

GPU PARALLELIZATION OF FUZZY MIN-MAX CLUSTERING NEURAL NETWORK FOR PATTERN RECOGNITION

P. S. DHABE & N. R. YENGUL

Department of Computer Science, Vishwakarma Institute of Technology, Pune, Maharashtra, India

ABSTRACT

Clustering is data mining technique, of grouping unlabelled data by identifying underlying structure in it. Traditional clustering algorithms like K-means requires the number of clusters to be formed as an input, and it is time-consuming iterative process. Therefore, there is a need for new adaptive clustering algorithms, with less time complexity. In this paper we propose two new things, one is to modify existing overlap and contraction of hyper boxes in unsupervised Fuzzy Min-Max clustering neural network and another is to accelerate its execution on modern GPUs using CUDA. The modifications to original FMN we call it as MFMN, found superior to FMN in terms of detection of overlap and contraction process. We also proposed GPU parallel implementation of MFMN using CUDA and obtained 21 and 6 times speedup, for training and testing, respectively and 95% gain in time for training phase. Thus, we recommend MFMN and its GPU parallelization for clustering larger data sets.

KEYWORDS: Fuzzy Logic, Artificial Neural Network, Clustering, Fuzzy Min-Max Neural Network, GPU

INTRODUCTION

The Fuzzy Set Theory [19] was proposed by Lofty Zadeh in 1965. Since then, cluster analysis is highly associated with fuzzy logic. The first fuzzy clustering algorithm known as a Fuzzy C –Means Clustering was proposed by J.C. Dunn in 1973 [20], and improved by J.C. Bezdek in 1981 [12]. Fuzzy set acquired human-like decision-making system, i.e. instead of giving the full membership of a particular class; it gives partial membership for more than one classes. Fuzzy Logic provides a simple way to arrive at a definite conclusion based on vague, ambiguous, imprecise, noisy or missing input information.

The fuzzy logic owns capability of handling uncertain information, and able to represent complex class boundaries. On the other hand, Neural Network shows an ability to learn from a process, and computationally efficient due to its parallel nature [21]. The fuzzy sets and neural network can effectively merged by utilizing neural network processing elements as a fuzzy set [1]. Neural network compares and input pattern with a set of stored exemplars, or stored vectors. The closer the input pattern matches with stored exemplars, the higher the output of corresponding processing element in the network. Assuming transfer function of each processing element of a neural network utilizes as fuzzy set membership function.

The Fuzzy Min-Max Neural Network (FMNN) is a special type of Fuzzy Neural Network. The FMNN was introduced by Simpson, in two parts – Classification [13] and Clustering [1] in 1992 and 1993 respectively. FMNN is applicable in many areas like color face segmentation [14], Satellite Infrared Image Clustering [15], Iris Recognition

[16], Oil Pipeline Leakage Detection [17], power quality monitoring [18]. FMNN is adaptive clustering algorithm. Unlike, most popular clustering algorithms, k-means, and C-means, one need to define the number of clusters prior to learning, but in FMN clusters are created dynamically during learning. In FMN pattern clusters are represented by Hyper box fuzzy sets. The hyperbox fuzzy set is represented by min point (near the origin) and max point (away from the origin) in n dimensional pattern space. Every hyperbox is associated with membership value. Belonging to a pattern in a hyper box fuzzy set depends only on two parameters hyper box size θ and sensitivity parameter γ , one represents the size of the hyperbox and other represents fuzziness of the hyperbox. This paper focuses on improving clustering ability of FMN and accelerating its performance by implementing it on GPU rather than Central Processing Unit (CPU) of system.

This paper is organized in such a way that, Section II describes, Simpson's FMN algorithm [1]. Section III analyses the FMN and identifies the problems in original algorithm followed by Section IV, which illustrates proposed modifications in FMN and its influence on performance. Section V, evaluates the performance of MFMN, on various datasets. Section VI gives detail explanation of, parallel implementation of MFMN, and performance improvement in terms of time, and with respect to the different input datasets are illustrated in the Result section. The following section briefly introduces about GPU and NVIDIA CUDA (Compute Unified Device Architecture).

Introduction to GPU and CUDA

GPU is a massively parallel hardware architecture, consisting of 1000s of the core, designed for accelerated computing. The main difference between CPU and GPU is, CPU have many powerful cores (< 100) and GPU is having less powerful multiple cores (> 1000). GPU cores are data parallel (i.e. simultaneous execution of the same function on elements of data) and CPU cores are task parallel (i.e. simultaneous execution of a different function on same of different elements of data).

