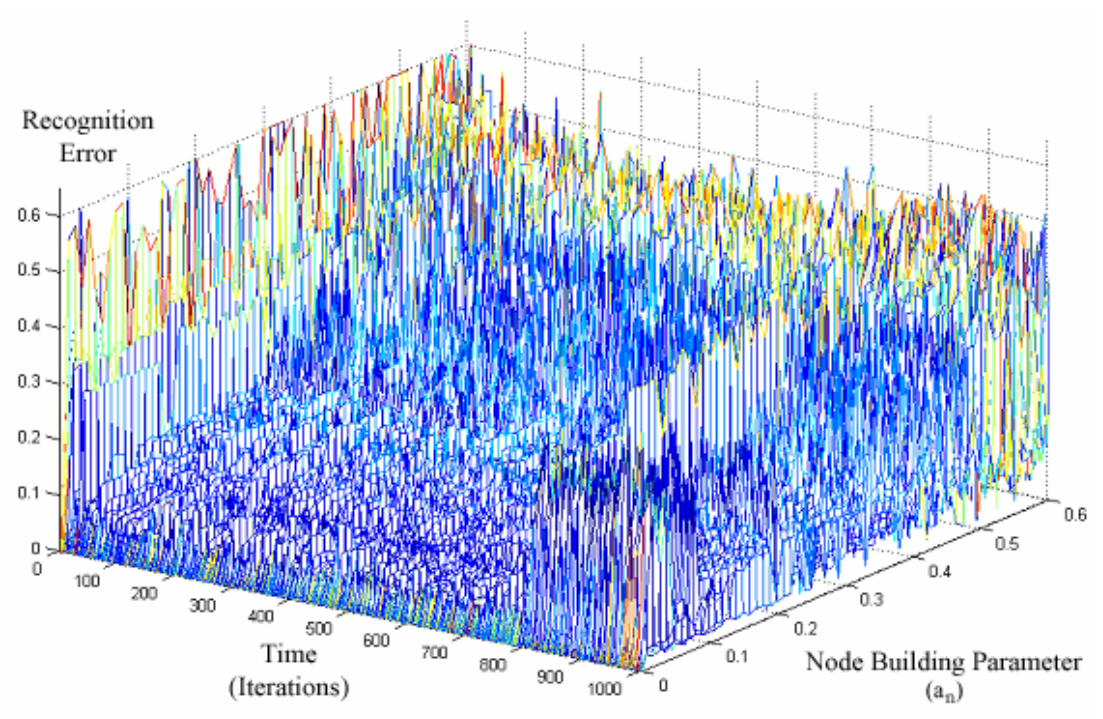

Figure 5.19 TPSOM Test 6: The effect of the node building parameter on the Recognition Error.

Figure 5.19 shows the effect of a_n on the Recognition Error of the network. For small values of a_n, the network creates many neurons and so the input space is well covered with neurons. Therefore, it is very likely that the TPSOM will have a neuron that represents the given class, leading to a low recognition error. As a_n increases, the smaller number of neurons in the network leads to individual neurons covering more and more of the input space. This continues until the error is very large. The instantaneous error value only quantifies errors in recognising a particular pattern and does not reveal whether the network has captured the temporal structure (shown later). In this respect, the TPSOM network performs in a similar manner to the (E)PSOM. This is because the algorithms use a_n in a similar manner: for the creation of new neurons.

