**International Journal of Information Retrieval Research**
Volume 12 • Issue 1

# Reliable Distributed Fuzzy Discretizer for Associative Classification of Big Data

Hepzi Jeya Pushparani, Sarah Tucker College, Manonmaniam Sundaranar University, India

Nancy Jasmine Goldena, Sarah Tucker College, Manonmaniam Sundaranar University, India

## ABSTRACT

Data mining is an essential task because the digital world creates huge data daily. Associative classification is one of the data mining tasks which is used to carry out classification of data, based on the demand of knowledge users. Most of the associative classification algorithms are not able to analyze the big data which are mostly continuous in nature. This leads to the interest of analyzing the existing discretization algorithms which converts continuous data into discrete values and the development of novel discretizer reliable distributed fuzzy discretizer for big data set. Many discretizers suffer the problem of over splitting the partitions. The proposed method is implemented in distributed fuzzy environment and aims to avoid over splitting of partitions by introducing a novel stopping criteria. Proposed discretization method is compared with existing distributed fuzzy partitioning method and achieved good accuracy in the performance of associative classifiers.

## KEYWORDS

Associative Classification, Big Data, Discretization, Distributed Fuzzy Discretization, Fuzzy Discretization

## INTRODUCTION

Every second , the world creates a large volume of data in different domains, with reference to the International Data Corporation (IDC) study forecasting that the global data sphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025. Large volumes of data beyond conventional system's storage and processing capacities are known as Big data(Minelli et al., 2013). Real world data is categorical, numerical , continuous and various formats. The most efficient task is the extraction of information from that data. Classification algorithms are developed to meet the growing demand of data .The art of integrating frequent pattern mining and classification is known as Associative classification (Abdelhamid et al., 2012; Baralis & Garza, 2012) Many studies have shown that associative classifications have specific advantages over other traditional classification approaches such as Decision Tree and Rule Induction(Wedyan, 2014). First associations are extracted from the dataset using frequent pattern mining algorithms(Aggarwal et al., 2014) and then the classification rules are created. Most of the frequent pattern mining algorithm works only on categorical attributes. In order to improve the speed and accuracy of associative classifier , an efficient discretizer is required to discretize real data.

Discretization is a task of data preprocessing that transforms continuous features into discrete one, helping to enhance learning performance. Most of the algorithms for data mining work on discrete values. So discretization is carried out prior to the process of classification. Supervised discretization methods use the class information to set partition boundaries while unsupervised discretization

DOI: 10.4018/IJIRR.289572

methods do not use class labels to pick cut points. Entropy based discretization method uses class information to compute and evaluate the split point, which is certainly supervised and separated from top to bottom. Association rule learners prefer multivariate discretization that can capture the interdependencies between attributes, while univariate discretization discretes each attribute in isolation which tends to dissatisfactory association rules (Ishibuchi et al., 2001).

Discretization with fuzzy set is known as fuzzy discretization which resolves soft boundary problem. Fuzzy discretization first discretizes quantitative attribute values into intervals(Ishibuchi et al., 2001). Each cutpoint is associated with the membership function .The membership function is used to determine the degree of each attribute value corresponding to each interval. In fuzzy discretization, a value can be discretized into more than one interval at the same time with varying degrees. Fuzzy discretization process has the following steps (1) Identification of cutpoints (2) partitions are created based on the cutpoints (3) Using triangle membership function attribute values are converted into fuzzified values.

Classical data preprocessing techniques are not enough to scale well when managing large volume of data. To deal the problem with big data, scalable distributed techniques are developed. The first distributed programming techniques to tackle this problem are MapReduce(Dean & Ghemawat, 2004) and its open-source version Apache Hadoop. Apache Spark(Karau, Holden ; Konwinski, Andy ; Wendell Patrick ; Zaharia, 2015) is a fast, memory based data processing tool for large scale data processing . Through the ability of this Spark, processes present in many Machine Learning (ML) problems may be speeded up. So this tool has become especially popular among researchers and business experts in machine learning. Our main objective is to prove that in these frameworks, proposed discretization algorithm Reliable Distributed Fuzzy Discretizer can be parallelized, providing strong discretization solutions for Big data analytics. An efficient discretizer gives good classification accuracy in association rule mining. In order to prove the effectiveness of our proposed discretizer, RDFD is compared with distributed MDLP discretizer.

Most of the studies recently proposed in the literature for mining big data combine the MapReduce model with the Apache Hadoop and Apache Spark cluster computing frameworks. With regard to classification problems, some recent works have proposed several distributed MapReduce versions of classical algorithms, such as SVM (Alham et al., 2011; Caruana et al., 2011), prototype reduction (Triguero et al., 2016), KNN(Goyal et al., 2020), associative (Bechini et al., 2016; Ducange et al., 2015), boosting (Palit & Reddy, 2012), decision trees (Dai & Ji, 2014; Wang et al., 2014; Zhang et al., 2012), naive bayes classifiers and neural networks(Schölkopf et al., 2007), investigating performance in terms of speedup (Schölkopf et al., 2007). Researchers are continuously investigating new algorithms, taking into account not only the accuracy of the classifiers, but also the scalability of the proposed approaches, only few works have integrated fuzzy set (Ducange et al., 2015; Lopez et al., 2014; Triguero et al., 2015).

