CLUSTERING MICROARRAY DATA USING THE SELF ORGANISING OSCILLATOR NETWORK

L. B. Jack and A. K. Nandi

Signal Processing and Communications Group, Department of Electrical Engineering and Electronics,
The University of Liverpool, Brownlow Hill, Liverpool, L69 3GJ

1. ABSTRACT

Clustering algorithms belong to an area of research that has many practical uses. Over the years, many different clustering algorithms have been proposed. Of these, the majority that are in common use today tend to be based on mathematical techniques which utilise the density of the data in data space. This has advantages for many scenarios, however there are occasions where density based clustering algorithms may not always be the most appropriate choice.

The Self-Organising Oscillator Network (SOON) is a comparatively new clustering algorithm [1], that has received relatively little attention so far. The SOON is distance based, meaning that clustering behaviour is different in a number of ways that can be beneficial. This paper examines the performance of the SOON with a biological dataset taken from microarray experiments on the Cell-cycle of yeast. The SOON is shown to be a useful addition to the available clustering algorithms, being able to highlight small (but potentially significant) clusters of interest in a dataset.

2. INTRODUCTION

Recently, there has been much interest in the development of techniques suitable for unsupervised clustering. The Kohonnen Self Organising Map (SOM) has been perhaps one of the most popular unsupervised clustering algorithms, used in many different applications [2, 3, 4]. Other alternatives include K-means clustering [5] and K-medioids [6].

Most of these techniques are density based; that is, the method by which cluster centres are chosen is based in some way around the density distribution of the data in dataspace. The practical upshot of density based approaches is that the centres chosen by the clustering techniques map the distribution of the data very closely, however they do not map the distributions of data in areas where the density of the data is low. With distance based clustering techniques, clusters are determined by a distance parameter. For any given centre point, all data points within a set distance will be regarded as members of the cluster.

The SOON algorithms are distance based, which means that they can select the number of clusters to match the data, rather than attempting to fit the data to a predetermined number of clusters.

3. BACKGROUND

Clustering algorithms are becoming more common as tools for use in many different real-life applications. In well bounded and understood datasets, it is relatively easy to determine how many different clusters should be found. Once this number of clusters is known, then it is relatively trivial to perform the clustering and interpret the results. However, when data is either noisy or not easily separable (which is often the case with many different real-world datasets, particularly of biological origin), it can be much more difficult to determine with confidence the "correct" number of centres for the clustering algorithm.

Cluster validation techniques can be used to attempt and determine the best match for this, however in many cases there is not necessarily a correct answer, and judging the quality of the clustering results becomes something of an abstract problem. Most clustering algorithms that find a predetermined number of clusters work on the basis of the density distribution of the data in dataspace. Centres are placed at locations which correspond to the most dense areas of the data in dataspace, creating a "gravity" effect, where high numbers of datapoints in one area of dataspace will tend to attract one or more cluster centres in an attempt to map this distribution. Areas which are sparsely populated will tend to be neglected, as there are not enough points in the vicinity to create sufficient "gravity" to attract a cluster centre towards them.

The SOON differs from density based methods in the way that the algorithm uses to determine what constitutes a cluster. Simply expressed, the SOON defines a distance, δ_0, which acts as the determinant of the cluster. From a given point, any other points which fall within the distance δ_0 are regarded as belonging to the same cluster, and will gradually synchronise their phase values (and hence firing times).

4. THEORY

The SOON algorithm, first proposed by Frigui *et al* [1], has its roots in a number of different biological processes that share the same physical characteristics. Fireflies flash at random when observed individually; however, when in large groups, the fireflies exhibit the characteristic of firing together when in groups that are phyiscally close to each other. Groups which are separated by distance will fire as disparate groups, each synchronised within itself. Heart pacemaker cells also share a similar behaviour, along with the menstruation cycles of groups of women in close proximity to each other. This behaviour, of self-organisation of components with an oscilliatory nature, gives rise to the name of the algorithm - the Self Organising Oscillator Network (**SOON**).

