

An Independent Rough Set Approach Hybrid with Artificial Bee Colony Algorithm for Dimensionality Reduction

¹Nambiraj Suguna and ²Keppana Gowder Thanushkodi
¹Department of Computer Science and Engineering,
²Director,
Akshaya College of Engineering and Technology,
Coimbatore, Tamil Nadu, India

Abstract: Problem statement: Dimensionality reduction is viewed as an important pre-processing step for pattern recognition and data mining. As the classical rough set model considers the entire attribute set as a whole to find the subset, comparing all possible combinations of sets of attributes is difficult. **Approach:** In this study, we have introduced an improved Rough Set-based Attribute Reduction (RSAR) namely Independent RSAR hybrid with Artificial Bee Colony (ABC) algorithm, which finds the subset of attributes independently based on decision attributes (classes) at first and then finds the final reduct. Initially the instances are grouped based on decision attributes. Then the Quick Reduct algorithm is applied to find the reduced feature set for each class. To this set of reducts, the ABC algorithm is applied to select a random number of attributes from each set, based on the RSAR model, to find the final subset of attributes. **Results:** The performance is analyzed with five different medical datasets namely Dermatology, Cleveland Heart, HIV, Lung Cancer and Wisconsin and compared with six other reduct algorithms. The reduct from the proposed approach reaches greater accuracy of 92.36, 86.54, 86.29, 83.03 and 88.70 % respectively. **Conclusion:** The experiments states that the proposed approach reduces the computational cost and improves the classification accuracy when compared to some classical techniques.

Key words: Artificial bee colony, rough set, k-nearest neighbor, genetic algorithm, dimensionality reduction, proposed approach, classification accuracy, data mining, Bees Colony Optimization (BCO), attribute reduction

INTRODUCTION

Dimensionality Reduction or Feature subset selection is one of the important steps in data mining (Ahmed *et al.*, 2009; Ngo and Nguyen, 2009; Selamat *et al.*, 2010; Shylaja *et al.*, 2010). Numerous features have been acquired and stored in databases due to the growth and development in real-time applications. Considering the entire features may slowdown the learning process and may reduce the performance of the classifier because of redundant and irrelevant features (Ahmed *et al.*, 2009). It is essential to reduce the dimensionality by selecting most relevant features which results in decreasing the measuring cost, transmission and storage cost and compact classification models. There are several techniques that have been proposed in the literature: Filter, wrapper and embedded (Selamat *et al.*, 2010), unsupervised (Shylaja *et al.*, 2010) and supervised (Ngo and Nguyen, 2009; Jensen and Shen, 2007) Rough set theory provides a

mathematical tool that can be used for both feature selection and knowledge discovery (Jensen and Shen, 2007). It helps us to find out the minimal attribute sets called 'reducts' to classify objects without deterioration of classification quality and induce minimal length decision rules inherent in a given information system. The idea of reducts has encouraged many researchers in studying the effectiveness of rough set theory in a number of real world domains. However, it is not possible in the theory to say whether two attribute values are similar and to what extent they are the same. In our previous work (Suguna and Thanushkodi, 2010a) the rough set approach hybrid with Bees Colony Optimization (BCO) had been proposed to find the better reducts. Another limitation in the rough set theory is that the feature subset is constructed starting from the entire dimension, which involves more computational cost. As an extension of our previous work, a novel Rough set approach is proposed in this study to find the reducts in order to reduce the

Corresponding Author: N. Suguna, Computer Science and Engineering, Akshaya College of Engineering and Technology, Coimbatore, Tamil Nadu, India

computational complexity and to acquire a more accurate feature subset. In this proposed method, initially, the instances are grouped based on the decision attribute and then the reduct is found for each class. The common attributes from all these reduct sets are grouped to form the core reduct and the remaining attributes are considered for further reduction. From each set of reducts, the Artificial Bee Colony (ABC) algorithm based Rough Set-based Attribute Reduction (RSAR) model is applied to receive the final reduct. The rest of the study is organized as follows: The following text presents the basis of QuickReduct algorithm followed by explaining the ABC and our proposed method. Then the experiments are conducted with the databases taken from UCI machine learning repository and results are presented. And the study is concluded with a discussion on the results.

