Fuzzy Rough Data Reduction Using SVD

Rama Devi Y, Venu Gopal P, and Sai Prasad PSVS

***Abstract*—Fuzzy rough data reduction algorithm proposed in [1] is not convergent on higher dimensional data due to its computational complexity increases exponentially as the number of input attributes and fuzzy sets increase. This paper shows how singular value decomposition can be used as a useful preprocessing method in order to achieve fuzzy rough reduct convergence on higher dimensional datasets. Eight datasets from UCI repository have been taken for the experimentation.**

***Index Terms*—Ant olony ptimization (ACO), Computational Complexity, Fuzzification, Fuzzy-Rough Reduct, Singular Value Decomposition (SVD).**

I. Introduction

Many problems in machine learning involve high dimensional descriptions of input features. A high-dimensional dataset increases the chances that a data-mining algorithm will find spurious patterns that are not valid in general. It is therefore not surprising that much research has been carried out on dimensionality reduction. However, existing work tends to destroy the underlying semantics of the features after reduction (e.g. transformation-based approaches) or require additional information about the given data set for thresholding (e.g. entropy-based approaches).

The task of feature selection is to significantly reduce dimensionality by locating minimal subsets of features, at the same time retaining data semantics. The rough set theory has been used as such a feature selection technique with some success for greatly reducing dataset dimensionality and selecting the relevant features in aiding classification tasks. The minimal subset of original features found by rough sets which are most informative is termed as reduct. But the traditional rough set theory methods reliance on discretized dataset, which is a source of information loss, is a significant drawback. It cannot handle real-valued or continuous-valued datasets, and it has also been observed that they are often inadequate for finding minimal reduction of features in a dataset

To overcome this, a new feature selection technique based on hybrid variant of rough sets has been proposed [1], which can handle both discrete-valued and real-valued datasets, and also performs further reduction of features.

However, the style of search process that is employed in this algorithm may terminate having found only locally optimal feature subsets. For this, a new feature selection technique based on hybridization of ant colony optimization and fuzzy-rough sets was presented in [3], which can find out globally optimal subsets with minimal reduct.

In this paper, we prove that the algorithm presented in [1] is not convergent on many real datasets due to its computational complexity increases exponentially when Cartesian product is taken to find out the number of equivalence classes with respect to number of attributes and fuzzy sets. The comparison between fuzzy rough and ant fuzzy rough algorithms has been done with respect to reduct and corresponding classification accuracy.

The rest of the paper is structured as follows: section 2 provides the theory behind fuzzy rough data reduction and its variant ant based feature selection. While section 3 briefs about the fuzzification method which we have used to fuzzify the dataset. Section 4 describes about the computational complexity of fuzzy rough reduct. Section 5 presents an algorithm for reducing the dimensionality of dataset using SVD. Section 6 demonstrates the experimental results on different datasets and Section 7 concludes the paper.

II. Fuzzy-Rough Attribute Reduction

In the same way that crisp equivalence classes are central to rough sets, fuzzy equivalence classes are central to the fuzzy-rough sets. For typical applications, this means that the decision values and the conditional values may all be fuzzy. The family of normal fuzzy sets produced by a fuzzy partitioning of the universe of discourse can play the role of fuzzy equivalence classes.

A. Fuzzy Lower and Upper Approximations

The fuzzy lower and upper approximations are fuzzy extensions of their crisp counterparts. Informally, in crisp rough set theory, the lower approximation of a set contains those objects that belong to it with certainty. The upper approximation of a set contains the objects that possibly belong. The definitions given in [5] diverge a little from the crisp upper and lower approximations, as the memberships of individual objects to the approximations are not explicitly available. As a result of this, the fuzzy lower and upper approximations are redefined as:

$$\mu_{\underline{P}X}(x) = \sup_{F \varepsilon U/P} \min(\mu_F(x), \inf_{y \varepsilon U} \max\{1-\mu_F(y), \mu_X(y)\}) \quad (1)$$

$$\mu_{\overline{P}X}(x) = \sup_{F \varepsilon U/P} \min(\mu_F(x), \sup_{y \varepsilon U} \min\{\mu_F(y), \mu_X(y)\}) \quad (2)$$

The tuple $< \underline{P}X, \overline{P}X >$ is called a fuzzy-rough set.

For an individual feature, *a*, the partition of the universe by {a} (denoted U/IND {a}) is considered to be the set of those fuzzy equivalence classes for that feature. For subsets of feature, the following is used:

Manuscript received January 14, 2011; revised May 12, 2011.

