Data Mining using Modified GFMM Neural Network

Supriya U. Kulkarni
B. Tech. (Fourth Year Student)
Department of Information Technology
S.G.G.S. Institute of Engineering and Technology
Nanded - 431606, Maharashtra, India

Balaji S. Shetty
Assistant Professor
Department of Information Technology
S.G.G.S. Institute of Engineering and Technology
Nanded - 431606, Maharashtra, India

ABSTRACT
The fuzzy neural networks are adaptive, learns quickly and are highly suitable in decision making where uncertainty is involved. In this paper the Modified General Fuzzy Min-Max Neural Network (MGFMMNN) is described which is experimented for the data mining tasks such as classification and clustering. The MGFMMNN utilizes fuzzy sets as pattern classes in which each fuzzy set is a union of fuzzy set hyperboxes. It is an extension of the general fuzzy min-max (GFMM) neural network proposed by Gabrys and Bargiala. The data mining tasks such as classification and clustering have been studied using MGFMMNN and Fisher Iris data set. Further, MGFMMNN is trained using Hepatitis Data Set to verify its classification and recognition ability. The results obtained are awfully persuading and confirms the effectiveness of the proposed system. The technique proposed is quick and reliably deployable in the applications that need classification and clustering.

General Terms
Data Mining, Fuzzy Systems, Neural Networks, Pattern Recognition.

Keywords
Classification, Clustering

1. INTRODUCTION
The classification and decision making is a form of data analysis that extracts information describing important data classes. Such models are entitled as classifiers which predict categorical class labels. For instance, we can construct a categorization model to sort out bank loan applications as either secure or unsafe. Such study can help us with a better understanding of the data at large and consequently in the decision making.

At present, the fuzzy neural networks (FNN), the synergistic combination of artificial neural networks (ANN) and fuzzy systems (FS), combines best of both the worlds, are commonly used in the classification and decision making [1]. The FNN combines the power of the ANN, such as learning, adaption, fault tolerance, parallelism and generalization with human being like thinking and reasoning using FS. An enormous literature is available on the FNNs which recommends various architectures and algorithms for the different applications.

Patrick K. Simpson proposed supervised learning neural network classifier known as fuzzy min-max neural network (FMN) that utilizes fuzzy sets as pattern classes where each fuzzy set is an aggregate (union) of fuzzy set Hyperboxes. This learning algorithm has the ability to learn on-line and in a single pass through the data. Its performance is evaluated for commonly used and well-known fisher iris data set [2]. Simpson has also proposed unsupervised fuzzy min-max clustering neural network (FMCN) in which clusters are implemented as fuzzy sets using membership function with a hyperbox core that is constructed from a min point and a max point [3]. Unlike FMN, the FMCN is unsupervised learning algorithm.

Gabrys and Bargiela have proposed general fuzzy min-max (GFMM) neural network for classification and clustering, which is a fusion of supervised and unsupervised learning [4]. In the continuation of fuzzy min-max neural network classifier, Kulkarni U. V. *et al.* have proposed fuzzy hyperline segment neural network classifier (FHLSNN). It utilizes fuzzy sets as pattern classes in which each fuzzy set is a union of fuzzy set hyperline segments [5]. The FHLSNN is supervised classifier. This classifier when applied for rotation invariant handwritten character recognition problem, it performed better compared to unsupervised four layer feed-forward fuzzy neural network (FNN) of Kwan and Cai, [6] and GFMM algorithm in terms of recognition rate, training time and recall time per pattern. \

U.V. Kulkarni *et al.* have also proposed unsupervised fuzzy hyperline segment clustering neural network (FHLSCNN). The performance of FHLSCNN is found superior over FMCN when applied for clustering of Fisher Iris data [7]. P. M. Patil, U. V. Kulkarni and T. R. Sontakke have proposed general fuzzy hyperline segment neural network (GFHLSNN). The GFHLSNN combined, FHLSNN and FHLSCNN, by combination of both supervised and unsupervised learning, which is referred as general learning. Therefore, like GFMM it can be used for pure classification, pure clustering and hybrid classification/ clustering [8].

The modifications in the architecture and learning algorithm of GFMM have been proposed to improve its performance by many researchers. They have suggested different ideas. Kim and Yang proposed a weighted fuzzy min-max neural network. The membership function proposed in this algorithm considers the occurrences of input pattern along with frequency of occurrences. In order to overcome low automation degree and to achieve the remarkable generalization capability Antonello Rizzi *et al.* proposed the two new learning algorithms for GFMM as the adaptive resolution classifier and also its pruning version.

