MR_IMQRA: An Efficient MapReduce Based Approach for Fuzzy Decision Reduct Computation

Kiran Bandagar, Pandu Sowkuntla(✉), Salman Abdul Moiz, and P. S. V. S. Sai Prasad

School of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, Telangana, India
bandagar.kiran30@gmail.com, pandu.sowkuntla@uohyd.ac.in, salman.abdul.moiz@gmail.com, saics@uohyd.ernet.in

Abstract. Fuzzy-rough set theory, an extension to classical rough set theory, is effectively used for attribute reduction in hybrid decision systems. However, it's applicability is restricted to smaller size datasets because of higher space and time complexities. In this work, an algorithm MR_IMQRA is developed as a MapReduce based distributed/parallel approach for standalone fuzzy-rough attribute reduction algorithm IMQRA. This algorithm uses a vertical partitioning technique to distribute the input data in the cluster environment of the MapReduce framework. Owing to the vertical partitioning, the proposed algorithm is scalable in attribute space and is relevant for scalable attribute reduction in the areas of Bioinformatics and document classification. This technique reduces the complexity of movement of data in shuffle and sort phase of MapReduce framework. A comparative and performance analysis is conducted on larger attribute space (high dimensional) hybrid decision systems. The comparative experimental results demonstrated that the proposed MR_IMQRA algorithm obtained good sizeup/speedup measures and induced classifiers achieving better classification accuracy.

Keywords: Fuzzy-rough sets · Hybrid decision systems · Attribute reduction · Iterative MapReduce · Apache Spark · Vertical partitioning

1 Introduction

The decision system with different types of attributes (e.g., categorical, real-valued, set-valued, and boolean) is called as Hybrid Decision System (HDS). Traditional approaches like rough sets [7] require discretization of numeric attributes to perform attribute reduction, which can result in significant information loss [4]. Extensions were proposed to classical rough set theory to overcome this problem. Dubois and Prade [2] developed fuzzy-rough sets and rough-fuzzy sets, as hybrid approaches combining strengths of fuzzy sets and rough sets together.

B. Deka et al. (Eds.): PReMI 2019, LNCS 11941, pp. 306–316, 2019.
https://doi.org/10.1007/978-3-030-34869-4_34

Out of these, fuzzy-rough sets have evolved as a standard approach for feature subset selection in hybrid decision systems.

Jensen et al. [5], proposed new approaches for fuzzy-rough attribute reduction, where, different algorithms were designed based on *attribute dependency degree measure* and *discernibility matrix* methods. Cornelis [1] proposed a selection of the subset of features with fuzzy decision reducts and designed a Modified Quick Reduct Algorithm (MQRA). Sai Prasad et al. [8] proposed an efficient approach IMQRA (Improved Modified Quick Reduct Algorithm) for fuzzy decision reduct computation based on MQRA [1] by incorporating a simplified computational model and positive region removal.

All the existing fuzzy-rough reduct computation algorithms are sequential and can only handle smaller size datasets. A little attention has been paid on parallel/distributed techniques for fuzzy-rough attribute reduction to deal with large-scale datasets, particularly high dimensional datasets. Therefore, it is the need of the hour to research the issue of fuzzy-rough set based attribute reduction in parallel/distributed approach.

With the objective of scalable fuzzy-rough set feature selection, in this paper, a novel MapReduce based fuzzy-rough Improved Quick Reduct Algorithm (MR_IMQRA) is proposed. It is implemented on iterative MapReduce framework of *Apache Spark* [12]. Existing classical rough set based MapReduce approaches for attribute reduction [11] use object space partitioning (horizontal partitioning technique) of the input data to the nodes of the cluster. This technique results in complicated shuffle and sort phase for the datasets having the larger attribute space (high dimensionality). In contrast, proposed MR_IMQRA is attribute space (vertical partitioning technique) partitioning based approach suitable for datasets of larger attribute space prevalent in the areas of Bioinformatics and document classification.

The rest of this paper is organized as follows. The related details of fuzzy-rough attribute reduction and the existing IMQRA algorithm are given in Sect. 2. The proposed MR_IMQRA algorithm is discussed in Sect. 3, along with MapReduce based implementation details. Comparative experimental results and analysis are provided in Sect. 4. Finally, the conclusion of this paper is given in Sect. 5.