CUDA is a parallel computing platform and programming model invented by NVIDIA in November 2007. CUDA is a powerful programming language, which accelerates sequential programs parallel. GPU programming is heterogeneous programming, it means GPU program consists of CPU plus GPU code [22]. The parallel code executes on GPU leaving the sequential code on CPU for execution. We have implemented MFMN in CUDA C/C++ and higher-level language C for GPU and CPU respectively.

ORIGINAL FUZZY MIN – MAX CLUSTERING NEURAL NETWORK

FMN is an unsupervised learning model. FMN constructs hyperboxes as fuzzy sets (Pattern Clusters). Every hyperbox is associated with min-max points and membership function. Hyperboxes are generated during learning as a processing elements of a neural network. An input layer F_r contains n processing elements (one for each dimension) and accepts h^{th} input training pattern $A_h = (a_{h1}, a_{h2}, \dots, a_{hm}) \in I^n$. An output layer F_e represents hyperbox fuzzy sets. The weights associated with link between input layer and output layer, are min-max points of corresponding hyperbox (two links for each connection). The min and max point of hyperbox is stored in vector V and W respectively.

The further sections in this paper will elaborate FMN in detail. The Section A gives formulation of fuzzy membership function and variation in sensitivity parameters γ . Section B and C describes training and testing phases of FMN algorithm respectively.

Fuzzy Membership Function

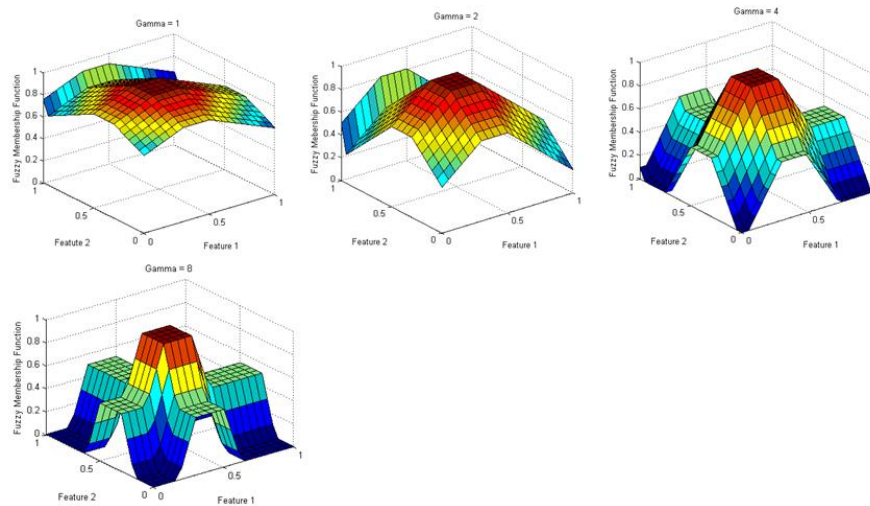
The membership function of hyper box defined in (1).

$$b_j(A_h, V_j, W_j) = \frac{1}{n} \sum_{i=1}^n [1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)] \quad (1)$$

Where $f(\bullet)$ is the two-parameter ramp threshold function?

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

The parameter $\gamma > 0$ is the sensitivity parameter that regulates how fast the membership values decreases when an input pattern A_h separated from hyperbox. The input test patterns which fall outside hyper box are insensitive to γ . The variations in the shape of fuzzy membership function for two dimensional input patterns can be best understood by varying γ , which is described in Figure 1-4. Min-max points are taken as $V_j = \{0.3, 0.3\}$ and $W_j = \{0.5, 0.5\}$ respectively. When γ is large the fuzzy set becomes more crisp and when γ is small fuzzy set becomes less crisp. By making γ large, it is possible to define a crisp cluster as a special case of the fuzzy cluster [1].



**Figure 1: Hyperbox Fuzzy Set For $\gamma=1$ Figure 2: Hyperbox Fuzzy Set For $\gamma=2$
Figure 3: Hyperbox Fuzzy Set For $\gamma=4$ Figure 4: Hyperbox Fuzzy Set For $\gamma=8$**

FMN Learning

The FMN Learning algorithm is a three stage expansion/contraction process [1]. The learning begins with selecting first input training pattern and finding expandable hyperbox from committed set of clusters. If it unable to find closest hyperbox which can expand to accommodate this pattern, it will create new hyperbox. If expandable hyperbox exists to accommodate input pattern, then hyperbox expansion process carried out according to (2), (3), (4), otherwise new hyperbox will be created. Every expanded hyperbox will undergo overlap test immediately after expansion process, which determines overlap with committed hyperboxes. Contraction process eliminates overlap, by redefining hyperbox with modified min-max points. The expansion-contraction process is described below.