Y Ramadevi is a Professor in Dept. of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India. (e-mail:yrdcse.cbit@gmail.com)

P Venugopal is a research associate with the Dr YRamadevi.(e-mail: venu062k5@gmail.com).

PSVS Sai Prasad is Assistant Professor in the Dept. of Computer sandInformation Sciences, University of Hyderabad, Hyderabad, India. (e-mail: saics@uohyd.ernet.in).

$$\frac{U}{P} = \otimes \left\{ a \in P : \frac{U}{IND(\{a\})} \right\} \quad (3)$$

Each set in U/*P* denotes an equivalence class. The extent to which an object belongs to such an equivalence class is therefore calculated by using the conjunction of constituent fuzzy equivalence classes say F_i, i = 1, 2… n:

$$\mu_{F1 \cap F2 \cap ..... Fn}(x) = \min(\mu F_1(x), \mu F_2(x), ..., \mu F_n(x)) \quad (4)$$

B. Fuzzy-Rough QuickReduct Algorithm

Fuzzy-Rough Feature Selection (FRFS) [1] builds on the notion of the fuzzy lower approximation set to enable reduction of datasets containing real-valued features. The process becomes identical to the crisp approach when dealing with nominal well-defined features.

The crisp positive region in the standard RST is defined as the union of the lower approximations. By the extension principle, the membership of an object x ε U, belonging to the fuzzy positive region has been defined by:

$$\mu_{POS_{P(Q)}}(x) = \sup_{\substack{X \epsilon U \\ Q}} \mu_{\underline{P}X}(x) \quad (5)$$

Using the definition of the fuzzy positive region, a dependency function (kappa) between a set of features *Q* and another set *P* can be defined as follows:

$$\gamma'_P(Q) = \frac{|\mu_{POS_{P(Q)}}(x)|}{|U|} = \frac{\sum_{x \epsilon \mu} \mu_{POS_{P(Q)}}(x)}{|U|} \quad (6)$$

The above kappa function will be acting as fitness function which will be used to filter out irrelevant attributes. This value ranges between 0 and 1, and higher the value of this function the more likely the attribute will be candidate for reduct.

Based on this function a fuzzy rough QuickReduct algorithm has been defined, which terminates when previous iteration best kappa value equal to the current iteration best value. So, basically it terminates when there is no increase in the dependency measure. For more details about this algorithm can be referred in [1]. However, the search process that is employed in this algorithm is greedy and it may end up having found only locally optimal feature subset. The next section 2.3 presents an algorithm based on ant colony optimization which can be used to find out globally optimal feature subset.

C. Ant Based Fuzzy-Rough Quick Reduct Algorithm

Ant colony optimization (ACO) is a meta-heuristic in which colonies of artificial ants cooperate in finding good solutions to difficult discrete optimization problems [8]. ACO is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time. Additionally, it can be the case that ants discover the best feature combinations as they proceed throughout the search space.

In this algorithm, each ant will be constructing a reduct and best reduct is chosen from the set of reducts which has highest kappa value. So, here again the kappa function defined in Eq. (6) will remain as fitness function. The number of ants can be set equal to the number of conditional attributes present in the dataset. A predefined kappa threshold needs to be set and if the best reduct found in the current generation doesn't satisfy this threshold value global updation is performed. The step by step details regarding this algorithm can be found in [7].

III. Fuzzification Method

Before applying fuzzy rough feature selection algorithms, the dataset need to be fuzzified. Fuzzification is the process where the crisp quantities are converted to fuzzy (crisp to fuzzy). By identifying some of the uncertainties present in the crisp values, we form the fuzzy values. The conversion of fuzzy values is represented by the membership functions.

There are different ways of doing fuzzification, and this process typically involves deriving membership functions and finding out parameter values for those functions which will depend upon the data under consideration. In this work, the Fuzzification method presented by SK Pal [4] has been used. If we perform fuzzification for a dataset using this standard method, the modified dataset will contain three columns per attribute consisting of membership values with respect to fuzzy sets: Low, Medium and High.

IV. Computational Complexity of Fuzzy-Rough Reduct

The computational complexity of fuzzy rough QuickReduct algorithm [1] increases exponentially when we want to calculate the dependency value (kappa) for different attribute combinations by taking the Cartesian product of fuzzy sets (fuzzy equivalence classes) representing each attribute. The following shows with an example how computational complexity increases as the number of fuzzy sets increase. Here, Na, Za, Pa represents fuzzy sets with respect to attribute 'a'.

