A method for finding minimal sets of features adequately describing discrete information objects

D. Sitnikov[1], O. Titova[1], O. Romanenko[1] & O. Ryabov[2]
[1]*Kharkov State Academy of Culture, Ukraine*
[2]*National Institute of Advanced Industrial Science and Technology, Japan*

Abstract

One of the classical Data Mining problems is the problem of classifying new objects on the basis of available information when the information associated with these objects does not allow identifying them unambiguously as elements of some set. In such cases using rough sets theory is often an effective solution. This theory operates with such concepts as "indiscernible" elements and relations. A rough set is characterized by lower and upper approximations for finding which the authors earlier suggested an original algebraic method. The given method uses only logic operations, which makes the process of searching logic rules very quick and efficient.

The upper and lower approximations of a rough set allow describing elements of this set as completely as it is possible from the viewpoint of available information. In this connection it seems interesting and important to find irreducible sets of features describing a rough set with the same "precision" as with the help of a full set of features (so called reducts). This problem is quite difficult and complicated and at present it does not have good solutions. Our paper continues research carried out by the authors earlier and we suggest a method for finding reducts based on eliminating non-salient features in the reverse order of their importance. The suggested procedure allows us to avoid exhaustive searching by extracting a predefined number of most significant reducts. In this paper we consider arbitrary features taking on their values from finite sets.

Keywords: rough set, low approximation, upper approximation, boundary region, reduct.

WIT Transactions on Information and Communication Technologies, Vol 42, © 2009 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA090141

1 Introduction

Modern Data Mining methods allow discovering non-trivial dependencies in large information arrays. Building classification and association models, regression analysis, discovering clustering and aggregation dependencies are typical problems in the field of Data Mining and Knowledge Discovery. At present there are many theoretical instruments facilitating the process of mining data. Rough sets theory suggested by Pawlak [1] is an effective mathematical tool that allows solving the problem of classifying new objects on the basis of available information. This theory takes into account the fact that information associated with the objects being classified does not allows often determining unambiguously to which class an object belongs.

A rough set can be characterized by its upper and lower approximations. The lower approximation contains elements about which it can be said that they *definitely* belong to the given set. The upper approximation contains elements that *may* belong to the set. The difference between the upper and lower approximations is defined as a boundary region the size of which can be considered as a measure of "roughness" for the set under consideration.

We earlier suggested an original algebraic approach for determining the upper and lower approximations for a given set [2]. This approach allows using only logic operations for calculating approximations, which substantially speeds up the process of generating logic rules reflecting dependencies in discrete data. Since Data Mining methods and technologies are designated for processing huge information volumes the suggested approach has substantial computational advantages as compared with the methods using other types of operations. We have extended our algebraic approach in [3] where not only binary features but also any discrete features have been considered.

The concept of reducts [5] is one of the basic concepts in rough sets theory. A reduct is defined as a minimal subset of attributes that describe a rough set without any information losses (in other words they describe the set with the same degree of precision as the full set of features). In the opinion of Z. Pawlak [5] finding reducts is an interesting and complicated problem. It must be said that an arbitrary rough set may have several reducts. In the case of a few reducts for a given set it is interesting to determine how important the features forming the reduct are. We understand the importance of features as a measure of their "classification strength" and suppose that most important features form most important reducts. Therefore we can compare reducts in some way according to their importance.

We use the fact that the suggested algebraic approach allows us to quickly generate approximations to efficiently calculate the importance of a feature [4]. Our method is based on the calculation of the changes in the boundary region after excluding a feature. The larger the boundary region becomes the more important the eliminated feature is.

In this paper we continue research started in our previous papers and suggest a method for determining the importance of non-binary features and the order in which we should eliminate non-salient features to find most significant reducts.

2 Calculating approximations for non-binary features

Our method of calculating approximations for arbitrary discrete information features has been described in [4]. The main idea of the method is as follows. Suppose we have a non-empty finite set of objects $U=\{a_1, a_2, \ldots, a_n\}$, called *universe*. Discrete functions $P_1(t)$, $P_2(t)$, …, $P_k(t)$ defined on the universe are called *coordinates*. The functions P_1, P_2, …, P_k can be interpreted as characteristic properties of the objects from U.