Expansion: Given a pattern $A_h \in A$, find the hyperbox that provides the highest degree of membership and allows expansion (if needed). The degree of membership is calculated using (1). For hyperbox B_j to expand to include pattern A_h , the following constraint must be met.

$$\sum_{i=1}^n (\max(w_{ji}, a_{hi}) - \min(v_{ji} - a_{hi})) \leq \theta \quad (2)$$

Where $0 \leq \theta \leq 1$ is the size of hyperbox.

If (5) is satisfied, the min point is adjusted using the equation

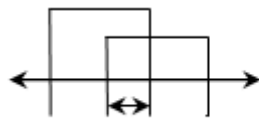
$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n \quad (3)$$

And the max point is adjusted using the equation

$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n \quad (4)$$

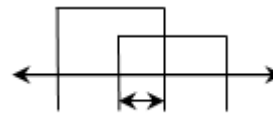
Overlap Test: In FMN, after every expansion process overlap (and containment) test is carried out with every committed hyperbox. If any overlap gets detected it will be removed by contraction process. FMN considers four overlapping cases and their respective contraction conditions. In these overlapping cases, B_j is considered as an expanded hyperbox in the previous stage and B_k is considered as a committed hyperbox in pattern space.

case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$ case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$



$v_{ji} \quad v_{ki} \quad w_{ji} \quad w_{ki}$

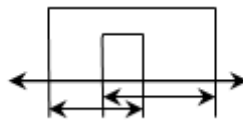
Overlap = $w_{ki} - v_{ji}$



$v_{ki} \quad v_{ji} \quad w_{ki} \quad w_{ji}$

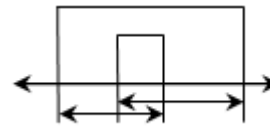
Overlap = $w_{ji} - v_{ki}$

case 3: $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$ case 4: $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$



$v_{ji} \quad v_{ki} \quad w_{ki} \quad w_{ji}$

Overlap = $\min(w_{ki} - v_{ji}, w_{ji} - v_{ki})$



$v_{ki} \quad v_{ji} \quad w_{ji} \quad w_{ki}$

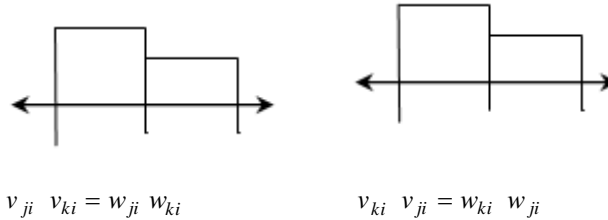
Overlap = $\min(w_{ki} - v_{ji}, w_{ji} - v_{ki})$

Contraction: If any overlap found after hyperbox expansion, it is eliminated on dimension by dimension basis as follows.

Case 1: If $v_{ji} < v_{ki} < w_{ji} < w_{ki}$, then contract using (5) Case 2: If $v_{ki} < v_{ji} < w_{ki} < w_{ji}$, then contract using (6)

$$v_{ki}^{new} = w_{ji}^{new} = \left(\frac{v_{ki}^{old} + w_{ji}^{old}}{2} \right) \quad (5)$$

$$v_{ji}^{new} = w_{ki}^{new} = \left(\frac{v_{ji}^{old} + w_{ki}^{old}}{2} \right) \quad (6)$$

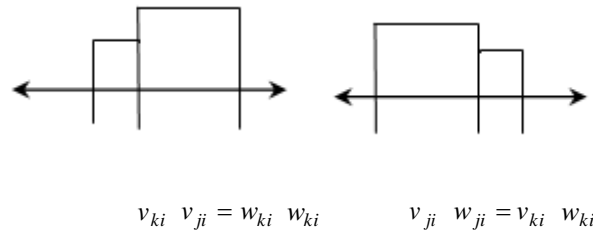


Case 3: If $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$, then the contraction is performed on the smaller of the two overlaps.

If $w_{ki} - v_{ji} < w_{ji} - v_{ki}$ then contract using (7) Else if $w_{ki} - v_{ji} > w_{ji} - v_{ki}$ then contract using (8)

$$v_{ji}^{new} = w_{ki}^{old} \quad (7)$$

$$w_{ji}^{new} = v_{ki}^{old} \quad (8)$$



Case 4: If $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$, then by symmetry, the same assignments under the same conditions in case 3 are applied here.

All three stages of FMN repeated to achieve cluster stability. Cluster stability is defined as all the hyper box min-max points not changing during successive representation of dataset [1].

FMN Testing

The trained neural network is then use for inference or reasoning for unknown pattern (test pattern). The Fuzzy Neural Network gives membership value for applied input pattern. We can use this membership information for decision making (e.g. Classification, Pattern Matching).