2 Related Work

This section provides related definitions, terminology and concepts for fuzzy-rough attribute reduction based on [2,5,9] and presents the existing work of Improved Modified Quick Reduct Algorithm (IMQRA) [8].

2.1 Fuzzy-Rough Attribute Reduction

Let $HDT = (U, C^h = C^s \cup C^r, \{d\})$ be a Hybrid Decision Table. Here U represents the set of objects, C^s is set of symbolic (categorical) attributes, C^r is set of numerical (real valued) attributes, and d is the symbolic decision attribute. In fuzzy rough sets [2,5,9], a fuzzy similarity relation is defined on objects for measuring the graded indiscernibility based on numeric attribute. For a numeric attribute $a \in C^r$, R_a represents fuzzy similarity relation, where $R_a(i,j)$, $\forall (i,j) \in U \times U$ gives fuzzy similarity for any pair of objects i, j. It is to be noted that, if an attribute is qualitative (categorical), then the classical indiscernibility relation is adopted, hence $a \in C^s$, $R_a(i,j)$ is taken as either 1 (if the object values are equal) or 0 (if the object values are not equal). The fuzzy similarity relation R can be extended for a set of attributes $P \subseteq C^h$ by using a specified t-norm Γ as given,

$$R_P(i,j) = \Gamma\left(R_a(i,j)\right) \ \ \forall i,j \in U \ and \ \forall a \in P \tag{1}$$

Many approaches are existed in the literature to construct similarity relation. In the proposed design, the following procedure is used to build similarity relation.

$$R_a(i,j) = max\left(min\left(\frac{a(i) - a(j) + \sigma(a)}{\sigma(a)}, \frac{a(j) - a(i) + \sigma(a)}{\sigma(a)}\right), 0\right) \tag{2}$$

Here, $\sigma(a)$ is standard deviation of attribute a. From Radzikowska-Kerry's fuzzy-rough set model [9], the fuzzy-rough *lower approximation* of a fuzzy set A on U can be defined by using fuzzy similarity relation R in U.

$$R \downarrow A(j) = \inf_{i \in U} I(R(i,j), A(i)) \tag{3}$$

where I is fuzzy implicator. From the *Lemma 1* of [8], the above (3) is simplified using the natural negation N_I of I for obtaining fuzzy-rough positive region based on $P \subseteq C^h$ as,

$$POS_P(j) = R_P \downarrow R_{d,j}(j) = \begin{cases} \min\limits_{i \in U_2(j)} \left(N_I(R_P(i,j))\right) & if \ \ U_2(j) \neq \phi \\ 1 & otherwise \end{cases} \tag{4}$$

Here, for an object, $j \in U$, the $U_1(j)$ represents the set of objects which belongs to the decision class of j and $U_2(j)$ represents the rest of the objects which belong to other decision classes. The resulting dependency degree measure is given as,

$$\gamma_P(\{d\}) = \frac{\sum_{i \in U} \ POS_P(i)}{|U|} \tag{5}$$

A fuzzy-rough *reduct* R is defined as minimal subset of attributes satisfying $\gamma_R(\{d\}) = \gamma_{C^h}(\{d\})$. The reduct generation can be done by using two control strategies, (i) *Sequential Forward Selection (SFS)*, and (ii) *Sequential Backward Elimination (SBE)*. In SFS strategy, reduct generation starts with an empty set, and attributes are incrementally added. It is possible in SFS strategy that the computed reduct may have some redundant attributes resulting as a super set of reduct (*superreduct*). In SBE strategy, reduct generation starts with whole attributes, and redundant attributes are removed one by one that results in minimal reduct. Even though SBE generates minimal reduct, the computational efficiency of the SFS strategy is more. In contrast to classical rough set approaches the redundancy in SFS reduct is very less owing to graded indiscernibility. Hence, the proposed algorithm is developed based on the attribute dependency degree measure approach that follows the SFS control strategy of the reduct generation, which has a less possibility of resulting in superreduct.