Following the basic principles of rough sets theory we should describe a given set $X \subseteq U$ in terms of the coordinates. Since there exists a one-to-one correspondence between the subsets of U and all the binary predicates defined on U we consider a predicate X(t) instead of set X, where X(t) equals 1 if and only if $t \in X$. Thus we should describe the finite predicate X(t) in terms of the coordinates $P_1, P_2, \ldots, P_k$ (table 1).

Table 1.

	a_1	a_2	…	a_n
P_1	δ_{11}	δ_{12}	…	δ_{1n}
P_2	δ_{21}	δ_{22}	…	δ_{2n}
…	…	…	…	…
P_k	δ_{k1}	δ_{k2}	…	δ_{kn}
X	λ_1	λ_2	…	λ_n

where $\delta_{1j} \in \{0,1..m_1\}$, $\delta_{2j} \in \{0,1..m_2\}$,...$\delta_{kj} \in \{0,1..m_k\}$, $\lambda_j \in \{0,1\}$, if $\delta_{ij}=w$ then $P_i(a_j)=w$, if $\lambda_j = 1$ then $X(a_j) = 1$, if $\lambda_j = 0$ then $X(a_j) = 0$.

In the general case the approximations for X can be calculated in accordance with the following formulae [3]:

$$I_* = (\lambda_1 \,\&\, P_1^{\delta_{11}} \,\&\, P_2^{\delta_{21}} \,\&\, \ldots \,\&\, P_k^{\delta_{k1}}) \vee (\lambda_2 \,\&\, P_1^{\delta_{12}} \,\&\, P_2^{\delta_{22}} \,\&\, \ldots \,\&\, P_k^{\delta_{k2}}) \vee \ldots \vee (\lambda_n \,\&\, P_1^{\delta_{1n}} \,\&\, P_2^{\delta_{2n}} \,\&\, \ldots \,\&\, P_k^{\delta_{nn}}) \quad (1)$$

$$I_* = (\lambda_1 \vee \neg P_1^{\delta_{11}} \vee \neg P_2^{\delta_{21}} \vee \ldots \vee \neg P_k^{\delta_{k1}}) \,\&\, (\lambda_2 \vee \neg P_1^{\delta_{12}} \vee \neg P_2^{\delta_{22}} \vee \ldots \vee \neg P_k^{\delta_{k2}}) \,\&\, \ldots \,\&\, (\lambda_n \vee \neg P_1^{\delta_{1n}} \vee \neg P_2^{\delta_{2n}} \vee \ldots \vee \neg P_k^{\delta_{nn}}) \quad (2)$$

where if $P_k(a_i) = \delta_{ij}$ then $P_k^{\delta_{ij}} = 1$ else $P_k^{\delta_{ij}} = 0$, if $P_k(a_i) = \delta_{ij}$ then $\neg P_k^{\delta_{ij}} = 0$ else $\neg P_k^{\alpha_n} = 1$ for any P.

Let us consider an example of calculating approximations (table 2). Suppose that characteristic functions P_1, P_2, P_3 describing properties of objects a_1, ...,a_5 can take on their values from the set {0,1,2}.

Table 2.

	a_1	a_2	A_3	a_4	a_5
P_1	1	0	2	0	0
P_2	0	2	0	0	2
P_3	0	2	1	1	2
X	0	1	0	1	0

Using formulae (1) and (2) we obtain the following logic expressions for approximations:

$$I^* = (P_1^0 \,\&\, P_2^2 \,\&\, P_3^2) \vee (P_1^0 \,\&\, P_2^0 \,\&\, P_3^1) \tag{3}$$

$$I_* = (\neg P_1^1 \vee \neg P_2^0 \vee \neg P_3^0) \,\&\, (\neg P_1^2 \vee \neg P_2^0 \vee \neg P_3^1) \,\&\, (\neg P_1^0 \vee \neg P_2^2 \vee \neg P_3^2) \tag{4}$$

The resulting approximations are represented in table 3:

Table 3.