Nandedkar and Biswas have suggested the use of overlapped compensatory neurons and the containment compensatory neurons to resolve the membership confusion in the overlapped area [9].

Reza Davtalab *et al.* proposed novel fuzzy Min-Max neural classifier that uses modified compensatory neurons. This classifier is suitable for online training, uses supervised single-pass method. In this technique for handling overlapping areas that are mainly created in borders, a modified compensatory node with a radios-based transition function is used. This modification improved the classification accuracy in discriminating cases [10]. H. Zhang *et al.* proposed data-core based fuzzy min-max neural network. In this algorithm innovative membership function for classifying neurons is

proposed on the basis of noise, the geometric center of the hyperbox and the data core and the performance is verified with different benchmark problems and also with pattern classification of oil pipeline [11].

The multilayer perceptron (MLP) with error backpropagation is awfully popular and widely used neural classifier used for lots of problems worldwide by most of the researchers [12]. However, the use of MLP with error backpropagation suffers from various drawbacks. This learning algorithm is inflexible and requires huge training time compared to modern FNNs proposed by different researchers as mentioned above.

This has motivated us to make use of FNN and to explore for some better solution to the problem of classification and clustering. In spite of various FNNs, the GFMM is chosen to explore it for data mining tasks as it is one of the most admired FNN algorithms.

This paper is organized as follows. In Section 2, the architecture of the GFMM [4] neural network is enlightened and reproduced with brief explanation for the convenience of the reader. Few modifications have been suggested in the original GFMM. The MGFMMNN learning algorithm with modifications is explained in Section 3. The experimental procedure, simulation results, description of data set and discussions on the results are presented in the Section 4. Finally, section 5 gives conclusions and future scope for further research. .

2. ORIGINAL GFMM

The GFMM neural network algorithm after training constructs the neural network as shown in [4]. The input layer consists of 2**n* nodes to accommodate fuzzy input pattern having lower and upper bound vectors. The processing nodes (neurons) in the input layer does not do any processing. Hence the transfer function results in output equal to input.

The second layer shows the hyperboxes created, and possibly expanded and contracted during training phase of the network. Hyperboxes are fuzzy sets and characterized by min and max points along with membership function.

The min and max points of the hyperboxes in this layer are stored in **V** and **W**, the two matrices that are also created during training phase. These min and max points of the hyperboxes are represented by the connections between first and second layer.

The transfer function of the nodes in this layer is membership function. Hence, every hyperbox uses membership function to compute its output which is fuzzy membership value. The hyperbox membership function uses its min, max points and the input pattern to calculate output. The output of hyperbox node is fuzzy and is one if pattern is included by fuzzy set hyperbox. Otherwise, the membership value decreases as the distance of the pattern increases form the hyperbox. The rate of decrease depends on the user defined sensitivity parameter, $0 < \gamma \leq 1$. During training the number of hyperboxes constructed in this layer depends on the user defined parameter Θ, which puts bound on the maximum size of the hyperboxes.

Each third layer node represents class. The output of the third layer node represents the degree (possibility value) with which the input pattern belongs to that particular class. The connections between second and third layer are binary. The weight assigned to the link/connection is one if hyperbox belongs to that class, else the weight is zero. The node $\boldsymbol{c_0}$ represents all the unlabeled hyperboxes from the second layer. Each class node in the output layer calculates output by taking fuzzy union of the weighted outputs of the hyperboxes of that class.

The steps in the GFMMNN learning algorithm are initialization, hyperbox growth, overlap check and hyperbox retrenchment. These three steps are just mentioned here for the convenience of the reader. It is advised to refer [4] for details of these learning steps in detail. In the proposed idea, which is discussed in the forthcoming section 3, few changes are suggested, leading to improvement in space and time complexity of the original GFMM.