2.2 Improved Modified Quick Reduct Algorithm (IMQRA)

Sai Prasad et al. [8] proposed IMQRA algorithm based on the MQRA (Modified Quick Reduct Algorithm) [1]. A brief description of this algorithm is given below. Detailed theoretical and experimental description can be found in [8].

According to this algorithm, the fuzzy similarity relation for attribute $a \in C^h \cup \{d\}$ is represented as a symmetric similarity matrix with dimensions $U \times U$ and having entries $R_a(i, j),\ \forall i, j \in U$. IMQRA starts with reduct set P initialized to an empty set, and in each iteration, attribute inducing maximum gamma gain is included into P. Objects achieving lower approximation membership of 1 are named as $ABSOLUTE\_POS_P$. It is proved in [8] that, removal of $ABSOLUTE\_POS_P$ does not affect the subsequent computations while resulting in significant space and time complexity gains. The algorithm terminates when P satisfies the reduct properties.

3 Proposed Work

The proposed MR_IMQRA algorithm is a scalable distributed/parallel version of IMQRA [8]. This section describes the proposed algorithm (given in Algorithm 1), along with its features. The proposed MR_IMQRA algorithm consists of two steps, namely, (i) Computation of distributed fuzzy-rough similarity matrix, and (ii) Fuzzy-rough reduct computation.

Algorithm 1. MR_IMQRA

Input: *HDT:* (U, $C^h = C^s \cup C^r$, {d}), R_{sim}*:* Fuzzy similarity relation, *N:* Fuzzy Negation, Γ: t-Norm.
Output: Fuzzy superreduct B
Procedure:
$AttrRdd\langle attr, attrData\rangle \leftarrow readAsRdd(HDT)$
$Dpartition \leftarrow U/\{d\}$
$simMatRdd\langle attr, R_{attr}\rangle \leftarrow AttrRdd$.**map**{$\langle attr, attrData\rangle$ =>
  Construct matrix R_{attr} from $attrData$ using R_{sim}on each pair of objects
  EMIT $\langle attr, R_{attr}\rangle$
}
$B \leftarrow \{\}$, $R_B \leftarrow \{\}$
$\gamma_B(\{d\}) \leftarrow 0$, $\gamma_{old} \leftarrow -1.0$
$posRegSum \leftarrow 0$
while $\gamma_B\{d\} > \gamma_{old}$ AND $\gamma_B\{d\} \neq 1$ AND $|nonAbsPos| > 0$ **do**
  broadcast(DPartition), **broadcast**(R_B), **broadcast**(B)
  $\gamma_{old} \leftarrow \gamma_B(\{d\}$
  PosRdd$\langle attr, |POS_{B\cup\{attr\}}(\{d\}|\rangle \leftarrow simRdd$.**map**{$\langle attr, R_{attr}\rangle$ =>
  if $attr \in B$ **then**
    EMIT $\langle attr, -1\rangle$
  else
    $R_{B\cup\{attr\}} = \Gamma(R_B, R_{attr})$
    Compute $POS_{B\cup\{attr\}}(\{d\}$
    EMIT $\langle attr, |POS_{B\cup\{attr\}}(\{d\}|\rangle$
  end if
  $\langle bA, |POS_{B\cup\{bA\}}|\rangle = PosRdd$.**reduce**{($\langle a1, |POS_{B\cup\{a1\}}|\rangle, \langle a2, |POS_{B\cup\{a2\}}|\rangle$) =>

  if $|POS_{B\cup\{a1\}}| > |POS_{B\cup\{a2\}}|$ **then**
    EMIT $|POS_{B\cup\{a1\}}|$
  else
    EMIT $|POS_{B\cup\{a2\}}|$
  end if
  }
  $B \leftarrow B \cup \{bA\}$
  $posRegSum \leftarrow |POS_{B\cup\{bA\}}|$
  $R_{bA} = simMatRdd$.**filter**{$\langle attr, R_{attr}\rangle$ => $(attrNo == bA)$}.**map**(_._2)
  $R_B \leftarrow \Gamma(R_B, R_{bA})$
  $\langle nonAbsPos, absPos\rangle$ =getAbsolute(R_B))
  $\gamma_B = \dfrac{|absPos| + posRegSum}{|U|}$
  $simMatRdd\langle attr, R_{attr}\rangle = simMatRdd$.**filter**{$\langle attr, R_{attr}\rangle$ => $(attr! = bA)$}
  Restrict DPartition and U to nonPos objects
end while
return B