ANALYSIS OF FMN

In this section, the FMN analyzed in terms of methodology as well as time complexity of FMN.

Problem in FMN Overlap Test

As described in section II, FMN can identify overlap as well as containment between newly expanded hyper box and all other committed hyper boxes in pattern space. Among all four cases, FMN can only detect overlap or containment of true hyper boxes. Point, line or another true hyper box touching boundary of expanded hyper box (Figure 5), cannot be detected by FMN.

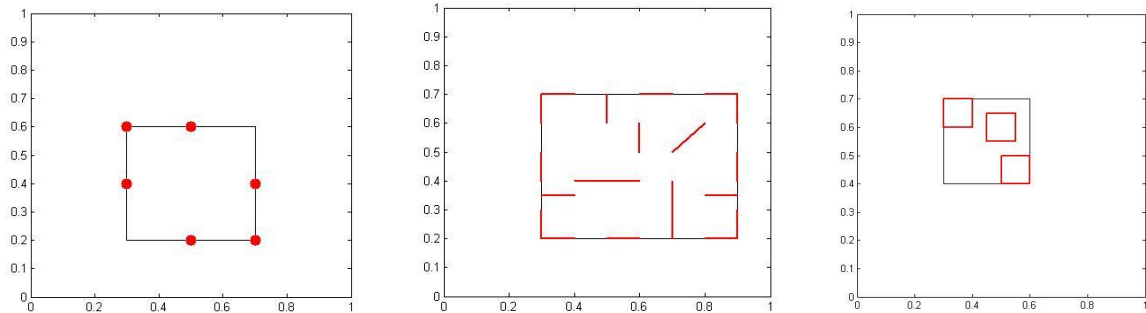


Figure 5: Overlapping Hyper boxes with another Hyper box

Problem in FMN Contraction Process

FMN Contraction process is not able to remove 100% overlap, due touching boundaries of hyper boxes. This will affect the performance of clustering algorithm. This may lead to having full membership value of a pattern that falls on the boundary of two contracted hyper boxes as shown in Figure 6.

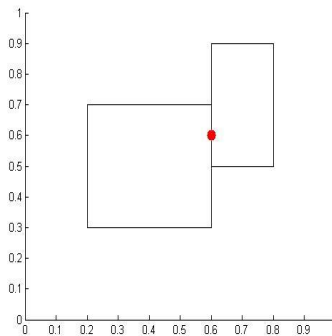


Figure 6: Overlapping Pattern on Boundary of Two Contracted Hyper boxes

Time Complexity of FMN

It is observed that, in FMN training stage, every input pattern will carry out expansion criteria with each committed hyper box in pattern space. It is observed in [2] that for a small value of θ , many numbers of hyper boxes are created and it depends on a number of instances and dimensions of the input dataset. The number of hyper boxes goes on increasing according to size of dataset for same hyper box size. This may lead to increase in training time if pattern space has a large number of hyper box fuzzy sets.

Based on this analysis, it is the necessity of FMN to modify overlap cases and contraction process for accurate clustering. Therefore, we introduced modifications in overlap test and contraction process of FMN, which are described in the following sections. θ

PROPOSED METHODOLOGY

This section describes MFMN algorithm. Section A and B gives detail description of modified overlap test and contraction process respectively. In FMN, for every new input pattern, expansion criteria checked for hyper box with highest membership value first. If this hyper box unable to accommodate new pattern then all other committed hyper boxes are considered for expansion process, according to their membership values. This will become time consuming task. Membership value of input pattern depends on sensitivity parameter γ and distance from hyper box. Hyper box located far away from input pattern unable to accommodate it due to membership value. Therefore, we are considering only hyper boxes with highest and second highest membership values along with some criterion. This leads to improve time complexity of an algorithm. We first consider highest membership value. If this hyper box exhausted without expansion then only consider hyper box with second highest membership value. Here we have introduce new parameter α which will decide whether to consider hyper box with second highest membership or not. The detail modified algorithm is described in next section.

Modified Overlap Test

In MFMN only overlapping case 3 and 4 are modified from FMN, keeping others untouched.

$$\text{case 3: } v_{ji} \leq v_{ki} \leq w_{ki} \leq w_{ji}$$

$$\text{case 4: } v_{ki} \leq v_{ji} \leq w_{ji} \leq w_{ki}$$

The modified overlap test can successfully able to detect overlapping conditions mentioned in Figure 5 as well as already existing overlapping conditions. To remove these overlaps we have modified contraction process also. This involves new parameter δ , is small gap between two hyper boxes after contraction process. This parameter δ is added to min point v_{ji} and subtracted from max point w_{ji} of previously expanded hyper box. Please note that for all the experimental studies δ set to $10 E^{-6}$ for the sake of simplicity.

