Data Mining using Rule Extraction from Kohonen Self-Organising Maps

James Malone, Kenneth McGarry, Stefan Wermter and Chris Bowerman

School of Computing and Technology, University of Sunderland, St Peters Campus, St Peters Way, Sunderland, SR6 ODD, UK
james.malone@sunderland.ac.uk

Abstract

The Kohonen self-organizing feature map (SOM) has several important properties that can be used within the data mining/knowledge discovery and exploratory data analysis process. A key characteristic of the SOM is its topology preserving ability to map a multi-dimensional input into a two dimensional form. This feature is used for classification and clustering of data. However, a great deal of effort is still required to interpret the cluster boundaries. In this paper we present a technique which can be used to extract propositional IF..THEN type rules from the SOM network's internal parameters. Such extracted rules can provide a human understandable description of the discovered clusters.

Keywords Kohonen Self-Organising Map, Rule Extraction, Data Mining, Knowledge Discovery

1 Introduction

Recently, there has been a great deal of interest in applying neural networks to data mining tasks [3, 17, 18]. Unfortunately, the architecture of most neural networks appears as a "black box" which requires a further stage of processing through extracting symbolic rules to provide a meaningful explanation of their internal operation [1, 16]. However, the majority of this work has concentrated on producing accurate models without considering the potential gains from understanding the details of these models [28, 19]. Recent work has to some extent addressed this problem but much work has still to be done [6, 13]. In this paper we outline our own methods of extracting rules from self-organising networks, assessing them for novelty, usefulness and comprehensibility.

The Kohonen SOM [11] is probably the best known of the unsupervised neural network methods and has been used in many varied applications. It is particularly suited to discovering input values that are novel and for this reason has been used in medical [22], industrial [10, 30] and commercial applications [31]. The SOM is often chosen because of its suitability for the visualization of otherwise difficult to interpret data. The SOM performs a topology-preserving mapping of the input data to the output units. This enables a reduction in the dimensionality of the input data, rendering it more suitable for analysis and therefore also contributing towards forming links between neural and symbolic representations [29].

The purpose of this paper is to demonstrate that a trained SOM can provide initial information for extracting rules that describe the cluster boundaries. Such an

automated procedure will require less skill from the users to interpret the 2-dimensional output layer. To demonstrate this, three data sets were used in the experimental work. These consisted of the Iris, Monks and LungCancer data [9]. These data sets were chosen since they are well known in the machine learning community and are frequently used in benchmarking new algorithms.

The paper is structured as follows. Section two describes the architecture of the Kohonen SOM and explains why it is particularly suited for data mining applications. Section three outlines how our knowledge extraction algorithm produces symbolic rules from the Kohonen SOM. Section four shows how the extracted rules can be used in the knowledge discovery process and discusses the implications of the experimental work. Section five presents the conclusions.

2 The Kohonen SOM Architecture

The basic Kohonen SOM has a simple 2-layer architecture. Since its initial introduction by Kohonen several improvements and variations have been made to the training algorithm. The SOM consists of two layers of neurons, the input and output layers. The input layer presents the input data patterns to the output layer and is fully interconnected. The output layer is usually organised as a 2-dimensional array of units which have lateral connections to several neighbouring neurons. The architecture is shown in Fig. 1.
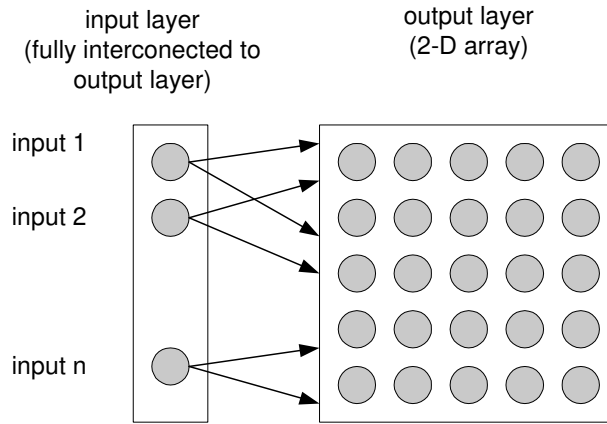

Fig. 1 Architecture of the SOM.