MATERIALS AND METHODS

Quick reduct: Rough set theory is an extension of conventional set theory that supports approximations in decision making. QuickReduct is one of efficient reduct algorithm presented in the literature. In QuickReduct, the reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same quality of classification as the original. A reduct is defined as a subset R of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$. A given dataset may have many attribute reduct sets, so the set R of all reducts is defined as:

$$R = \{X : X \subseteq C, \gamma_R(D) = \gamma_C(D)\}$$

The intersection of all the sets in R is called the core, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the dataset. In RSAR, a reduct with minimum cardinality is searched for; in other words an attempt is made to locate a single element of the minimal reduct set $R_{min} \subseteq R$:

$$R_{min} = \{X : X \in R \forall Y \in R | |X| \leq |Y|\}$$

The QuickReduct algorithm given in Fig. 1, attempts to calculate a minimal reduct without exhaustively generating all possible subsets. It starts with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in dependency, until this produces its maximum possible value for the dataset.

```

QUICKREDUCT (C,D)
C, The set of all conditional features
D, The set of decision features
(1) R ← { }
(2) do
(3)     T ← R
(4)     ∀x ∈ (C - R)
(5)     if  $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$ 
(6)         T ← R U { x }
(7)     R ← T
(8) until  $\gamma_R(D) == \gamma_C(D)$ 
(9) Return R
    
```

Fig. 1: The Quickreduct algorithm

Independent QuickReduct (IQR): The problem of finding a minimal reduct of an information system has been a subject of much research (Suguna and Thanushkodi, 2010b). The most basic solution to locating such a reduct is to simply generate all possible reducts and choose any with minimal cardinality. Obviously, this is an expensive solution to the problem and is only practical for very simple datasets. Most of the time only one minimal reduct is required, so all the calculations involved in discovering the rest are pointless. To improve the performance of the above method, an element of pruning can be introduced. By noting the cardinality of any pre- discovered reducts, the current possible reduct can be ignored if it contains more elements. However, a better approach is needed, one that will avoid wasted computational effort. Note that an intuitive understanding of QuickReduct implies that, for a dimensionality of n, $(n^2+n)/2$ evaluations of the dependency function may be performed for the worst-case dataset. According to the QuickReduct algorithm, the dependency of each attribute is calculated and the best candidate chosen. The next best feature is added until the dependency of the reduct candidate equals the consistency of the dataset (1 if the dataset is consistent). This process, however, is not guaranteed to find a minimal reduct. Using the dependency function to discriminate between candidates may lead the search down a non-minimal path. It is impossible to predict which combinations of attributes will lead to an optimal reduct based on changes in dependency with the addition or deletion of single attributes. It does result in a close-to-minimal reduct, though, which is still useful in greatly reducing the dimensionality of the dataset.

Normally all the reduct algorithms start with an empty set and adds in turn, one feature at a time, which requires greater computations. Here we are reducing this computation time as follows: Initially the feature space is clustered based on decision attributes and then the reduct is found for each cluster.

- (1) Initialize the food source positions
- (2) Each employed bee produces a new food source in her food source site and exploits the better source
- (3) Each onlooker bee selects a source depending on the quality of her solution, produces a new food source in selected source site and exploits the better source
- (4) Determine the source to be abandoned and allocate its employed bee as scout for searching new food sources
- (5) Memorize the best food source found so far
- (6) Repeat steps 2-5 until the stopping criterion is met

Fig. 2: Artificial Bee Colony (ABC) algorithm

```

IQRBEE (C,D)
C, the set of all conditional features;
D, the set of decision features.
(1) Cluster the domain
(2) Find the reduct for each class
(3) Construct the core reduct and reduct sets
(4) Select the initial parameter values for ABC
(5) Initialize the population (xi)
(6) Calculate the objective and fitness value
(7) Find the optimum feature subset as global.
(8) do
    a. Produce new feature subset (vi)
    b. Apply the greedy selection between xi and vi
    c. Calculate the fitness and probability values
    d. Produce the solutions for onlookers
    e. Apply the greedy selection for onlookers
    f. Determine the abandoned solution and scouts
    g. Calculate the cycle best feature subset
    h. Memorize the best optimum feature subset
(9) repeat // for maximum number of cycles
    
```

Fig. 3: Artificial bee colony based reduct algorithm

For example if we have M number of feature rows, N_C number of conditional attributes and N_D number of decision attributes, the feature rows are clustered based on decision attributes at first. For each cluster, the reduct is received as R_i , where $i=1,2,\dots,N_D$. From this set of reducts, the most common attributes are taken out as the core reduct (R_c). Then the ABC algorithm is applied to select the random number of features from each cluster (R_i) to find the optimum feature subset as described in the following text. The core idea has been explained in Fig. 3.