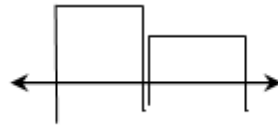
Modified Contraction Process

The modified contraction process is given below.

Case 1: If $v_{ji} < v_{ki} < w_{ji} < w_{ki}$, then contract using (9) and (10)

$$v_{ki}^{new} = \left(\frac{v_{ki}^{old} + w_{ji}^{old}}{2} \right) \quad (9)$$

$$w_{ji}^{new} = \left(\frac{v_{ki}^{old} + w_{ji}^{old}}{2} \right) - \delta \quad (10)$$

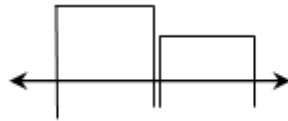


$$v_{ji} \quad v_{ki} \quad w_{ji} \quad w_{ki}$$

Case 2: If $v_{ki} < v_{ji} < w_{ki} < w_{ji}$, then contract using (11) and (12)

$$v_{ji}^{new} = \left(\frac{v_{ji}^{old} + w_{ki}^{old}}{2} \right) + \delta \quad (11)$$

$$w_{ki}^{new} = \left(\frac{v_{ji}^{old} + w_{ki}^{old}}{2} \right) \quad (12)$$

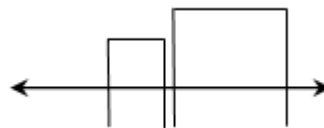


$$v_{ki} \quad v_{ji} \quad w_{ki} \quad w_{ji}$$

Case 3: If $v_{ji} \leq v_{ki} \leq w_{ki} \leq w_{ji}$, then the contraction performed on the smaller of the two overlaps.

If $w_{ki} - v_{ji} < w_{ji} - v_{ki}$ then contract using (13)

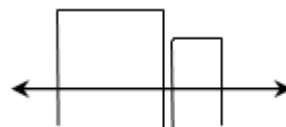
$$v_{ji}^{new} = w_{ki}^{old} + \delta \quad (13)$$



$$v_{ji} \quad v_{ki} \quad w_{ki} \quad w_{ji}$$

Else if $w_{ki} - v_{ji} > w_{ji} - v_{ki}$ then contract using (14)

$$w_{ji}^{new} = v_{ki}^{old} - \delta \quad (14)$$



$$v_{ji} \quad v_{ki} \quad w_{ki} \quad w_{ji}$$

Case 4: If $v_{ki} \leq v_{ji} \leq w_{ji} \leq w_{ki}$, then by symmetry, the same assignments under the same conditions in case 3 are applied here.

Experimental Studies

Various experiments carried out to evaluate the operations and performance of MFMN. In this experimental study we had used 4 datasets; details of the datasets summarized in Table 1. All these experiments were carried out on GPU server with 16 GB RAM, Intel's Xeon CPU E5-2620V3, 2.64 GHz, windows 7, 64 bit OS with single NVIDIA's Tesla K20 GPU with 2496 cores.

Table 1: Description of Datasets

| Dataset | Instances | Features | Source |
|---------------|-----------|----------|---------|
| Online Retail | 313999 | 6 | [3] |
| Iris | 150 | 4 | [5] |
| Butterfly | 58 | 2 | Created |
| Sample | 51 | 2 | Created |

Manually Created Datasets

For evaluation of MFMN, two data sets artificially created – Sample Data set (Figure 7) and Butterfly dataset (Figure 8). Both the datasets created within the range 0 to 1, so need of normalization.

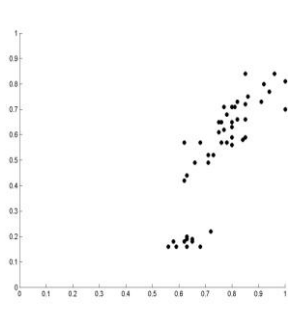


Figure 7: Sample Dataset

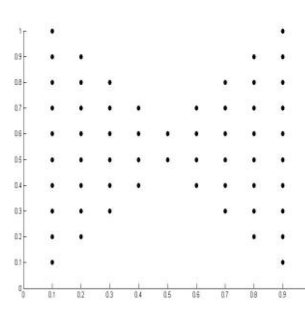


Figure 8: Butterfly Dataset

The MFMN applied on both these data sets to illustrate the performance of the same. After application of FMN on butterfly dataset, we found some overlapping conditions, which we were not able to identify by FMN. However, MFMN successfully detected those overlaps and able to remove overlaps with modified contraction process.