Each output neuron, by means of these lateral connections, is affected by the activity of its neighbours. The activation of the output units according to Kohonen's original work is given by equation 1. The modification of the weights is given by equation 2:

$$O_j = F_{\min}(d_j) = F_{\min}\left(\sum_i (X_i - W_{ji})^2\right) \tag{1}$$

$$\Delta W_i j = O_j \eta \left(X_{i-W_{ji}}\right) \tag{2}$$

where: O_j = activation of output unit j, X_i = activation value from input unit, W_{ji} = lateral weights connecting to output unit, d_j = neurons in neighbourhood, F_{min} = unity function returning 1 or 0, η = gain term decreasing over time.

The lateral connections enable the SOM to learn "competitively", this means that the output neurons compete for the classification of the input patterns. During training the input patterns are presented to the SOM and the output unit with the nearest weight vector will be classed as the winner. Equation 1 shows how the Euclidean distance measure is used to select the winning neuron. An important factor within SOM competitive learning is the neighbourhood function. This function determines the number of output neurons activated by a given input pattern. The neighbourhood size generally shrinks as training progresses until only one neuron is active. This property is very important for topology preservation.

The unsupervised learning ability of the SOM enables it to be used for exploratory data analysis since no a-priori structural information about the data is necessary. One of the most important characteristics of SOMs is the ability to preserve the topology of the original target object. The target object is generally of higher dimensionality than the layer of network output units and therefore a degree of dimensionality reduction occurs [26].

Problems can occur when the structure of the modelled data is too dissimilar from the output layer's structure. This effect is known as topology distortion and has a direct impact on the quality and interpretability of the SOM. A detailed analysis of the effects of topology distortion was made by Li et al based on topology theory [12]. Further work by Villmann [27] has demonstrated that an analysis of topology preservation can lead to improved performance. Villmann was able to use such an analysis to build a SOM type network that grew the required number of output units rather than pre-specify a possible architecture. Previous work also involves growing a network structure [7].

The key application of the SOM is for visual inspection of the formed clusters. Therefore, recent work by Rubio on the development of new techniques for visual analysis of the SOM is of interest [21]. These techniques include what Rubio calls a grouping neuron (GN) algorithm that alters the positions of the neurons according to their reference vectors. The objective is to impose an ordering on the map that enables faster computation time and allows simplification of neuron labelling.

However, a technique that does not require visual inspection in order to perform knowledge extraction is desirable.

3 Knowledge Extraction and Representation

Very little work has appeared in the literature regarding rule extraction from unsupervised/SOM type networks [24]. This is surprising considering the importance of unsupervised methods when applied to exploratory data analysis applications. Bahamonde used the SOM to cluster symbolic rules for their semantic similarity but not as a direct extraction process from the SOM's internal weights and interconnections [2]. Despite their powers of visualization, SOMs cannot provide a full explanation of their structure and composition without further detailed analysis.

One method towards filling this gap is the *unified distance matrix* or U-matrix technique of Ultsch [25]. The U-matrix technique calculates the weighted sum of all Euclidean distances between the weight vectors for all output neurons. The resulting values can be used to interpret the clusters created by the SOM.

The following example shows how the U-matrix is computed, for simplicity a 4x1 sized SOM is considered. This linear SOM will produce a 7x1 U-matrix, composed of the following elements:

$U = [U(1), U(1, 2), U(2), U(2, 3), U(3), U(3, 4), U(4)]$

Where: $U(i,j)$ is the distance between the neurons and $U(k)$ is the average of the surrounding inter-neuron weights. The $U(k)$ values are easily calculated e.g.

$$U(2) = \frac{(U(2,3) + U(3,4))}{2} \tag{3}$$

Fig. 2 shows the boundaries produced by a SOM trained on the Iris data set.
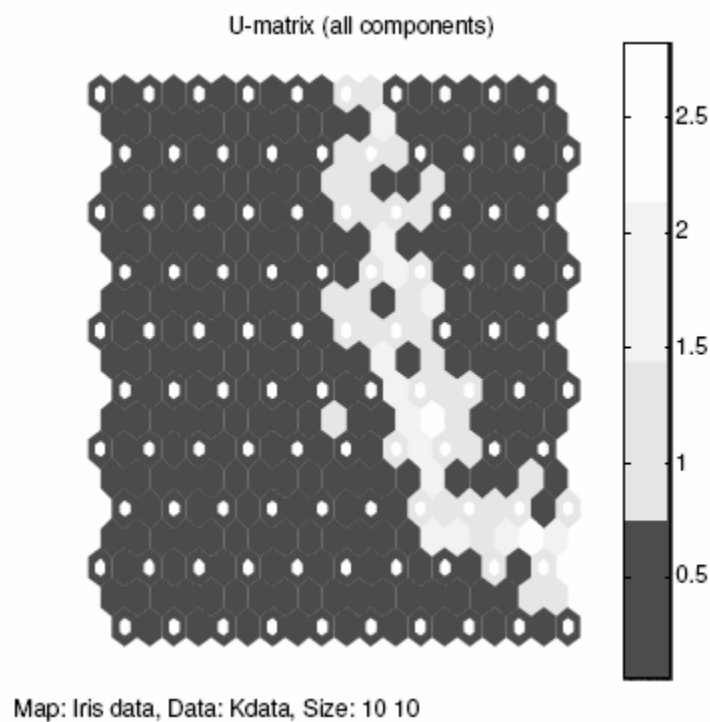