3.1 Computation of Distributed Fuzzy-Rough Similarity Matrix

As mentioned earlier, the proposed algorithm uses the vertical partitioning technique to distribute the input data to the nodes of the cluster. To realize this technique, a necessary preprocessing step is to be done on the input dataset. The input dataset is converted into the form, such that the rows correspond to the attributes, and the column corresponds to the objects. Each row is prefixed with an attribute number for preserving the attribute identity in the partitioning of the dataset. Algorithm receives input data in two portions of conditional attributes data and decision attribute data.

The portion containing the information of conditional attributes is read as in RDD form $AttrRdd\langle attr, attrData\rangle$. Here, the key $attr$ corresponds to the attribute number, and the value $attrData$ corresponds to the object information of the attributes. (Note: An RDD in Apache Spark represents a *Resilient Distributed dataset* for performing parallel operations over several partitions of data in the cluster. The notation, $RDD < key, value >$ represents the structure of each object of RDD in the pair of key and $value$). As the entire attribute information is available within a single partition, the requisite similarity matrices for all the conditional attributes can be computed in parallel using a single *map()* operation. Here, for each record of $AttrRdd$, the corresponding similarity matrix is constructed using Eq. (2) and a new transformed $RDD : simMatRdd\langle attr, R_{attr}\rangle$ is constructed, where the value R_{attr} corresponds to the similarity matrix of $attr$.

3.2 Fuzzy-Rough Reduct Computation

The fuzzy-rough similarity matrices, computed in the earlier section, acts as the input for this fuzzy-rough reduct computation. Initially, the reduct set B and the associated similarity matrix R_B is set to $NULL$, the $gamma$ value of the previous iteration γ_{old} is set to -1.0, $gamma$ value of current iteration $\gamma_B(\{d\})$ is set to zero. In each iteration, decision equivalence classes $Dpartition$, current reduct set B, and reduct similarity matrix R_B are broadcasted to all the nodes of the cluster, as every partition requires this information for further computations. The computation in an iteration of MR_ IMQRA requires computation of $POS_{B\cup\{attr\}}(\{d\})$ for all $attr \in C^h - B$ and inclusion of best attribute into B.

In an iteration of the proposed algorithm, if an attribute is already in B, then a dummy $\langle key, value\rangle$ pair is generated as $\langle attr, -1\rangle$, so that it is not considered subsequently into the reduct. For every attribute $attr \in C^h - B$, the computation of $R_{B\cup\{attr\}}$ is done using *t-norm* operation. The creation of $R_{B\cup\{attr\}}$ is done locally and the corresponding memory is removed after computation of $POS_{B\cup\{attr\}}(\{d\})$. Then a key-value pair $\langle attr, |POS_{B\cup\{attr\}}(\{d\}|\rangle$ is generated. Through the *reduce()* operation, the global best attribute bA is selected and added to the reduct set B. The *reduce()* operation of Apache Spark involves local *reduce()* followed by global *reduce()*. Therefore, in every partition the local best attribute is selected and only it's corresponding key-value pair is communicated to the global Reducer. Hence, the proposed vertical partitioning

based approach has a minimum data transfer across shuffle and sort phase in an iteration.

In the Driver, we need to update R_B as B is included with bA; this requires the availability of R_{bA} in the Driver. Hence a *filter()* operation is applied on $simMatRdd$ to select a record corresponding to bA, and it's associated similarity matrix is fetched to driver and updation of R_B is done using *t-norm* operation. In this way MR_IMQRA algorithm continues till γ_B value reaches to 1 or γ_B remains unchanged for the last m number of iterations (hence indicating that it can not get a better *gamma* measure by further adding more *attributes*) or $nonAbsPos$ has become zero. If the 2^{nd} terminating condition meets, then it removes m lastly added *attributes* from B. At the end it returns the final reduct set B.