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	2	0	0
P_2	0	2	0	0	2
P_3	0	2	1	1	2
X	0	1	0	1	0
I^*	0	1	0	1	1
I_*	0	0	0	1	0

3 Importance of attributes

We define the *importance* of a feature P_i as a percentage showing how the "roughness" of the set X increases after this feature is eliminated:

$$V(P_i) = \frac{\Delta(BN_I)}{M(X)} * 100\% , \tag{5}$$

where ($\Delta(BN_I)$) represents the change in the boundary region when the feature Pi is excluded; M(X) is the number of elements in the set X. It should be noted that in the same way we can define the importance of a feature set, for example, $V(P_1, P_3, P_7)$. For this example the change in the boundary region is calculated after the features P_1, P_3, P_7 have been eliminated *simultaneously*.

If $V(P_i) \geq$ minDeterioration the feature P_i is salient, if $V(P_i) <$ minDeterioration the feature P_i is non-salient. It should be noted that although several features taken separately can be non-salient, their combination can be salient. minDeterioration is a preset threshold value that depends on concrete data and problems being solved. This threshold can be defined by an analyst in accordance with his/her objectives.

Consider an example of such calculations. For table 3 the boundary region contains 2 elements (the upper approximation contains 3 objects and the lower approximation contains 1 object; 3-1=2). Let us exclude the feature P_1 and calculate new approximations (table. 4):

Table 4.

	a_1	a_2	a_3	a_4	a_5
P_2	0	2	0	0	2
P_3	0	2	1	1	2
X	0	1	0	1	0
I^*	0	1	1	1	1
I_*	0	0	0	0	0

Now the boundary region contains 4 elements (the upper approximation contains 4 elements, the lower approximation is empty). Thus the importance of the feature P_1 is

$$V(P_1) = \frac{(4-2)}{2} * 100\% = 100\% .$$

Let us now calculate the importance of the feature P_2, for which we delete the corresponding row from table 3 to obtain table 5:

Table 5.

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	2	0	0
P_3	0	2	1	1	2
X	0	1	0	1	0
I^*	0	1	0	1	1
I_*	0	0	0	1	0

The boundary region now contains 2 elements and the importance of the feature P_2 is

$$V(P_2) = \frac{(2-2)}{2} * 100\% = 0\% .$$

Let us finally make similar calculations for the feature P_3 (table 6):

The importance of the feature P_3 is

$$V(P_3) = \frac{(2-2)}{2} * 100\% = 0\% .$$

Thus on the basis of the above calculations we can state that the feature P_1 is salient and the other features P_2 and P_3 are non-salient since after any of them have been eliminated the quality of data description will not deteriorate. It should be noted that we have considered a highly simplified example where some features are non-salient for any value of minDeterioration. In other cases we can obtain features that are salient for particular values of this threshold and non-salient for different values.

Table 6.

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	2	0	0
P_2	0	2	0	0	2
X	0	1	0	1	0
I^*	0	1	0	1	1
I_*	0	0	0	1	0

4 A method for searching reducts

Information on the importance of discrete features allows eliminating some attributes and some subsets of attributes from the original set of features. An interesting problem is to find irreducible sets of features adequately describing a given set, so called *reducts* In the general case the set of features can have many different reducts. Different criteria for "best" reducts can be suggested but obviously the task of obtaining minimal sets of most important features seems interesting. Since we can now numerically measure the importance of a feature with the help of rough approximations, a simple approach to searching reducts can be suggested.

At the first stage we should calculate the importance of each feature and select salient and non-salient features by comparing them to minDeterioration. Then we can try either to find all possible reducts or select only a given number of "best" reducts.

In the first case we should consider eliminating all possible subsets from the set of non-salient attributes. Every time the elimination procedure stops as soon as the resulting table gives us the importance exceeding minDeterioration. Let a list of attributes $P_1, P_2, ..., P_k, P_{k+1}, ...P_n$ be sorted in the order of increasing their importance and suppose that the importance of the first k attributes is less than

minDeterioration. Then the number of subsets to be tested is $\sum_{m=1}^{k} C_n^m$ i.e. we encounter the necessity of exhaustive searching. Nevertheless if j (j<k) attributes have been eliminated and the change in the boundary region is greater than the threshold allowed ($\Delta(BN_I)$ >minDeterioration) this branch of searching should not be followed at other iterations, which speeds up the procedure to a certain degree.

