Article

Using data clustering as a method of estimating the risk of establishment of bacterial crop diseases

Michael J. Watts
School of Earth and Environmental Sciences, University of Adelaide, North Terrace, SA 5005, Australia
E-mail: mjwatts@ieee.org

Received 16 February 2011; *Accepted 1 March 2011*; *Published online 1 April 2011*
IAEES

Abstract
Previous work has investigated the use of data clustering of regional species assemblages to estimate the relative risk of establishment of insect crop pest species. This paper describes the use of these techniques to estimate the risk posed by bacterial crop plant diseases. Two widely-used clustering algorithms, the Kohonen Self-Organising Map and the k-means clustering algorithm, were investigated. It describes how a wider variety of SOM architectures than previously used were investigated, and how both of these algorithms reacted to the addition of small amounts of random 'noise' to the species assemblages. The results indicate that the k-means clustering algorithm is much more computationally efficient, produces better clusters as determined by an objective measure of cluster quality and is more resistant to noise in the data than equivalent Kohonen SOM. Therefore k-means is considered to be the better algorithm for this problem.

Keywords bacterial diseases; crop; risk; establishment; data clustering.

1 Introduction

There is a pressing need to identify which species of bacterial crop diseases pose the most risk to a particular geographical region. Identification of pre-emerged invasive diseases would allow more directed prevention or eradication and control if by chance the species were to establish. Computational modelling is one of the tools that can be used to assist decision makers in assessing the threat posed by various invasive species. Data clustering (Everitt et al., 2001) is an important and widely used method of data analysis and data mining that groups similar items together into subsets, or clusters, where each item in a cluster is more similar to the other items in the cluster than it is to items outside the cluster. By determining the cluster a geographical region, as represented by its species assemblage, belongs to, it is possible to infer which species may become established in that region. Worner and Gevrey (2006) and Gevrey et al (2006) were the first to use clustering with respect to invasive species, in this case insect pests. In their study a Kohonen Self-Organising Map (SOM, Kohonen (1990)), which is a type of artificial neural network (ANN, Kasabov (1996)) model was used to cluster insect species assemblages found in geographic regions. The rationale behind this approach is that regions that have climatic or other environmental properties in common are likely to have similar species assemblages (Worner and Gevrey, 2006). If a particular region in a cluster does not have an insect species present, yet that species is present in a large proportion of geographical regions that have highly similar assemblages (that is, the other regions in the cluster), then it is likely that the region possesses an environment conducive to the establishment of that species if it were introduced (Worner and Gevrey, 2006; Gevrey et al, 2006). A quantitative "risk

weighting" for each species was thus derived from the frequency at which each species appeared in a cluster, where a large risk weighting value for a species meant a large risk of that species establishing in the region, and the risk weighting of each species is the same for all regions in that cluster. SOM have been previously used in several ecological applications (Foody, 1999). These applications include modelling similar diatom distribution patterns across France (Park et al, 2006), characterisation of the spatial distribution of fish (Brosse, Grossman and Lek, 2007), analysis of the spatial distribution of invertebrates (Céréghino, Giraudel and Compin, 2001), and ecological community ordination (Giraudel and Lek, 2001). An advantage of Kohonen SOM is that they perform vector quantisation, that is, vectors are projected from their original high dimensional space into a lower dimensional space, which allows for easier visualisation of the resulting clusters.

In the paper Watts and Worner (2009) the results of clustering insect assemblages with Kohonen SOM were critically compared with the results of an alternative clustering algorithm, the k-means algorithm (Lloyd, 1982). Although this work found that k-means was in some ways superior to SOM, several issues were left unaddressed. These issues were as follows:

1. The application of the method beyond insects was not investigated
2. The effect of noise (small random changes) in assemblages for each algorithm was not investigated
3. Only one size of SOM was investigated

The work reported in this paper addresses these issues. It compares the performance of a range of SOM map sizes with the performance of equivalent k-means algorithms over assemblages of bacterial crop diseases. It also investigates the effects of adding noise to the assemblages. The following research questions are investigated:

1. Which algorithm produces the 'best' clusters?
2. What effect does adding varying levels of noise have on clustering performance?
3. Which species of bacterial crop disease, that are not already listed as established, have the greatest risk of establishment for New Zealand?

2 Method

2.1 Data

Data was sourced from the CABI Crop Protection Compendium 2003 (CABI, 2003). This comprised the presence and absence of 114 bacterial crop disease species within 459 geographical regions listed within the compendium. These regions represent all of the world's landmass. While there were more than 114 disease causing bacterial species listed in the compendium, only those species recorded as present in more than 5% of the geographic regions were retained. There were no empty regional assemblages, that is, no vectors in the data set consisted entirely of absences.