3.3 Absolute Positive Region Removal in MR_IMQRA

The absolute positive region objects are those objects which achieve the total positive region membership of 1 [8]. The removal of such objects does not affect the computations of remaining iterations and reduces the space complexity of the algorithm efficiently. As an RDD is immutable, the removal of these objects from the respective similarity matrices will become complex and requires the creation of a new RDD. Therefore, in MR_IMQRA, the removal of absolute positive region objects is done only from $Dpartition$ and U. In the driver, using $getAbsolute$ function on R_B, $nonPos$ and $absPos$ objects are determined and $Dpartition$ and U are restricted to $nonPos$ objects. Hence, the rest of the computations are restricted to only non-positive region objects in mappers. In this way, MR_IMQRA becomes a real implementation of IMQRA algorithm by incorporating the absolute positive region removal aspect that gives computational advantages.

4 Experimental Results and Analysis

In this section, experiments are conducted to illustrate the utility of the proposed MR_IMQRA algorithm for scalable fuzzy-rough set based attribute reduction.

4.1 Experimental Setup

The experiments are conducted on a cluster of five nodes, out of which one node is master (driver), and the rest of the nodes are workers (slaves). Every machine has Intel Core i5-7500 Processor with a clock frequency of 3.4 GHz, having 8 GB of RAM and all the nodes are installed with Ubuntu 18.04 LTS, java 1.8.0_191, Apache Spark 2.3.1, and Scala 2.11.8.

As mentioned in earlier sections, the proposed algorithm is suitable for the datasets having moderate object space and larger attribute space (i.e., high dimensional datasets). Accordingly, the datasets are chosen and downloaded from GitHub [6]. The description of the datasets is given in Table 1.

Table 1. Datasets used in the experiments

Dataset	Objects	Features	Classes
Ovarian	253	15156	2
Yeoh	248	12625	6
Chin	118	22215	2
Buyczynski	127	22283	3

Table 2. Comparative results of MR_IMQRA with MR_MDLP_IQRA

Dataset	MR_IMQRA		MR_MDLP_IQRA	
	Computational Time(s)	Reduct length	Computational Time(s)	Reduct length
Ovarian	15.69	3	816.01	2
Yeoh	19.35	5	862.20	4
Chin	11.34	4	801.38	4
Burczynski	14.03	5	794.198	5

4.2 Comparison of MR_IMQRA and MR_MDLP_IQRA

In the literature, it is observed that no significant work is done in MapReduce based fuzzy-rough set attribute reduction. Hence, to assess the importance of the vertical partitioning technique in MR_IMQRA algorithm, a fusion of two approaches MR_MDLP [10] (for scalable discretization of numerical attributes with MapReduce) and MR_IQRA_IG [11] (for computation of reduct on categorical dataset obtained from MR_MDLP) are done and represented as MR_MDLP_IQRA. The source code of the MR_MDLP is made available in GitHub [3].

Table 3. Classification accuracy of MR_IMQRA, and MR_MDLP_IQRA (in %)

Dataset	MR_IMQRA		MR_MDLP_IQRA	
	SVM	Random forest	SVM	Random forest
Ovarian	98.68	98.68	68.42	31
Yeoh	74.67	72.00	28.00	24.00
Chin	80.56	75.00	61.11	69.00
Burczynski	53.85	69.23	58.97	48.71

Experiments are conducted on algorithms, MR_IMQRA, and MR_MDLP_IQRA for the given datasets. The obtained computational time (in seconds) and reduct length are given in Table 2. From the results, it can be observed that MR_IMQRA algorithm is taking considerably less computational time