Han and Kamber (2011) stated that the major concerns of data classification were predictive accuracy, speed, robustness, scalability, and interpretability. To achieve this, the existing associative classifiers are analyzed to find out efficient discretizer. Association rule mining is performed in three steps. First frequent item sets are extracted from the training set. Then the rules are mined from the frequent item set. Finally association rules are pruned. Associative classification may use Apriori algorithm or FP-growth algorithm for generating association rules. Apriori algorithm(Agrawal, Rakesh; Srikant, 2013) is continuously repeated to scan database, find out all frequent item sets, until it does not produce new candidate item sets. Apriori algorithm does not filter prior candidate item sets, that reduces the amount of candidate item sets to scan. Therefore, it needs many times to complete scanning a database. In implementing efficiency, Apriori algorithm is not completely efficient.

FP-growth algorithm was proposed by Han et al.(2000), it can be one of the representation of the item sets which do not require candidate generations. It does not need association length to proceed phases which generate candidate item sets in Apriori algorithm. However, mining with Apriori algorithm does not achieve the goal efficiently because it may need many times to scan database and generates lots of candidate itemsets. Therefore, FP-growth preceedes the first scan in transaction database, then later filters the frequent itemsets and gradually increases support. Next, in the second scan, it establishes FP-tree structure by the transaction database. Then, a header table is used to allocate each item node in FP-tree. Last, a Header table mines a conditional pattern tree which finds out all frequent item sets in recursive method. It is a very efficient and memory saving algorithm. Fuzzy associative classification approaches based on Apriori algorithm which is slow in terms of computation while comparing to FP growth because of huge number of candidate generation (Fazzolari et al., 2014; Lucas et al., 2012; Pach et al., 2008) .

Before the extraction of frequent itemset, continuous attributes are discretized into discrete values using any one of discretizer. Discretizer may be supervised or unsupervised. Equal interval width is the simplest discretization method which divides the range of observed values for a variable into k equal sized bins where k is user supplied parameter. As Catlett(1991) points out, this type of discretization is vulnerable to outliers that may drastically skew the range. In Another method, equal frequency discretization divides a continuous variable into k bins where (given m instances) each bin contains m/k adjacent values. Since these unsupervised methods do not utilize instance labels in setting partition boundaries, it is likely that classification information will be lost by binning as a result of combining values that are strongly associated with different classes into the same bin (Kerber, 1992).

M.Zeinalkhani and M.Eftekhari (2014) proposed two step method for fuzzy discretization. In First step each domain of each attribute is divided into non fuzzy partition. In second step each non fuzzy partition is transformed into fuzzy partition and membership function is generated by the following methods partition width, standard deviation of examples, the newly introduced parameters coverage rate of partition (PCR) and coverage rate of neighbour partition (NPCR). In addition, Fuzzy Entropy Based Fuzzy Partitioning method is introduced. Zeta discretization algorithm with PCR membership function outperforms the others in terms of the accuracy and complexity for construction of fuzzy decision tree.

Segatori et al. (2018) introduced a distributed fuzzy discretizer based on fuzzy information entropy for managing big data in Map Reduce environment. Strong fuzzy partition with triangular membership function is generated for each continuous attribute by using the distributed version of the well-known method proposed (Fayyad & Irani, 1993). Madhu G et al.,(2014) proposed new non parametric discretization method ZDISC based on Z-score discretization in five biomedical datasets and compared with other discretization methods such as Ameva, Baysian, CACC, CADD, CAIM, Chi2,Chimerge, ExtChi2, Fayyad and Irani discretization (MDL) and PKID. C4.5 and SVM classifiers are used for classification task and proved that ZDISC is superior than other discretization methods mentioned above in terms of classifier accuracy.

Fuzzy Clustering is different from classical clustering in which each instance belongs to many number of clusters with associated membership function. A big data clustering method based on the map reduce framework using an ant colony approach to decompose the big data into several data partitions to be used in parallel clustering(Yang & Li, 2013). Ant colony clustering algorithm using MapReduce leads to the automation of the semantic clustering to improve the data analysis task. This algorithm is developed and tested on data sets with large number of records (up to 800 K) and showed acceptable accuracy with good speedup. MR-FCM algorithm scales well with increasing data set sizes as shown by the scalability analysis conducted.

This paper is organized as follows. Background Section provides a basic description of the distributed fuzzy discretization. Material and Method Section presents each phase of the proposed approach the experimental setup and discusses the results that are obtained on real-world dataset. Last Section is presented with conclusion.

## BACKGROUND

### Fuzzy Discretization

Discretization is a preprocessing technique in which the continuous attributes are converted into discrete intervals. For example, given a dataset with x attributes, n instances, m continuous attributes ($m \subset x$ ) and C class labels, the disjoint intervals created by the discrtetization of a continuous attribute ($\{[i_0,i_1],(i_1,i_2],(i_2,i_3],...,(i_{k-1,}i_k]\}$).

Where $i_{1,}i_{2,}...i_{k-1}$ are the cutpoints and k is the number of intervals created for the continuous attribute m. The total number of discrete values are k. The values of m are placed into the any one of the interval.