4.1 Oscillator basics

The basic unit of clustering under the SOON algorithm is the oscillator. Mathematically, an oscillator is defined using the equation

$$f_i(\phi) = \frac{1}{b}\ln[1 + (e^b - 1)\phi] \qquad (1)$$

where b is a constant value used to control the curve of the oscillator. The output of the oscillator, x_i is bounded in the range $[0,1]$, for all values of $f_i(\phi)$. This is achieved using a limiting function

$$B(x) = \begin{cases} x & \text{if} \quad 0 \leq x \leq 1 \\ 0 & \text{if} \quad x < 0 \\ 1 & \text{if} \quad x > 1 \end{cases}$$

Oscillators which are physically close to each other should over time synchronise together to fire as one. This requires the clustering algorithm to adjust the phase of individual oscillators such that the oscillators which are physically close will take on similar and then synchronised phase. Some form of adjustment of the phase values is required in order to allow this process to occur. As a final stage to one iteration of the training algorithm, the output value of an oscillator will be adjusted $x_j(t^+)$, using the formula

$$x_j(t^+) = B(x_j(t) + \varepsilon_i(\phi_j)) \tag{2}$$

$\varepsilon_j(\phi_j)$, the coupling strength for a given phase ϕ_j is the key to the operation of the whole algorithm. At this stage, adjustments are made to the state variables (and hence, ultimately the phase values) by applying an adjustment which considers the distance an oscillator is from the winning oscillator. Those oscillators physically near to the winning oscillator are made more likely to fire at the same time as the winning oscillator by adjusting the phase *towards* that of the winning oscillator, while those further away have the phase values adjusted so as to push them *down* the phase curve, away from the winning oscillator. Dependent upon the individual problem under consideration, the exact formulation of the coupling function used to calculate these adjustment values may vary, however, for the problem under consideration in this paper, the following coupling function was used.

$$\varepsilon_i(\phi_j) = \begin{cases} C_E[1 - (\frac{d_{ij}^2}{\delta_0})] & \text{if} \quad d_{ij}^2 \leq \delta_0 \\ -C_I[(\frac{d_{ij}^2 - \delta_0}{\delta_1 - \delta_0})] & \text{if} \quad \delta_0 < d_{ij}^2 \leq \delta_1 \\ -C_I & \text{otherwise} \end{cases} \tag{3}$$

Having decided on a limit distance δ_0, δ_1 is set to be five times δ_0. The coupling function promotes all oscillators which have a distance less than δ_0, increasing the phase value by C_E, the constant of excitation multiplied by a fraction that represents the distance between the winning oscillator and the oscillator under consideration, and δ_0. The phase of all those with distance d_{ij} in the interval $\delta_0 < d_{ij} \leq \delta_1$ are inhibited by some fraction of C_I, the coefficient of inhibition. All values of $d_{ij} > \delta_1$ are hard limited to $-C_I$. C_E is typically relatively small, of the order 0.1-0.2. C_I is normally set to the value C_E/N, where N is the number of datapoints under consideration, as any given datapoint is likely to be inhibited more often than it is likely to be excited.

Control of the cluster size is achieved through the manipulation of the δ_0 parameter. Small values will lead to a high number of small, tight clusters, while larger values of δ_0 will create a smaller number of larger clusters. Extremely large values will cause only one cluster to be formed, as this will swallow up all smaller clusters.

4.2 The SOON-2 Algorithm

Several variants of the SOON algorithms have now been proposed, however all share a common basic form. The variant algorithm used in this paper is the SOON-2 algorithm, which incorporates modifications that make it suitable for use whitc high quantities of data.

The SOON-1 algorithm uses all training points as initial oscillators. By reducing this to a smaller number of centres, and distributing them throughout the data space, a series of *prototypes* can be created. These points may be either existing points in the training set, or alternately may be selected to highlight specific areas of interest in data-space, increasing the likelihood of clusters being created in that area. This is of particular interest in microarray analysis, where certain gene expression profiles may be of interest due to biological or physiological processes that are thought to be of significance in a particular operation.

Every datapoint under consideration is allocated an oscillator. A series of prototypes are chosen such that they are distributed either evenly through the data, or in areas of specific interest. The number of prototypes is normally significantly less than the number of datapoints under consideration. All prototypes start with a spheroid enclosure of the same radius, which is set as a parameter on commencement of the algorithm. The algorithm determines which of the oscillators is the next to fire by examining the individual phases of each oscillator and selecting the one with phase ϕ_i closest to 1; the nearest prototype to the winning oscillator is found, and the distance of all datapoints to the nearest prototype is calculated. The phase of all oscillators is increased by a set amount $(1 - \phi_i)$. Having adjusted all phases as necessary, the new state x_i of each oscillator is calculated using equation 1. From the state variables, the coupling strengths (ε_i) are calculated using the coupling function (equation 3), which allows the state variables to be adjusted based on the distance of each oscillator to the winning oscillator. The state variables are then adjusted using equation 2, giving the revised output values. The new phase values can then be calculated using the inverse of the oscillator function, i.e.

$$\phi_j = f^{-1}(x_j) \tag{4}$$

At the end of this cycle of the algorithm, points which were physically close to the oscillator which fired will move closer together, gradually tending to synchronise, whilst those which were further away will move away from the winning oscillator. The newly synchronised oscillators are used to adjust the centre of the nearest oscillator, and the process repeats. Alternatively, certain oscillators will be too far away from any others to form a cluster, and will essentially remain as individual clusters. Dependant upon the noise inherent in the dataset, this might lead to the construction of relatively large numbers of individual oscillators, in the case of extremely noisy data, or in the case of clean data, relatively few unsynchronised oscillators.

