Rule Extraction using Fuzzy Clustering for a Sedimentary Rock Data Set

T.D. Gedeon, H. Kuo, and P.M. Wong

Abstract -- In this paper, the fuzzy classification method developed by Abe and Lan (1995) has been improved. This method extracts fuzzy rules directly from numerical data. This paper shows how pre-processing input data using clustering may help the classification accuracy in some cases. The proposed method is compared with Abe & Lan's fuzzy classification method with a data set obtained from an oil reservoir in the North West Shelf in Australia and the Fisher Iris data.

I. Fuzzy Systems for Pattern Classification

Pattern classification is an important area in many engineering sciences. Although neural networks are suitable for the classification of data set with a large number of input dimensions, they often suffer from the need for the optimisation of network structure and extensive computation time for practical applications (Wong et al,. 1997).

In many circumstances, fuzzy systems present a more practical approach for pattern classification. They do not necessarily rely on trial and error used to construct neural networks. Complex physical systems can be reduced to a set of fuzzy rules, which are often appealing to engineering scientists.

II. . Fuzzy Classification using Hyperboxes

The idea of representing the existence of data for a class by a set of hyperboxes (or rules) was discussed in Simpson (1992). Abe and Lan (1995) developed a supervised fuzzy classification method (or simply the "AL" method) based on the use of hyperboxes to solve the problem where different classes overlap each other.

Two types of hyperboxes were introduced in the AL method: activation hyperboxes and inhibition hyperboxes. An activation hyperbox represents the existence region for a class, whereas an inhibition hyperbox contains the overlapping data within the activation hyperbox. In this approach, the hyperbox is drawn by taking the maximum and minimum values of the input vectors for a class.

During training, fuzzy hyperboxes are defined recursively for each level. For example, let us assume there are three classes, a, b and c in the training set with a certain number of input vectors x with m dimensions *(x1,...,xm)*. Fuzzy rules are generated for every class pair combination, *(a, b)*, *(a, c)*, *(b, a)*, *(b, c)*, *(c, a)* and *(c, b)*. The first element in the pair is the leading class. We start with class pair *(a, b)* at the top level, 0. An activation hyperbox, *Aaa*, will be drawn to cover the region occupied by class *a*, and similarly for class b to get *Abb*. If these two hyperboxes do not overlap, then the following rules are obtained:

Rule 1: If **x** is in *Aaa* then **x** is class *a*.

Rule 2: If **x** is in *Abb* then **x** is class *b*.

However, if *Aaa* and *Abb* do overlap, an inhibition hyperbox, Iab, will be drawn to represent the overlapping region between *Aaa* and *Abb*. In this case, the following rules are generated instead:

Rule 1: If **x** is in *Aaa* and not in *Iab*, then **x** is class *a*.

Rule 2: If **x** is in *Abb* and not in *Iab*, then **x** is class *b*.

The same process will be repeated for any input vector **x** which is in *Iab*, and a new level is created. Activation and inhibition hyperboxes (i.e. fuzzy rules) will be created in the same manner in this new level. The recursion will stop when there is no overlapping between the activation hyperboxes or the number of the remaining input vectors are too few to form hyperboxes (ie. when there is only one vector left).

This results in a pair of fuzzy rules generated by the recursion at every level. So, every class pair will have a list of fuzzy rules.

In testing, the data set is tested to see the likelihood of its belonging to a particular class. A test vector **x** is passed into the fuzzy rule lists for a class pair to generate a list of degree of membership (DOM) for each level. The maximum of the list is taken as the DOM of **x** for that class pair.

T.D. Gedeon is with the School of Information Technology, Murdoch University, Western Australia
H. Kuo is with the Macquarie Bank
P.M. Wong is with the Petroleum Engineering, University of New South Wales, Sydney, Australia

Therefore, a test vector **x** will have a list of DOMs for every class pair. Then, the conflict is solved for every class individually. The minimum DOM from the class pairs with the same leading class is taken as the likelihood/DOM of **x** for that leading class.

In the previous example, the fuzzy rules from (*a*, *b*) will determine the DOM of **x** when compared to class *b*. Similarly, the rules obtained from (*a*, *c*) will determine the DOM of **x** when compared to class *c*. The smallest value is taken as the DOM of **x** for class *a*.

This process is repeated for every class. In the end, every vector **x** will have a DOM value for every class. Vector **x** is classified as the class with the highest DOM.

DOM is 1 for **x** located within the activation hyperbox. For **x** outside the activation hyperbox, it is determined by x_k where distance to the surface of the hyperbox is the maximum among those of $x_1,\ldots,x_m$, to approximate the real distance. For example, in Fig. 3 below the horizontal distance is zero and is ignored. Thus, in the AL method, the DOM is calculated based on only one dimension. The membership function of the activation hyperbox is shown in Fig. 1.