Artificial bee colony based reduct (BeeIQR):

Artificial Bee Colony (ABC) algorithm, for real parameter optimization, is a recently introduced optimization algorithm and simulates the foraging behaviour of bee colony, for unconstrained optimization problems (Karaboga and Basturk, 2008; Srichandum and Rujiranyong, 2010). For solving constrained optimization problems, a constraint handling method was incorporated with the algorithm. In a real bee colony, there are some tasks performed by specialized individuals. These specialized bees try to maximize the nectar amount stored in the hive by performing efficient division of labour and self-organization. The minimal model of swarm-intelligent

forage selection in a honey bee colony, that ABC algorithm adopts, consists of three kinds of bees: Employed bees, onlooker bees and scout bees. Half of the colony comprises employed bees and the other half includes the onlooker bees. Employed bees are responsible from exploiting the nectar sources explored before and giving information to the other waiting bees (onlooker bees) in the hive about the quality of the food source site which they are exploiting. Onlooker bees wait in the hive and decide a food source to exploit depending on the information shared by the employed bees. Scouts randomly search the environment in order to find a new food source depending on an internal motivation or possible external clues or randomly. Main steps of the ABC algorithm simulating these behaviors are given in the Fig. 2.

The above procedure can be implemented for feature reduction. Let the bees select the feature subsets at random and calculate their fitness and find the best one at each iteration. This procedure is repeated for number of iterations to find the optimal subset.

As discussed earlier, after choosing the core reduct (R_c), with the remaining attributes at each R_i , the employed bee produces the feature subset in random. Consider a domain which contains N_D number of unique decision values, then the same number of bees (p) has been chosen as the population size. From this population half of the bees are considered as employed bees and the remaining half are considered as onlooker bees. For each employed bee, a random subset from one reduct set is assigned. The random sets assigned to all the bees are combined to form the feature subset. For example, consider a database that contains 10 numbers of conditional attributes (c_1, c_2, \dots, c_{10}) and 3 numbers of decision attributes with 500 records. Initially the records are clustered into 3 groups based on the decision attribute and then the reduct is applied for each group. For example, if we are obtaining the reduct as:

$$R_1 = \{c_1, c_3, c_4, c_9\}$$

$$R_2 = \{c_3, c_4, c_8\}$$

$$R_3 = \{c_3, c_4, c_6, c_7, c_{10}\}$$

From these reducts, the common attributes are chosen as core reduct; in this example, $R_c = \{c_3, c_4\}$ and the remaining attributes are removed from each reduct:

$$R_1 = \{c_1, c_9\}; \quad R_2 = \{c_8\}; \quad R_3 = \{c_6, c_7, c_{10}\}$$

In the next step, 3 bees are employed to construct a reduct, by selecting random subsets from these reducts and combining them with the core to find the optimum one. For example:

$$\Rightarrow R_c + \text{Bee1} = \{c_1\} + \text{Bee2} = \{c_8\} + \text{Bee3} = \{c_6, c_{10}\}$$

$$\Rightarrow \{c_3, c_4, c_1, c_8, c_6, c_{10}\}$$

This reduct is evaluated using the ABC algorithm. In the second step of the algorithm, for each employed bee, whose total number equals to the half of the number of food sources, a new source is produced by:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

where, ϕ_{ij} is a uniformly distributed real random number within the range $[-1, 1]$, k is the index of the solution chosen randomly from the colony ($k = \text{int}(\text{rand} * N) + 1$), $j = 1, \dots, D$ and D is the dimension of the problem. After producing v_i , this new solution is compared to solution x_i and the employed bee exploits the better source. In the third step of the algorithm, an onlooker bee chooses a food source with the probability and produces a new source in selected food source site. As for employed bee, the better source is decided to be exploited.