Map: Iris data, Data: Kdata, Size: 10 10

Fig. 2 U-matrix showing cluster boundaries of Iris data set.

3.1 The Rule Extraction Process

Our proposed rule extraction process is outlined in fig. 3 using pseudo-code.

```
Train SOM with data set
Produce U-matrix for components as a total and individually
Specify error threshold for boundaries
for i = number of classes{
        Calculate boundary[i] from total component U-matrix
        for j = no. of individual components{
                calculate position boundary[i] on individual component[j] U-matrix
                if individual component[j] boundary is within error threshold then{
                        map individual component[j] boundary to component's map values
                        take the mean of each unit value along the border
                        use value as a rule to represent that particular cluster boundary
                }
        }
}
```

Fig. 3 Rule extraction algorithm.

3.2 Identifying Cluster Boundaries

Boundaries from the components/U-matrix are selected beginning at the first candidate border unit (in our experiments this is the top left most unit, however the starting position is not critical), extract the value of the currently selected unit to its adjacent units. To extract the boundary we must first calculate which of the two neighbouring units would be the strongest candidate to form such a boundary. This is achieved by using relative differences of the candidate border unit (see Fig. 4). The two selected neighbouring units with the highest relative difference are identified as candidate boundary units.

(a)   (b)

Fig. 4 a Candidate border units and, **b** the six neighbouring units (a,b,c,d,e,f) to the selected unit (x).

The difference between the distance of the mean of the current unit (x) and two other candidate boundary units to the mean of the distance of the remaining neighbouring units (in this instance four others), divided by the range of those remaining neighbours ranges. We have called this measure the Boundary Difference Value (BDV).

$$BDV = \frac{M_L - M_O}{R_O} \qquad (4)$$

where M_L is mean of 3 selected candidate boundary units, M_O is mean of other remaining neighbouring units and R_O is range of remaining neighbouring units.

Once each combination of candidate boundary units has been calculated, the units with the highest BDV are those selected as the most likely units to form a boundary. This process is repeated for each unit on the U-matrix until the strongest boundary candidates have been selected for each one. The next step involves searching for the highest BDV and forming the boundary along the subsequent highest BDV neighbouring units. Fig. 5 illustrates this process; the line indicating the boundary which is automatically formed from this process. In Fig. 5, the size of the circle is directly proportional to the value of the BDV, i.e. the largest circle indicates the highest BDV value. This process is repeated until the number of boundaries is sufficient to identify all of the clusters (i.e. the number of classes required). In Fig. 5, two clusters are required so therefore only one boundary is necessary.
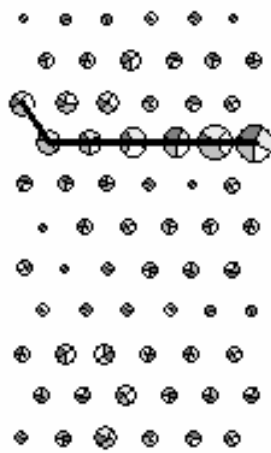

Fig. 5 BDV Plot with boundary indicated by solid line.

3.3 Identifying Key Input Features

Once the process of selecting boundaries is complete, they are now related to the components (input features) to identify the important ones. To undertake this, two steps are required; (i) the boundaries for each individual component must be identified and, (ii) the already extracted boundaries from the total U-matrix are compared with each of these individual component boundaries. If a match is found between a component's boundary and that of the total U-matrix boundary then that component is considered important and is of significance to the clustering process. Similarity between boundary patterns for a component and the U-matrix have been found to indicate that those components have an influence upon the clustering process and are therefore considered important. Fig. 6 illustrates this process; boundaries are shown as white lines on the components (input features) and on the BDV plot.
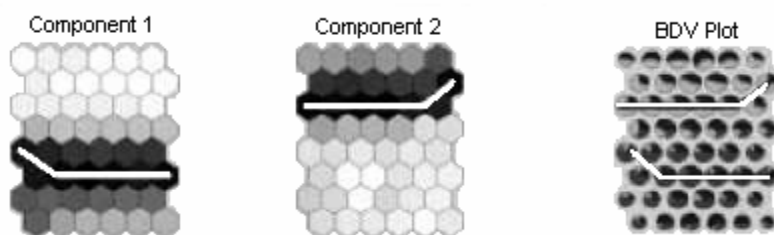

Fig. 6 Selecting important components.