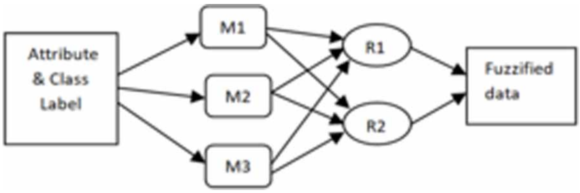
Fuzzy discretization is the best way to represent the continuous attributes in terms of linguistics. First each attribute is sorted before applying discretization. Random points within the attribute values are chosen and using fuzzy class information entropy, cut points are selected. Using the cutpoints multiple partitions are created. Each partition has two cutpoints lower cut point and upper cutpoint. Each partition is assigned linguistic terms. Triangular membership function is used to describe the degree of membership for each value.

### Distributed Fuzzy Discretization

Map reduce is a distributed computing environment for associative classification in big data. Map and Reduce are the two phases of this distributed environment. The input program is distributed into many nodes and one of the nodes is served as a master and others are slave. Input data are splitted into many chunks and distributed to all the slave nodes. Each node has many mappers and reducers. In map phase mapper works on a subset of data and produces <k,v> pairs. In reduce phase, reducer collects the matching pairs from the mappers and combines them to produce final output.

Fuzzy Discretization is done in distributed framework using MapReduce as shown in Figure 1. M number of mappers and N number of reducers are used to perform the fuzzy partitioning. Each attribute is partitioned using the map reduce tools. Much number of mappers and reducers are involved to create partitions in the attributes of big data. Each mapper has <key,value> a pair as attribute value and class label. Mappers return the fuzzy centres. By using the fuzzy centres the reducer converts the attributes values into fuzzified data.

**Figure 1. Map reduce framework for fuzzification process**



## MATERIAL AND METHOD

### Proposed Reliable Distributed Fuzzy Discretizer for Big data

Procedure1 is the main procedure of Reliable distributed fuzzy discretizer. Four random points within the attribute values are selected and then the sub procedure Find_partition is called for the cutpoint selection, by passing the attribute values and random points.

Cutpoint selection is an important step in discretization process. Valid cutpoint is identified by using the class information fuzzy entropy and the minimum weighted fuzzy entropy point is selected as a cutpoint among the chosen random points. Partitioning among the cutpoint is performed. The values above the cutpoint is denoted as left_p and the values below the cutpoint is denoted as right_p. Procedure Generate_Partition is called by passing left_p and right_p respectively. This is done by mappers in distributed environment.

1) *Map phase:* In Map phase attributes value and class label is passed to the multiple number of mappers. Within this phase two procedures are used for creating fuzzy centres. They are Find_partition and Generate_Partition. These two procedures are used by the m number of mappers and the partitions are created. Map phase is shown in Figure 2. M1, M2, M3 are mappers.
2) *Reduce Phase:* In reduce phase, fuzzy centres are used to fuzzify the attribute values and the linguistic variables are assigned using the strong triangular membership function. Fuzzy centres are the output of mappers. Fuzzification process is done by the reducers using input provided by mappers and fuzzified data is returned as output of reducers. Reduce phase is shown in Figure 3. R1, R2 are reducers.

## Sparkle Information Gain

Recursively cutpoints are identified for splitting the continuous features and partitions are done till the stopping criteria is reached. Stopping criteria is important to avoid over splitting. Information Gain is used as a stopping criteria for getting fuzzy centres. Sometimes this information gain leads to the problem of over splitting. To avoid this, a novel criteria sparkle gain is introduced in additionally with information gain. This also plays a vital role in deciding fuzzy centre. Current partition undergoes partitioning process if there is particular percentage of class labels. For example, Given a previous partition $P_i$ is replaced by current partition $P_c$ on the interval v if and only if

$$FuzzInfoGain\left(P_c\right) > \frac{\log_2\left(\left|S_v\right|-1\right)}{\left|S_v\right|} + \frac{\Delta\left(P_c;v\right)}{\left|S_v\right|} \tag{1}$$

$$\Delta\left(P_c;v\right) = \log_2(3^{L_C}-2) - \left[\sum_{j=1}^{|P_i|} L_{i,j} * FE\left(A_{i,j}\right) - \sum_{j=1}^{|P_c|} L_{c,j} * FE\left(A_{c,j}\right)\right] \tag{2}$$

$$FuzzyInfoGain = WFEnt\_Prev\_partition - \ WFent\_currpartition \tag{3}$$

Where $L_c$ and $L_i$ are the number of class labels of current and previous partition respectively. $FE(A_{i,j})$ and $FE(A_{c,j})$ are the fuzzy entropy of previous partition and current partition respectively. In Reliable distributed fuzzy discretizer along with the above criteria (2) , sparkle gain is also calculated.

