

Data Centroid Based Multi-Level Fuzzy Min-Max Neural Network

Shraddha Deshmukh*, Sagar Gandhi*, Pratap Sanap and Vivek Kulkarni

Abstract—Recently, a multi-level fuzzy min max neural network (MLF) was proposed, which improves the classification accuracy by handling an overlapped region (area of confusion) with the help of a tree structure. In this brief, an extension of MLF is proposed which defines a new boundary region, where the misclassification is more frequent. A methodology to classify patterns more accurately is presented. Our work enhances the testing procedure by means of data centroids, eventually, increasing the fuzziness in the decision making. We exhibit an illustrative example, clearly highlighting the advantage of our approach. Results on standard datasets are also presented to evidentially prove a consistent improvement in the classification rate.

Index Terms—Classification, Neurofuzzy, Fuzzy min-max neural network, Hyperbox.

I. INTRODUCTION

FUZZY set theory was proposed with the intended use to the fields of pattern classification and information processing [1]. Indeed, it has attracted many researchers, and their applications to real-life problems are of a great significance. Simpson [2] presented the fuzzy min max neural network (FMM), which makes the soft decisions to organize hyperboxes by its degree of belongingness to a particular class, which is known as a membership function. Hyperbox is a convex box, completely represented by min and max points. FMM classification results are completely characterized with the help of a membership function. Along with this elegant proposal, [2] also presented the characteristics for a good classifier, among which, nonlinear separability, overlapping classes and tuning parameters have proved to be of a great interest to a research community.

Simpson also presented a clustering approach using FMM in [3]. But many problems in real-life require

both classification and clustering. To address this issue, GFMM [4] brought this generality. Besides generality, the more significant contribution has proved to be modification to the membership function. The presented membership function computes the belongingness to the hyperbox so that the membership value decreases uniformly as we move away from the hyperbox. Another weakness of FMM was the patterns belonging to overlapped region, where the rate of misclassification is considerably high. The tuning parameter, theta (θ), which controls the size of a hyperbox, has a great impact on this overlapped region. Smaller theta values produce less overlaps producing high training accuracy, but the efficacy of the network gets compromised, and for larger theta values, accuracy gets decreased.

Multiple approaches were presented to tackle this problem. Earlier, the process of contraction [1][4] was employed, which used to eliminate all the overlapping regions. This method had the intrinsic problem of representing patterns not belonging to any of the hyperbox, in turn lessening the accuracy. Exclusion/Inclusion Fuzzy Classification (HEFC) network was introduced in [5], which further reduced the number of hyperboxes and increased the accuracy. Inclusion hyperboxes were used to represent patterns belonging to the same class, while exclusion hyperboxes were used to denote the overlapped region, treated as if it is a hyperbox. This notion is used as it is in almost all the newly introduced models [6][7][8][9]. Fuzzy Min-Max Neural Network Classifier with Compensatory Neurons (FMCN) was acquainted in [7]. Authors categorized the overlap into three parts, namely, full containment, partial overlap and no overlap, and then a new membership function to accommodate belongingness based on the compensation value. Authors also analyzed that neatly taking care of overlapped region automatically brings the insensitivity to the hyperbox size parameter, θ .

Data Core based Fuzzy Min-Max Neural Network (DCFMMN) [8] further improved upon FMCN. Authors eliminated the need of overlap categorization. They also suggest a new membership function based on noise, geometric center and data cores of the hyperbox. Wherein DCFMMN improved the accuracy in few cases, there are some serious drawbacks.

*: These authors contributed equally.

The authors are with Persistent Systems Ltd., India (e-mail: shraddhadeshmukh2@gmail.com; gandhi.sagar.gs@gmail.com; pratap_sanap@persistent.com; vivek_kulkarni@persistent.com)

Persistent System Ltd. (www.persistent.com) is a publicly listed company. Persistent LABS is the R&D group of Persistent Systems Ltd., which focuses on R&D activities involving Machine Learning, Video Analytics, Internet of Things (IoT) in addition to many other areas.