Once the important variables have been identified, the final stage of the process is to extract the variables which will be used to form the rules. For each identified important component, the boundaries which have already been correlated to each component are now translated onto the plots of the actual component values. A single rule is formed by simply extracting the values from the positions of the previously extracted boundaries and then taking the mean of these values. Fig. 7 illustrates this process; in Component 1 the mean value of the units upon which the boundary line falls is calculated as 0.04.
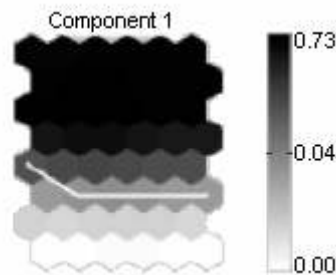

Fig. 7 Extracting values from the component plane.

4 The Data Mining Process

In order for the knowledge extraction process previously described to produce useful results, a further step is required; that of data mining. A popular and insightful definition of data mining and knowledge discovery states that, to be truly successful data mining should be *"the process of identifying valid, novel, potentially useful, and ultimately comprehensible knowledge from databases"* that is used to make crucial business decisions [4]. Furthermore, on evaluation, this leads us to conclude that the following properties must be present within a successful application of data mining

- Nontrivial; rather than simple computations, complex processing is required to uncover the patterns that are buried in the data.
- Valid; the discovered patterns should hold true for new data.
- Novel; the discovered patterns should be new to the organisation.
- Useful; the organisation should be able to act upon these patterns and enable it to become more profitable, efficient, etc.
- Comprehensible; the new patterns should be understandable to the users and add to their knowledge.

Therefore the most important issue in knowledge discovery is how to determine the value or interestingness of the patterns generated by the data mining algorithms. Two approaches can be used; (i) objective measures, which require the application of some data-driven method such as the coverage, completeness or confidence of the extracted rule set and, (ii) subjective or user-driven measures which rely on the domain expert's beliefs and expectations to assess the discovered rules. The rules extracted from the SOM using our proposed technique were evaluated by objective interestingness measure.

4.1 Objective Interestingness Measures

Freitas enhanced the original measures developed by Piatetsky-Shapiro [20] by proposing that future interestingness measures could benefit from by taking into account certain criteria [5, 6].

- Disjunct size; the number of attributes selected by the induction algorithm is of interest. Rules with fewer antecedents are generally perceived as being easier to understand. However, those rules consisting of a large number of antecedents may be worth examining as they may refer either to noise or a special case.
- Misclassification costs; these will vary from application to application as incorrect classifications can be more costly depending on the domain, e.g. a false negative is more damaging than a false positive in Cancer detection.
- Class distribution; some classes may be under represented by the available data and may lead to poor classification accuracies, etc.
- Attribute ranking; certain attributes may be better at discriminating between the classes than others. So discovering attributes present in a rule that were thought previously not to be important is probably worth investigating further.

Work by McGarry used several of these techniques to evaluate the importance of rules extracted from RBF neural networks to discover their internal operation [14, 15, 16]. However, the statistical strength of a rule or pattern is not always a reliable indication of novelty or interestingness. Those relationships with strong correlations usually produce rules that are well known, obvious or trivial. Additional methods must be used to detect interesting patterns.

4.2 Subjective Interestingness Measures

Subjective measures have been developed in the past and generally operate by comparing the user's beliefs against the patterns discovered by the data mining algorithm. There are techniques for devising belief systems and typically involve a knowledge acquisition exercise from the domain experts [13]. Other techniques use inductive learning from the data and some also refine an existing set of beliefs through machine learning. Silberschatz and Tuzhilin view subjective interesting patterns as those more likely to be unexpected and actionable [23].

4.3 Rule Generation

The data mining and knowledge discovery process begins with the training of the Kohonen network. The three data sets used are described in Table 1.