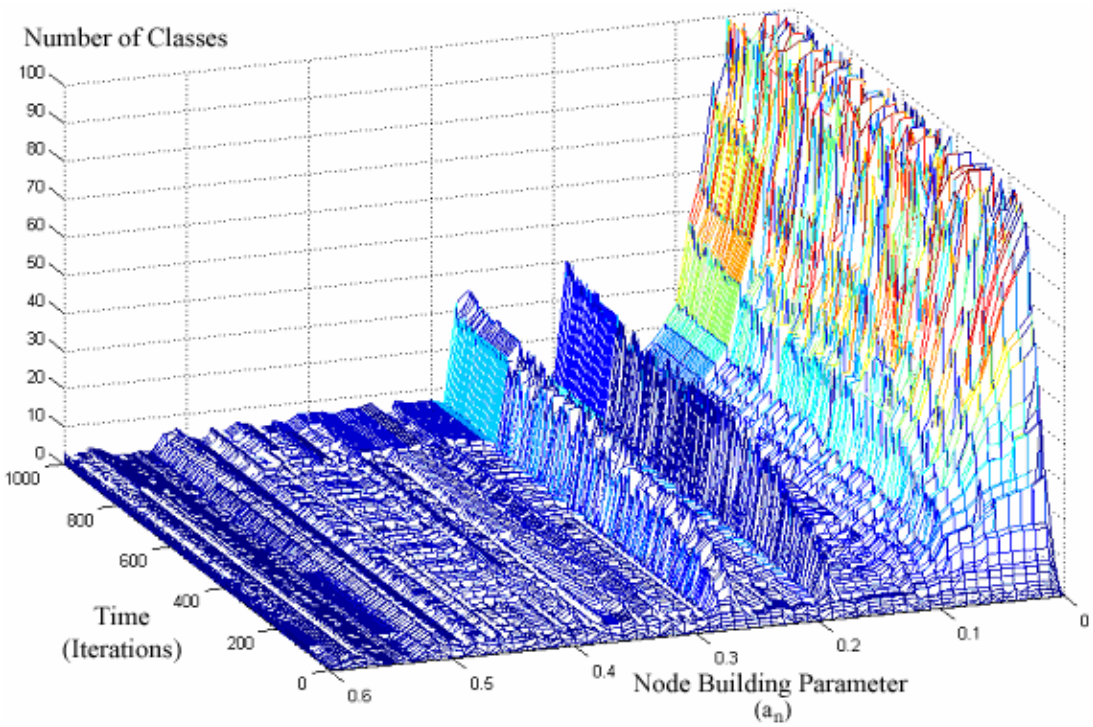

Figure 5.20 TPSOM Test 6: The effect of the Node Building Parameter on the last class seen (Number of states).

Figure 5.20 shows the effect of a_n on the last class seen in the network over time. When the results are displayed in this way, the last class seen is equivalent to the numbers of different classes (states) used by the network. For small values of a_n, the network creates many classes and vice versa for large values. At a_n values of 0.28 and 0.18, two ridges can be seen. These ridges represent a situation where the TPSOM is unable to classify the input initially. The network takes time to reach a stable state; when it does, it operates correctly and contains the correct number of classes, their labels being of high values. These high values suggest that the network expanded and contracted its topology many times during the learning process, requiring a new class number for each expansion.

From a perspective of class association matrices between input and the TPSOM, small values of a_n create sparse matrices with several TPSOM classes representing one input class. For values of a_n between 0.2 and 0.4 the TPSOM yields matrices much like that in

Table 5.3. Table 5.29 shows the relationship between the Input set class number and the arbitrary class number assigned to the class for an a_n value of 0.59. At this large

value, the TPSOM classes many different input classes as the same class. The three persistent, common classes, input set class numbers 1,2 and 3 (columns 1,2 and 3) appear to be classified well by TPSOM. For example, input class 2 (column 2) is represented by TPSOM class 2 (row 2), which has only 1 misclassification with 85 correct classifications. For all the other classes (both rows and columns) there are many column entries for a given row. This shows that the TPSOM incorrectly classified different input classes as one TPSOM class, highlighting a breakdown in the correct functioning of the network at high values of a_n.

Table 5.29 TPSOM Test 6: The relationship between the Input set class number and the class number provided by the TPSOM for an a_n of 0.59.

		Input Set Class Number							
		0	1	2	3	4	5	6	7
TPSOM Class	0	1	0	0	0	10	6	8	237
	1	0	236	1	1	59	0	126	0
	2	0	0	85	0	6	0	0	0
	3	0	0	0	86	0	35	103	0

5.4.6.1 Test Summary

This test has shown that for small values of a_n, the network has a very small Recognition error but uses many neurons to achieve this. Classification is not possible at this end of the scale due to the huge number of neurons sharing similar points in the input space. As a_n increases, the TPSOM begins to operate correctly until a value of approximately 0.35 is reached, at which point the ability for the TPSOM to distinguish between classes begins to decrease. At very large values of a_n, the network classifies different input classes as the same class, which is undesirable.

