

Attribute Reduction using Forward Selection and Relative Reduct Algorithm

P.Kalyani

Associate Professor in Computer Science,
SNR Sons College,
Coimbatore, India.

Dr. M.Karnan

Professor and Head, Department of CS
Tamilnadu Engineering College,
Coimbatore, India.

ABSTRACT

Attribute reduction of an information system is a key problem in rough set theory and its applications. Rough set theory has been one of the most successful methods used for feature selection. Rough set is one of the most useful data mining techniques. This paper proposes relative reduct to solve the attribute reduction problem in rough set theory. It is the most promising technique in the Rough set theory, a new mathematical approach to reduct car dataset using relative reduct algorithm. The redundant attributes are eliminated in order to generate the effective reduct set (i.e., reduced set of necessary attributes) or to construct the core of the attribute set. The technique was originally proposed to avoid the calculation of discernibility functions or positive regions, which can be computationally expensive without optimizations. This paper analyses the efficiency of the proposed backward relative reduct algorithm against forward selection algorithm. The experiments are carried out on car data base of UCI machine learning repository.

Keywords: Rough set theory, Data mining, Knowledge discovery, Feature selection, Forward selection, backward elimination

1. INTRODUCTION

The main aim of feature selection (FS) is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In real world problems FS is a must due to irrelevant or misleading features. For instance, by removing these factors, learning from data techniques. Given a feature set size n , the task of FS can be seen as a search for an optimal feature subset through the competing 2^n candidate subsets. Rough set theory introduced by Pawlak [1][2] during the early 1980s spans a quarter century. The rough set approach to approximation of sets leads to useful forms of granular computing that are part of computational intelligence. The definition of what an optimal subset is may vary depending on the problem to be solved. Although an exhaustive method may be used for this purpose, this is quite impractical for most datasets. Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity.

However, the degree of optimality of the final feature subset is often reduced. Many problems in machine learning involve high dimensional descriptions of input features. 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

or require additional information about the given data set for threshold. A technique that can reduce dimensionality using information contained within the data set and that preserves the meaning of the features is clearly desirable. Rough set theory (RST) can be used as such a tool to discover data dependencies and to reduce the number of attributes contained in a data set using the data alone and no additional information.

Given a data set with discretized attribute values, it is possible to find a subset (termed reduct) of the original attributes using RST that are the most informative; all other attributes can be removed from the data set with minimal information loss. The main advantage of the RS is that although one can put additional background or additional preliminary information into the computation, it is not required. Recently, RS has been applied to many fields such as data mining [3][4], decision support systems [5][6], knowledge acquisition [7] etc. From the dimensionality reduction perspective, informative features are those that are most useful in determining classifications from their values.

However, it is most often the case that the values of attributes may be both crisp and real-valued, and this is where traditional rough set theory encounters a problem. It is not possible in the original theory to say whether two attribute values are similar and to what extent they are the same; for example, two close values may only differ as a result of noise, but in RST, they are considered to be as different as two values of different order of magnitude.

It is, therefore, desirable to develop techniques to provide the means of data reduction for crisp and real value attributed data sets which utilizes the extent to which values are similar.

2. DIMENSIONALITY REDUCTION

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task, or redundant. For example, if the task is to classify customers as to whether or not they are likely to purchase a popular new CD at All Electronics when notified of a sale, attributes such as age or music taste. Although it may be possible for a domain expert to pick out some of the useful attributes, this can be a difficult and time-consuming task, especially when the behavior of the data is not well known. Leaving out relevant attributes or keeping irrelevant attributes may be detrimental, causing confusion for mining algorithm employed.

Dimensionality reduction reduces the data size by removing such attributes from it. Typically, methods of attribute subset selection are applied. The goals of attribute subset selection are applied. The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability

distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Miming on a reduced set of attributes has an additional benefit. It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

An exhaustive search for the optimal subset of attributes can be prohibitively expensive, especially as d and the number of data classes increase. Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection. These methods are typically greedy in that, while searching through attribute space, they always make what looks to be the best choice at the time. Their strategy is to make a locally optimal choice in the hope that this will lead to a globally optimal solution. Such greedy methods are effective in practice and may come close to estimating an optimal solution.

Basic heuristic methods [8] of attribute subset selection include the following techniques.