**Table 1** Description of the data sets used in the experimentation.

Data Set	No. of Examples	No. of Attributes	No. of Classes	Description
Iris	150	4	2	Data describes three 3 iris flower classes, two of which are not linearly separable from each other. Attributes are petal length and width and sepal length and width.
Monks	124	6	2	This data contains a binary classification (monk or not_monk) over a six-attribute discrete domain. Attributes describe the robot's appearance, e.g. head_shape, body_shape.
LungCancer	32	56	3	The data describes 3 types of pathological lung cancers. The Authors give no information on the individual variables.

The process is completed when the symbolic rules are extracted. Interpreting the cluster boundaries of the network in the form of rules makes explicit to the user such parameters as the important input variables and so leads to knowledge discovery. To understand how these values can now be used as rules to describe the clusters, we first look at how the data has been mapped in the SOM. Fig. 8 shows visually how the different classes in the Monks data set have clustered in the SOM. It becomes apparent using the rule extraction technique that there are several clusters which each relate to a single class and it is this clustering process that we are trying to represent in the extracted rules. By projecting the boundaries over the histogram it can be seen that the rules can partition a cluster.

(a)

9
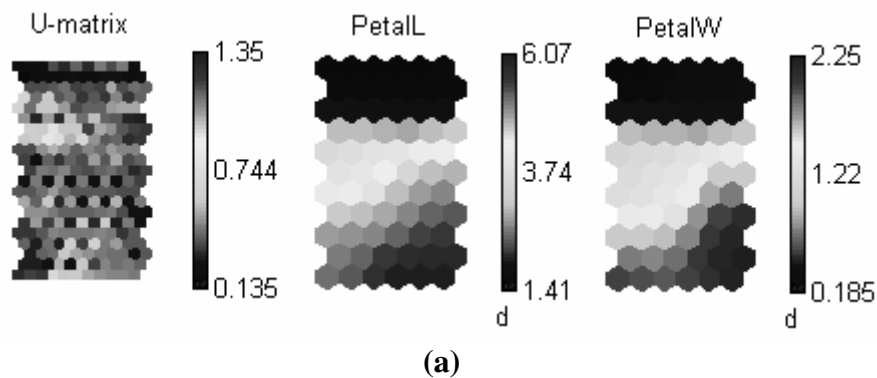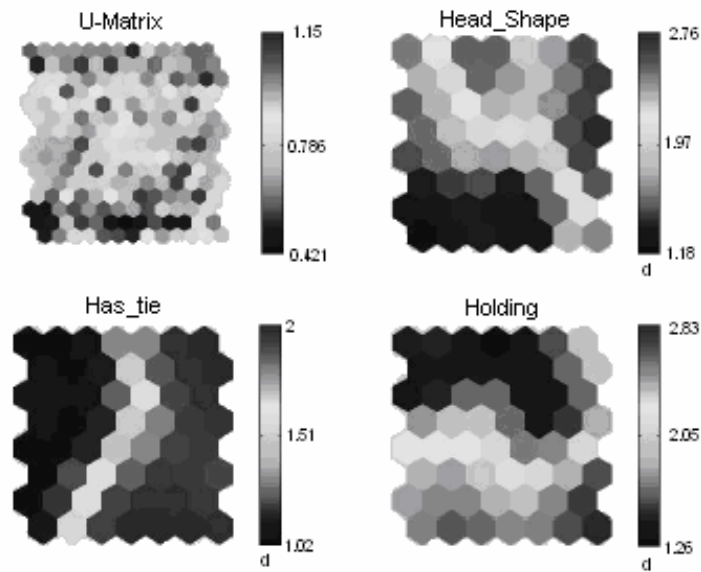
(b)
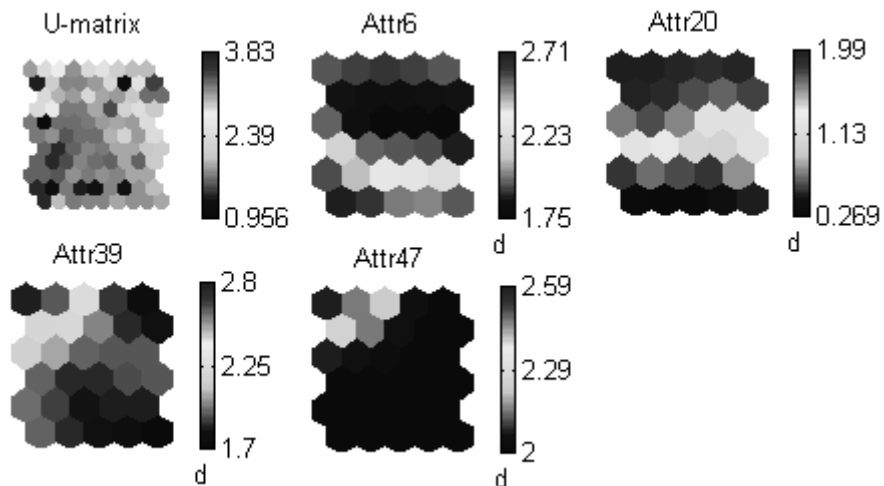
(c)