5. MICROARRAY DATASET

The microarray array data is taken from the Stanford Yeast Cell-Cycle Project [7]. The number of clusters present within this dataset is not clearly defined, as the data is not clearly separable, however there are a number of groups present within the data the represent different biological processes within the cell cycle of the yeast organism. There have been a number of papers which describe these clusters, along with their biological meanings. The initial data consists of 17 observations over time on approximately 6400 different genes

during the cell cycle process. After normalisation and pre-filtering in order to remove minimally variant genes, the number of genes available for consideration drops to just over 1000 (1002 genes).

6. EXPERIMENTS

The SOON was tested using a Euclidean distance measure, giving spheroid clusters. Using prototypes, set to one half of the total number of datapoints available for clustering, the algorithm was allowed to stabilise, whereupon the clusters were examined. Any cluster with fewer than six members was discarded. At all stages, the coupling function was kept constant, as given in equation 3. The constant of excitation was set to 0.1, with $CI = CE/N = 0.1/1002$. The value of δ_0 was varied between 0.01 and 0.31 in increments of 0.05.

7. RESULTS & DISCUSSION

Figure 1 shows the results of a clustering run using the Euclidean distance measure and the yeast data. The horizontal axis of the plots represent the time course in ten minute intervals, while the vertical axis represents the gene expression magnitude after normalisation.

As can be seen, there are a number of different clusters that make themselves clear as a result of the clustering. Clusters 9 and 177 both show the same characteristics, containing 67 and 49 members in total. Clusters 36 forms a cluster containing 12 members. Clusters 17 and 118 form a cluster with 41 and 10 members. Clusters 6, 50, 58, 74, 188 and 221 all exhibit a broadly similar general trend where a low initial expression level is replaced by a sharp peak at around timepoint 10, followed by a decrease, and then a increase towards time point 16.

These four groups of clusters broadly correspond to phases G1, S, G2 and M respectively, as identified by Cho [8] and Spellman [9]. Additionally, the clusters also match fairly closely in shape to those identified by Tamayo [10].

Of interest is cluster 100 on the figure which describes a small cluster of rather distinctive behaviour; the gene remains predominantly stable around level 0 for most of the experiment, however at timepoint 10 there is a sudden trough to approximately -3 in magnitude. This occurs at the same time as several other groups of genes are peaking in the opposite direction. A cluster with this form does not appear in the results given by Tamayo[10] using a SOM; this may in part be due to the fact that the variation filter used in these experiments gave different results to that of Tamayo, selcting 1002 genes rather than the 823 used in their experiments. However, carrying out tests using a MATLAB based SOM toolbox on the same dataset (Figure 2) also failed to highlight this relatively small cluster as a point of interest - cluster 6 being the closest match. This gives an indication of the extra information that the SOON algorithm can highlight within a dataset.

Experiments with the other sizes of δ_0 have shown that the clusters 9 and 177 appear across the different results, indicating that the algorithm is relatively robust, and choice of the δ_0 parameter is not absolutely crucial. Rather, it offers a degree of control over the granularity of the clusters. This, of course moves the problem of selecting the correct numbers to one of selecting the most appropriate value of δ_0, however, it is felt that this offers more control over the clusters created than for density based approaches.

8. CONCLUSIONS

The SOON is a clustering algorithm that would appear to offer certain properties that are beneficial when examining microarray data. Particularly, the ability of the algorithm to generate an arbitray number of clusters dependent upon the data, rather than pre-selecting a value is welcome. The algorithm is also able to highlight small clusters of genes that are physically close to high density areas of data with different profiles. This allows the SOON to highlight clusters that might not be as clearly visible using other clustering techniques like the SOM.

9. ACKNOWLEDGEMENTS

The authors would like to acknowledge the invaluable assistance of Prof. A. Cossins, School of Biological Sciences, University of Liverpool for his assistance with regard to some of the biological aspects of this paper. Dr. L. Jack is supported by the BBSRC.