5.4.7 Test 7: The Effect of the Learning Rate

This test shows the effect of the Learning Rate, b_r on the operation of the TPSOM. The learning rate acts in a similar manner to the Link Ageing Parameter (a_r) in the (E)PSOM algorithms. The learning rate controls the effect of each pattern on the link probabilities. A high learning rate (near to 1.0) will mean that the effect of a given pattern is quite large and vice versa. The network parameters used for this test can be found in Table 5.30.

Table 5.30 TPSOM Test 7 parameters.

Parameter	Value
Node Building Parameter (a_n)	0.29
Link Removal Threshold (a_r)	0.1
Learning Rate (b_r)	0.001-1.0

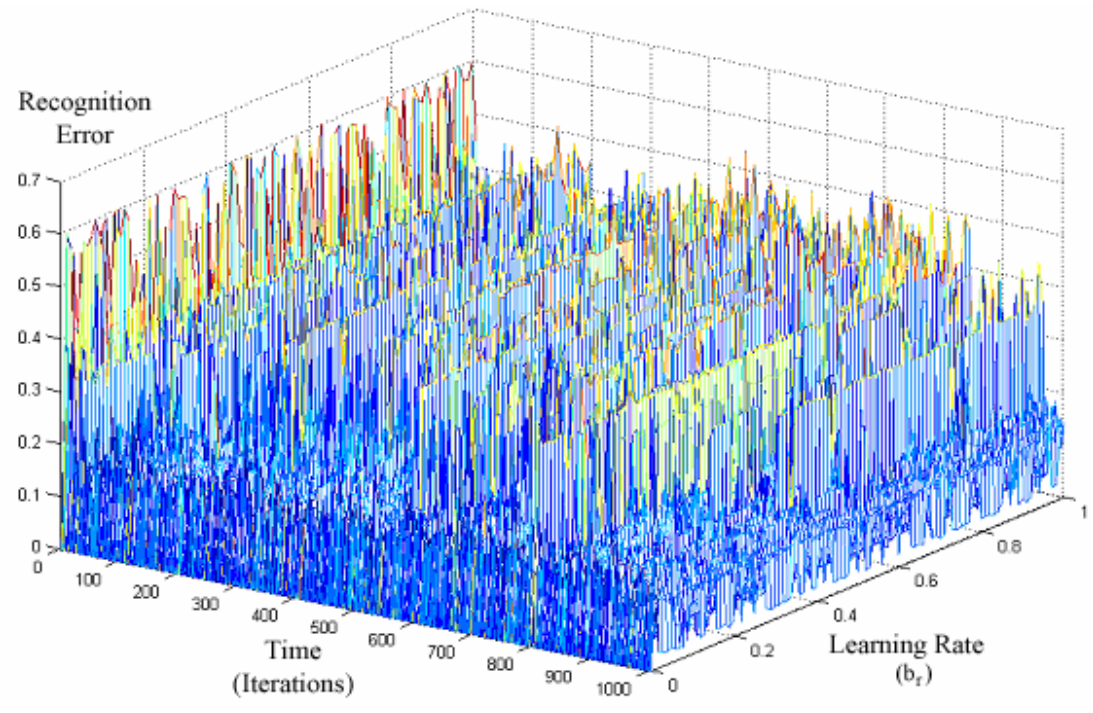

Figure 5.21 TPSOM Test 7: The effect of the Learning Rate on the Recognition Error.

Figure 5.21 shows the effect of b_r on the recognition error of the network. The operating range of the network is quite small (between 0.0001 and 0.05). This suggests that the TPSOM algorithm is not robust with respect to its learning rate. Beyond this, the network over-fits the input data and coherent structures are not created. Values of 0 cause the network to reject the input pattern and are therefore not considered.