$$Sparkle\ \ Gain = L_c \ > 0.8 * L_s \tag{4}$$

Figure 2. Map phase of fuzzy partitioning

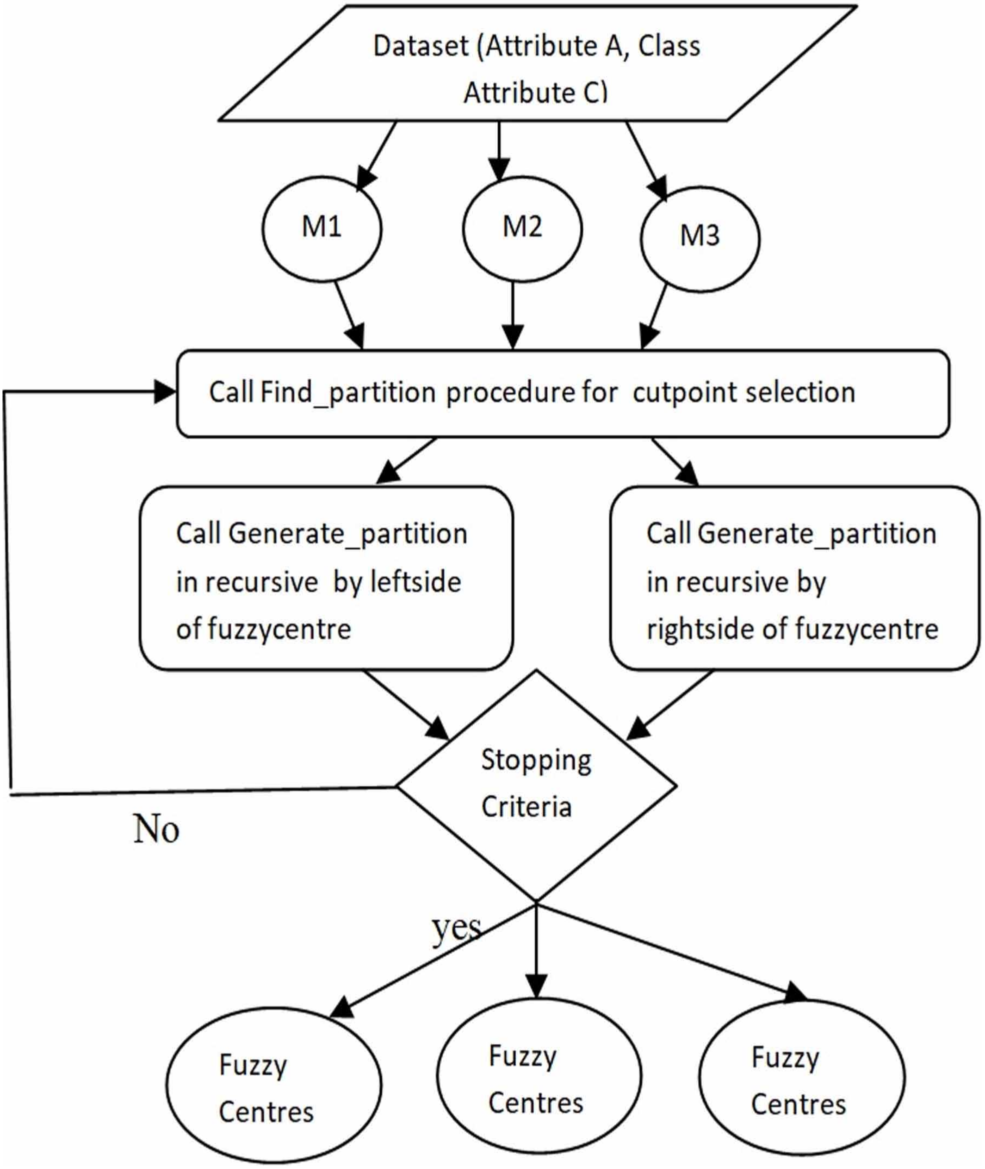
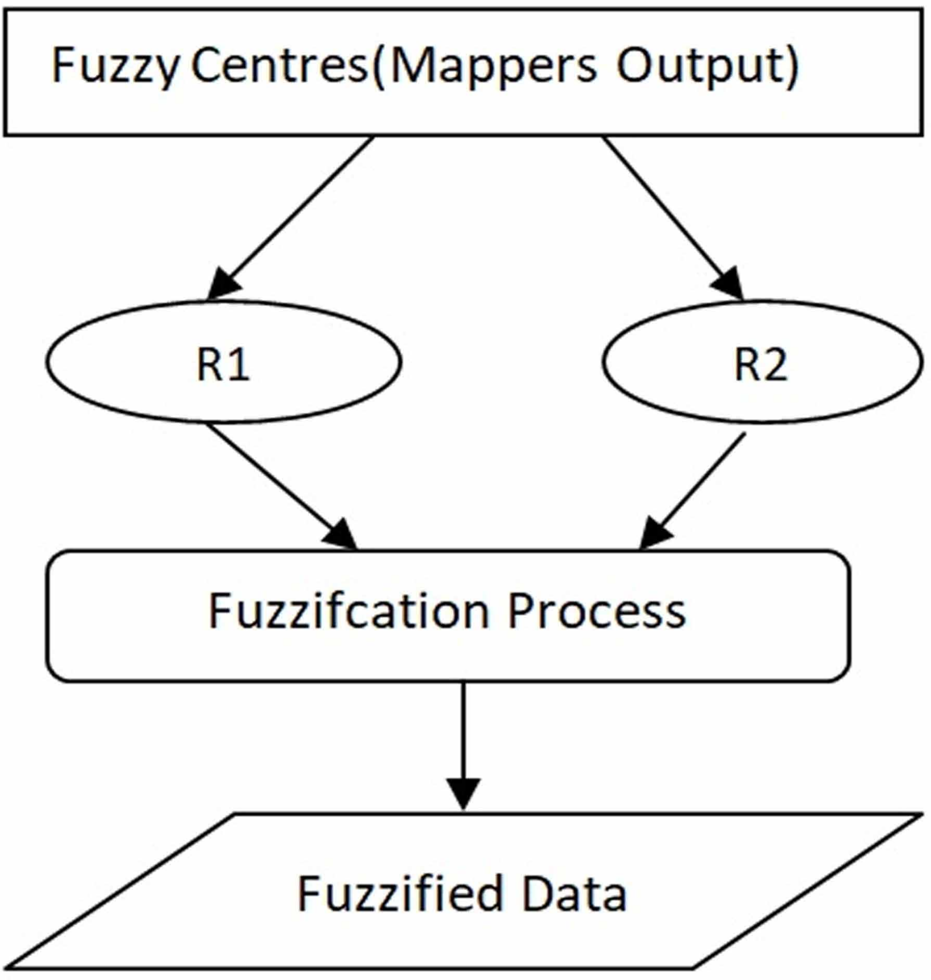