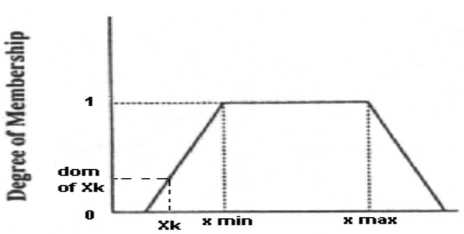

Fig. 1: Membership function (Abe and Lan, 1995).

If x_k is nearest to the surface of hyperbox in its dimension, it is clear that as x_k moves further away from the surface of the hyperbox, the DOM decreases until it reaches zero.

III. DRAWBACKS OF THE AL METHOD

Although the AL method is fast to train and has good performance for data sets with distinct characteristics, this method has some drawbacks when the data classes are highly overlapped, which is not uncommon in real world data sets.

According to the AL method, a hyperbox is drawn taking the maximum and minimum values of vectors of a class. If the maximum and minimum values are close (ie. the vectors are close to each other), the test data will be less likely to fall into this hyperbox and be classified as this class. It may lead to a wrong classification. Four examples are shown to illustrate the drawbacks.
In the first example (Fig. 2), circled **b** would be classified as class *a*, because it exists entirely within *Aaa*. However, **b** is closer to the centre of *Abb* than *Aaa*. This case is likely to happen when **b** is not used in training. This behaviour leads to poor generalisation by the AL method.

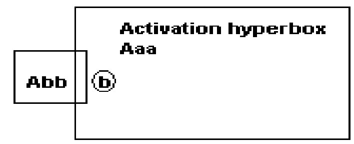

Fig. 2: A case where *Abb* is small.

In addition, the DOM function is based on only one dimension. Therefore, it may not represent the real relationship between the test data and the region occupied by the class.

In the second example (Fig. 3), the AL method would classify the circled **b** as class *a*, since its distance to *Aaa* is closer than the distance to *Abb.* However, this is clearly wrong because circled **b** is closer to the centre of *Abb* (or any other **b**).

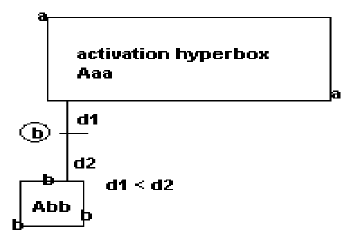

Fig. 3: A case in which d1 and d2 are the maximum distance between circled **b** and the surface of the hyperboxes.

In the third example (Fig. 4), the AL method would not be able to classify the circled **b**, because **b** has equal DOM from both of the classes due to its equal maximum distances in the vertical dimension to *Aaa* and *Abb*. But clearly, circled **b** is closer to the centre of *Abb* than *Aaa*.

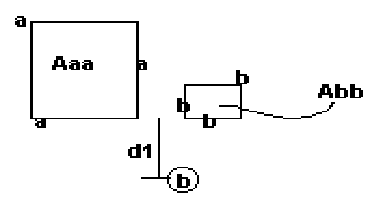

Fig. 4: A case where the distance d1 between circled **b** and the two hyperboxes would not help.

In the last example (Fig. 5), circled **b** would be classified as class *a*, because it exists entirely within *Aaa*.

The case in presented Fig. 5 is in fact a complex one. If the DOM of circled **b** is determined only by its distance to the centre of *Aaa* and *Abb*, **b** may not be correctly classified, because circled **b** is closer to the centre of *Aaa* than *Abb.*

An improved AL method is introduced in the next section to solve the misclassification problem in the four cases presented.

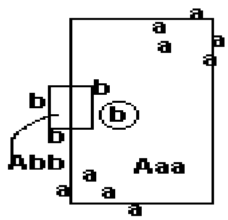

Fig. 5: A case in which the distance from circle **b** to the centre of *Aaa* and *Abb* would not help.

IV. FUZZY CLUSTERING CLASSIFICATION METHOD

This paper proposes a fuzzy clustering classification method (or simply the "FCC" method), in which the misclassification problem mentioned in the previous section will be solved.

The FCC method is based on the AL method. It consists of three parts as shown in Fig. 6. Each of the three parts is represented as a block. The first part performs input clustering. The second part is a fuzzy classification system, which is similar to the AL method. Activation and inhibition hyperboxes are also used to generate fuzzy rules in the same way as the AL method. The final part merges the outputs from the fuzzy classification system to give a final classification decision.