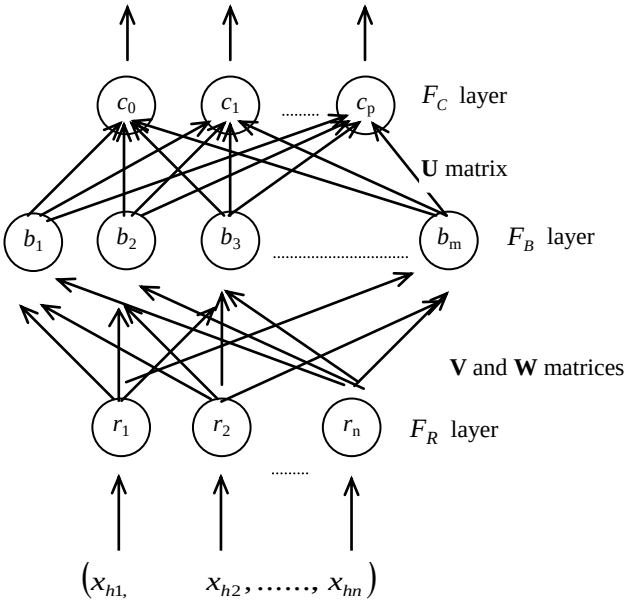

Fig 1: Topology of MGFMMNN Neural Network

3. MGFMMNN ALGORITHM

The fuzzy input pattern may have lower and upper bounds as defined in [4]. Therefore, Gabrys and Bargiala have used 2**n* nodes in the input layer to accept the input pattern with these two bounds. However, it is very rare to see such patterns and most of the times while experimenting with real world applications, in geometrical sense, the input pattern is a point in *n*-dimensional Euclidean space. Therefore, it is assumed that input pattern is a point with equal lower and upper bounds. Hence input is represented as,

$$X_h = [X_h^l \; X_h^u], \text{ but where } X_h^l = X_h^u . \qquad (1)$$

This assumption reduces the requirement of the number of nodes in the input layer to *n*. The modified GFMM neural network uses only *n* nodes in the input layer as shown in figure 1. The input layer is designated as F_R layer and accepts *n* dimensional input pattern whose lower and upper bound are always equal. This proposed modification requires changes in the representation of hyperbox nodes in the second F_B layer and as in [4] this layer consists of the hyperboxes that are created during training. The representation of hyperboxes is as used by Simpson [2-3]. For the convenience of the reader this representation is reproduced in figure 2, where the implementation of a hyperbox and its associated membership function as a neural network assembly is shown for the j^{th} hyperbox node. The input nodes accept each dimension of the h^{th} input, $A_h = X_h^l = X_h^u$. There are two connections from each input node to the output node, one connection represents the min value for that dimension and the other connection represents the max value for that dimension.

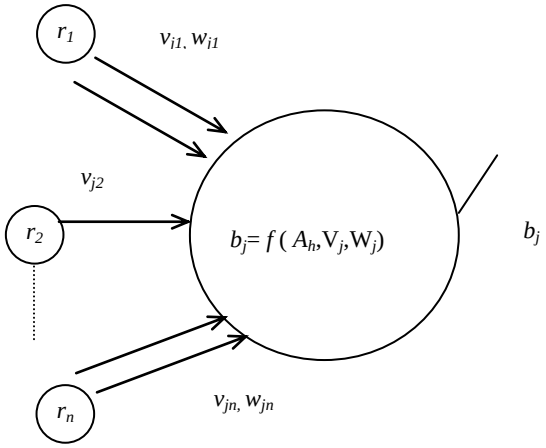

Fig 2: Hyperbox Implementation

This proposed modification not only reduces the input nodes to half but also leads to improvement in the space and time complexity of the algorithm. The input representation in [4] requires space to store lower and upper bounds even if all the samples in data set are points. This representation requires twofold space.

The proposed modification further leads in improvement of time complexity. It can be seen by calculating the time saved during training and recall phases, if the network is implemented in software. Let D be the number of patterns in the dada set used for training, τ be the time in seconds required for computation of output for each of the nodes in input layer. Hence, time saved in one iteration is $n\tau$. Total time saved during training for one pass with D data samples will be $n\tau D$. Assuming training is performed in N passes the total time saved will be $N*\tau*n*D$ seconds. However, during recall phase time saved per pattern will be only $n\tau$ seconds.

As in [4], the connections between F_B and F_C are binary and each F_C node represents class.

Further, the equations of case 1 and 2 that are used in [4] during overlap test and removal are modified as,

$$\text{case 1: } v_{ji} < v_{ki} \le w_{ji} < w_{ki} \text{ and}$$

$$\text{case2 : } v_{ki} < v_{ji} \le w_{ki} < w_{ji}, \qquad (2)$$

since it is observed that previous conditions in [4] are unable to find dimension of minimum overlap if $v_{ki} = w_{ji}$ in case 1 and $v_{ji} = w_{ki}$ in case 2. The correction is obtained by equation (2). Except these changes all other things have been kept consistent with the original GFMM neural network algorithm [4].

4. EXPERIMENTAL RESULTS

The Fisher Iris Database is [13] standard and very well-liked data set which is used by numerous researchers for verifying the performance of the their proposed algorithms for classification and clustering. This is one of the best known databases to be found in the pattern recognition literature containing samples of three categories. One class is linearly separable from the other 2; however, the latter are not linearly separable from each other.