The algorithm stops when no attributes can be excluded. In the above example we can eliminate separately P_2 and P_3. The attempt of excluding these attributes simultaneously leads to table 7.

Table 7.

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	2	0	0
X	0	1	0	1	0
I^*	0	0	0	0	0
I_*	0	0	0	0	0

From this table it can be seen that the importance of the pair (P_2, P_3) is

$$V(P_2, P_3) = \frac{(2-0)}{2} * 100\% = 100\%,$$

from which it follows that these attributes cannot be eliminated together. Thus for our example we obtain only two possible reducts: (P_1, P_2) and (P_1, P_3).

In the second case that seems more interesting and practically important (when it is sufficient to find a given number of most significant reducts) the search procedure described above can be improved. The importance of each attribute is recalculated after every elimination step, i.e. the importance of any attribute is calculated for the table obtained after excluding a feature. Also at each step the attribute with the minimal (recalculated) importance is eliminated. If two attributes have the same value of importance the algorithm randomly selects an attribute to be excluded. When no attributes are left for elimination the rest of attributes form a reduct. The algorithm stops when a requested number of reducts have been found. If this number is greater than the number of non-salient attributes from the original table, when all of them have been used the search procedure starts again from the least important attribute in the original table. After the attribute has been eliminated the algorithm does not consider the element with the least importance (this branch has already been used) but selects the attribute for which the recalculated importance is next to the minimal one etc.

The general idea of the suggested method is simple and clear. At each step we eliminate the least important attribute. When it is not possible we take the attribute next to it in the order of increasing importance. Using this method we

avoid exhaustive searching since the number of search branches is equal to the requested number of reducts.

5 Conclusions and discussion

The suggested method for determining the importance of a discrete information feature allows us to significantly reduce the number of features used for describing a rough set, which leads to more concise and understandable logic rules obtained on the basis of approximations. The developed procedure of discovering a given number of most significant reducts avoids exhaustive searching since the number of search branches is equal to the requested number of reducts. If a user requests not only a given number of reducts but also restricts the number of features in each reduct the suggested procedure will exclude all found reducts the number of features in which is greater than the requested one.

Of course a natural question can be posed. Why the resulting reducts obtained with the help of the suggested method are most significant? May be other reducts containing different features can be more useful for some problems? Of course it may happen, but we have suggested a method based on eliminating at each step features whose classification strength is minimal in comparison with the other ones. If such features non-salient from the viewpoint of the suggested procedure seem salient from other viewpoints necessary add-ons to the suggested procedure can be developed, which will allow us to avoid eliminating some interesting features.

References

[1] Pawlak Z.; (1995) *Rough set approach to knowledge-based decision support* / Proc. of the 14 European Conference on Operational Research Jerusalem, Israel.

[2] D. Sitnikov, O. Ryabov.; (2004), *An algebraic approach to defining rough set approximations and generating logic rules,* in Zanasi, A.; Ebecken, N.; Brebbia, C.; (eds), Data Mining V, Malaga, Spain, pp. 179-188.

[3] D. Sitnikov, O. Ryabov; O. Titova, O. Romanenko. (2007), *A generalized algebraic approach to finding rough set approximations and generating logic rules,* in Zanasi, A.; Ebecken, N.; Brebbia, C.; (eds), Data Mining VIII, pp. 3-12

[4] D. Sitnikov; O. Titova, O. Romanenko, O. Ryabov. (2008), *An approach to finding reduced sets of information features describing discrete objects based on rough sets theory* in Zanasi, A.; Ebecken, N.; Brebbia, C.; (eds), Data Mining IX, pp. 3-11

[5] Pawlak Z.; (1997) *Rough set approach to knowledge-based decision support*. European Journal of Operational Research, 99, pp. 420-432.