Persistent LABS is Recognized and Approved R&D Center by Department of Scientific and Industrial Research (DSIR), Govt. of India.

- 1) DCFMN introduces two new user controlled variables, ϵ and λ . ϵ is used to suppress the influence of the noise and λ is used to control the descending speed of the membership function. These two variables greatly impact the performance of the model and naturally, defining their values is a tedious job.
- 2) There exists an underlying assumption that noise within all the hyperboxes is similar, which may not be true. Moreover, the sequence of the training exemplars plays a role as well.
- 3) MLF conveys that this membership function is not always preferred, in that, it does not work well for high percentage of samples belonging to overlapped area.

Multi-level fuzzy min max neural network (MLF) [9] addresses the problem of overlapped region with an elegant approach. It uses separate levels for overlapping regions, and monotonically decreases the hyperbox size (θ). For most cases, MLF produces 100% training accuracy.

Though MLF achieves a significant milestone, entertaining testing accuracy is rather more important than training accuracy, as it greatly sways the usage of the algorithm in practical scenarios. In this brief, we identify and define a new boundary region, where misclassification rate is substantial. To the best of our knowledge, this kind of approach is presented for the first time, at least we did not come across any similar published work. Hence we propose a method, based on data centroids, to evidentially prove that handling this newly introduced area of confusion between hyperboxes of different classes significantly increases the testing accuracy.

The paper is organized as follows. MLF is reviewed in Section II. We introduced D-MLF algorithm in Section III. An illustrative example and comparative results of D-MLF with MLF model are presented in Section IV and V, respectively. Finally, conclusion is given in Section VI.

II. MULTI-LEVEL FUZZY MIN-MAX NEURAL NETWORK

Multi-level fuzzy min max neural network (MLF) is a classifier which efficiently caters misclassification of patterns belonging to overlapped region by maintaining a tree structure, which is a homogeneous tree [9].

In MLF training phase, exemplars are continuously recurred to form the hyperboxes and overlaps, each recursion resulting in one level. This recursive procedure is carried till the predefined maximum depth or till overlap exists. Hyperbox expansion, based on hyperbox

size controlling parameter (θ), is validated using equation (1) and expansion is carried out by equation (2).

$$\forall_i = 1 \dots D (\max(w_b^i, a^i) - \min(v_b^i, a^i)) \leq \theta \quad (1)$$

$$\begin{aligned} v_b^i &\leftarrow \min(v_b^i, a^i); & 1 \leq i \leq D \\ w_b^i &\leftarrow \max(w_b^i, a^i); & 1 \leq i \leq D \end{aligned} \quad (2)$$

Where, v_b^i and w_b^i are min point and max point of hyperbox B respectively, a^i is the i^{th} dimension of pattern A and D is the number of dimensions. Also, prior to each recursion, θ is updated using equation (3)

$$\theta_n = \mu \theta_p \quad (3)$$

Where, θ_n and θ_p thetas for next level and previous level, respectively and μ , being the value between 0 and 1, ensures that size of hyperbox in overlapped region is less than its previous level.

In the testing phase, overlap regions are first traversed recursively, to discover appropriate subnet to which a test pattern belongs to. Thence, in that level, a class of hyperbox having highest membership value with the hyperboxes in the discovered subnet, is selected as a predicted class.

MLF is able to achieve higher accuracy rates than previous FMM methods. This is due to an elegant treatment to the boundary region a confusion area. But, after training, there exists a room for yet another boundary. The region where membership function generates very close by values, it becomes difficult to assign a class with high degree of assurance. As per our experiments, MLF, and all the previous classifiers, do not perform well in this area. Hence, a definition of this new region, and a methodology to solve it is proposed.