1. Stepwise forward selection: The procedure starts with an empty set of attributes. The best of the original attributes is determined and added to the set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.
2. Step wise backward elimination: The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.
3. Combination of forward selection and backward elimination: The stepwise forward selection and backward elimination methods can be combined so that at each step, the procedure selects the best attribute and removes the worst from remaining attributes

3. FORWARD SELECTION

The forward selection algorithm, attempts to calculate a minimal reduct without exhaustively generating all possible subsets. The problem of finding a minimal reduct of an information system has been the subject of much research [9]. 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.

Central to RSAR is the concept of indiscernibility. Let $I=(U,A)$ be an information system, where U is a non-empty set of finite objects (the universe) and A is a non-empty finite set of attributes such that $a : U \leftarrow Va$ for every $a \in A$. With any P there is an associated equivalence relation $IND(P)$:

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 $\square R (D) = \square C(D)$.

The forward selection algorithm implies that, for a dimensionality of n , $(n2+n) = 2$ evaluations of the dependency function may be performed for the worst-case dataset.

According to the forward selection 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).

Example: Car Data set
Conditional attribute values:

- Buying (B) – v-high, high, med, low to assign 4,3,2,1
- Maint (M) – v-high, high, med, low to assign 4,3,2,1
- Doors (D) – 2,3,4,5 more
- Persons – 2, 4, more
- Lug boot - small, med, big to assign 1, 2, 3
- Safety – low, med, high to assign 1, 2, 3

Decision attributes values:

- Unacc - 0
- Acc - 1
- Good - 2
- V-good - 3

Table 1: Car Data set

Object	B	M	D	P	L	S	C
1	4	4	2	2	1	1	1
2	4	4	2	2	1	2	1
3	4	4	2	2	1	3	1
4	4	4	2	2	2	1	1
5	4	4	2	2	2	2	1
6	1	1	5	5	3	3	4
7	1	1	5	5	3	2	3
8	1	1	5	5	3	1	1
9	1	1	5	5	2	3	4
10	1	1	5	5	2	2	3
11	1	1	5	5	2	1	1
12	1	1	5	5	1	3	3
13	1	1	5	5	1	2	2
14	1	1	5	5	1	1	1
15	1	1	5	4	3	3	4
16	1	1	5	4	3	2	3
17	1	1	5	4	3	1	1
18	1	1	5	4	2	3	4
19	1	1	5	4	2	2	3
20	1	1	5	4	2	1	1

To illustrate the operation of these, an example data set will be used. Here, the table consists of six conditional features (b, m, d, p, l, and s), one decision feature (class) and twenty objects. The task of feature selection here is to choose the smallest subset of these conditional features so that the resulting reduced data set remains consistent with respect to the decision feature. A data set is consistent if for every set of objects whose attribute values are the same, the corresponding decision attributes are identical.
 $Ind(B) = \{1,2,3,4,5\}, \{6,7,8 \dots\dots 20\}$

$$\text{Pos}(B)/(C) = \text{Ind}(B) \subseteq \text{Ind}(C) \\ = \{1, 2, 3, 4, \text{ and } 5\}$$

Therefore, $\gamma(B)/(C) = 5/20$ similarly, other columns are calculated as follows

$$\gamma(M)/(C) = 5/20, \quad \gamma(D)/(C) = 5/20, \\ \gamma(P)/(C) = 5/20, \quad \gamma(L)/(C) = 5/20 \\ \gamma(S)/(C) = 7/20.$$

The maximum degree of dependency value is taken. In this case safety is the highest dependency.

Therefore $R \leftarrow \{\text{safety}\}$ and $\gamma_{\text{best}} = 7/20$ until the dependency degree is equal to 1.

According to the FORWARD selection algorithm, the dependency of each attribute calculated, and the best candidate is chosen. As attribute S generates the highest dependency degree, then attribute is chosen and the sets $\{S,B\}, \{S,M\}, \{S,D\}, \{S,P\}, \{S,L\}$ are evaluated.

Take the next combination and find out the degree of dependency as follows.

$$\gamma(S, B)/(C) = 10/20, \quad \gamma(S, M)/(C) = 10/20, \\ \gamma(S, D)/(C) = 10/20, \quad \gamma(S, P)/(C) = 14/20 \\ \gamma(S, L)/(C) = 13/20$$

Take the maximum degree of dependency value. In this iteration (Safety, Persons) is the highest dependency degree but the value is not equal to 1. So go to the next iteration. As attribute $\{S,P\}$ generates the highest dependency degree, then attribute is chosen and the sets $\{S,P,B\}, \{S,P,M\}, \{S,P,D\}, \{S,P,L\}$ are evaluated.