This data set is prepared by Ronald Fisher. In all four features of 150 irises samples of three classes/types are recorded. Class 1 is Setosa; Class 2 is Verginica; and class 3 is Versicolor. Features measured are petal width (PW), petal length (PL), sepal width (SW), and sepal length (SL) along with its class. This data set is downloaded from repository [13].

The algorithm is implemented using Java with Eclipse as an integrated development environment. The MGFMMNN algorithm uses normalized pattern space i.e. I^n. Therefore, the data set is normalized such that all the values are relatively scaled to lie in the range 0 to 1.

A. Example 1—Classification of Fisher Iris Data Set

In the first experiment all 150 available data patterns have been used for training and testing; however in the second experiment 25 randomly selected patterns from each class have been used for training and the remaining 75 for testing.

The first experiment created 82 hyperboxes for $\Theta = 0.037$ with no misclassifications. To verify the effect of sensitivity parameter γ on learning, the experiment was carried out by varying γ from 1 to 50 in steps of 5, it created same number of hyperboxes. However, further increase in its value resulted in increase of number of hyperboxes because for large values of gamma fuzzy set hyperboxes starts returning very low membership for the patterns falling outside the hyperboxes. Few of these results obtained are listed in the Table 1.

Table 1. Effect of sensitivity parameter on learning

$\Theta = 0.037$						
γ	1	10	20	30	40	50
Hyperboxes	82	82	82	82	115	115

In the range $0 < \gamma \leq 1$, the number of hyperboxes remained consistent to 82 yielding no misclassification. Hence, in normalized pattern space, I^n, one can optimally use the value in the range, $0 < \gamma \leq 1$.

In the second experiment, MGFMMNN created 28 hyperboxes for $\Theta = 0.0632$, and $\gamma = 1$ with no misclassifications. In this experiment to verify the effect of Θ, i.e. the maximum size of the hyperbox on the learning, it is varied in the range $0 \leq \Theta \leq 0.1$ in steps of 0.01 and few readings with steps of 0.1. The number of hyperboxes created along with misclassifications are noticed for these different values of Θ. The results obtained are listed in Table 2.

Table 2 shows that increase on bound of maximum size of hyperbox, decreases the number of hyperboxes created during training process with increase in resulting misclassifications.

Table 2. Effect of hyperbox size on learning

$\gamma = 1$		
Θ	Hyperboxes	Misclassifications
0	74	0
0.01	74	0
0.02	62	0
0.03	49	0
0.04	33	0
0.05	33	0
0.06	29	0
0.07	21	1
0.08	19	4
0.09	14	3
0.1	14	3
0.2	7	6
0.3	3	7

B. Example 2—Clustering of Fisher Iris Data Set

To check the performance of clustering, Fisher Iris data set is used and modified GFMM neural network is trained using 150 patterns without using class information means treating all patterns as if belonging to unlabeled class. The confusion matrices obtained after clustering are given through tables 3 to 5.

Table 3. No confusion for 120 hyperboxes

Hyperbox size = 0.0251 Number of Hyperboxes = 120 Overall Confusion = 0 %			
Class	1	2	3
1	100%		
2		100%	
3			100%

Table 4. 16.66% confusion for 81 hyperboxes

Hyperbox size = 0.03 Number of Hyperboxes = 81 Overall Confusion = 16.66 %			
Class	1	2	3
1	100%		
2	30%	68%	2%
3	18%		82%

Table 3 and 4 shows that increase on bound of maximum size of hyperbox, decreases the number of hyperboxes created during training process with increase in resulting misclassifications hence in overall confusion after clustering. Similar results obtained by further increasing hyperbox size resulting decrease in number of hyperboxes resulting in further increase in confusion are shown in Table 5 and 6.

Table 5. 29.33% confusion for 57 hyperboxes

Hyperbox size = 0.04 Number of Hyperboxes = 57 Overall Confusion = 29.33%			
Class	1	2	3
1	100%		
2	52%	48%	
3	56%		44%

Table 6. 44% confusion for 39 hyperboxes

Hyperbox size = 0.05 Number of Hyperboxes = 39 Overall Confusion = 44 %			
Class	1	2	3
1	100%		
2	72%	28%	
3	60%		40%

To have the more clarity the chart is provided in figure 3 which shows that during clustering as we increase hyperbox size, number of hyperboxes decrease, however overall confusion increases.

