A Logic Based Feature Selection Method for Improving the Accuracy of Data Mining Classification Algorithms

MEHMET HACIBEYOGLU[1], AHMET ARSLAN[1], SIRZAT KAHRAMANLI[2]
Department of computer engineering
Selcuk University[1], Mevlana University[2]
Selcuklu/Konya
TURKEY
hacibeyoglu@selcuk.edu.tr, ahmetarslan@selcuk.edu.tr, sirzat@selcuk.edu.tr

Abstract: - Real life data sets often contain noisy data which makes the subsequent data mining process difficult. The feature selection preprocessing step can be simplified the datasets by eliminating the features that are redundant for classification process, with pertinent features would reduce the size of dataset and afterwards allow more apparent analysis of extracted rules pattern and rules. This paper introduces a method called logic function-based feature selection (L.FB.F.S) for improving the accuracy of data mining classification algorithms. The goal of the feature selection is to find the minimal subsets of attributes of the dataset that can be used for classification tasks by removing both the irrelevant and redundant features. Primarily, L.FB.F.S finds the all MSs of a dataset. Secondly, one of MS with the best classification ability is selected for improving the accuracy of data mining classification algorithms.

Key-Words: - Feature selection, attribute reduction, classification, Boolean function

1 Introduction

In most of information systems (IS) such as data mining, decision support techniques for pattern recognition and neural network training, the data tables called as datasets are used. A basic problem for many practical applications of information systems is the selection of minimal subsets of attributes (MSAs) for sufficient classification of objects in the considered dataset [1-3]. Classification is a widely used technique in various fields, including data mining [4] whose goal is classify a large dataset into predefined classes, using supervised learning algorithms [5]. The abundance of potential features constitutes a serious obstacle to the efficiency of most data mining classification algorithms. Such popular methods as k-nearest neighbors, C4.5 [6] and back propagation [7] are slowed down by the presence of many features, if the most of these features are irrelevant or redundant to the learning task [8]. Moreover, some algorithms may be confused by irrelevant and noisy features and construct poor classifiers [9]. One of the solutions to this problem is preprocessing the datasets with feature selection algorithm (FSA). The feature selection (FS) refers to choosing MSAs from the set of original attributes. The purpose of the FS is to identify the significant features, eliminate the irrelevant of dispensable features and build a good learning model [10]. Furthermore, FS gives the following benefits to dataset: speeding up the data mining algorithms, improving the efficiency and precision of data classification rules, reducing the dimensionality of feature space, facilitating the data collecting process and reducing the amount of the memory needed for storing the datasets [2,11-13]. After FS process the number MSAs may be as large as $\binom{n}{n/2}$ [2,3,11]. Usually MSAs have different cardinalities and those that of least cardinality are called reducts [2,11]. A dataset may have more than one reduct. Anyone of them can be used to replace the original dataset. Finding all of the MSAs from a dataset is an NP-Hard problem [14].

Numerous FSAs have been proposed for improving the classification accuracy rate during the last few decades and several extensive comparative studies have been conducted. However, most of them based on rough-set theory, was introduced by Pawlak [15] in 1982. In [16], a new approach is presented to construct a good ensemble of classifiers using rough set theory and database operations. This method was used to compute a set of reducts which included the entire indispensable attribute required for the decision categories. For each reduct, a reduct table was generated by removing those attributes which are not in reduct. Next, a novel rule induction algorithm was used to compute the maximal generalized rules for each reduct table and a set of reduct classifiers were formed based on the

corresponding reducts. In [17] introduced an application of rough set method for feature selection (FS) in pattern recognition. They proposed a new FSA to the result of principle component analysis (PCA) used for feature projection and reduction. Finally, rough set methods had shown ability to reduce significantly the pattern dimensionality and had proven to be viable data mining techniques as a front end of neural classifiers. In [18] described a new rough set model and refined the core attributes and reducts based on relational algebra to take the advantages of the very efficient set-oriented database operations. Using this model and definitions, they presented two algorithms for computing core and reduct. These algorithms were also being applied in a real-life application with very large datasets. These models are efficient and scalable compared with traditional rough set models.

In this paper, we propose logic function-based feature selection (L.FB.F.S) approach. L.FB.F.S method derives the MSAs from a dataset and we use MSAs for accelerating the classification process and improving the classification accuracy rate.

2 The Logic Function Based Feature Selection Method

An IS can be represented as $S=\{O,C\cup D\}$, where, $O=\{O_i\}_{i=1}^{M}$ is a finite set of objects, $C=\{A_j\}_{j=1}^{N}$ is a finite set of condition features and D is a decision feature. Each condition feature has a domain of values $Vc_{ji}=\{A_j(O_i)_{i=1}^{M}\}_{j=1}^{N}$ and the decision feature has a domain of value $Vd_i=\{D(O_i)\}_{i=1}^{M}$, where Vc_{ji} and Vd_i are the values that the features A_j and D take on object O_i, respectively [19]. Such as IS usually considered a dataset represented by a data table in which the i_{th} row represents a piece of information about the object O_i and the j_{th} column represents the values of the feature A_j. Table 1 shows an example of a IS with seven objects $O=\{O_1,O_2,\ldots,O_7\}$, four condition features $C=\{A_1,A_2,A_3,A_4\}$ and one decision feature $D=\{D\}$.