Dataset (Attribute A, Class Attribute C)
M1
M2
M3
Call Find_partition procedure for cutpoint selection
Call Generate_partition in recursive by leftside of fuzzycentre
Call Generate_partition in recursive by rightside of fuzzycentre
Stopping Criteria
No
yes
Fuzzy Centres
Fuzzy Centres
Fuzzy Centres

Figure 3. Reduce phase of fuzzy partitioning



Where $L_C$ and $L_S$ are the number of unique class labels of current partition and entire dataset respectively. If 0.8 percentage of unique classes of the dataset is present in the current partition, previous partition is replaced by current partition.

**Procedure 1:** Reliable Distributed Fuzzy Discretization

```
------------------------------------------------------------------
Input : attribute A , the class attribute   C, random point cp
Output :Discretized data for attribute A
begin
1. low← min(A); high← max(A)
2. Choose  four random points within the limit of low and high.
3. For each point(cp) do the following-
(i) Call the Find_partition procedure by passing the arguments A,cp,C.
(ii) Save the entropy , partitioned data based on the given point cp.
```

```
4. Select minimum entropy among the given four points and consider
that point as the first cutpoint. The selected cutpoint is known
as attr_range.
5. Fix it as the first fuzzy centre(low, attr_range, high)
6. Call Generate_Partition for the left side data of the fuzzy
centre.
7. Call Generate_Partition for the right side data of the fuzzy
centre.
End.
Procedure :Find_partition
Input   : Attribute A, cp , class attribute C
Output : Weighted Fuzzy entropy of right_p, left_p, min_WFent
Begin
1. Split the attribute value into two partitions. left_p and
right_p. left_p has the attribute value less than cp and right_p
has the attribute values greater than cp.
2. Similarly the class labels corresponding to the attribute
values are also partitioned.
3. For each partition P, calculate the weighted fuzzy entropy.
```

$$WFent = len(P)/len(A)*(-FE)$$

$$FE = \sum_{i=1}^{k} \frac{i^{th} classfrequency}{len(P)} * log_2^{\left(\frac{i^{th} classfrequency}{len(P)}\right)}$$

```
where FE is the fuzzy entropy of the partition and k is the number
of class labels within the partition
4. Return minimum weighted entropy min_WFent among two partitions
and the weighted fuzzy entropy of the splitted partitions left_p,
right_p.
End.
Procedure: Generate_Partition
Input: prev_partition, curr_partition, prev_fuzzycentre
Output: fuzzy_centre
Begin
1. low ← min(Currpartition)
2. high ← max(Currpartition)
3. Select any four random points within the range low and hgh.
4. Call the Find_partition procedure by passing the above points
one by one.
5. Select the best cutpoint using the minimum weighted fuzzy
entropy.
6. Calculate  FuzzyInfoGain using eq.(4)
7. Check the stopping criteria is reached .If so returns the
fuzzy_centre of current partition.
8. Otherwise call the Generate_Partition by passing the leftside
values of the fuzzy centre and also call the Generate_Partition
```

```
procedure by passing the right side values of the fuzzy centre.
End.
```

### Association Rule Mining and Classification

Association rules are created using the distributed approach specified in PFP-Growth (Li et al., 2008). For each class label, n number of rules are created. PFP-Growth algorithm generates frequent FCARs with high fuzzy confidence. Fuzzy Support counting, fuzzy FP-Growth and Rules selection are the three map reduce phases of fuzzy association rule mining process in distributed environment. In distributed fuzzy support counting phase, dataset is scanned and frequent fuzzy sets are identified using the support threshold minSupp. The number of occurrences of each class label is also counted in this phase. Distributed Fuzzy FPGrowth phase, mines fuzzy class association rules whose support, confidence values are higher than *minSupp*, *minConf* thresholds respectively. In distributed rule selection phase, the top *K* non redundant rules are selected from the final rule base for each class label to reduce the number of rules. Fuzzified rules for all continuous values are created. Classification process is done using rule based classifiers by using the Fp-rules. Classification accuracy is also calculated.