A histogram of the prevalences of each species is presented in Fig. S.1 (supplementary material). This shows a log-normal distribution, with a large number of species with relatively low prevalence, and only a few of greater prevalence. As the technique used here is data-driven and utilises the associations between regional assembles, it was important to verify that the species assemblages were not random. This was done via a null model analysis (Gotelli, 2000). The software used for this analysis was EcoSim version 7 (Gotelli and Entsminger, 2006). Ten thousand iterations of the independent swap, sequential swap and random knight's tour algorithms were performed and the C-score, V-ratio, number of checker boards and number of species combinations were all evaluated. The default setting of retaining degenerate matrices was retained. The results of each run are presented in Table S.1 (supplementary material). These show that the assemblages were significantly non-random (p=0.001).

2.2 Kohonen self-organising maps

The Kohonen Self-Organising Map (SOM, Kohonen (1990)) is an artificial neural network that learns representations of data via an unsupervised learning algorithm. That is, while many other ANN learn to model target data, the SOM learns the patterns within the data itself. It consists of two layers of artificial neurons: the input layer, which accepts the external input signals, and the output layer (also called the output map), which is usually arranged in a two-dimensional structure. The structure of the Kohonen SOM is shown in Figure S.2 (supplementay material). Every input neuron is connected to every output neuron, and each connection has a weighting value attached to it. When an input vector is presented to the SOM, the Euclidean distance between the input vector and the incoming weight vector of each output map neuron is calculated. The output neuron with the smallest distance is declared the winner (This is also known as the Best Matching Unit (BMU)). SOM learning is an iterative process, during which training examples are propagated through the network, and connection weights modified according to equation (1).

$$w_{i,j}(t+1)=w_{i,j}(t)+h(t)(x_i-w_{i,j}(t)) \qquad (1)$$

where $w_{i,j}(t)$ is the connection weight from input i to map neuron j at time t, x_i is element i of input vector x, and h is the neighbourhood function, as defined in equation (2).

$$h(t)=\alpha \exp(-d^2/(2\sigma^2(t))) \qquad (2)$$

where α is the learning rate, which decays towards zero as time progresses, d is the Euclidean distance between the winning unit and the current unit j, and σ is the neighbourhood width parameter, which also decays towards zero.

The SOM algorithm is essentially a clustering algorithm that will assign training examples to neurons, where each neuron is equivalent to the centre of one cluster. The winning neuron for each vector thus determines which cluster the vector belongs to.

The SOM simulator used was custom written in C++. The Euclidean distance measure was used and the output map was laid out in a rectangular configuration. In Worner and Gevrey (2006) a SOM output map of 108 neurons was used in a rectangular map of nine by twelve neurons, where the size was determined by equation (3):

$$n=5v^{0.5} \qquad (3)$$

where n is the number of output map neurons and v is the number of training vectors. In this work, to address the third criticism listed in Watts and Worner (2009), a wider range of map dimensions was used. The dimensions of the output maps used are listed in Table S.2 (supplementay material) while the number of training epochs was calculated as 500 times the number of output map neurons, as recommended in Kohonen (1997). The map sizes ranged from what was considered to be the smallest usable map with two by four neurons, up to nine by twelve neurons, which was the size suggested by equation 3 and was the size used in Watts and Worner (2009) and Worner and Gevrey (2006).

2.3 k-means clustering

The k-means algorithm (Lloyd, 1982) belongs to a family of algorithms known as optimisation clustering algorithms. In this family of algorithms, clusters are formed such that some criterion of cluster goodness is optimised. That is, the examples are partitioned into clusters such that the clusters are optimal according to

some measure. The name comes from the fact that k clusters are formed, where the centre of the cluster is the arithmetic mean of all vectors within that cluster.

The k-means algorithm is as follows:

1. Select k seed examples as initial centres (randomly generated vectors can also be used).
2. Calculate the distance from each cluster centre to each example.
3. Assign each example to the nearest cluster.
4. Calculate new cluster centres, where each new centre is the mean of all vectors in that cluster.
5. Repeat steps 2-4 until a stopping condition is reached.

In the experiments reported here, the initial centres were vectors that were randomly selected from the data set, and the stopping criterion was based on the movement of the cluster centres: when vectors no longer changed clusters between iterations (the clusters had stabilised), the algorithm terminated. The number of clusters was set equal to the number of SOM output map neurons that were evaluated.

The disadvantage of k-means compared to SOM is that it does not perform vector quantisation, that is, it does not naturally result in a form that can be easily visualised. The advantage of k-means over SOM is that it is more computationally efficient and can thus run much faster.

2.4 Generating risk lists

The goal of the experiments reported here was to generate lists of bacterial crop disease species that were ordered according to the risk they pose to the target region, which in this case was New Zealand. The same method was used to generate these lists from the clusters that were generated by SOM and the clusters that were generated by k-means.