FMN and MFMN applied on sample dataset (Figure 8). Hyper boxes formed after application of FMN and MFMN, shown in Figure 9 and Figure 10 respectively. It was observed from Figure 10, that MFMN removes 100% overlap, i.e. after contraction process small gap δ separates two hyper box. This experiments were carried out for $\theta = 0.2$ and $\alpha = 0.05$.

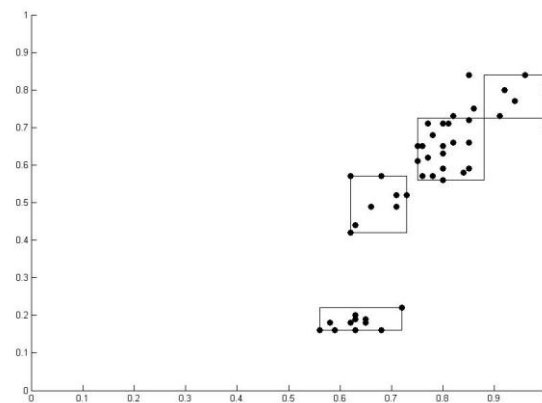


Figure 9: Results of FMN Algorithm after Contraction Process

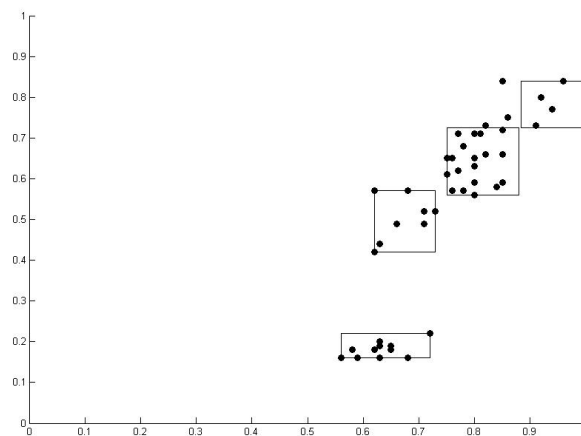


Figure 10: Results of MFMN Algorithm after Contraction Process

Standard Iris Dataset

We have used well-known dataset in the field of pattern recognition literature -Fisher Iris dataset [5]. Iris dataset have 150 instances along 4 features with 3 classes (50 for each class). In Iris dataset, 2 and 3 classes having similar features distinguishing from class 1 [6]. The iris dataset normalized to rescale between ranges 0 to 1, as a requirement of FMN algorithm. The dataset normalized by dividing each value of pattern with maximum value along that feature. The entire dataset used for clustering as well as for testing. In this study θ varied from 1.0 to 0.07 in step of 0.1 until 5 and after that in step of 0.01. At $\theta = 0.07$, we achieved 100% accuracy. In this experiment γ was set to 4 and α to 0.05. On an average 2 epochs were required to stabilize clusters. As referred in Figure 11, average confusion per class, decreases with increase in number of hyper boxes (i.e. decrease in θ).

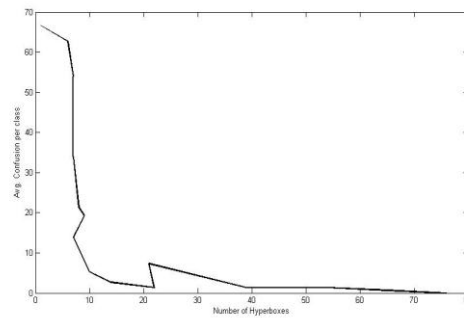


Figure 11: Number of Hyperboxes Vs Average Confusion per Class

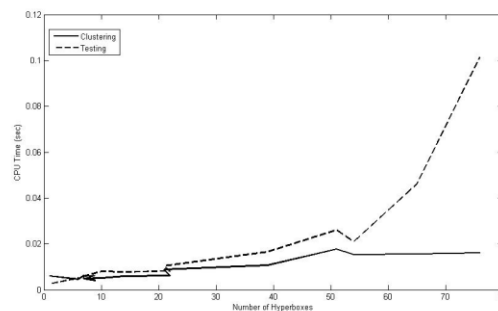


Figure 12: Number of Hyperboxes vs Elapsed CPU Time (sec)

It is observed from experiments that, as number of hyperboxes increases with respect to θ , time required for clustering as well as testing increases (Figure 12). For iris dataset at $\theta = 0.07$, the number of hyperboxes created were 76 and average CPU time was 16.17 msec for clustering. In this experimental study, we have referred General Fuzzy Min-Max Neural Network for Clustering and Classification (GFMM) [4]. GFMM is a hybrid-learning algorithm, applicable for pure classification, pure clustering or hybrid classification clustering. We achieved 100% clustering accuracy for less number of hyperboxes (76) as compared to FMN (117) and GFMM (86) under same circumstances.