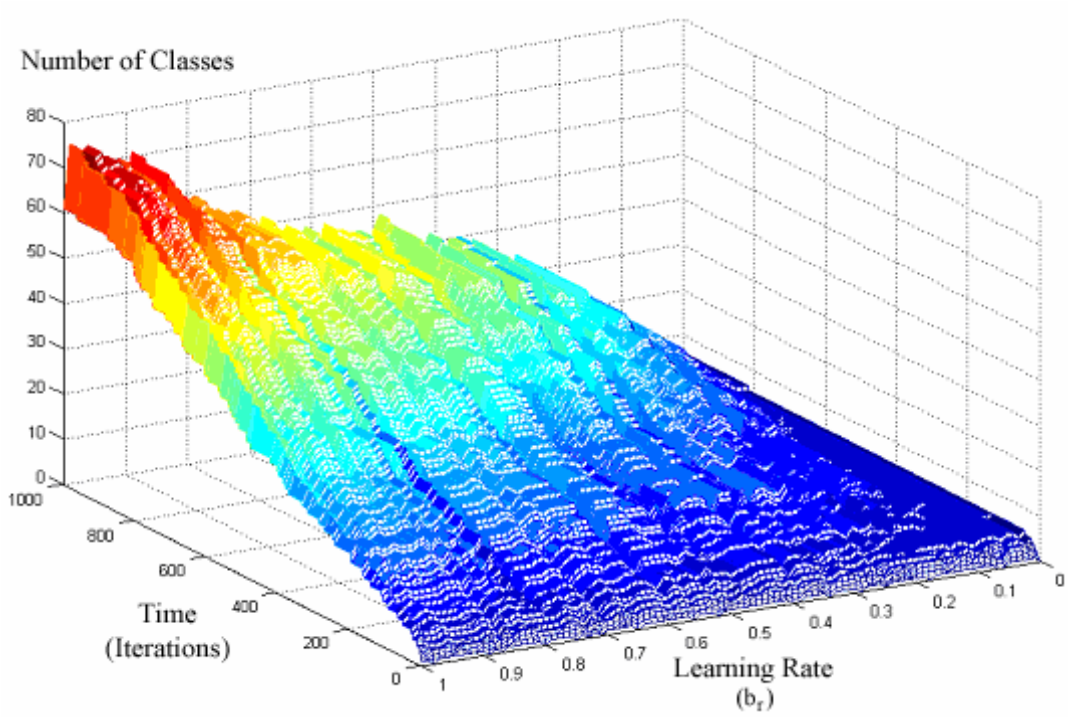

Figure 5.22 TPSOM Test 7: The effect of the Learning Rate on the number of classes within the TPSOM.

Figure 5.22 shows the effect of b_r on the number of classes within the TPSOM at any given time. For small values of b_r (< 0.08) the effect of each input is small, and over-fitting does not occur. The number of classes remains small and representative of the input set. As b_r increases beyond this value, each new input has a significant effect on the network which leads to old links being broken. In many cases, neurons associated with classes have no incoming links and are removed from the network. This volatile behaviour does not allow the network to retain information between presentations of different patterns from the same class.

The class association matrices for small values of b_r are similar to those given by Table 5.6, showing correct classification. As b_r increases, the class association matrices become very large as the network struggles to classify the patterns from classes that occur infrequently (either regular or random). The persistent classes are classified correctly until b_r reaches a value of 0.8. At this point, the two iterations between each presentation of this class are too an interval long for the network to retain the information.

5.4.7.1 Test Summary

This test showed the effect of the Learning Rate (b_r) on the operation of the network. It was shown that b_r controls the effect of a given pattern on the network. A small b_r means that each pattern shown to the network has little effect on the probability of the links connected to the focus. A large value of b_r means that the network over-fits by altering the probability of the links to such an extent that links are broken and neurons associated with persistent classes are removed. The operating range of the b_r parameter was found to be between 0.001 and 0.05. This is a much smaller range than for its (E)PSOM counterpart (the Link Ageing Parameter) on a similar data set (PSOM Test 4).

5.4.8 Test 8: Plain English Vowel Pairs

The aim of this test to create a topology based on pairings of vowels as shown in Section 5.3. The letters were then exchanged for their numerical position in the alphabet (A=1, E=5, I=9, O=5, U=21). Gaps in the text were represented by '0'. These numbers were shown to the TPSOM.

For this test alone, the TPSOM required two alterations:

1. A node must be able to link itself to represent instances of the same vowel twice. This was achieved by a circular link which starts and ends at the same node.
2. The network needed to deal with character spaces. When a character space is presented to the network, the knowledge of the previous character seen is lost and the next character presented is dealt with as if it was the first character seen. This allows the character pairs to be distinct from each other.

Table 5.31 TPSOM Test 8 Parameters.