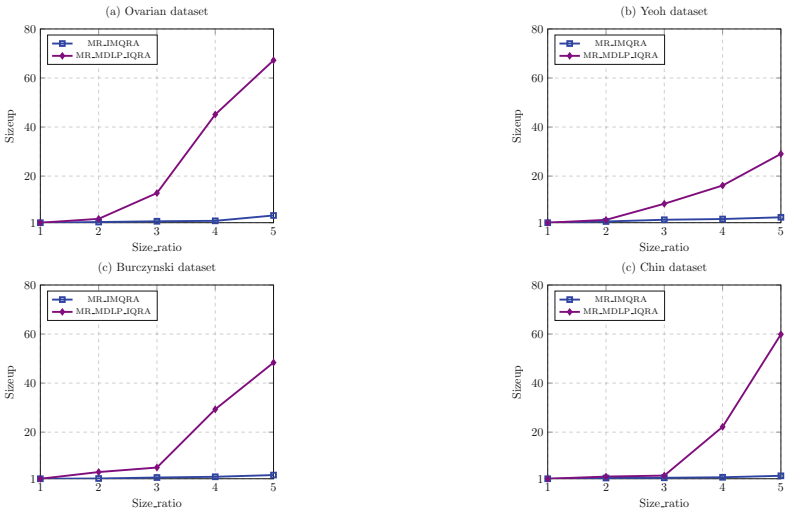

Fig. 1. Sizeup of MR_IMQRA and MR_MDLP_IQRA for different datasets

and almost giving similar reduct length like MR_MDLP_IQRA for all the datasets. The less computational times of MR_IMQRA are contrary to expectation as MR_IMQRA has a theoretical time complexity of $O(|C^h|^2|U|^2)$, where as MR_MDLP_IQRA has a time complexity of $O(|C^h|^2|U|log|U|)$. This phenomenon occurred because of vertical partitioning in MR_IMQRA leading to simplified shuffle and sort phase. In contrast, the horizontal partitioning in MR_MDLP_IQRA results in complex shuffle and sort phase, especially for high dimensional datasets.

Classification accuracy results using SVM, and Random forest classifiers for both algorithms, MR_IMQRA, and MR_MDLP_IQRA are given in Table 3 using 70% training data and 30% testing data. From the table, it is observed that MR_IMQRA achieved significantly higher classification accuracies than MR_MDLP_IQRA in both classifiers. It is observed that, both approaches are resulting in unrelated reducts. The classification analysis establishes that, MR_IMQRA has better potential in selection of relevant attributes in comparison to MR_MDLP_IQRA in which information loss due to discretization is affecting the selection of relevant reduct.

4.3 Performance Evaluation

Sizeup and speedup are the metrics used to asses the performance of the parallel algorithms. The sizeup experiments are conducted for different sizes of the datasets on the same cluster and are represented as follows,

$$Sizeup = \frac{Time\ taken\ by\ a\ dataset\ with\ corresponding\ size\_ratio}{time\ taken\ by\ a\ dataset\ of\ base\ size}.$$

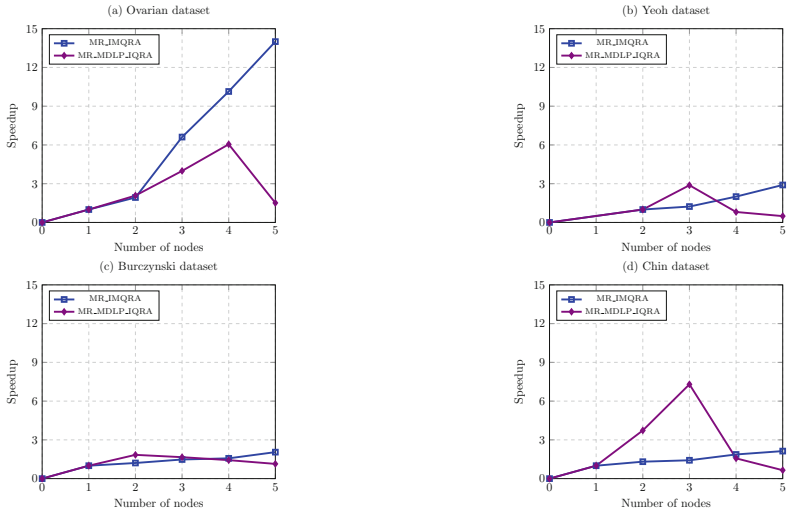

Fig. 2. Speedup of MR_IMQRA and MR_MDLP_IQRA for different datasets

Where, $Size\_ratio = \frac{H_{size}}{H_{base\_size}}$.
Here, H_{base_size} represents base dataset size and H_{size} represents current dataset size. The number of computers kept as five nodes. Each dataset size is increased with 20%, 40%, 60%, 80%, and 100% of attributes in the dataset. Figure 1 shows the sizeup performance results of MR_IMQRA and MR_MDLP_IG algorithms for different datasets with varying sizes of attribute space. Sizeup results shown in Fig. 1 establish that MR_IMQRA obtained a sub-linear sizeup measures in contrast to quadratic sizeup measures in MR_MDLP_IQRA.