## RESULT AND DISCUSSION

To characterize the behaviour of the proposed approach for fuzzy discretization of continuous attributes in distributed environment for big data, the experimental study focuses on two aspects. First, the classification accuracy of our approach and existing approach are analyzed. Then, We investigate a model complexity of discretizer in terms of rules generated. We test our algorithm on a well-known big dataset Cover Type extracted from the UCI repositories. This dataset has 581210 instances and 54 attributes which are continuous and categorical. Only 12 continuous attributes are taken for the experiment.All the experiments have been carried out on a system with 4-Core CPU (Intel i5-8250U CPU @ 1.60GHz, 1801 Mhz, 8 Logical Processor(s)) 16 GB RAM and a 1TB Hard Drive. Our experiment is carried out on windows10 and python 3.6.5 spyder.

Distributed fuzzy partitioning of continuous attributes is an essential step in associative rule generation to improve the performance of associative classification. Reduced training set is obtained by random extraction of percentage P of objects from the overall training set is an input to the RDFD. This random selection leads to achieve good results in big data. Twenty mappers and twenty reducers are used for this experiment. Data and program are passed to the mappers and the fuzzifed data are received from Reducers. Metrics is used to assess the performance of classifiers are shown in Table 1.

Discretization performance is reflected in terms of classification accuracy. Proposed reliable distributed fuzzy discretizer is compared with the existing distributed fuzzy partitioning method. This is shown in Figure 4(a).Proposed RDFD achieved good accuracy in terms of classifier performance which results shown in Table.2. Experiments conducted on 12000 random selection of CoverType dataset with the support ranges from 40 to 90 and the confidence 0.6, 0.7, 0.8. Accuracy 79.62 is the highest by using RDFD whereas 79.42 achieved in DFP. Average accuracy based on support is calculated and the results shown in Figure 4(a). RDFD achieved good accuracy than DFB. Average accuracy based on confidence is also calculated and results are shown in Table 2 and Table 3. RDFD has high accuracy than DFP.

Figure 4(b,c&d). shows the association rules in RDFD and existing DFP. Average number of rules generated in both the approaches are compared. Average number of rules generated in RDFD and DFP is given in Table 2 and Table 3 respectively. RDFD has more number of rules while the confidence is 0.6. DFP generates huge number of rules than the proposed discretizer for the confidence 0.7 and 0.8. In Figure 4(c&d), large number of rules are generated in DFP than RDFD with the confidence 0.7 and 0.8 respectively. Association rules generated in RDFD is lesser than in DFP because of the sparkle gain calculation in addition to fuzzy info gain. This reduces the complexity of this fuzzy model.

Table 1. Parameters used in experiments

| Method | Support | Confidence |
|---|---|---|
| RDFD | 40 to 90 | 0.6 to 0.8 |
| DFP | 40 to 90 | 0.6 to 0.8 |

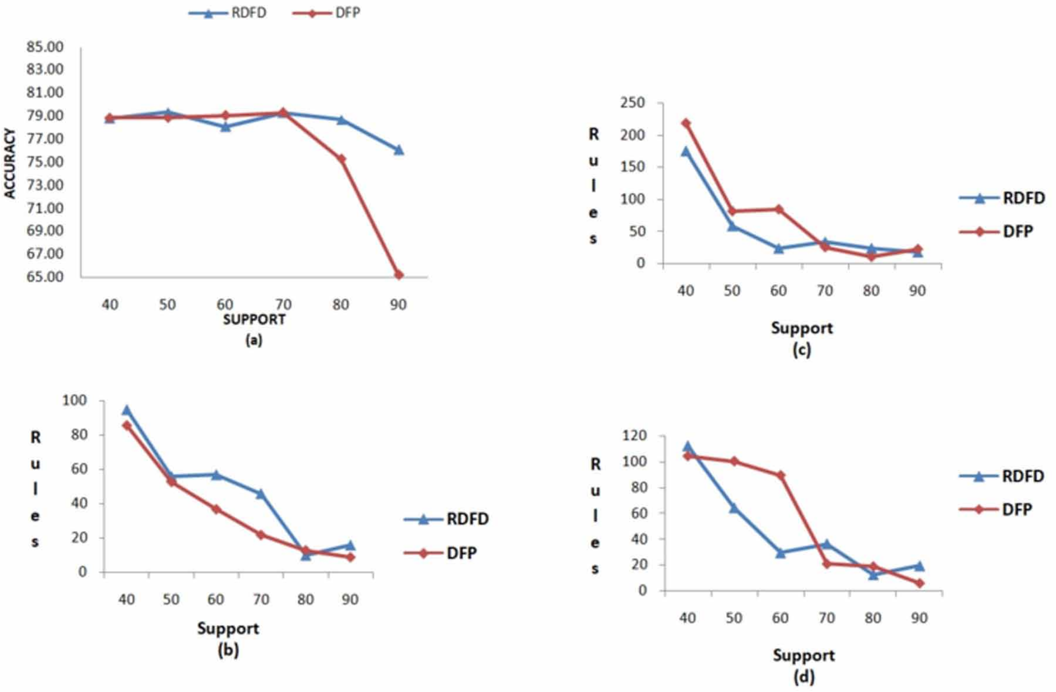
Table 2. Accuracy and rules obtained in reliable distributed fuzzy discretization