Fig. 8 Histogram showing component planes used to form the rules and their values with scale (down right hand side of each histogram) for a SOM trained with the various data sets; **a** Iris data set – components describe the petal length and width, **b** Monks data set – components describe various characteristics of a robot, such as whether they were holding an object and the head shape, and **c** LungCancer data set – components describe various characteristics of the three forms of lung cancer. The shaded hexagons indicate the value of the SOM at a particular location.

The extracted rules are in the format of symbolic, propositional rules in conjunctive normal form. Each rule consists of an antecedent that describes the cluster characteristics and a consequent pertaining to a cluster. Class label information may be assigned to make the rules more readable. For example, the Monks Data set produced the following rules presented in Fig. 9.

1	If PetalL < 2.8 and
	If PetalW < 0.75 then Setosa
2	If PetalL > 2.8 and
	If PetalL< 5 then Versicolor
3	If PetalL > 5 then Virginica

(a)

1	If Headshape < 2 and
	If Holding ≥ 1.7 then Not Monk
3	If Headshape < 2 and
	If Holding <1.7 then Monk
4	If Headshape ≥ 2 and
	If Has_tie < 1.5 then Monk
5	If Headshape ≥ 2 and
	If Has_tie ≥ 1.5 and
	If Holding < 1.7 then Not Monk
6	If Headshape ≥ 2 and
	If Has_tie ≥ 1.5 and
	If Holding ≥ 1.7 then Monk

(b)

1	If Attr20 <1 and
	If Attr6 >2.5 and
	If Attr39 <=2.25 and
	If Attr47 <= 2.3 then class1
2	If Attr20 >=1 and
	If Attr6 <=2.5 and
	If Attr39 <=2.25 and
	If Attr47 <= 2.3 then class2
3	Else default is class 3

(c)

Fig. 9 Rules extracted from the SOM trained on the various data sets; **a** Iris data set, **b** Monks data set and **c** lung cancer data set.

Rules were extracted using our described technique from SOM's trained on the three data sets and the results are presented in Table 2.

Table 2 Experimental results indicating accuracy and number and coverage of disjuncts of extracted rules for each data set.

Data set	No classes	Overall Rule Accuracy(%)	No. Disjuncts	Coverage of Disjuncts	Concept
Iris	3	95.3	1	50	Setosa
			1	50	Versicolor
			2	25, 25	Virginica
LungCancer	3	71.9	1	8	CancerClass1
			1	10	CancerClass2
			1	14	CancerClass3
Monks	2	61.3	3	13, 32, 31	Monk
			2	33, 15	Not_Monk

5 Discussion

The results presented in Table 2 show a variation in the accuracy of the rules to classify the data. An important feature to consider when applying our described rule extraction process is that the rules produced are an accurate representation of the SOM produced for each particular data set. As such, if the trained SOM does not readily cluster the data, then the rules produced will reflect this. For example, the highest accuracy achieved in literature through use of KNN method upon the LungCancer data set is 77% [9]. Our rule extraction method provides rules which are 71.9% accurate. Since 77% can be seen as the threshold, the rules extracted simply reflect this and hence are not ~100% accurate. Fig. 10 shows the clustering process achieved by our SOM experiments for the LungCancer and Iris data sets. A visual inspection of (a) can quickly reveal that the training process has not resulted in completely linearly separable clusters for LungCancer data whilst for (b) the data has clustered more readily. For example, in (a) class 3 generally appears in the top left hand corner of the map, however there is a small cluster at the top right and the bottom right. Such features are responsible for the inherent error rate within the rules extracted and the difference in such an error rate between data sets.