As was discussed in the Introduction, the fundamental assumption made in this paper is that regions that have similar species assemblages have similar environmental conditions that encourage or discourage establishment of certain species. The algorithm for finding risk rankings from clusters is based on that assumption, and is as follows:

- for each repetition
    - Find the cluster the target region is in (the target cluster).
    - Identify all regions that are in the target cluster (the neighbour regions). These assemblages form the target matrix.
    - Calculate the frequency each species appears in the target matrix (the risk factors).
    - Use these frequencies to calculate the ranks of each species, where higher frequency species rank more highly than species with lower frequencies.
- Calculate the mean and standard deviation of the ranks for each species.
- Order species by their mean ranks.

Thus, by this algorithm, species that frequently appear in regions that are similar to the target region are given a higher rank than species that do not. Rankings are assigned in descending order. Species with the same risk values are given an average ranking. For example, if three species with the same risk values are ranked 18, 17 and 16, each will be given the rank of 17, and the following species assigned the rank 15 (unless that species also shares a risk value with other species).

Risk lists were also generated directly from SOM weights, as was done for insects in (Worner and Gevrey, 2006; Gevrey et al, 2006; Watts and Worner, 2009), whereby the connection weight associated with each input (species) is taken as the species risk weighting. The rationale behind this approach was that the weights of the SOM represent cluster centres. Note that risk lists were not generated from the centres of k-means clusters, because the centre of a k-means cluster is the same as the mean of all items in that cluster.

2.5 Measuring cluster quality

There are many measures of cluster quality (Hansen and Jaumard, 1997), most of which are based on the distance between cluster centres or the distance between items in the cluster and items outside the cluster. We used the quantisation error, which is the mean distance between each vector and the centre of its cluster, to measure the quality of each clustering run. This measure, which is described in equation (4), yields a single value for all clusters, as opposed to other measures listed in Hansen and Jaumard (1997) that yield a measurement for each cluster. For this measure, a lower score is better.

$$Q=\sum d_{i,c}/v \qquad (4)$$

where Q is the quantisation error, $d_{i,c}$ is the Euclidean distance between vector i and the centre c of the cluster it belongs to, and v is the number of vectors in the data set.

The computational efficiency of each algorithm was also considered, that is, which algorithm performed the task the fastest. It is well-known that k-means is faster than SOM, especially over the relatively large number of variables used in this problem. This is because the number of calculations required in each iteration of k-means is less, and the number of iterations tends to be much smaller (Kasabov, 1996).

The variation of results between runs was considered by measuring the Jaccard distance between each unique pair of neighbour vectors, where a neighbour vector had a one for each region that was in the target region's cluster, and a zero in all other positions. That is, for each run, a 459-element vector was constructed where each element corresponded to one of the regions being clustered (one element for each region). A one was entered in that element if the corresponding region was present in the same cluster as the target, that is, if the region was a neighbour region. At the completion of all trials for that experiment, the Jaccard distance was measured between each unique pair of vectors and the mean and standard deviation of distances found. This directly measured the variation between the clustering runs in terms of the contents of the clusters. The Jaccard similarity was used because it is a simple and well-known measure of similarity between two binary vectors. To further compare the results of SOM and k-means, the Euclidean distance was measured between both the species risk weightings and regional neighbour frequencies produced by each algorithm. The distance between the species risk weightings were measured to determine the similarity of risk assigned to the species by each algorithm. The distance between the neighbour frequencies was measured because the final risk weightings are determined by the frequency at which other regions are clustered with the target region.

2.6 Adding noise

Noise was added by randomly changing a presence to an absence, or vice versa. Noise was added either to the target assemblage, or to the entire assemblage matrix. When adding noise to the entire matrix, the assemblages that the noise was added to were randomly selected. The number of ‘bits’ of noise (the number of presences or absences that were flipped) varied: when adding noise to the target assemblage only, between one and sixteen bits were flipped. When adding noise to the entire matrix, the number of bits ranged from one to sixteen and from 459 to 7344. The second range was calculated so that the proportion of bits flipped in the entire matrix was equal to the proportion of bits flipped in the target row, when only that row had noise added. This was done so that the affect of the overall proportion of noise could be investigated.

3 Results

Table 1 presents the results of measuring the cluster quality for Kohonen SOM. Although there are no trends clearly apparent, the minimum quantisation error of 3.01 was found for the 4×6 map size. Two things about the numbers presented in this table are striking: firstly, the values of the Jaccard distance between neighbour

vectors, which were zero up until the 8×12 output map; secondly, the values of the standard deviations for the other measures, which were also zero up until the 8×12 output map. This shows that for output map sizes less than 8×12 neurons, there was no variation between runs. In other words, the results were identical for each run of the SOM algorithm, up to a map size of 8×12 or larger[1]. The variation that arose at the larger map sizes was large: the coefficient of variation (CV) for the quantisation error ranged only from 10.62 to 12.17%, the CV for target size ranged from 36.13% for the 9×12 map to 52.72% for the 8×12 map. The differences between the risk weights derived from connection weights and those derived from clusters varied but were generally higher for the larger maps. As the 4×6 map size yielded the lowest quantisation error, this configuration was used for the following experiments. Also, the difference between the species weightings derived from connection weights and the species weightings derived from clusters was lowest for the 4x6 map. For comparison purposes, therefore, twenty four clusters were selected for the k-means clusters.