The indiscernibility relation is calculated for each feature subset as objective value (f_i). This value has to be maximized. From this objective value, the fitness value is calculated for each bee, as given in the following equation:

$$\text{fit}_i = \begin{cases} 1 / (1 + f_i) & \text{if } f_i \geq 0 \\ 1 + \text{abs}(f_i) & \text{otherwise} \end{cases}$$

The probability is calculated by means of fitness value using the following equation:

$$P_i = \frac{\text{fit}_i}{\sum_{j=1}^N \text{fit}_j}$$

where, fit_i is the fitness of the solution x_i . After all onlookers are distributed to the sources, sources are checked whether they are to be abandoned. If the number of cycles that a source cannot be improved is greater than a predetermined limit, the source is considered to be exhausted. The employed bee associated with the exhausted source becomes a scout and makes a random search in problem domain by the following equation:

$$x_{ij} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) * \text{rand}$$

The pseudocode of our proposed method is given as.

The following parameters we have used in our proposed method:

The population size (number of bees), p	Equal to the number of Classes
The dimension of the population	$p \times N$
Lower bound	1
Upper bound	N
Maximum number of iterations	1000
The number of runs	3

RESULTS AND DISCUSSION

The performance of the reduct approaches discussed in this study has been tested with 5 different medical datasets, downloaded from UCI machine learning data repository. Table 1 shows the details about the datasets used in this study.

Table 2 shows the reducts obtained from our proposed method for each dataset. The underlined attributes in the final reduct are the wavers, that is, at some iterations, they occur in the reduct and at some other iterations, they do not.

Table 3 shows the reduct results of the methods on the 5 different medical datasets. It shows the size of the reduct found for each method. The proposed IQRBee method is compared with general RSAR, Entity based Reduct (EBR), Genetic RSAR, Ant RSAR, Particle Swarm Optimization based RSAR (PSORSAR) and with our previous work (BeeRSAR). The QuickReduct and EBR methods produced the same reduct every time, unlike GenRSAR, AntRSAR, PSORSAR and BeeRSAR found different reducts and sometimes different reduct cardinalities. As it is illustrated in the results, the proposed IQRBee method comes out with very minimal reduct than the others, which shows its superior performance. An improved K-Nearest Neighbor (KNN) algorithm (Ibrahim *et al.*, 2009) named as Genetic KNN (GKNN) classifier is employed to analyze the classification performance (Suguna and Thanushkodi, 2010b).

Table 1 Datasets Used for Reduct

Dataset name	Total number of instances	Total number of features	Total number of decision attributes
Dermatology	366	34	6
Cleveland heart	300	13	4
HIV	500	21	3
Lung Cancer	32	56	3
Wisconsin	699	09	2

Table 2: Attributes in the Reduct from IQRBee

Dataset	Features	IQRBee		
		Core reduct	Final reduct	No. of attributes
Dermatology	34	{ 1 }	{ 1,3,5,8,15,24,33 }	6-7
Cleveland Heart	13	{ 1 }	{ 1,5,7,8,9 }	5
HIV	21	{ 1,3,9,12,15 }	{ 1,3,5,7,9,12,15 }	6-7
Lung Cancer	56	{ 1,4 }	{ 1,4,8,14 }	4
Wisconsin	09	{ 1,8 }	{ 1,4,6,8 }	4

Table 3: Reducts found for the datasets

Dataset	Features	RSAR	EBR	Ant RSAR	Gen RSAR	PSO-RSAR	Bee RSAR	IQRBee
Dermatology	34	10	10	8-9	10-11	7-8	7	6-7
Cleveland heart	13	7	7	6-7	6-7	6-7	6	5
HIV	21	13	13	10-11	11-13	9-10	8	6-7
Lung cancer	56	4	4	4	6-7	4	4	4
Wisconsin	09	5	5	5	5	4-5	4	4