Take three attribute combinations and find out the degree of dependency as follows

$$\gamma(S, P, B)/(C) = 14/20, \\ \gamma(S, P, M)/(C) = 14/20, \\ \gamma(S, P, D)/(C) = 14/20, \\ \gamma(S, P, L)/(C) = 20/20.$$

Take the maximum degree of dependency value. In this iteration (Safety, Persons, Lug_boot) is the highest dependency degree but the value is equal to 1. The final reduct set for the above table is (Safety, Persons, Lug_Boot). In fig.1, the final reduct stage is illustrated using the example data set.

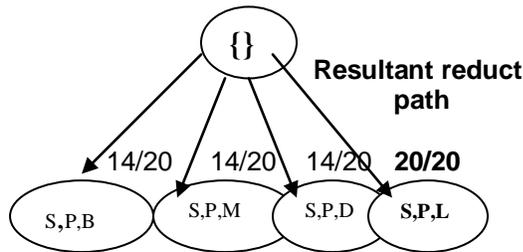


Fig.1. Final reduct branches of the search space

Table 2: An Example of Reducts:

(i).Data sets:

Object	B	M	D	P	L	S	C
1	4	4	2	2	1	1	1
2	4	4	2	2	1	2	1
3	4	4	2	2	1	3	1
4	4	4	2	2	2	1	1
5	4	4	2	2	2	2	1
6	1	1	5	5	3	3	4
7	1	1	5	5	3	2	3
8	1	1	5	5	3	1	1
9	1	1	5	5	2	3	4
10	1	1	5	5	2	2	3
11	1	1	5	5	2	1	1
12	1	1	5	5	1	3	3
13	1	1	5	5	1	2	2
14	1	1	5	5	1	1	1
15	1	1	5	4	3	3	4
16	1	1	5	4	3	2	3
17	1	1	5	4	3	1	1
18	1	1	5	4	2	3	4
19	1	1	5	4	2	2	3
20	1	1	5	4	2	1	1

(ii) A Reduced Dataset

Object	S	P	L	C
1	1	2	1	1
2	2	2	1	1
3	3	2	1	1
4	1	2	2	1
5	2	2	2	1
6	3	5	3	4
7	2	5	3	3
8	1	5	3	1
9	3	5	2	4
10	2	5	2	3
11	1	5	2	1
12	3	5	1	3
13	2	5	1	2
14	1	5	1	1
15	3	4	3	4
16	2	4	3	3
17	1	4	3	1
18	3	4	2	4
19	2	4	2	3
20	1	4	2	1

4. RELATIVE REDUCT

In a feature selection method based on an alternative dependency measure is presented. Jensen, Q. Shen [10] have developed the relative reduct algorithm. In, a feature selection method based on an alternative dependency measure is presented. The technique was originally proposed to avoid the calculation of discernibility functions or positive regions, which can be computationally expensive without optimizations.

The authors replace the traditional rough set degree of dependency with an alternative measure, the relative dependency, defined as follows for an attribute subset R: Pseudo code of the Reduct algorithm is given below.

Equation:

$$K_R(D) = \frac{|U|IND(R)|}{|U|IND(RUD)|}$$

Backward elimination of features where attributes are removed from the set of considered attributes if the relative dependency equals 1 upon their removal. Attributes are considered one at a time, starting with the first, evaluating their relative dependency.

R to the set of conditional attributes {B, M, D, P, L, and S}. Next, the elimination of attribute B is

$$K(M, D, P, L, S)(C) = 1$$

The relative dependency is equal to 1. Attribute B can be removed from the current reduct candidate

$R \leftarrow \{M, D, P, L, S\}$. The algorithm then considers the elimination of attribute M from R.

$$K(D, P, L, S)(C) = 1$$

The relative dependency is equal to 1. Attribute M is removed from R. $R \leftarrow \{D, P, L, S\}$. The next step evaluates the removal of D from the reduct candidate.

$$K(P, L, S)(C) = 1$$

The relative dependency is equal to 1. Attribute D is removed from R. $R \leftarrow \{P, L, S\}$. The next step evaluates the removal of P from the reduct candidate.

$$K(L, S)(C) = 20/13$$

At this does not equal to 1, attribute P is not removed from R. The algorithm then evaluates the elimination of attribute L from R. $R \leftarrow \{P, S\}$.

$$K(P, S)(C) = 20/11$$

The relative dependency does not equal to 1, attribute L is not removed from R. The algorithm then evaluates the elimination of attribute L from R.