Table 1 Results for Kohonen SOM and k-means. 'Quantisation Error' is the mean and standard deviation of the cluster quantisation error. 'Target Size' is is the mean and standard deviation of the number of assemblages in the target cluster. 'Jaccard' is the mean jaccard distance between each unique pair of neighbour vectors, where a neighbour vector has a one for each region that is in the target region's cluster, and a zero in all other positions. 'Weight Diff.' is the difference (Euclidean distance) between the species mean weightings as determined from the weights and species mean weightings as determined from the weights and species mean weightings as determined from the clusters.

SOM				
Map Size	Quantisation Error	Target Size	Jaccard	Weight Diff.
2×4	3.99/0	18/0	0/0	5.73
3×5	3.90/0	4/0	0/0	5.82
4×6	3.01/0	9/0	0/0	5.45
5×7	3.9/0	4/0	0/0	6
6×9	3.91/0	3/0	0/0	5.91
7×11	3.10/0	5/0	0/0	5.81
8×12	3.39/0.36	4.59/2.42	0.67/0.27	6.03
9×12	3.45/0.42	3.1/1.12	0.59/0.29	6.08
k-Means				
Clusters	Quantisation Error	Target Size	Jaccard	
8	2.29/0.02	23.29/8.51	0.64/0.20	
15	2.18/0.02	14.66/5.60	0.70/0.19	
24	2.09/0.01	11.14/4.57	0.78/0.17	
35	2.02/0.01	8.76/4.05	0.79/0.17	
54	1.92/0.02	6.87/3.14	0.77/0.19	
77	1.81/0.02	5.35/2.66	0.77/0.19	
96	1.71/0.03	4.34/2.28	0.74/0.20	
108	1.65/0.03	3.84/2.14	0.70/0.22	

Table 1 also presents the results of measuring the cluster quality for k- means. Here it can be seen that the quantisation error and target cluster size all steadily decrease as the number of clusters increases, while the Jaccard measurement increases. Whereas there was zero variation in these measures for the SOM, the CV over the quantisation errors for k-means ranged from 0.48% for twenty-four clusters to 1.82% for 108 clusters. The CV over the target cluster size went from 36.54% for eight clusters to 55.73% for 108 clusters. The CV of the

[1]Although this was not investigated in Watts and Worner (2009), the criticism therein of Worner and Gevrey (2006) using only one trial is borne out, as a 9×12 output map was used there.

target cluster entropy ranged from 78.12% for twenty-four clusters to 119.98% for 108 clusters, while the CV of the Jaccard measure ranged from 21.52% for thirty-five clusters to 31.28% for eight clusters. Thus, while the mean performance measures for SOM were not as good as for k-means, k-means yielded much more variation.

Table 2 Region neighbour frequencies for Kohonen SOM and k-means.

SOM		k-means	
Region Name	Frequency	Region Name	Frequency
New Zealand	1	New Zealand	1
Romania	1	United Kingdom	0.52
Victoria	1	Romania	0.49
Western Australia	1	France	0.47
Bulgaria	1	Italy	0.45
Russian Federation	1	Germany	0.42
Ontario	1	Netherlands	0.39
Canada	1	Victoria	0.36
Colombia	1	Western Australia	0.34
		Greece	0.33
		Switzerland	0.32
		South Australia	0.29
		Canada	0.29
		Bulgaria	0.28
		Hungary	0.28
		Iran	0.28
		New South Wales	0.27
		Denmark	0.26
		Spain	0.24
		Yugoslavia	0.24
		Queensland	0.21
		USA	0.21
		Israel	0.2
		Russian Federation	0.2
		Poland	0.19
		South Africa	0.19
		Australia	0.17
		Turkey	0.17
		Ontario	0.16
		Egypt	0.16
		Japan	0.14
		Zimbabwe	0.14
		Austria	0.13
		Tasmania	0.13
		India	0.12

The top eighty ranked species, in terms of threat posed to New Zealand, as determined by SOM, are presented in Table S.3 (supplementary material). These risk values were derived from 4×6 SOM clusters, rather than weights, so that a direct comparison could be performed with the results of k-means. As there was no variation between trials of the SOM, the standard deviation of the risk values in this table is zero. Species that were recorded as already present in New Zealand dominate the top of the list, while non-established species are present lower down the list. The top eighty ranked species as determined by k-means are presented in Table S.4 (supplementay material). As there was variation between the runs of this algorithm, there was

variation in the species risk weightings. Species that had a lower risk weighting were observed to have larger variation in those weightings. This was expected as species that had low weightings would also have been of lesser frequency in the data set.