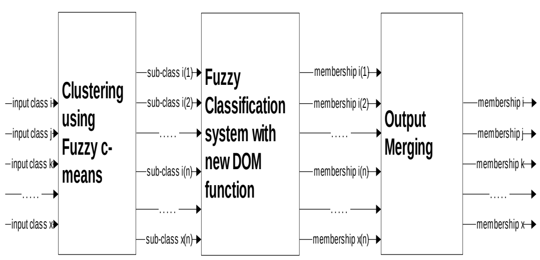

Fig. 6: The structure of FCC.

In FCC, we adopt a new DOM function. Instead of taking the maximum distance to the surface of the hyperbox among all dimensions, FCC calculates the DOM by using the distance between the vector **x** and the centre of the hyperbox. The centre of an activation hyperbox is **c** with *m* dimensions ($c_1, \ldots, c_m$). The distance between **x** and **c** is:

$$d_c(\mathbf{x}) = \sqrt{(x_1 - c_1)^2 + \ldots + (x_m - c_m)^2}$$

And the DOM of **x** is simply – $d_c(\mathbf{x})$, which means that the larger the distance, the smaller the DOM.

Hence, for the fuzzy rules generated at different levels for a class, a list of DOM values (in FCC, the values are negative) will be produced by these rules. Thus, the final DOM of **x** for a class is the maximum of the DOM values.

In this way, as **x** moves further away from the centre of an activation hyperbox, the DOM will become smaller. The likelihood/DOM of **x** for a particular class is determined by how close it is to the centre of the occupied region/hyperbox, not just to the surface of the hyperbox. By using this DOM function, a better generalisation can be achieved, because the test set vectors do not have to occur within the activation hyperbox to achieve a high DOM value. The misclassification problems in the first three examples in the previous section can be avoided using the new DOM function.

In the last example, however, the use of the new DOM function alone cannot solve the problem, because circled **b** is sitting near the centre of *Aaa*. This gives a high DOM of **b** for class *a*.

A technique to "break up" the activation hyperbox of class *a* is required in this case. The first part of FCC uses the fuzzy c-means method (Bezdek, 1981; Sugeno and Yasukawa, 1995) to perform fuzzy clustering of the input vectors in training (Fig. 6). Each class is clustered into *n* sub-classes. The number *n* can be changed to suit different data sets. Each sub-class is treated as a class of its own in the fuzzy classification system.

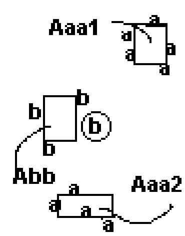

Fig. 7: The new hyperboxes after clustering.

The fuzzy classification system will determine the DOMs of **x** for all the sub-classes (Fig. 6). If the DOM of **x** is denoted as $da_i(x)$, for sub-class a_i, for $i = 1, \ldots, n$, then the DOM of **x** for class *a* is:

$$da(\mathbf{x}) = \max(da_1(\mathbf{x}), \ldots, da_n(\mathbf{x}))$$

A test vector **x** is classified as class *a* if a sub-class derived from *a* has the highest DOM by the fuzzy classification system.

By applying FCC, the new rules generated in the last example from the previous section are shown in Fig. 7. It is clear that circled **b** will be correctly classified in this case.

V. EXPERIMENT RESULTS AND ANALYSIS

Four experiments have been performed. The first experiment uses the Fisher Iris data as a benchmark to show the performance of the proposed FCC method. The remaining experiments consist of classification of an oil data set obtained from three oil wells in Australia. The data set is split randomly to form training and test sets.

Fisher iris data

Iris data was used in this experiment as a benchmark to compare the performance of the FCC and the AL methods. Due to the page limitations, we will only show part of the FCC results.

The first 25 and remaining 25 vectors are used for training and testing from each of the three classes respectively, giving a total of 75 vectors for training and 75 vectors for testing. Each vector has 4 dimensions. Three of the four dimensions are shown as a 3D plot in Fig. 8. The three different colour grading represent the three classes. The plot shows that the Iris data set presents a relatively "easy" problem, because the degree of overlapping between classes is low.

Table 1 shows the performance of the AL and FCC methods. The number of clusters required in each of the three classes (C1,C2,C3) can be varied by the user. This produces a different number of fuzzy rules (hyperboxes) after training. The number in parentheses represents the number of input vectors used in training and testing for each class. The recognition rate is defined as the number of correct classifications divided by the number of input vectors.

For example, the (1,1,1) configuration indicates that only one cluster was created in each class. This is equivalent to the AL method but using the new DOM function. The FCC method generated 4 fuzzy rules. This configuration gave 100% recognition for class 1 data, 88% in class 2 and 92% in class 3 in testing. This is equivalent to an overall recognition rate of 93%.