| **Support/Confidence** | **Accuracy** | | | **Rules** | | |
|---|---|---|---|---|---|---|
| | **0.8** | **0.7** | **0.6** | **0.8** | **0.7** | **0.6** |
| **40** | 77.68 | 79.26 | 79.62 | 95 | 176 | 113 |
| **50** | 79.42 | 79.39 | 79.31 | 56 | 59 | 65 |
| **60** | 75.6 | 79.28 | 79.42 | 57 | 24 | 30 |
| **70** | 79.33 | 79.17 | 79.42 | 46 | 34 | 37 |
| **80** | 79.42 | 79.42 | 77.34 | 10 | 24 | 13 |
| **90** | 69.54 | 79.42 | 79.2 | 16 | 18 | 20 |
| **Average** | **76.83** | **79.32** | **79.07** | **47** | 56 | 46 |

Table 3. Accuracy and rules obtained distributed fuzzy partitioning

| **Support/ Confidence** | **Accuracy** | | | **Rules** | | |
|---|---|---|---|---|---|---|
| | **0.8** | **0.7** | **0.6** | **0.8** | **0.7** | **0.6** |
| **40** | 79.07 | 79.11 | 78.42 | 86 | 219 | 105 |
| **50** | 77.85 | 79.42 | 79.39 | 53 | 82 | 101 |
| **60** | 79.28 | 78.93 | 78.98 | 37 | 85 | 90 |
| **70** | 79.42 | 79.42 | 79.1 | 22 | 26 | 21 |
| **80** | 79.42 | 77.58 | 68.93 | 13 | 11 | 19 |
| **90** | 57.79 | 79.42 | 58.48 | 9 | 23 | 6 |
| **Average** | 75.47 | 78.98 | 73.88 | 37 | **74** | **57** |

Figure 4. (a) Classifier accuracy based on Discretizer RDFD,DFP (b) Fp rules with confidence=0.8 , (c) Fp rules with confidence=0.7 , (d) Fp rules with confidence=0



## CONCLUSION

In this paper, Reliable Distributed Fuzzy discretizer is introduced for performing the fuzzy discretization task in big data in distributed environment. To enhance the performance of associative classification in big data, this work has been done and compared with the existing approach distributed fuzzy partitioning. In terms of accuracy, this work achieved good performance than DFP. For reducing model complexity a novel criteria has been introduced which is also proved that in terms of fuzzy centres generation. Less number of fuzzy centre's lead to minimize the generated association rules. Proposed approach RDFD has less number of rules compared to the existing approach. Even though less number of rules generated , proposed distributed approach achieves good accuracy in big dataset . Random selection of cutpoint leads the problem of unwanted splitting of attribute values. This problem has been avoided with the help of sparkle gain . In future this work can be carried out in terms of improving the scalability and minimize runtime for associative classification task.

## REFERENCES

Abdelhamid, N., Ayesh, A., Thabtah, F., Ahmadi, S., & Hadi, W. (2012). MAC: A multiclass associative classification algorithm. *Journal of Information and Knowledge Management*, *11*(2). https://doi.org/10.1142/S0219649212500116

Aggarwal, C. C., Bhuiyan, M. A., & Al Hasan, M. (2014). Frequent pattern mining algorithms: A survey. In *Frequent Pattern Mining* (pp. 19–64). Springer International Publishing. 10.1007/978-3-319-07821-2_2

Agrawal, Rakesh; Srikant, R. (2013). Fast Algorithms For Mining Association Rules In Datamining. *International Journal of Scientific & Technology Research*, *2*(12), 13–24.

Alham, N. K., Li, M., Liu, Y., & Hammoud, S. (2011). A MapReduce-based distributed SVM algorithm for automatic image annotation. *Computers & Mathematics with Applications (Oxford, England)*, *62*(7), 2801–2811. https://doi.org/10.1016/j.camwa.2011.07.046

Baralis, E., & Garza, P. (2012). I-prune: Item selection for associative classification. *International Journal of Intelligent Systems*, *27*(3), 279–299. https://doi.org/10.1002/int.21524

Bechini, A., Marcelloni, F., & Segatori, A. (2016). A MapReduce solution for associative classification of big data. *Information Sciences*, *332*, 33–55. https://doi.org/10.1016/j.ins.2015.10.041

Caruana, G., Li, M., & Qi, M. (2011). A MapReduce based parallel SVM for large scale spam filtering. *Proceedings - 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, 4*, 2659–2662. 10.1109/FSKD.2011.6020074

Catlett, J. (1991). *Megainduction : Machine learning on very large databases* (PhD Thesis). Basser Department of Computer Science, University of Sydney.