The regional neighbour frequencies, that is, the frequency with which regions appeared in the target cluster, are listed in Table 2 for SOM and k-means. The results in this table again show that there was no variation at all between runs for the SOM algorithm, with exactly the same regions being mapped into the target cluster each time. This lack of variation explains why there were only nine regions in the cluster, including the target. There are thirty-five regions listed including New Zealand for k-means. As there was much more variation from the k-means algorithm, there was a much greater range of frequencies of appearance for these neighbouring regions.

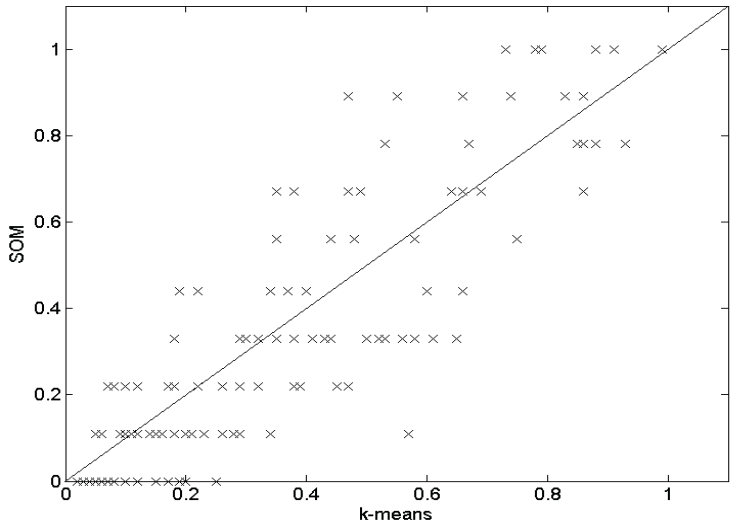

Fig. 1 Species risk weightings derived from SOM vs risk weightings derived from k-means for 4×6 map/24 clusters. The line indicates 1:1 agreement between the two algorithms.

The results of comparing the risk weightings from SOM clusters with the risk weightings from k-means are presented in Table S.5 (supplementary material). A plot of the mean risk values derived from a 4×6 SOM versus the mean risk values from 24-cluster k-means is presented in Fig. 1, where a strong similarity can be seen. The correlation coefficient of these two series was 0.8681, which indicates a strong level of similarity between the two algorithms. That is, in terms of mean risk weightings, the two algorithms produced highly similar results, despite the large difference in speed of the two algorithms.

3.1 Adding noise to the target assemblage

Table 3 presents the results of adding noise to the target region assemblage only. It is immediately apparent that adding any noise, even changing one presence to an absence or vice-versa, is enough to disrupt the SOM clustering process. Target cluster size for both SOM and k-means was significantly smaller (two-tailed *t*-test, p =0.001) after adding noise, but quantisation errors were not significantly changed (which is expected, as the number of bits that were changed were a small proportion of the bits in the overall clusters). However, for the SOM the CV of the quantisation errors altered from zero (for the 4x6 map with pristine data) to a minimum of 10.38% (for two bits of noise) and a high of 14.47% (for one bit of noise). The CV of the target cluster size ranged from 43.99% for seven bits of noise to 99.87% for 14 bits of noise, while the CV of the Jaccard distance measured ranged from 23.08% for fourteen bits to 40.85% for one bit. This shows that while adding

any noise is enough to cause a significant disruption to the clustering process, this disruption was not necessarily greatly increased by adding more noise. That is, adding any noise disrupted the SOM clustering, but more noise did not disrupt it any further.

The results for the k-means showed a resistance to disruption: the CV of the quantisation error was 0.48% for all levels of noise, while the CV of the target size ranged from 37.13% (for eight bits of noise) to 50% (for sixteen bits), and the CV for the Jaccard measure ranged from 18.52% (for ten bits) to 23.68% (for three bits). These variations were not significantly larger than those displayed by the k-means carried out over the initial data set, which suggests that the k-means algorithm was more resistant to noise than the SOM.

Table 3 Results of adding noise to target assemblage for Kohonen SOM and k-means. 'Bits' is the number of presences or absences that were flipped. Other column labels are as in Table 1.