Table 4: Classification performance of reducts

Dataset	Dermatology	Cleveland heart	HIV	Lung cancer	Wisconsin
IQRBee	92.36 ± 0.22	86.54 ± 0.36	86.29 ± 0.18	83.03 ± 0.18	88.70 ± 0.35
BeeRSAR	91.70 ± 0.74	84.70 ± 0.74	85.70 ± 0.74	82.37 ± 0.39	84.70 ± 0.74
PSORSAR	88.89 ± 0.52	84.89 ± 0.52	84.89 ± 0.52	79.63 ± 0.31	84.89 ± 0.52
AntRSAR	85.32 ± 0.34	86.32 ± 0.34	85.32 ± 0.34	79.53 ± 0.37	85.32 ± 0.34
GenRSAR	86.39 ± 0.42	85.39 ± 0.42	84.39 ± 0.42	78.89 ± 0.71	86.39 ± 0.42
EBR	78.89 ± 0.21	79.71 ± 0.17	77.76 ± 0.79	77.63 ± 0.28	81.12 ± 0.18
RSAR	76.03 ± 0.27	77.07 ± 0.31	75.07 ± 0.54	77.95 ± 0.14	78.60 ± 0.26

Table 4 shows the comparison of the classification accuracy of our proposed approach with the existing methods. It is clearly shown that the reducts from IQRBee reaches greater accuracy than the other methods.

CONCLUSION

Rough set theory provides a recognized context for dimensionality reduction in data mining. In this study, a novel approach of Rough Set-based Attribute Reduction (RSAR) is proposed for feature selection to obtain a more accurate reduct. Initially, the instances are grouped based on the class attribute and then the reduct is found for each group. The intersection operation is performed to select the common attributes from all these reducts to generate the core reduct. With the remaining attributes, the Artificial Bee Colony (ABC) algorithm based RSAR model is applied to obtain the final reduct. Experiments are carried out on five different datasets from the UCI machine learning repository. The performance of the reduct is analyzed with Genetic k-Nearest Neighbor (GKNN) classifier and compared with six different algorithms. The results show that our proposed method outperforms the other existing methods.

REFERENCES

Ahmed, E.F., W.J. Yang and M.Y. Abdullah, 2009. Novel method of the combination of forecasts based on rough sets. *J. Comput. Sci.*, 5: 440-444. DOI: 10.3844/jcssp.2009.440.444

Ibrahim, N.K., R.S.A.R. Abdullah and M.I. Saripan, 2009. Artificial neural network approach in radar target classification. *J. Comput. Sci.*, 5: 23-32. DOI: 10.3844/jcssp.2009.23.32

Jensen, R. and Q. Shen, 2007. Fuzzy-rough sets assisted attribute selection., *IEEE Trans. Fuzzy Syst.*, 15: 73-89. DOI: 10.1109/TFUZZ.2006.889761

Karaboga, D. and B. Basturk, 2008. On the performance of Artificial Bee Colony (ABC) algorithm. *Applied Soft Comput.*, 8: 687-697. DOI: 10.1016/j.asoc.2007.05.007

Ngo, C. and H. Nguyen, 2009. A tolerance rough set based semantic clustering method for web search results. *Lect. Notes Comput. Sci.*, 8: 453-464. DOI: 10.3923/itj.2009.453.464

Salamat, A., C. Phetchanchai, T. Saba and A. Rehman, 2010. Index financial time series based on zigzag-perceptually important points. *J. Comput. Sci.*, 6: 1389-1395. DOI: 10.3844/jcssp.2010.1389.1395

Shylaja, B.S., V.M.A. Rajam, V.U. Maheswari and A. Siromoney, 2010. Performance study of threshold variations in temporal decision systems for routing in vehicular ad hoc networks. *J. Comput. Sci.*, 6: 1479-1484. DOI: 10.3844/jcssp.2010.1473.1478

Srichandum, S. and T. Rujiranyong, 2010. Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. *Am. J. Eng. Applied Sci.*, 3: 7-14. DOI: 10.3844/ajeassp.2010.7.14

Suguna, N. and K. Thanushkodi, 2010a. A novel rough set reduct algorithm for medical domain based on bee colony optimization. *J. Comput.*, 2: 49-54.

Suguna, N. and K. Thanushkodi, 2010b. An improved k-nearest neighbour classification using genetic algorithm. *Int. J. Comput. Sci. Issues*, 7: 32-35.