Parameter	Value
Node Building Parameter (a_n)	0.4
Link Removal Threshold (a_r)	0
Learning Rate (b_r)	0.005

The Node Building Parameter (a_n) was set arbitrarily. This could be easily set because it was known that the minimum distance between any two classes is 1. A

link removal threshold (a_r) of 0 was set to allow the network not to forget any of the text. The intention was for the TPSOM to represent the whole 9 chapters, rather than retain a representation of the most recent chapters. The optimal learning rate of 0.005 was found empirically (discussed below).

Figure 5.23 shows the TPSOM network representing the vowel pairs for the first 9 Chapters of Great Expectations. Each node is labelled with the vowel it represents for simplicity. The corresponding state transition matrix is given in Table 5.32. This state transition matrix should be compared to Table 5.1, which shows the correct values for the state transition matrix.

By inspection, it can be seen that the TPSOM has not managed to replicate the state transmission matrix shown in Figure 5.1. For example, the transition between E to E in Table 5.1 is 0.342, however, the TPSOM has shows this link as 0.0 in Table 5.32

.

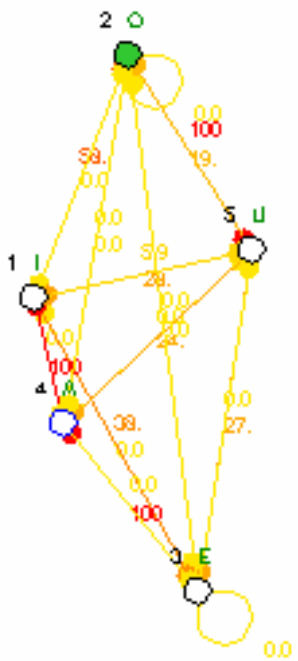

Figure 5.23 TPSOM Test 8: The TPSOM representing the vowel pairs for the first 9 chapters of Great Expectations.

Table 5.32 TPSOM Test 8: State Transition Matrix.

	A	E	I	O	U
A	0	0	1	0	0
E	1	0	0	0	0
I	0	0.36	0	0.581	0.059
O	0	0	0	0	1
U	0.245	0.274	0.289	0.192	0

The TPSOM diagram (Figure 5.23) shows that, at some point, there were instances of some vowel pairs but not regular enough for the TPSOM to represent that knowledge. These links are set to zero and should be ignored; normally, they would be removed but this features was switched off to facilitate the representation of a large, static dataset. While tuning the learning rate (b_r), a brief investigation of its effects was performed. For large learning rates (between 0.001 and 0.1), the network would only satisfactorily represent the most recent vowel pairs rather than the pairs of the entire dataset.

Those links that become zero, perhaps due to being infrequent in part of the text, will never move from being zero. This is because the link length is a multiplicand of the alteration value. This can be counter-acted by having the links removed and then replaced by new ones. This is the role of the link removal process; however, using this system reduces the network's ability to classify over a large number of iterations.

For small learning rates, the link probabilities do not change much from the initial link probabilities. Therefore, for very small values (such as 0.000001), training does not change the values from their initial node-built levels, the values of all the links being identical.

The reason for this problem is the coarse way in which a new link is added to the given node. This coarse process has the effect of losing much of the past knowledge stored in the list. A solution to this problem might be an alteration in the algorithm or better tuning of the learning rate.

5.4.8.1 Test Summary

This test showed that the TPSOM could only extract the main features of Great Expectations by representing the probability that one vowel followed another. Many of the statistically significant properties of the dataset were not captured by the TPSOM. This test demonstrated that when the number of data points that the TPSOM is required to represent is large, it is unable to correctly learn the data distribution of the input space. This is due a combination of tuning, the coarse link creation and the update algorithm. Improvements might include adding a parameter to the strength of the link creation algorithm, giving old links greater chance of survival.

5.5 Chapter Summary

In this chapter the TPSOM was tested on a number of data sets. First, the TPSOM was tested on artificial datasets to demonstrate its operation and sensitivity to its parameters. Secondly, it was used to explore the Radar dataset to see if any pattern could be found. As expected, none was found. Lastly, it was tested on vowel pairs taken from a long text where the statistical properties are known. The TPSOM could not replicate these properties, due to the size of the dataset and the course nature of the learning algorithm.