SOM				
Bits	Quantisation Error	Target Size	Jaccard	Weight Diff.
1	3.18/0.46	7.56/3.79	0.71/0.29	6.25
2	3.18/0.33	7.27/3.5	0.74/0.25	5.77
3	3.23/0.35	6.29/3.25	0.73/0.22	5.81
4	3.33/0.4	6.66/3.49	0.75/0.21	5.96
5	3.23/0.34	6.35/3.37	0.75/0.21	5.85
6	3.35/0.45	6.3/3.5	0.75/0.2	6.06
7	3.29/0.41	6.91/3.04	0.75/0.21	5.92
8	3.26/0.41	6.81/3.66	0.75/0.21	5.91
9	3.30/0.37	6.94/4.16	0.75/0.22	5.94
10	3.28/0.37	6.93/3.55	0.76/0.2	5.83
11	3.36/0.49	6.7/5.11	0.76/0.19	5.72
12	3.31/0.38	6.18/4.85	0.77/0.19	5.54
13	3.41/0.46	6.4/3.9	0.77/0.19	5.78
14	3.34/0.41	7.56/7.55	0.78/0.18	5.43
15	3.34/0.39	6.3/4.45	0.77/0.19	5.68
16	3.32/0.39	7.17/5.58	0.78/0.19	5.52
k-means				
Bits	Quantisation Error	Target Size	Jaccard	
1	2.09/0.01	10.97/4.48	0.78/0.18	
2	2.09/0.01	11.19/4.91	0.79/0.17	
3	2.09/0.01	10.68/4.93	0.76/0.18	
4	2.10/0.01	10.45/4.44	0.79/0.17	
5	2.1/0.01	10.5/4.45	0.8/0.17	
6	2.1/0.01	10.04/3.96	0.79/0.16	
7	2.1/0.01	10.72/4.57	0.79/0.17	
8	2.09/0.01	10.75/3.99	0.8/0.16	
9	2.1/0.01	10.49/5.0	0.80/0.16	
10	2.09/0.01	10.14/4.3	0.81/0.15	
11	2.09/0.01	9.96/4.22	0.81/0.16	
12	2.09/0.01	11.17/4.9	0.8/0.17	
13	2.1/0.01	10.25/5.18	0.8/0.18	
14	2.1/0.01	10.81/5.07	0.81/0.16	
15	2.1/0.01	9.21/4.08	0.81/0.17	
16	2.1/0.01	10.44/5.22	0.81/0.16	

3.2 Adding noise to the entire matrix

Table 4 presents the results of adding noise to the entire matrix. The quantisation errors of the SOM showed a steady increase as the amount of noise increased, although it was not greater than that in Table 3. The variation

over the CV decreased with noise: the maximum CV over quantisation error was 15.74% for five bits of noise, while the minimum was 4.11% for 7344 bits. The target cluster size showed much more disruption, with CV ranging from 48.19% for three bits to 151.97% for 7344 bits. The quantisation errors of the k-means are again comparable to those in Table 3, with the variation again staying quite low, ranging from a low of 0.42% for 459 bits of noise to 1.06% for 6426 bits.

The variation over the target cluster sizes ranged from 33.04% for nine bits of noise to 66.78% for 6885 bits. Again, these results showed that k-means was more resistant to noise than SOM.

Table 4 Results of adding noise to the entire matrix for Kohonen SOM and k-means. Column headings are as for Table 3.

SOM			
Bits	Quantisation Error	Target Size	Jaccard
1	3.28/0.37	7.23/3.77	0.73/0.25
2	3.31/0.39	7.1/3.52	0.72/0.23
3	3.32/0.38	7.45/3.59	0.73/0.22
4	3.36/0.46	6.56/3.81	0.74/0.22
5	3.43/0.54	6.22/3.55	0.73/0.21
6	3.35/0.4	6.66/4.06	0.74/0.2
7	3.37/0.43	5.98/3.48	0.73/0.21
8	3.35/0.4	7/3.99	0.74/0.21
9	3.34/0.43	6.5/3.89	0.75/0.21
10	3.34/0.38	6.57/3.46	0.73/0.19
11	3.39/0.4	6.69/3.87	0.74/0.21
12	3.37/0.40	7.15/4.13	0.73/0.2
13	3.34/0.39	6/3.62	0.74/0.2
14	3.34/0.39	6.37/3.79	0.74/0.2
15	3.33/0.46	5.84/3.60	0.73/0.21
16	3.36/0.45	6.94/3.72	0.74/0.19
459	3.46/0.356	7.47/3.76	0.74/0.2
918	3.56/0.33	6.49/3.64	0.76/0.17
1377	3.71/0.32	6.87/4.04	0.76/0.17
1836	3.84/0.31	7.42/4.18	0.76/0.17
2295	3.93/0.29	7.1/4.22	0.76/0.16
2754	4.06/0.29	7.57/4.08	0.77/0.15
3213	4.13/0.27	7.53/3.99	0.78/0.14
3672	4.23/0.27	7.77/4.24	0.79/0.14
4131	4.27/0.24	7.35/4.15	0.79/0.14
4590	4.4/0.24	7.26/4.2	0.78/0.15
5049	4.47/0.24	7/3.72	0.79/0.13
5508	4.53/0.23	8.12/4.23	0.8/0.12
5967	4.62/0.22	7.72/4.05	0.8/0.13
6426	4.73/0.22	7.02/4.84	0.81/0.13
6885	4.80/0.21	7.5/5.03	0.82/0.12
7344	4.87/0.2	12.95/19.68	0.86/0.11

k-means			
Bits	Quantisation Errors	Target Size	Jaccard
1	2.09/0.01	10.76/4.13	0.78/0.18
2	2.1/0.01	10.93/4.8	0.76/0.18
3	2.1/0.01	11.11/4.12	0.76/0.19
4	2.1/0.01	11.79/4.75	0.78/0.18
5	2.1/0.01	10.62/3.85	0.79/0.18