In this experiment, the AL method was able to achieve a high recognition rate of 92%. The FCC results using (1,1,1) improved only slightly and insignificantly (93%) compared to the AL method. Note that some configurations could produce better results in one class than others. For example, (1,1,4) was good for class 3 but not good for class 2. The best configuration was (4,4,4) which resulted in a recognition rate of 99%.

Note that the present conclusions were made based on the availability of the test results. For real application, we may need to obtain the best configuration for each class and the overall performance using the training set.

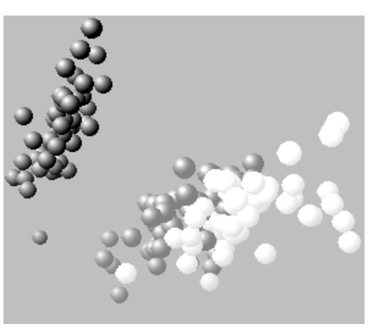

Fig. 8: Fisher Iris data.

Training				Testing			
No. of clusters created in			No. of rules	Recognition rate (%)			
C1 (25)	C2 (25)	C3 (25)		C1 (25)	C2 (25)	C3 (25)	Total (75)
b) AL			4	100	88	88	92
1	1	1	4	100	88	92	93
4	4	4	66	100	100	96	99
1	1	4	15	100	84	100	95

Table 1: Fish Iris data test results.

Oil data set

The second data set was obtained from three oil wells in the North West Shelf, offshore western Australia. The data set is more complicated than the Iris data set, because it has many highly overlapping regions. The data set contains a set of rock samples extracted from the wells, which were manually classified by an expert geologist as *fracture*, *ok* and *good*. It is an important process because the quality of the rock samples would determine the representativeness of subsequent measurements performed on the samples for reservoir engineering analysis.

The objective is to develop a classifier to determine the quality of the rock samples using 11 types of well measurements (inputs). The performance can be evaluated by comparing the predictions with the expert classification.

Three 3D plots displaying the distribution of three inputs (out of 11) from wells 1, 2 and 3 are shown in Figs. 9, 10 and 11 respectively. In these figures, black spheres represent data belonging to *fracture* class, *good* are gray and *ok* are white.

In the experiment with the well 1 data set, as the number of clusters of a class increases in training, the recognition rate increases for that class. This is because the test vectors will be more likely to fall near the centre of one of the clusters.

By applying clustering, it was interesting to see that it was possible to "tune" the recognition rate for each class individually in testing by changing the cluster numbers in training.

Well 1 data set

Table 2 shows the experiment result comparison between the AL and FCC methods. The (1,1,1) configuration gave a base recognition performance of 47% for *fracture*, 60% for *good* and 40% for *ok*. With (4,2,1) configuration, it was able to have a recognition rate of 93% for *fracture* and 100% for *good* in testing. Due to the increased cluster sizes in *fracture* and *good* classes, the test data was more likely to be classified as *fracture* or *good*. Hence the recognition rate of *ok* data in testing reduced.

The (1,1,4) gave a recognition rate of 75% for *ok* test data, but only 20% and 33% for the other two classes for the same reason mentioned above.

With the use of more clusters in all three classes, such as the (5,5,6) configuration, the overall recognition rate was improved to 70%. It was a 20% improvement compared to the AL method (50%).

Well 2 data set

Table 3 tabulates the results from AL and FCC methods. The (5,5,6) configuration gave a much greater recognition rate (75%) compared to the (1,1,1) configuration (41%) or the AL method (61%).

Well 3 data set

The major characteristic of this data set is that there are only 9 vectors in *fracture* in the training set. This normally means that the test vectors will be less likely to fall into the activation hyperbox. This was shown in the AL method, in which all the test vectors were classified as either *good* or *ok*. The recognition rate for *fracture* was 0%.

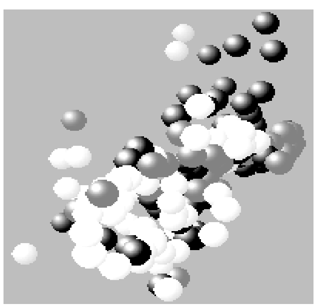

Fig. 9: Oil data for well 1.

Training				Testing			
No. of clusters created in			No. of rules	Recognition rate (%)			
Frac (39)	Good (26)	OK (51)		Frac (15)	Good (15)	OK (20)	Total (50)
d) AL			8	47	47	55	50
1	1	1	8	47	60	40	48
4	2	1	24	93	100	10	62
1	1	4	18	20	33	75	46
5	5	6	120	80	73	45	70

Table 2: Well 1 data test results, in which 116 vectors are used in training and 50 in testing.