III. DATA CENTROID BASED MULTI-LEVEL FUZZY MIN-MAX NEURAL NETWORK

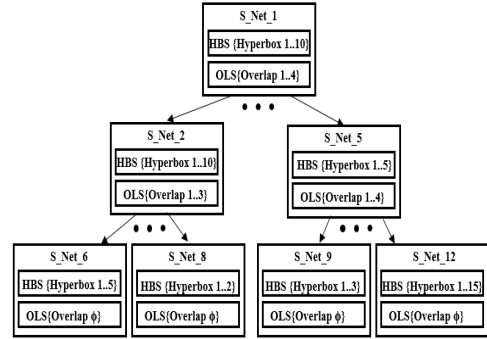


Fig. 1. D-MLF Subnet Tree Structure

In this section, we give details about a newly proposed algorithm, specifically, we define a new boundary region

generated due to trained network and propose a solution to correctly classify test patterns belonging to it.

Figure 1 describes the D-MLF structure, each node in S_net contains two segments, Hyperboxes Segment (HBS) and Overlapped Segment (OLS). HBS represents hyperboxes generated in that level, whereas OLS represents overlaps in that level. Along with hyperbox information, Data Centroid (DC).

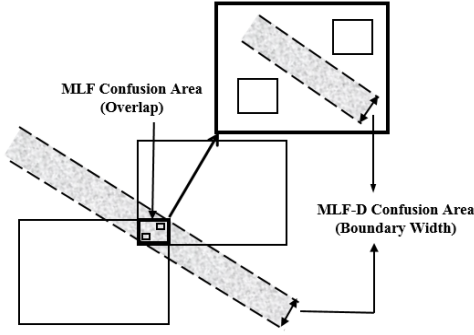


Fig. 2. Confusion area for classification

Figure 2 shows the area of confusion considered by MLF and D-MLF. We introduce a boundary region that exists between any two hyperboxes, where, according to our experiments, the rate of misclassification is comparatively high. In the proposed method, the recommendations of MLF are intact, in addition to it, we use distance with the data centroids to improve a classification rate in the anew boundary region.

A. D-MLF Training Phase

Similar to the MLF learning procedure, D-MLF maintains S_nets using HBS and OLS structures. First, all the patterns are passed through, resulting in creation and expansion of hyperboxes using equations (1) and (2). Then each hyperbox is checked with the rest of hyperboxes to detect the overlap using equation (4).

$$\forall i = 1 \dots D(\min(w_b^1, w_b^2) - \max(v_b^1, v_b^2) > 0) \quad (4)$$

Where w_b^1 and w_b^2 are the max points and v_b^1 and v_b^2 are the min points of the two hyperboxes, among which overlap is tested. Moreover, D-MLF adds a new step at the learning phase, known as data centroid (DC) computation, where DC of all input patterns belonging to each hyperbox is maintained in the HBS. DC is computed as follows:

$$DC_h = \frac{1}{n} \sum_{i=1}^n P_{hi} \quad (5)$$

Where DC_h is the data centroid of the h^{th} hyperbox, n is number of patterns belonging to h^{th} hyperbox and P_{hi} is the i^{th} pattern in h^{th} hyperbox.

If there exists an overlap, patterns belonging to the overlapped region are again sent to training procedure, where HBS and OLS creation takes place for the next level. This process of recursion is followed afterward to train all the patterns. Due to computation of OLS and process of finding patterns belonging to OLS, D-MLF and MLF are not single pass algorithms. In general, given the n overlaps in the first level, entire training data has to be traversed n times. Thereafter, in the subsequent stages, data belonging to overlapped region is traversed in order of magnitude of number of overlaps in that region. This is a novel finding, and contradictory to what MLF authors have mentioned [9].

Note that, the patterns belonging to overlapped region are not part of the DC computation. This step makes sure that training patterns balloting for more than one class are omitted in the final decision making.