6	2.1/0.01	11.31/4.64	0.79/0.17
7	2.1/0.01	10.4/4.48	0.78/0.17
8	2.1/0.01	10.97/3.9	0.78/0.18
9	2.1/0.01	10.08/3.33	0.76/0.19
10	2.1/0.01	11.32/4.51	0.78/0.17
11	2.1/0.01	10.4/3.65	0.77/0.18
12	2.1/0.01	11.72/5.31	0.78/0.17
13	2.1/0.01	10.82/4.2	0.77/0.18
14	2.11/0.01	10.92/4.14	0.77/0.19
15	2.1/0.01	10.54/4.35	0.79/0.16
16	2.11/0.01	10.83/4.4	0.77/0.18
459	2.36/0.01	10.06/4.02	0.81/0.15
918	2.54/0.01	11.53/5.02	0.77/0.18
1377	2.7/0.02	11.19/4.43	0.8/0.15
1836	2.84/0.02	11.1/5.15	0.81/0.14
2295	2.96/0.02	11.95/6.28	0.81/0.13
2754	3.08/0.02	11.08/5.47	0.83/0.12
3213	3.18/0.03	10.69/5.83	0.83/0.13
3672	3.28/0.02	11.73/5.97	0.82/0.13
4131	3.39/0.03	11.8/6.55	0.83/0.11
4590	3.47/0.03	12.23/6.83	0.81/0.12
5049	3.55/0.04	14.01/7.47	0.82/0.12
5508	3.63/0.03	13.91/8.46	0.83/0.11
5967	3.71/0.04	13.82/8.27	0.83/0.12
6426	3.77/0.04	14.13/8.89	0.85/0.1
6885	3.83/0.04	14.54/9.71	0.85/0.1
7344	3.91/0.04	13/17.04	0.88/0.1

4 Discussion

The first research question investigated was which algorithm produced the best clusters over this data set. There are two criteria to consider when determining which algorithm produced the best clusters. Firstly, which algorithm produced clusters with the best objective measures of cluster quality? Secondly, which algorithm was the most consistent in terms of the regions that were clustered together?

The quantisation errors were lower for k-means than for Kohonen SOM. There was however, a complete absence of variation in the clusters for SOM output maps smaller than 8x12. That is, while k-means produced better clusters, SOM produced clusters that were identical for every run. Even with randomly initialised connection weights, the SOM algorithm still produced clusters that were consistent between runs. So, k-means was the best algorithm in terms of the clusters it produced, but SOM was the best in terms of consistency. This means that the judgement of which algorithm was the best overall must be made with consideration of the effects of adding noise to the assemblages.

The second research question investigated in this work was concerned with the effect of adding noise (randomly changing presences to absences or vice versa) to the assemblage data. It was shown that adding noise to the species assemblage vectors caused a disruption to the clustering for both algorithms. This disruption was most pronounced for the Kohonen SOM compared to k-means. Changing a single species presence flag (changing either a presence to an absence, or an absence to a presence) caused a disruption of the clustering process. The variation to the clustering caused by this disruption, as measured by the Jaccard coefficient between neighbour vectors, was the same as the variation shown by k-means before the addition of noise. That is, adding a single bit of noise was sufficient to cause as much variation in the SOM as was apparent in k-means. Disruption of the k-means algorithm was less apparent, with there being very little difference between the results produced by clustering noisy and noiseless data. For both algorithms, there was

no apparent difference between adding noise to the target assemblage vector (which meant that the noise was added to a single vector between runs) and adding noise to the entire assemblage matrix (which meant that the noise was randomly distributed throughout the matrix) when the number of bits was equivalent. Only when the number of bits of noise approached a large fraction of the total matrix did the disruption greatly increase. Overall, while the SOM algorithm was vulnerable to the addition of noise, the k-means algorithm was more resistant to its effects. When combined with the results of the cluster quality measures, this leads to the conclusion that k-means was the better algorithm for this problem for this data set, especially when the much greater speed and simplicity of the k-means algorithm is considered.