Example, if P = {a, b}, U/IND ({a}) = {Na, Za}, U/IND ({b}) = {Nb, Zb}, then:

U/P = {Na∩Nb, Na∩Zb, Za∩Nb, Za∩Zb}

Similarly, if P = {a, b}, U/IND ({a}) = {Na, Za, Pa}, U/IND ({b}) = {Nb, Zb, Pb}, then:

U/P = {Na∩Nb, Na∩Zb, Na∩Pb, Za∩Nb, Za∩Zb, Za∩Pb, Pa∩Nb, Pa∩Zb, Pa∩Pb}

From the above, each set in U/P denotes an equivalence class. The computational complexity will be $O(m^n)$, where 'n' represents number of attributes and 'm' represents number of fuzzy sets. So for {a, b}, with respect to three fuzzy sets the number of equivalence classes are 9 and similarly, if we want to calculate equivalence classes for 15 attributes, the number of equivalence classes after computing Cartesian product will be 3^{15} (1,43,48,907). Now for each fuzzy equivalence class, the expression described in eq. has to be calculated and then supremum of that has to be taken. In this work, we have observed that after 3^{12} the memory was exhausted. To overcome this, the next section 5 presents

SVD which was used as a preprocessing method before applying fuzzy rough approaches on higher dimensional datasets.

V. Singular Value Decomposition

Singular value decomposition is an important factorization of a rectangular real or complex matrix into a series of linear approximations that expose the underlying structure of the matrix. Applications which employ the SVD include computing the pseudo inverse, least squares fitting of data, matrix approximation, and determining the rank, range and null space of a matrix. It is a matrix factorization technique commonly used for projecting high dimensional (sparse) data into a low dimensional (dense) space.

The SVD of m × n matrix **P** with rank 'k' is its factorization into a product of three matrices

$$\mathbf{P} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^{\mathbf{T}} \tag{7}$$

Where, U is an m-by-m unitary matrix over k, the matrix S is m-by-n diagonal matrix with nonnegative real numbers on the diagonal, and V^T denotes the conjugate transpose of V an n-by-n unitary matrix over K. Such a factorization is called a singular-value decomposition of P [6].

It is possible to reduce dimensionality by choosing the k most significant dimensions from the factor space which is then used for estimating the original vectors. Many researchers have emphasized the usage of SVD as dimensionality reduction technique. In particular, Ravi Kanth et al. [9] showed that the low dimensional data obtained to use the SVD support fast searching because overhead of computation is reduced. The following section describes an SVD algorithm for dimensionality reduction.

A. SVD Dimensionality Reduction

Let P be m x n matrix, then SVD dimensionality reduction of matrix P is performed by following steps:

(i) Calculate the SVD of P and obtain the matrices U, S and V^T of dimension m ×r, r ×r, and r×n, respectively. Here, 'r' represents rank of the matrix.

(ii) Compute product of matrix 'U' with square root of matrix 'S' i.e.,

$$U \times \sqrt{S} \tag{8}$$

Call the resultant matrix as of size m x k

Now, reduce the dimensionality until the following condition is satisfied:

$$\frac{Trace\ (S, 1{:}k1)}{Trace(S)} * 100 > 90\% \tag{9}$$

where, k1<k

Check the above condition for different values of k1 and get the transformation matrix of size m x k1. Suppose, for the value of 65 Eq. (9) is greater than 90%, then the Eq. (8) matrix m x k will reduce to 90x65 where m=90. In this work, the SVD of the matrix (or the dataset) has been calculated using the function in MATLAB 7.6.

In the above expression greater than 90% makes sure that there will be no information loss in the transformed dataset.

VI. Experimentation Results

Eight datasets from UCI repository [2] have been taken as described in Table I consisting of real valued features. These datasets have been divided in to training (60%) and testing (40%). The conditional attributes values of the training set have been fuzzified using the fuzzification method described in section 3. Initially, all datasets were given input to fuzzy rough and ant fuzzy rough algorithms in order to find out the reduct. But, from the Tables 2 and 3 it can be observed that for some datasets the reduct convergence has been achieved and for some it is not. So, SVD dimensionality reduction technique (as described in section 3) was used as a preprocessing method to achieve convergence. The reduced datasets classification accuracy was checked on five different classifiers: JRIP, PART, Naïve Bayes, Random Tree and CART. The classification accuracy of original dataset has also been taken which has not undergone feature selection process for comparison.