$R \leftarrow \{P, L\}$.

$$K(P, L)(C) = 20/5$$

At this does not equal to 1. Attribute S is not removed from S. The algorithm terminates and the output reducts $R \leftarrow \{P, L, and S\}$.

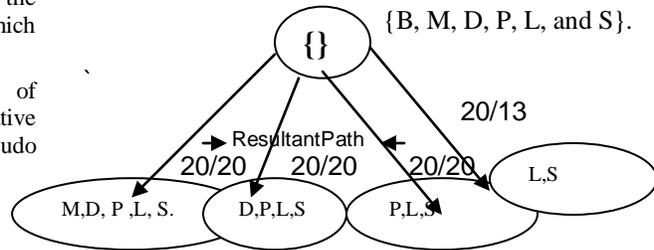


Fig. 2 Relative reducts branches of the search space

The relative reduct branch explain first try to remove the attribute B. The dependency degree of {M, D, P, L, S} =1. So B is removed. Next try to remove the attribute M. The dependency degree of {D, P, L, S} =1. So M is removed.

Next try to remove the attribute D. The dependency degree of {P, L, S} =1. so D is removed. The remaining attributes dependency degree is not equal to 1. So the attributes P, L, S is not removed from the set.

A Reduced Dataset:

Object	S	P	L	C
1	1	2	1	1
2	2	2	1	1
3	3	2	1	1
4	1	2	2	1
5	2	2	2	1
6	3	5	3	4
7	2	5	3	3
8	1	5	3	1
9	3	5	2	4
10	2	5	2	3
11	1	5	2	1
12	3	5	1	3
13	2	5	1	2
14	1	5	1	1
15	3	4	3	4
16	2	4	3	3
17	1	4	3	1
18	3	4	2	4
19	2	4	2	3
20	1	4	2	1

5. COMPUTATIONAL RESULTS

Table 3: Comparison Results

Car data set	Instances	No. of Attributes	Forward selection		Relative reduct	
			No. of reduct	Times in (seconds)	No. of reduct	Times in (seconds)
Cartest-20	20	6	3	56.230000	3	33.087000
Car dataset 1-250	250	6	4	65.544000	5	72.454000
Car dataset 1-500	500	6	5	129.737000	5	99.713000

6. CONCLUSION

In this paper, we presented an algorithm for attribute reduction. It is based on the rough set theory using backward elimination. Illustration of the algorithm processing and experiment results indicate that the algorithm proposed by this paper is effective and efficient, especially for large data sets. The technique was originally proposed to avoid the calculation of discernibility functions or positive regions, which can be computationally expensive.

REFERENCES

- [1] Z. Pawlak (1982) Rough sets. *International Journal of Computer and Information Sciences*, vol.11, pp. 341–356.
- [2] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishing, Dordrecht, 1991.
- [3] P. J. Lingras, “Data mining using extensions of the rough set model,” *J.Amer. Soc. Inf. Sci.*, vol. 49, pp. 415–422, May 1998.
- [4] N.O. Attoh-Okine, “Rough set application to data mining principles in pavement management database,” *J. Comput. Civil Eng.*, vol. 11, pp.231–237, Apr. 1997.
- [5] Z. Pawlak, “Rough set approach to knowledge-based decision support,” *Eur. J. Oper. Res.*, vol. 99, pp. 48–75, Jan. 1997.
- [6] Liu Qin. *Rough Set and Rough Reasoning*. Beijing: Science Press, 2001
- [7] L. K. Terje, “Rough modeling—Extracting compact models from large databases,” *Mastersthesis, Knowledge Systems Group*, Norwegian Univ. Sci. Technol., Trondheim, Norway, 1999.
- [8] Chouchoulas, J. Halliwell and Q. Shen. On the Implementation of Rough Set Attribute Reduction. *Proceedings of the 2002 UK Workshop on Computational Intelligence*, pp. 18-23. 2002.
- [9] J.J. Alpigini, J.F. Peters, J. Skowronek, N. Zhong (Eds.): *Rough Sets and Current Trends in Computing*, Third International Conference, RSCTC 2002, Malvern, PA, USA, October 14-16, 2002, *Proceedings. Lecture Notes in Computer Science 2475 Springer 2002*, ISBN 3-540-44274-X.
- [10] R. Jensen, Q. Shen (2004) Semantics-preserving dimensionality reduction: Rough and fuzzy rough based approaches. *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp.1457–1471.
- [11] Blake, C.L. and C.J. Merz, 1998. *UCI Repository of machine learning databases*. Irvine, University of California, <http://www.ics.uci.edu/~mlnar/>.