As determined by the k-means algorithm, the five bacteria species or groups which are not recorded as being present in New Zealand, and pose the greatest threat as ranked by their risk weightings, are*: Xanthomonas hortorum pv. pelargonii*; *Pseudomonas syringae pv. morsprunorum*; aster yellows phytoplasma group; *Xanthomonas axonopodis pv. malvacearum*; and pear decline phytoplasma. All of these had a risk weighting greater than 0.5. The following were also in the top five non-established species as determined by the Kohonen SOM: aster yellows phytoplasma group; *Xanthomonas axonopodis pv. malvacearum*; *Xanthomonas hortorum pv. pelargonii*. The risk weightings of these were all above 0.6. As they are very common in regions with similar species assemblages as New Zealand, as determined by two differnt algorithms, these five species are considered to pose a significant threat of establishment within New Zealand.

The most significant advantage offered by the SOM algorithm is the ability to easily visualise the clusters that result. This is because SOM perform vector quantisation, that is, they reduce the dimensionality of the vectors into the two- dimensional space of the output map. While methods of quantising the clusters that result from k-means exist, such as Sammon projection (Sammon, 1969), it is desirable to produce a single visualisation across all trials. This, however, is an open research issue, as the variation in cluster contents between trials of the k-means algorithm complicates this process.

A challenge with a study of this type lies in verifying the predictions made. It is neither ethical nor desirable to release plant pathogens into the New Zealand agricultural system just to see if they will establish: thus, it is not possible to perform controlled experiments to verify these predictions. Although the k-means algorithm is more resistant to noise than SOM, as a data-driven technique it is still vulnerable to problems with the data. If there are wide-spread problems with the data used to construct the clusters, then the results will be unreliable. These problems can include mis-identification of species and unrecorded results of successful control or eradication campaigns against established species.

Future work will further investigate the potential of establishment of the five diseases named above via the construction of models that relate regional environmental variables to regional species presence or absence. Additional clustering algorithms, such as the Evolving Clustering Method (Song and Kasabov, 2001) will also be investigated.

5 Conclusions

The work reported in this paper has shown that, for the task of predicting the invasiveness of bacterial crop diseases, the k-means clustering algorithm is better than the Kohonen Self-Organising Map algorithm. It was shown that k-means produced better clusters, as measured by an objective cluster measure, and was more resistant to the effects of noise in the data. While the SOM produced more consistent clusters at smaller output map sizes, and made it easier to visualise the resulting clusters, k-means is considered to be the best algorithm for this task, at this time.

Acknowledgements

Data from the Crop Protection Compendium was used with permission of CAB International, Wallingford, UK.

References

Brosse S, Grossman GD, Lek S. 2007 Fish assemblage patterns in the littoral zone of a European reservoir. Freshwater Biology, 52: 448-458

Crop Protection Compendium. 2003. Global Module(5th Edition). CAB International, Wallingford, UK

Céréghino R, Giraudel JL, Compin A. 2001. Spatial analysis of stream invertebrates distribution in the Adour-Garonne drainage basin (France), using Kohonen self organizing maps. Ecological Modelling, 146: 167-180

Everitt BS, Landau S, Leese M. 2001. Cluster Analysis (Fourth Edition). Arnold Publishing, London

Foody GM. 1999. Applications of the self-organising feature map neural network in community data analysis. Ecological Modelling, 120: 97-107

Gevrey M, Worner S, Kasabov N, Pitt J, Giraudel J-L. 2006. Estimating risk of events using SOM models: A case study on invasive species establishment. Ecological Modelling, 127: 361-372

Giraudel JL, Lek S. 2001. A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination. Ecological Modelling, 146: 329-339

Gotelli NJ. 2000. Null model analysis of species co-occurrence patterns. Ecology, 81(9): 2606-2621

Gotelli N, Entsminger G. 2006. EcoSim: Null models software for ecology (Version 7). Acquired Intelligence Inc. & Kesey-Bear. Jericho, VT 05465

Hansen P, Jaumard B. 1997. Cluster analysis and mathematical programming. Mathematical Programming, 79: 191-215

Kasabov NK. 1996. Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering. MIT Press, MIT

Kohonen T. 1990. The self-organizing map. Proceedings of the IEEE, 78(9): 1464-1479

Kohonen T. 1997. Self-Organizing Maps. Springer

Lloyd SP. 1982. Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2): 129-136

Park Y-S, Tison J, Lek S, Giraudel J-L, Coste M, Delmas F. 2006. Application of a self-organizing map to select representative species in multivariate analysis: A case study determining diatom distribution patterns across France. Ecological Informatics, 1: 247-257

Sammon JW. 1969. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers, 18(5) : 401-409

Song Q, Kasasbov NK. 2001. ECM, a novel on-line, evolving clustering method and its applications. Proceedings of the 5th Biannual Conference on Artificial Neural Networks and Expert Systems

Watts MJ, Worner SP. 2009. Estimating the risk of insect species invasion: Kohonen self-organising maps versus k-means clustering. Ecological Modelling, 220(6): 821-829

Worner SP, Gevrey M. 2006. Modelling global insect pest species assemblages to determine risk of invasion. Journal of Applied Ecology, 43: 858-867