In the Table III, No Convergence (NC) refers to either the algorithm was going into infinite search loop or the memory has been exhausted. Table II shows how the fuzzy rough algorithm is convergent on lower dimensional datasets. From the Tables II and III, clearly it can be observed that fuzzy rough data reduction algorithm is not convergent on all real datasets, specifically on higher dimensional datasets due to its computational complexity. Also, Table III shows that on some datasets ant fuzzy rough algorithm was converging and on some fuzzy rough algorithm was converging to reduct.

Table I: Datasets Specification

Dataset	No. of Instances	Attributes	No. of Tuples		No. of Decision Classes
			Training	Testing	
Iris	150	5	90	60	3
Fruit	16	5	10	16	2
Glass	214	10	130	84	6
Abalone	4177	9	2506	1671	27
Water	521	39	312	209	3
Vehicle	846	19	507	339	4
Statlog	2310	20	1386	924	7
Spambase	4601	58	2760	1841	2

TABLE II: FUZZY ROUGH REDUCT CONVERGENCE ON LOWER DIMENSIONAL DATASETS (WITHOUT SVD)

Dataset	Original Features	FRQR		Ant FRQR	
		Reduct	kappa	Reduct	Kappa
Iris	5	{1,2,3,4}	0.4693	{1,2,3,4}	0.4693
Fruit	5	{2}	0.6000	{1,2}	0.5000
Glass	10	{1,2,3,4,5,6,8,9}	0.4752	{1,3,4,5,6,7,8,9}	0.4745
Abalone	9	{1}	0.5000	{1, 2}	0.5000

FRQR- Fuzzy-Rough QuickReduct

TABLE III: FUZZY-ROUGH REDUCT CONVERGENCE USING SVD ON HIGHER DIMENSIONAL DATASETS

Dataset	No. of features	FRQR	Ant FRQR		SVD features	SVD based FRQR		SVD based Ant FRQR	
			Reduct	kappa		Reduct	kappa	Reduct	kappa
Water	39	NC	{28,31,36}	0.4275	10	{1,2,3,4,5,6, 7,8,9,10}	0.5149	{1,2,5}	0.2795
Vehicle	19	NC	{9,8,11}	0.2866	10	{1,2,3,4,5,6, 7,8,9,10}	0.5155	{2,1,8}	0.2484
Stat log	20	NC	NC	NC	5	{1,2,3,4,5}	0.2579	{1,2,3, 4,5}	0.2579
Spambase	58	{41,46}	NC	NC	6	{1,2,3,4,5,6}	0.3084	{1,2,3, 4,5,6}	0.3084

TABLE IV: SVD BASED CLASSIFICATION ON WATER DATASET

Classifier	Original Accuracy		SVD based FRFS Accuracy		SVD based Ant FRFS Accuracy	
	Train	Test	Train	Test	Train	Test
JRIP	95.51	81.81	86.63	76.19	74.86	74.60
PART	99.67	83.73	83.95	74.60	74.86	74.60
Naive Bayes	86.85	86.60	82.88	76.98	74.86	71.42
Random Tree	100	78.46	100	69.04	100	65.87
CART	84.61	78.94	83.42	77.77	74.86	74.60

TABLE V: SVD BASED CLASSIFICATION ON VEHICLE DATASET

Classifier	Original Accuracy		SVD based FRFS Accuracy		SVD based Ant FRFS Accuracy	
	Train	Test	Train	Test	Train	Test
JRIP	76.72	64.60	77.30	58.12	52.63	40.88
PART	89.34	65.48	90.78	55.66	75.98	50.24
Naive Bayes	46.54	42.77	62.82	56.65	45.06	37.43
Random Tree	100	60.76	100	59.11	100	49.75
CART	92.89	66.37	91.11	59.11	67.10	53.20

TABLE VI: SVD BASED CLASSIFICATION ON STATLOG DATASET

Classifier	Original Accuracy		SVD based FRFS Accuracy		SVD based Ant FRFS Accuracy	
	Train	Test	Train	Test	Train	Test
JRIP	98.34	94.69	81.94	73.69	81.94	73.69
PART	99.20	94.15	95.42	80.90	95.42	80.90
Naïve Bayes	80.73	80.08	71.48	70.09	71.48	70.09
Random Tree	100	92.31	100	83.42	100	83.42
CART	97.83	92.96	92.41	81.80	92.41	81.80