In terms of objective interestingness measures, the number of disjuncts for each data set is low. Each class is represented by, at most, 3 disjuncts and the lowest coverage for a single disjunct is 8 and has, therefore, avoided the problems associated with small disjuncts [8].
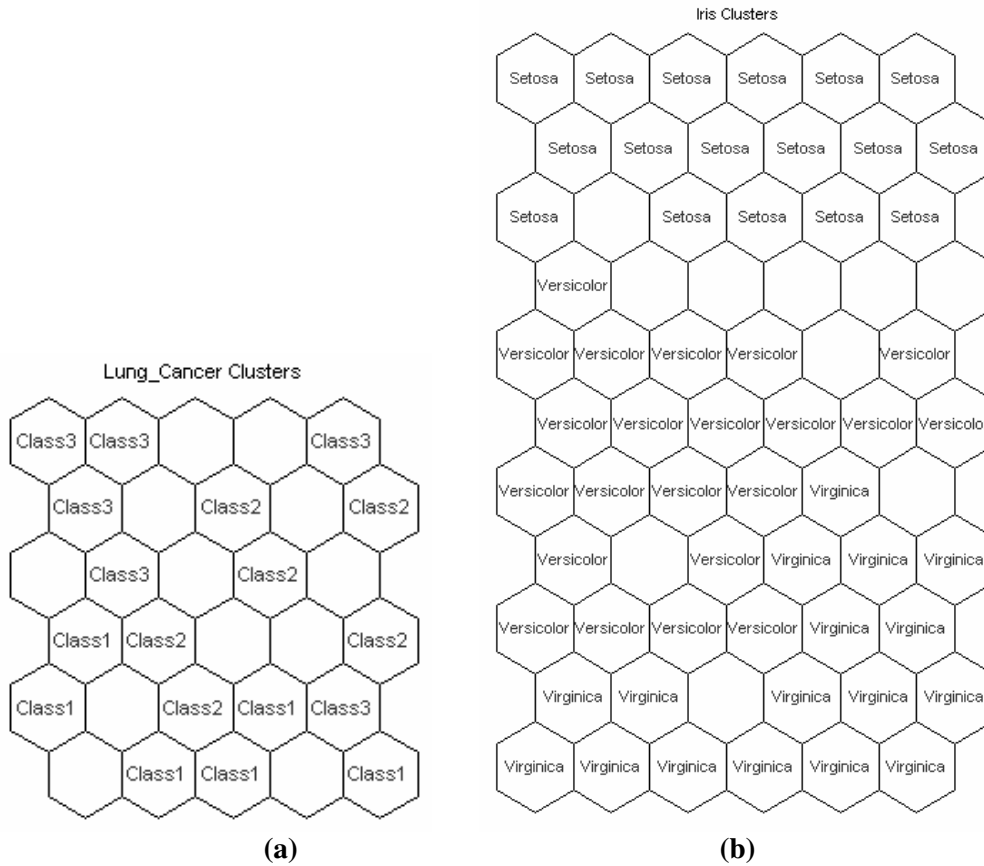
Fig. 10 SOM Clustering of **a** Lung_Cancer data set and **b** Iris data set

6 Conclusions

In this paper, we have proposed a technique for the automatic extraction of rules from trained SOMs. The technique performs an analysis of the U-matrix translated from a trained SOM to form boundaries which are then related to individual components. The technique proposed succeeds in extracting (i) the important components which are responsible for the clustering and (ii) the important values of these components which can be used to describe the clusters formed. These individual component boundaries are then used to extrapolate values which can then form the basis of propositional IF..THEN type rules. These simple rules can be easily exploited by an expert or decision support system and are easily interpretable to an expert. The use of rule extraction from SOMs should be viewed as a complementary approach to the available data visualization techniques. When used in conjunction with such methods as U-matrix analysis, it will provide a comprehensive exploratory data analysis tool.

Whilst the rules extracted accurately represent the trained SOM in cases where clustering has readily occurred, they also reflect any deficiencies where clustering did not occur. Therefore, one important factor when assessing the applicability of our technique is the underlying accuracy of the clustering process performed by the SOM. Clearly, in cases where the SOM fails to readily cluster the data, the resulting rules will reflect this underlying inaccuracy.

7 Acknowledgments

References

[1] R. Andrews, J. Diederich and A. Tickle. The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057-1068, 1998.

[2] A. Bahamonde. Self-organizing symbolic learned rules. In *Proceedings of the International Conference on Artificial and Natural Neural Networks*, Lanazrote, Canaries, 1997.

[3] Y. Bengio, J. Buhmann, M. Embrechts and J. Zurada. Introduction to the special issue on neural networks for data mining and knowledge discovery. *IEEE Transactions on Neural Networks*, 11(3):545547, 2000.

[4] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth. From data mining to knowledge discovery: an overview. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1-34. AAAI-Press, 1996.

[5] A. Freitas. On objective measures of rule suprisingness. In *Principles of Data Mining and Knowledge Discovery: Proceedings of the 2nd European Symposium,Lecture Notes in Artificial Intelligence*, volume 1510, pages 1-9, Nantes, France, 1998.