Large Dataset

We used Online Retail Dataset [3] for evaluation of performance of MFMN. This is a transnational dataset of one year, of a UK-based online retail organization. This dataset applied on MFMN to identify highest profitable customers. This dataset originally having 541909 instances over 8 attributes. This dataset were filtered, refined and normalized before applying for MFMN. It is unlabeled dataset, we have pre-processed it to find out the classes, using Regency, Frequency and Monetary (RFM) model as mentioned in [5], [6]. RFM is a segmentation technique used in Business Intelligence, to identify most valuable customers. After all the pre-processing, we obtained training dataset of 313999 instances over 6 attributes, with 3 classes – Most Profitable Customers (22373), Regular Customers (266794) and Least Profitable Customers (24832).

This training dataset then applied for illustration of MFMN clustering. Similar to experimentation with iris dataset, CPU time increases with increase in number of hyper boxes. Average confusion per class decreases with increase in number of hyper boxes (Figure 13).

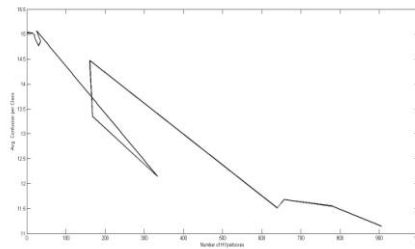


Figure 13: Number of Hyper boxes vs Average Confusion per class

NEED OF PARALLELISM

It is observed from previous section, CPU time increases with increase in instances and/or dimensions of dataset, also CPU time directly related to number of hyper boxes. Therefore, we had decided to parallelize the modified algorithm. We had implemented the MFMN parallel on GPU. As GPU is data parallel device, it is best suited for this task. As per experimental studies carried out in this paper, for large number of hyper boxes, time required to compute membership value of every hyper box for single input pattern is considerably large. Moreover, this time increases for multiple numbers of instances. Similar, situation occurs while finding out hyper box with maximum membership value. Therefore, we have chosen this part of algorithm for parallelization. In parallel algorithm, for single input pattern, calculation of membership value for all existing hyper boxes done by launching number of threads equal to number of hyper boxes. Identification of maximum membership value is implemented using CUDA reduction kernel [9], [10].

RESULTS

This section illustrates performance of MFMN implemented on CPU and GPU. For obvious comparison, for all dataset study γ set to 1, $\theta = 0.1$ and $\alpha = 0.05$. We have separately analyzed membership function and function for finding out maximum membership value for CPU and GPU. Computing membership value for an input pattern, for all committed hyper boxes in online retail dataset, we obtained $8.83\times$ speedup with 88.67% gain in time, similarly for iris we get $1.4\times$ speedup with 28.53% gain in time. It is observed that for small dataset CPU performs better as compared to GPU. On contrary, for large dataset GPU performs better as compare to CPU. It is observed that, for finding out maximum value among 639 membership values (In terms of Online Retail Dataset) $3.47\times$ speedup observed with 71.22% gain in time. For Iris dataset with 51 hyper boxes, CPU performed $1.25\times$ faster than GPU. By observing accelerated performance of both these functions, we have decided to use these parallel functions in clustering and testing process of MFMN and obtained results shown in Table II, III.

Table 2: Performance of FMN Training on CPU and GPU

| Dataset | Clusters | CPU Time (msec) | GPU Time (msec) | Speedup | Gain (%) |
|---------------|----------|-----------------|-----------------|---------|----------|
| Online Retail | 639 | 1774007.32 | 82254.6013 | 21.57 | 95.36 |
| Iris | 51 | 3.996 | 1.947136 | 2.05 | 51.27 |
| Butterfly | 19 | 0.34 | 0.993984 | 0.34 | -192.35 |
| Sample | 11 | 0.184 | 0.770048 | 0.09 | -318.50 |

Table 3: Performance of FMN Testing on CPU and GPU

| Dataset | Clusters | CPU Time (msec) | GPU Time (msec) | Speedup | Gain (%) |
|---------------|----------|-----------------|-----------------|----------|----------|
| Online Retail | 639 | 29780.76 | 4922.28320 | 6.050192 | 83.47 |
| Iris | 51 | 73.34 | 14.2078 | 5.161756 | 80.62 |

CONCLUSIONS

In this paper, we propose MFMN, which is modified version of FMN, in terms of overlap test and contraction of hyper boxes. MFMN is superior to FMN for overlap and contraction. The modified overlap test can able to detect hyper boxes with touching boundaries. In addition, contraction process in MFMN, successfully removes overlaps by creating small gap between two hyper boxes. It is observed from experiments, MFMN was able to achieve 100% accuracy with less number of hyper boxes for Iris dataset, as compared with FMN and GFMN. The GPU parallelization of MFMN is 21 times faster than serial implementation with 95.36% gain for online retail dataset. We can expect better performance of MFMN, for larger datasets. Thus, we recommend MFMN and its GPU parallel implementation for clustering of larger data sets.