Algorithm 1 D-MLF Training Pseudo code

Net = D-MLF-train(net, θ)

```

{
1: if (stop condition) then
2:   for all (h as hyperbox Net) do
3:      $DC_h = h.centroid / h.membercount$ 
4:   end for
5:   return null
6: end if
7: for all (sample  $\epsilon$  data) do
8:   if ((sample  $\leq$  hyperboxes)  $\vee$  ( $\exists$  hyperbox h: (h + sample  $<$   $\theta$ ))) then
9:     h.centroid += sample;
10:    h.membercount += 1;
11:   else
12:     create new hyperbox h;
13:     h.centroid = sample;
14:     h.membercount = 1;
15:   end if
16: end for
17: for all (i as an overlapped area) do
18:   sdata = samples which inhabit in i region;
19:   for all (s  $\epsilon$  sdata) do
20:     if (s  $\epsilon$  hyperbox  $H_i$ ) then
21:       Hi.centroid -= s;
22:       Hi.membercount -= 1;
23:     end if
24:     create an overlap-box as  $OL_i$  and add to OLS
25:      $Snet_i = D-MLF-train(sdata, \mu\theta)$ ;
26:     link  $OL_i$  to  $Snet_i$  with link  $e_i$ ;
27:   end for
28: end for
}
```

B. D-MLF Testing Phase

The original MLF used a decision making based on the subnets decision. The selected subnet need not be a leaf node in the tree. We do not alter this model, rather enhance the process of how subnet marks the choice.

Membership function mentioned in the equation (11) is used against overlapped boxes. After recursively traversing the OLS an appropriate subnet is discovered, to which test pattern belongs to. A membership function explained in the equation (6) is used, this time, to compute the membership with the hyperboxes within the selected subnet.

$$b_j(A_h) = \min(\min([1 - f(a_h^i - w_i^j, \gamma_i)] [1 - f(v_i^j - a_h^i, \gamma_i)]))$$

$$f(x, \gamma) = \begin{cases} 1 & \text{if } x\gamma > 1 \\ x & \text{if } 0 \leq x\gamma \leq 1 \\ 0 & \text{if } x\gamma < 0 \end{cases} \quad (6)$$

Where b_j represents belongingness of sample A_h with j^{th} hyperbox. x is a difference between min and max point with sample A_h and γ is a tuning parameter to control fuzziness.

Within these membership values, hyperboxes with highest two values are selected to define a boundary. Medial region of these hyperboxes, controlled by ψ , is treated as a boundary region. ψ is a user controlled variable, mentioned in the percentage value. At this point, it is necessary to check if test pattern belongs to the boundary region. We define $\alpha 1$ and $\alpha 2$ as incident angles between test pattern and two hyperboxes, respectively. Inclusion value is evaluated as follows:

$$Inclusion Value = \begin{cases} 1, & (\alpha 1 < 90) \text{ and } (\alpha 2 < 90) \\ 0, & \text{Otherwise} \end{cases} \quad (7)$$

Further, Based on the inclusion value, output class is chosen. If pattern exists in the area outside of the defined boundary, we simply follow a path of MLF, and classify the pattern based on the maximum membership value, which is already computed. If the pattern belongs to the boundary region, Euclidean Distance [10] between test pattern and the data centroids of the selected hyperboxes is computed. Hence, centered on the inclusion value, the output of the network is denoted as either the class of maximum g_s among all the hyperboxes, or as a minimum of the distances of the topmost two hyperboxes

$$Output = \begin{cases} g_s^i, & \text{if inclusive value} = 0 \\ f_s^i, & \text{if inclusive value} = 1 \end{cases} \quad (8)$$

Where g_s^i is given by;

$$g_s^i = \begin{cases} c_s^i, & \text{if } e_s o_s = 1 \\ 0, & \text{Otherwise} \end{cases} \quad (9)$$