Dai, W., & Ji, W. (2014). A MapReduce Implementation of C4.5 Decision Tree Algorithm. *International Journal of Database Theory and Application*, *7*(1), 49–60. https://doi.org/10.14257/ijdta.2014.7.1.05

Dean, J., & Ghemawat, S. (2004). *MapReduce: Simplified Data Processing on Large Clusters*. OSDI.

Ducange, P., Marcelloni, F., & Segatori, A. (2015). A MapReduce-based fuzzy associative classifier for big data. *IEEE International Conference on Fuzzy Systems,* 1–8. 10.1109/FUZZ-IEEE.2015.7337868

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1022–1027.

Fazzolari, M., Alcalá, R., & Herrera, F. (2014). A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm. *Applied Soft Computing*, *24*, 470–481. https://doi.org/10.1016/j.asoc.2014.07.019

Goyal, J., Khandnor, P., & Aseri, T. C. (2020). A Comparative Analysis of Machine Learning classifiers for Dysphonia-based classification of Parkinson's Disease. *International Journal of Data Science and Analytics*, 1–15. 10.1007/s41060-020-00234-0

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining. Concepts and Techniques* (3rd ed.). The Morgan Kaufmann Series in Data Management Systems.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *SIGMOD Record*, *29*(2), 1–12. https://doi.org/10.1145/335191.335372

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2001). Fuzzy data mining: Effect of fuzzy discretization. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 241–248. 10.1109/icdm.2001.989525

Karau, H., Konwinski, A., & Zaharia, M. (2015). Learning Spark: Lightning-Fast Big Data Analytics. O'Reilly Media, Inc.

Kerber, R. (1992). Chimerge Discretization of Numeric Attributes., 123-128. - References - Scientific Research Publishing. *Proceedings of the 10th National Conference on Artificial Intelligence*.

Li, H., Wang, Y., Zhang, D., Zhang, M., & Chang, E. Y. (2008). PFP: Parallel FP-growth for query recommendation. *RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems*, 107–114. 10.1145/1454008.1454027

Lopez, V., del Rio, S., Benitez, J. M., & Herrera, F. (2014). On the use of MapReduce to build linguistic fuzzy rule based classification systems for big data. *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1905–1912. 10.1109/FUZZ-IEEE.2014.6891753

Lucas, J. P., Laurent, A., Moreno, M. N., & Teisseire, M. (2012). A fuzzy associative classification approach for recommender systems. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, *20*(4), 579–617. https://doi.org/10.1142/S0218488512500274

Madhu, G., Rajinikanth, T. V., & Govardhan, A. (2014). Improve the classifier accuracy for continuous attributes in biomedical datasets using a new discretization method. *Procedia Computer Science*, *31*, 671–679. https://doi.org/10.1016/j.procs.2014.05.315

Minelli, M., Chambers, M., & Dhiraj, A. (2013). Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses. In Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses. John Wiley and Sons. https://doi.org/10.1002/9781118562260.

Pach, F., Gyenesei, A., & Abonyi, J. (2008). Compact fuzzy association rule-based classifier. *Expert Systems with Applications*, *34*(4), 2406–2416.

Palit, I., & Reddy, C. K. (2012). Scalable and Parallel Boosting with MapReduce. *IEEE Transactions on Knowledge and Data Engineering*, *24*(10), 1904–1916. https://doi.org/10.1109/TKDE.2011.208

Schölkopf, B., Platt, J., & Hofmann, T. (2007). Map-Reduce for Machine Learning on Multicore. In Advances in Neural Information Processing Systems 19*: Proceedings of the 2006 Conference* (pp. 281–288). MIT Press.

Segatori, A., Marcelloni, F., & Pedrycz, W. (2018). On Distributed Fuzzy Decision Trees for Big Data. *IEEE Transactions on Fuzzy Systems*, *26*(1), 174–192. https://doi.org/10.1109/TFUZZ.2016.2646746

Triguero, I., Peralta, D., Bacardit, J., García, S., & Herrera, F. (2015). MRPR: A MapReduce solution for prototype reduction in big data classification. *Neurocomputing, 150*(Part A), 331–345. 10.1016/j.neucom.2014.04.078

Triguero, I., Peralta, D., Bacardit, J., García, S., & Herrera, F. (2016). MRPR: A MapReduce Solution for Prototype Reduction in Big Data Classification. *Neurocomputing*.

Wang, S., Zhai, J., Zhu, H., & Wang, X. (2014). Parallel ordinal decision tree algorithm and its implementation in framework of mapreduce. *Communications in Computer and Information Science*, *481*, 241–251. https://doi.org/10.1007/978-3-662-45652-1_25

Wedyan, S. (2014). Review and Comparison of Associative Classification Data Mining Approaches. *International Journal of Computer, Information. Systems and Control Engineering*, *8*(1), 34–45.

Yang, J., & Li, X. (2013). MapReduce based method for big data semantic clustering. *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, 2814–2819. 10.1109/SMC.2013.480

Zhang, C., Li, F., & Jestes, J. (2012). Efficient parallel kNN joins for large data in MapReduce. *ACM International Conference Proceeding Series*, 38–49. 10.1145/2247596.2247602