FUTURE WORK

There are some situations in which, input pattern shows exact same membership function value for more than one hyper box. These situations are not considered in this work, which provides scope for future work. The size of hyper box θ , is still user defined, making it dynamic subject to future work.

REFERENCES

1. Patrick K. Simpson, "Fuzzy Min-Max Neural Networks – Part 2: Clustering," IEEE Trans. On Fuzzy Systems, Vol. 1, No. I, pp. 32-45, Feb. 1993
2. Dr. Daqing Chen. (2015, Nov 6). [Online]. Available: <https://view.officeapps.live.com/op/view.aspx?src=http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx>
3. R. A. Fisher. (1988, Jul 1). [Online]. Available: <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>
4. G. Babrys and A. Bargiela, "General Fuzzy Min-Max Neural Network for clustering and classification," IEEE Trans. Neural Netw., vol. 11, no. 3, pp. 769-783, May 2000.
5. D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: A case study of RFM model-based Customer segmentation using data mining," Journal of Database Marketing & Customer Strategy Management, vol. 19, no. 3, pp. 197–208, 2012.
7. Dr. Sankar Rajagopal, "Customer Data Clustering Using Data Mining Technique," *International Journal of Database Management Systems (IJDMS)*, Vol. 3, No. 4, Nov. 2011.
8. Wei JT, Lin SY, Wu HH. A review of the application of RFM model. African Journal of Business Management.

- 2010;4(19):4199–4206.
9. Mark Haris. (2007). [Online]. Available: <http://developer.download.nvidia.com/assets/cuda/files/reduction.pdf>
 10. John Nickolls, Ian Buck, Michael Garland, Kevin Skadron, “Scalable Parallel Programming with CUDA,” Queue, vol.6 no.2, March/April 2008.
 11. Andale. (2015, Dec 14). *RFM and Customer Value*. [Online]. Available: <http://www.statisticshowto.com/rfm-customer-value/>
 12. Jiawei Han, Micheline Kamber and Jian Pei, “Cluster Analysis: Basic Concepts and Methods,” in *Data Mining: Concepts and Techniques*, 3rd ed. Waltham: Morgan Kaufmann, 2012, ch. 10, sec. 10.2.1, pp. 451-454.
 13. J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
 14. Patrick K. Simpson, “Fuzzy Min-Max Neural Networks – Part 1: Classification,” *IEEE Trans. Neural Networks*, Vol. 3, pp. 776-786, Sept. 1992.
 15. Jaun Wachs, Helman Stern and Mark Last, “Color Face Segmentation using a Fuzzy Min-Max Neural Network,” *International Journal of Image and Graphics*, Vol. 2, No. 4, pp. 587-601, Sep. 2002.
 16. Barnali Goswami, Gupinath Bhandari, Sanjay Goswami, “Fuzzy Min-Max Neural Network for Satellite Image Clustering,” 3rd *Int. Conf. on Emerging Applications of Information Technology*, 2012.
 17. S. S. Chowhan, G. N. Shinde, “Iris Recognition Using Fuzzy Min-Max Neural Network”, *IJCE*, Vol. 3, No. 5, Oct. 2011.
 18. Y. j. Ma, J. h. Liu and W. Zeng-guo, "Modified fuzzy min-max neural network for clustering and its application on the pipeline internal inspection data," *2016 35th Chinese Control Conference (CCC)*, Chengdu, 2016, pp. 3509-3513.
 19. Seera Manjeevan, Chee Peng Lim, Chu Kiong Loo, Harapajan Singh, "A modified fuzzy min-max neural network for data clustering and its application to power quality monitoring." *Applied Soft Computing* 28 C, pp. 19-29, March 2015.
 20. L. A. Zadeh, “Fuzzy Sets”, *Information and Control*, No. 8, pp. 338-353, 1965.
 21. J.C. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,” Vol. 3, no. 3, Sep. 1973.
 22. Jacek M. Zurada, “Neuron Modelling for Artificial Neural Systems,” in *Fundamental Concepts and Models of Artificial Neural Systems*, Ed. St. Paul: West Publisher Company, 1992, ch. 2, sec. 2.1, pp. 30-36.
 23. *CUDA C Programming Guide*. [Online] Available: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/#axzz4idraNnUK>