TABLE VII: SVD BASED CLASSIFICATION ON SPAMBASE DATASET

Classifier	Original Accuracy		SVD based FRFS Accuracy		SVD based Ant FRFS Accuracy	
	Train	Test	Train	Test	Train	Test
JRIP	94.31	93.15	90.45	87.31	90.45	87.31
PART	98.33	93.69	92.02	87.04	92.02	87.04
Naive Bayes	80.57	79.30	78.14	78.44	78.14	78.44
Random Tree	99.96	90.65	100	86.32	100	86.32
CART	96.92	92.01	91.84	86.50	91.84	86.50

But, it can be seen that after applying SVD, the convergence has been achieved. Ant based approach convergence will be more than the fuzzy rough because the algorithm is approximate. In Table III, the SVD features are with respect to algorithm described in section V (A).

The reduced dataset found by fuzzy rough and ant fuzzy rough based on SVD were divided in to training and testing, and then the classification accuracies were taken w.r.t different classifiers. Classification accuracies for fuzzy and ant fuzzy is same on statlog and spambase datasets because the reduct found by these approaches is similar. And, if you compare the SVD based accuracies of fuzzy rough and ant fuzzy rough w.r.t original are slightly less, but these accuracies have been achieved with far less number of attributes compared to original. Even though the results have not been included for SVD based classification on lower dimensional datasets, but it was observed that results were almost same with respect to both the approaches. Also, the focus of this work was more on the convergence rather than improving the classification accuracy of the algorithm.

VII. CONCLUSIONS

This paper identified how fuzzy rough QuickReduct algorithm is not able to achieve convergence on higher dimensional datasets due to its computational complexity. It also demonstrated how SVD can be adopted as a useful preprocessing tool to achieve convergence on some datasets. The comparison between fuzzy rough and ant fuzzy rough algorithms based on SVD has been done with respect to reduct and corresponding classification accuracy. Other preprocessing methods can also be adopted here to achieve fuzzy rough reduct convergence. Also, one needs to use high performance computing facility in order to avoid infinite search loop and out of memory problems to achieve convergence on all datasets. It is also expected that to some extent the convergence can be achieved by modifying the termination criteria of the fuzzy rough algorithm.

ACKNOWLEDGEMENT

We thank Prof. C.R.Rao, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, for his valuable guidance throughout this work.

REFERENCES

[1] J. Richard and Q. Shen “Fuzzy-rough sets for descriptive *dimensionality* reduction”, *Fuzzy Systems, Proceedings of the 2002 IEEE International Conference*, pp.29- 34, 2002.

[2] UCI Repository of machine learning database. *http://archive.ics.uci.edu/ml/datasets.html*.

[3] R.Jensen *and* Q. Shen “Fuzzy-Rough Data Reduction with Ant Colony Optimization”, *Fuzzy Sets and Systems*, 149(1):5-20. 2005.

[4] S.K. Pal, S. Mitra and, “Multilayer Perceptron, Fuzzy Sets, and Classification”, *IEEE Transactions on Neural Networks*, Vol. 3, No.5, September 1992.

[5] Dubois and H. Prade “Putting rough sets and fuzzy sets together”, In R. Slowinski (Ed.), *Intelligent Decision Support*, Kluwer Academic Publishers, pp. 203–232. 1992.

[6] Jae Kyeong Kim, Yoon Ho Cho “Using Web Usage Mining and SVD to Improve E-commerce Recommendation Quality”, *PRIMA*, pp. 86-97, 2003.

[7] Dr. Y. Rama Devi, P. Venugopal, PSVS Sai Prasad “Impact Analysis of Jensen and SK Pal Fuzzification in Classification” *Proceedings of the 1st Amrita ACM-W Celebration of Women in Computing (A2CWiC)*, Amrita University, Sep 16-17, 2010.

[8] M. Dorigo, T Stutzle, Ant Colony Optimization. Harlow, *England*: *Addison-Wesley*, 1999.

[9] K. V. Ravi Kanth , Divyakant Agrawal , Ambuj Singh, Dimensionality reduction for similarity searching in dynamic databases, *Proceedings of the 1998 ACM* SIGMOD *international conference on Management of data*, pp.166-176, June 01-04, 1998, Seattle, Washington, United States.