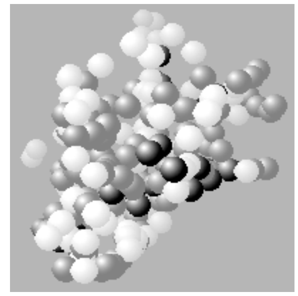

Fig. 10: Oil data for well 2.

Training				Testing			
No. of clusters created in			No. of rules	Recognition rate (%)			
Frac (18)	Good (70)	OK (64)		Frac (15)	Good (35)	OK (25)	Total (75)
f) AL			9	40	66	68	61
1	1	1	9	73	51	8	41
4	2	1	25	100	57	4	48
1	3	1	15	40	97	4	55
1	1	4	19	47	17	92	48
5	5	6	122	93	83	52	75

Table 3: Well 2 data test results, in which 152 vectors are used in training and 75 in testing.

In the FCC method, however, the experiments demonstrated that the fuzzy rules generated in training would not be "biased" to a class with a relatively large number of the vectors in the training set. It could recognise the class with a relatively small number of training vectors in testing.

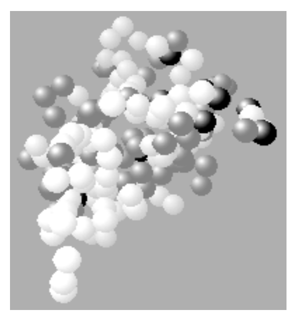

Fig. 11: Oil data for well 3.

Training				Testing			
No. of clusters created in			No. of rules	Recognition rate (%)			
Frac (9)	Good (34)	OK (48)		Frac (8)	Good (31)	OK (46)	Total (85)
h) AL			6	0	55	59	52
1	1	1	6	75	45	26	38
3	2	2	24	88	45	41	47
1	3	1	13	63	84	22	48
3	1	6	46	75	10	83	55
3	5	6	93	63	45	70	60

Table 4:Well 3 Oil data test results, in which 91 vectors are used in training and 85 in testing.

The FCC method calculates the DOM based on the centre, not the distance to the surface of the hyperbox. This gives a better generalisation ability. Table 4 shows that a high recognition rate (up to 88%) for *fracture* was achieved using the FCC method.

The (3,5,6) configuration gave a 60% overall recognition rate. It increased the recognition rate of *fracture* and *ok* data in this case. However there is a small reduction for *good* data. The improvement was 8% compared to the AL method.

VI. Conclusion

Fuzzy rule based classifiers are fast to train and can be used in complex data sets. This paper proposes an improved classifier, namely the fuzzy clustering classification (FCC) method. It is developed based on the work presented by Abe and Lan (1995). The major differences are the use of an improved degree of membership function and clustering of inputs prior to fuzzy classification. The performance of the FCC and AL are showed using the Fisher Iris data set and an oil data set.

In the Iris data set, the AL method produces good results due to the simplicity of the data set. The FCC method does not show any marked improvement.

The oil data set is relatively complex. The FCC method produces 8% to 20% improvement over the AL method in our experiments. The FCC also provides a flexible option to configure the fuzzy network so that the maximum classification may be obtained for a given class.

In order to develop a better classifier, further work needs to be done to determine the optimal cluster configuration for a given problem. The incorporation of the boundary expansion technique (Abe and Lan, 1995) may also enhance the accuracy of classification.

VII. Acknowledgments

The authors would like to acknowledge the technical contributions from Prof L.T. Kóczy of the Technical University of Budapest. He was a Visiting Professor from January to March 1999 at the University of New South Wales, where he conducted research in fuzzy rules interpolation.

VIII. References

Abe, S. and Lan, M.S. (1995) “A Method for Fuzzy Rules Extraction directly from Numerical Data and Its Application to Pattern Classification,” *IEEE Trans. on Fuzzy Systems*, 3(1): 18-28.

Bezdek, J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press: New York.

Simpson, P.K. (1992) “Fuzzy Min-Max Neural Networks – Part 1: Classification,” *IEEE Trans. on Neural Networks*, 3(5): 776-786.

Sugeno, M. and Yasukawa, T. (1993) “A Fuzzy-Logic-Based Approach to Qualitative Modeling,” *IEEE Trans. on Fuzzy Systems*, 1(1): 7-31.

Wong, P.M., Gedeon, T.D. and Taggart, I.J. (1997) “Fuzzy ARTMAP: A new tool for lithofacies recognition,” *AI Applications*, 10(3): 29-39.