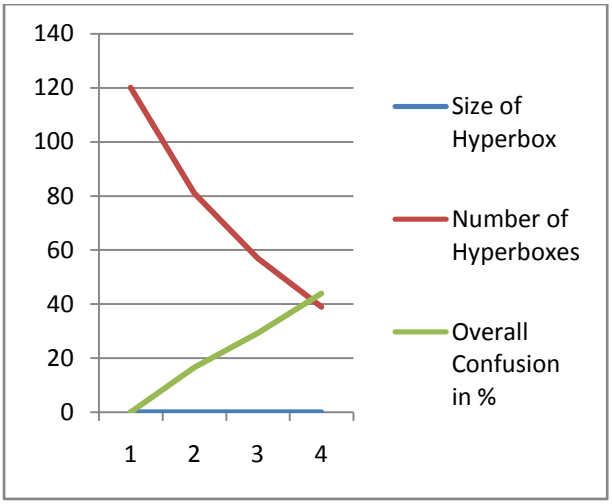

Fig 3: Performance of clustering

C. Example 3— Classification using Hepatitis Data Set

The number of instances in this data set are 155 [13]. The number of attributes including class are 20. The data set includes two classes, Die and Live. Some of the attributes in few of the instances are missing. Therefore, these missing attributes are replaced by the average value of that attribute over all the instances of that class.

Train data consisting of 120 instances is used during learning. As required it took few passes during which Θ is adjusted appropriately to train the MGFMMNN. Remaining 35 sample are used for testing the recognition ability of the classifier. The trained MGFMMNN yields 100% classification accuracy. However, recognition rate obtained during testing is 88%.

5. CONCLUSION AND FUTURE SCOPE

Like original GFMM, the MGFMMNN can be used for data mining tasks such as classification and clustering. As long as there are no matching sample belonging to two different classes, the recognition rate for training set is 100%.

Since all the manipulations involve only simple compare, add, and subtract operations like GFMM, the resulting algorithm is extremely efficient due to suggested modifications. The assessment of space and time complexity shows that the MGFMMNN is superior than original GFMM.

Further modifications in original GFMM are possible by removing bound on maximum size of hyperbox and allowing creation of hyperboxes till they grow without overlapping and hence indirectly the algorithm automatically shall put bound on the size instead of user defined approach by providing it in the beginning.

6. REFERENCES

[1] J. M. Zurada, Introduction to Artificial Neural Systems, Bombay: Jaico Publishing House, 1994.

[2] Simpson, P. K. 1992. Fuzzy min-max neural networks – Part 1: Classification. IEEE Trans. on Neural Networks. Vol. 3, No. 5, 776-786.

[3] Simpson, P. K. 1993. Fuzzy min-max neural networks – Part 2: Clustering. IEEE Trans. on Fuzzy Systems. Vol. 1, No. 1, 32-45.

[4] Gabrys, B. and Bargiela, A. 2000. General fuzzy min-max neural network for clustering and classification. IEEE Trans. Neural Networks. Vol. 11, No. 3, 769-783.

[5] Kulkarni, U. V., Sontakke, T. R., and Randale, G. D. 2001. Fuzzy hyperline segment neural network for rotation invariant handwritten character recognition. In Proceedings of IEEE: INNS: IJCNN 2001 Joint International Conference on Neural Networks. Washington DC, USA. Vol. 4, 2918-2923.

[6] Kwan, H. K. and Cai, Y. 1994. A Fuzzy neural network and its application to pattern recognition. IEEE Transactions on Fuzzy Systems. Vol. 2, No. 3, 185-192.

[7] Kulkarni, U. V., Sontakke, T. R., and Kulkarni, A. B. 2001. Fuzzy hyperline segment clustering neural network. Electronics Letters. Vol. 37, No. 5, 301-303.

[8] Patil, P. M., Kulkarni, U. V., and Sontakke, T. R. 2002. General Fuzzy Hyperline Segment Neural Network. IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia. Volume 4. 6.

[9] Nandedkar, A. V., and Biswas, P. K. 2007. A fuzzy min-max neural network classifier with compensatory neuron architecture. IEEE Transactions on Neural Networks. Volume 18. 42-54.

[10] Reza Davtalab, Mir Hossein Dezfoulian, and Muharram Mansoorizadeh. 2014. Multi-level fuzzy min-max neural network classifier. IEEE Transactions on Neural Networks. Volume 25. 470-482.

[11] Zhang, H., Liu, J., Ma, D. and Wang, Z. 2011. Data-core-based fuzzy min-max neural network for pattern classification. IEEE Transactions on Neural Networks. Volume 22, 2339-2352.

[12] Bose, N. K. and Liang, P. 1998. Neural Network Fundamentals with Graphs, Algorithms, and Applications. New Delhi: Tata McGraw-Hill.

[13] UCI repository of machine learning databases. 1998. University of California, Irvine. http://www.ics.uci.edu/mlearn/MLRepository.html