REFERENCES

[1] Mohammed Ben Hadj Rhouma and Hichem Frigui, "Self-Organization of Pulse-Coupled Oscillators with Application to Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 1–16, Feb. 2001.

[2] Kimme Kiviluoto, "Predicting bankruptcies with the self-organizing map," *Neurocomputing*, vol. 21, no. 1-3, pp. 191–201, 1998.

[3] A. Ultsch, "Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series," 1999.

[4] Janne Nikkila *et al*, "Analysis and visualisation of gene expression data using self-organizing maps," *Neural Networks*, vol. 15, pp. 953–966, 2002.

[5] J MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967, vol. 1, pp. 281–297, University of California Press.

[6] L Kaufman and P Rousseeuw, *Finding Groups in data: an introduction to cluster analysis*, John Wiley and Sons, New York, 1990.

[7] "The yeast cell cycle analysis project," http://genome-www.stanford.edu/cellcycle.

[8] R J Cho *et al*, "A genome-wide transcriptional analysis of the mitotic cell cycle," *Molecular Cell*, vol. 2, no. 1, pp. 65–73, Jul 1998.

[9] P. T. Spellman, G. Sherlock, M. Quang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell-cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, December 1998.

[10] P. Tamayo *et al*, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 2907–2912, March 1999.

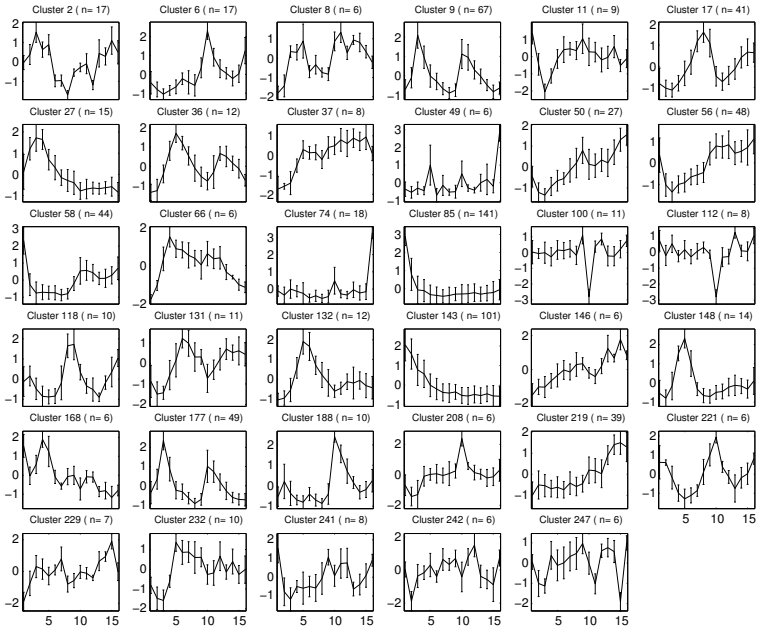

Figure 1: Clustering results using Euclidean distance of 0.2 on the yeast data

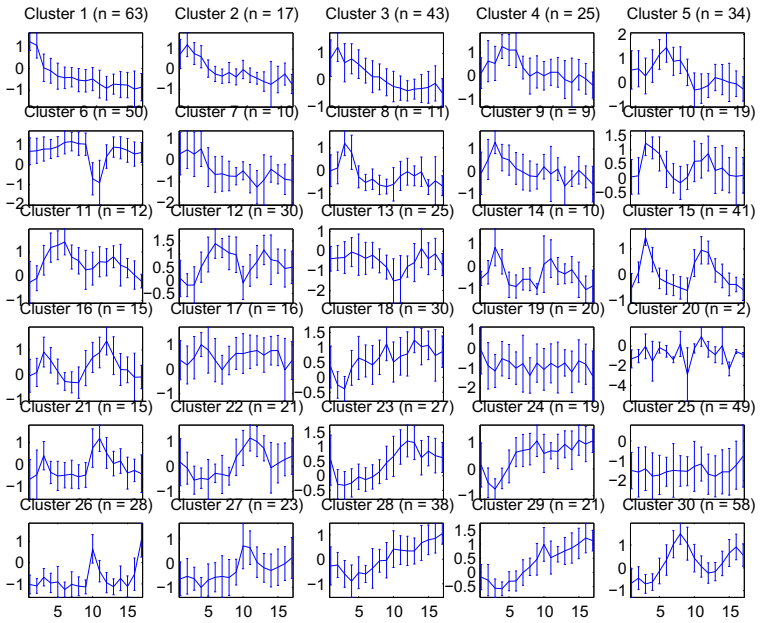

Figure 2: Clustering results using a Self Organising Map