Where c_s^i is the i^{th} class membership for the test sample in s subnet, e_s is edge between subnet s and the corresponding overlap box that enables the subnet if test sample is in this overlap box. And o_s is the output of OLS, which is given by equation (10)

$$o_s = \max_{1 \leq j \leq m} l_j \quad (10)$$

Where m is number of overlap boxes in OLS and l_j is membership function of the j^{th} overlap box for test sample A_h , given by equation (11)

$$l_j = \begin{cases} 1, & \text{if } A_h \text{ is inside } OL_j \\ 0, & \text{Otherwise} \end{cases} \quad (11)$$

And f_s^i is given by equation(12)

$$f_s^i = \min(D_s^1, D_s^2) \quad (12)$$

Where D_s^i is the Euclidean Distance computed amongst sample s and the data centroid of the topmost i^{th} hyperbox using equation (13)

$$Euclidean Distance = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (13)$$

Algorithm 2 D-MLF Testing Pseudo code

Out = D-MLF-test(net, sample)

```
{
1: for all ( $ol_i$  as OL box in OLS) do
2:   if (sample is in  $ol_i$ ) then
3:     out = D-MLF-test ( $S_{net_i}$ , sample);
4:     return null;
5:   end if
6: end for
7: MV = [];
8: for all ( $B_i$  as hyperbox in HBS) do
9:   MV += membership (sample,  $B_i$ );
10: end for
11: [h1, h2] = [max(MV), max(MV (MV < max(MV)))]
12: if (Inclusion value == 1) then
13:   d = EuDistance(sample, h1.DC, h2.DC);
14:   out = min(d).class;
15: else
16:   out = max (MV).class;
17: end if
}
```

IV. ILLUSTRATIVE EXAMPLE

In this illustration, we describe the effectiveness of the proposed model, clearly pointing out the identification and handling of the stated area of confusion.

Figure 3 illustrates the 2-diamentional data space. We consider 14 data samples for training and 6 data samples

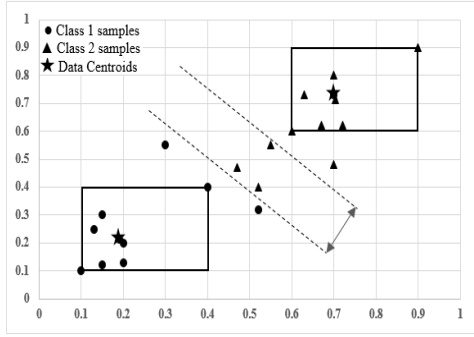


Fig. 3. Illustrative example signifying advantage of D-MLF

for testing. Hyperbox size parameter (θ) is fixed at 0.3 and a boundary parameter (ψ) is fixed at 5%. Both MLF and D-MLF create two hyperboxes at 0th layer. D-MLF also computes data centroids (DC) for each hyperbox, $h1$ and $h2$. Here, data centroids of $h1$ and $h2$ are $DC1 = (0.19, 0.21)$ and $DC2 = (0.70, 0.71)$, respectively. Patterns which do not belong to boundary region are classified correctly by MLF. But when it comes to boundary region, it fails to correctly classify the patterns. Whereas the proposed D-MLF works better in the boundary region as well, as its decision making is not completely based on the membership value, but it also considers data centroids.

It can be noted that the patterns in the above example are not uniformly spread out. Which is a very common scenario in real-world examples. It occurs because of the dominance of the parameters such as outliers, temporal nature of the variables, etc. Due to them, most of the times, the patterns within the overall data, and in case of Fuzzy Min Max hierarchy, within hyperboxes, will not be steadily spread across all the dimensions. As demonstrated above, our proposed method treats them elegantly, without many of the modifications to the state of the art.

V. EXPERIMENTAL RESULTS

Performance of proposed method (D-MLF) is studied on the basis of the classification rate. Various experiments were carried out to test D-MLF on different standard datasets.