[6] A. Freitas. On rule interestingness measures. *Knowledge Based Systems*, 12(5-6):309-315, 1999.

[7] B. Fritzke. Growing self-organizing networks - why? In *Proceedings of the European Symposium on Artificial Neural Networks*, pages 61-72, Brussels, 1996.

[8] R. C. Holte, L. E. Acker and B. W. Porter. Concept learning and the problem of small disjuncts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 813-818, Detroit, 1989.

[9] Z. Q Hong and J. Y. Yang "Optimal Discriminant Plane for a Small Number of Samples and Design Method of Classifier on the Plane", *Pattern Recognition*, 24(4):317-324, 1991.

[10] J. Koh, M. Suk and S. M. Bhandarkar. A multilayer self-organizing feature map for range image segmentation. *Neural Networks*, 8(1):67-86, 1995.

[11] T. Kohonen, E. Oja, O. Simula, A. Visa and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358-1383, 1996.

[12] X. Li, J. Gasteiger and J. Zupan. On the topology distortion in self-organizing feature maps. *Biological Cybernetics*, 70:189-198, 1993.

[13] B. Liu, W. Hsu, L. Mun and H. Y. Lee. Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):817-832, 1999.

[14] K. McGarry. The analysis of rules discovered by the data mining process. In *Proceedings of 4th International Conference on Recent Advances in Soft Computing*, pages 255-260, Nottingham, UK, 2002.

[15] K. McGarry, S. Wermter and J. MacIntyre. The extraction and comparison of knowledge from local function networks. *International Journal of Computational Intelligence and Applications*, 1(4):369-382, 2001.

[16] K. McGarry, S.Wermter and J. MacIntyre. Knowledge extraction from local function networks. In *Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pages 765-770, Seattle, USA, 2001.

[17] S. Mitra and Y. Hayashi. Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3):748-768, 2000.

[18] S. Mitra and S. Pal. Data mining in soft computing framework: a survey. *IEEE Transactions on Neural Networks*, 13(1):314, 2002.

[19] K. Ono. New approaches for data mining in corporate environments. In *Proceedings of the 2nd International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pages 49-65, London, UK, 1998.

[20] G. Piatetsky-Shapiro and C. J. Matheus. The interestingness of deviations. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, 1994.

[21] M. Rubio and V. Gimenez. New methods for self-organising map visual analysis. *Neural Computation and Applications*, 12(3-4):142-152, 2003.

[22] P. K. Sharpe and P. Caleb. Self organising maps for the investigation of clinical data: A case study. *Neural Computing and Applications*, 7:65-70, 1998.

[23] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of the 1ˢᵗ International Conference on Knowledge Discovery and Data Mining,* pages 275-281, 1995

[24] A. Ultsch and D. Korus. Automatic acquisition of symbolic knowledge from subsymbolic neural nets. In *Proceedings of the 3rd European Conference on Intelligent Techniques and Soft Computing*, pages 326-331, 1995.

[25] A. Ultsch, R. Mantyk and G. Halmans. Connectionist knowledge acquisition tool: CONKAT. In J. Hand, editor, *Artificial Intelligence Frontiers in Statistics: AI and statistics III*, pages 256-263. Chapman and Hall, 1993.

[26] J. Vesanto and Esa Alhonemi. Clustering of the self-organizing map. *IEEE Trans-*
actions on Neural Networks, 11(3):586-600, 2000.

[27] T. Villmann. Estimation of topology in self-organizing maps and data driven growing of suitable network structures. In *Proceedings of the 6th European Conference on Intelligent Techniques and Soft Computing*, pages 235-244, Aachen, Germany, 1998.

[28] D. Watkins. Discovering geographical clusters in a U.S. telecommunications company call detail records using Kohonen self organising maps. In *Proceedings of the 2nd International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pages 67-73, London, UK, 1998.

[29] S. Wermter and R. Sun. *Hybrid Neural Systems*. Springer, Heidelberg, Germany, 2000.

[30] U. Witkowski, S. Ruping, U. Rucket, F. Schutte, S. Beineke and H. Grotstollen. System identification using self-organizing feature maps. In IEE, *Proceedings of the 5th International Conference on Artificial Neural Networks*, pages 100-105, Cambridge, England, 1997.

[31] C. X. Zhang and D. A. Mlynski. Mapping and hierarchical self-organizing neural networks for VLSI placement. *IEEE Transactions on Neural Networks*, 8(2):299-314, 1997.