Speedup experiments are conducted for the same datasets on different sizes of the cluster and it is represented as follows,

$$Speedup(n) = \frac{Computational\ time\ taken\ by\ a\ single\ node}{Computational\ time\ taken\ by\ a\ cluster\ of\ n\ nodes}.$$

Figure 2 shows the speedup results of MR_IMQRA and MR_MDLP_IQRA algorithms for different datasets with varied nodes from 1 to 5. MR_IMQRA has obtained the best speedup values in Ovarian dataset. In all the datasets MR_IMQRA has a steady increase in speedup measure values with increase in number of nodes, where in oscillations are observed in the results of MR_MDLP_IQRA. The results emperitically establish that proposed MR_IMQRA is recommended as a scalable solution for fuzzy-rough set reduct computation in high dimensional datasets.

5 Conclusion

The proposed work introduces a MapReduce based MR_IMQRA algorithm for attribute reduction in datasets of lesser object space and larger attribute space

(high dimensional datasets) prevalent in Bioinformatics and document classification. MR_IMQRA is a distributed version of IMQRA algorithm and uses vertical partitioning to distribute the input dataset. The impact of vertical partitioning technique and the removal of the absolute positive region is shown vividly in the experimental analysis by obtaining reduct in lesser computational time and with reasonable sizeup and speedup values in comparison to horizontal partitioning based MR_MDLP_IQRA. The proposed algorithm also induced significantly better classifiers. In future, a MapReduce based SBE approach will be augmented to MR_IMQRA to remove existence of redundant attributes, if any resulting from SFS approach.

Acknowledgement. This work is supported by Department of Science and Technology (DST), Government of India under ICPS project [grant number: File DST/ICPS/CPS-Individual/2018/579(G)].

References

1. Cornelis, C., Jensen, R., Martn, G.H., Slezak, D.: Attribute selection with fuzzy decision reducts. Inf. Sci. **180**(2), 209–224 (2010)
2. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets*. Int. J. Gen. Syst. **17**(2–3), 191–209 (1990)
3. Lin, H.: MDLP-discretization (2017). https://github.com/hlin117/mdlp-discretization. Accessed 21 Nov 2017
4. Hu, Q., Yu, D., Xie, Z.: Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recogn. Lett. **27**(5), 414–423 (2006)
5. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. IEEE Trans. Fuzzy Syst. **17**(4), 824–838 (2009)
6. Ramey, J.: Datamicroarray (2016). https://github.com/ramhiser/datamicroarray/tree/master/data. Accessed 11 Jan 2016
7. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**(5), 341–356 (1982)
8. Prasad, P.S.V.S.S., Rao, C.R.: An efficient approach for fuzzy decision reduct computation Trans. Rough Sets **17**, 82–108 (2014). https://doi.org/10.1007/978-3-642-54756-0_5
9. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets Syst. **126**(2), 137–155 (2002)
10. Ramrez-Gallego, S., et al.: Data discretization: taxonomy and big data challenge. Wiley Interdisciplinary Rev.: Data Min. Knowl. Discov. **6**, 5–21 (2015)
11. Sai Prasad, P.S.V.S., Bala Subrahmanyam, H., Singh, P.K.: Scalable IQRA_IG algorithm: an iterative mapreduce approach for reduct computation. In: Krishnan, P., Radha Krishna, P., Parida, L. (eds.) ICDCIT 2017. LNCS, vol. 10109, pp. 58–69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-50472-8_5
12. Zaharia, M., et al.: Apache spark: a unified engine for big data processing. Commun. ACM **59**, 56–65 (2016)