Standard datasets such as Iris, Glass, Wine, Wisconsin Breast Cancer (WBC), Wisconsin Diagnostic Breast Cancer (WDBC) and Ionosphere were used. These datasets were obtained from the UCI repository of machine learning databases [11]. In these experiments, hyperbox size parameter (θ) was chosen as 0.2, 0.5 and 0.9. This was to perform the measurements across the spectrum. As we increase the size of the hyperbox, the number of overlaps increase, and so does the misclassification rate. We split the data evenly for training

and testing. The average results are shown over 100 experiments. For each iteration, training and testing data is chosen randomly.

Table 1 shows results, we compare our results to MLF method, as it has been already proven to perform better than the previously proposed FMM methods [9].

TABLE I
RESULTS

Dataset	Theta	MLF	MLF-D
Iris	0.2	96	97.33
	0.5	93.33	96
	0.9	94.67	96
Wine	0.2	91.01	93.25
	0.5	95.5	98.87
	0.9	89.89	95.5
Glass	0.2	85.04	89.72
	0.5	87.85	92.52
	0.9	88.78	89.72
WBC	0.2	94.18	95.35
	0.5	94.48	95.34
	0.9	93.05	94.35
WDBC	0.2	92.6	94.71
	0.5	90.84	93.31
	0.9	92.25	93.31
Ionosphere	0.2	87.8	88.40
	0.5	87.84	90.92
	0.9	87.29	89.05

VI. CONCLUSION

In this brief, we introduced a new boundary region and distance based MLF classification method to handle patterns belonging to that boundary region. A data centroid based method, D-MLF, minimizes significance of outliers and similar errors in decision making. It has been evidentially proven that the proposal outperforms all the previously proposed FMM methods. More importantly, we have proposed a model suited for data in the real world, extending the state of the art. D-MLF will help humongous application areas such as Security, Natural Language Processing, Biomedical Reasoning, etc.

REFERENCES

- [1] L. A. Zadeh, "Fuzzy sets", Information and Control, vol. 8, no. 3, pp. 338-353, 1965.
- [2] P. K. Simpson, "Fuzzy min-max neural networks. Part 1. Classification", IEEE Trans. Neural Network, vol. 3, no. 5, pp. 776786, Sep. 1992.
- [3] Simpson, P. K., "Fuzzy Min-Max Neural Networks - Part 2: Clustering", IEEE Trans Fuzzy Systems 1, 3245 1993.
- [4] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification", IEEE Trans. Neural Networks, vol. 11, pp. 769783, 2000.
- [5] Bargiela, W. Pedrycz, and M. Tanaka, "An inclusion/exclusion fuzzy hyperbox classifier", Int. J. Knowl. Based Intell. Eng. Syst., vol. 8, no. 2, pp. 9198, 2004.
- [6] A. Rizzi, M. Panella, and F. M. F. Mascioli, "Adaptive resolution min-max classifiers", IEEE Trans. Neural Netw., vol. 13, no. 2, pp. 402414, Mar. 2002.

- [7] A. V. Nandedkar and P. K. Biswas, "A fuzzy min-max neural network classifier with compensatory neuron architecture", *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 4254, Jan. 2007.
- [8] H. Zhang, J. Liu, D. Ma, and Z. Wang, "Data-core-based fuzzy minmax neural network for pattern classification", *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2339-2352, Dec. 2011.
- [9] R. Davtalab, M. H. Dezfoulian and M. Mansourizade, "Multi-level fuzzy min-max neural network classifier", *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 470-481, Mar. 2013.
- [10] W. Bezdel and H. J. Chandler, "Results of an analysis and recognition of vowels by computer using zero-crossing data", *Proc. Inst. Elec. Eng.*, vol. 112, pp. 2060-2066, Nov. 1965.
- [11] K. Bache and M. Lichman., UCI machine learning repository, School Inf. Comput. Sci., Univ. California, Irvine, CA, USA., 2013. [Online Available] <http://archive.ics.uci.edu/ml>