Table1 An example of an IS

	A_1	A_2	A_3	A_4	D
O_1	L	2	C	5	2
O_2	L	3	S	6	1
O_3	L	3	C	5	2
O_4	H	2	C	6	1
O_5	H	3	C	5	1
O_6	H	3	S	6	1
0_7	L	2	S	6	1

L.FB.F.S selects the MSAs that contain the relevant features from condition set C with respect to the decision set D. The proposed method has four steps, as shown in Figure 1.

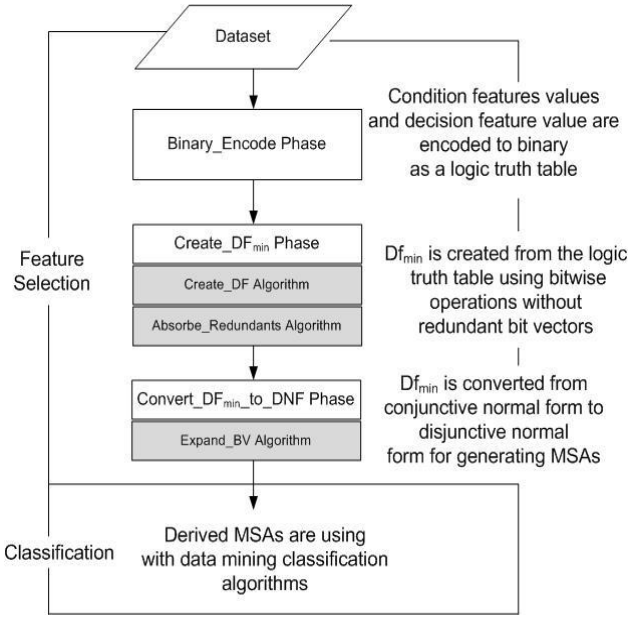

Fig. 1 The flowchart of the proposed L.FB.F.S approach

2.1 Binary Encode Phase

In this algorithm, decision feature and condition features are encoded to binary format, respectively. Primarily, the numbers of bits needed are calculated for encoding the each element of set V_i that belongs to either feature A_i or decision D.

$$n_i=\lceil \log_2|V_i| \rceil \quad (1)$$

For example, consider the domain values of a condition feature A_3 shown as $V_3=\{C,S\}$, that is in Table 1. For this feature $|V_3|=2$ and $n_3=1$ values are obtained according to formula (1). The initial binary encoded value for the first element of V_i is

n_i bit zero code that can be formulated as $\{O\}^{n_i}$. Also the next binary encoded value is one more than the current encoded value. According to these rules we can represent the $V_3 = \{C, S\}$ as shown $V_3 = \{0,1\}$. After binary encoding the values of all condition features and decision feature, Table 2 is obtained from Table 1 is shown as the following:

Table2 The binary encoded truth table image of Table 1

	A1	A2	A3	A4	D
O1	0	0	0	0	0
O2	0	1	1	1	1
O3	0	1	0	0	0
O4	1	0	0	1	1
O5	1	1	0	0	1
O6	1	1	1	1	1
07	0	0	1	1	1

Definition 1 (Bit Vector) A bit vector BV_m is a bit string $b_1, b_2, \ldots, b_n$ with all the condition bits are concatenated in the m. object. For example in Table 2, the bit vector BV_3 for O_3 is 0100 and BV_7 for O_7 is 0011.

2.2 Create_DF$_{min}$ Phase

Discernibility matrix and discernibility function (DF) are mainly used for FS on ISs. In rough set theory, the discernibility matrix of an IS is a symmetric $n \times n$ matrix with entries h_{jk} as shown below. Each entry consists of the set of attributes upon which objects O_j and O_k differ [20].

$$h_{jk} = A_i \in C \mid A_i(O_j) \neq A_i(O_k), \quad i \in \{1,2,\ldots n\} \; and \; \mathrm{j,k} \in \{1,2,\ldots \mathrm{m}\} \tag{2}$$

From discernibility matrix, the discernibility function can be defined. This is concise notation of how each object within the dataset may be distinguished from the others. A discernibility function f_D is a boolean function z boolean variables $A_1^*, \ldots, A_z^*$ defined as [20]:

$$f_D(A_1^*, \ldots, A_z^*) = \wedge \{\vee h_{jk}^* \mid 1 \leq k \leq j \leq |S|, h_{jk} \neq \varnothing\} \tag{3}$$

Our proposed algorithm constitutes the discernibility function from Table 2 with boolean functions by not using the formulas (2) and (3). We use the sets $S_{on}(D)$ and $S_{off}(D)$ as follows:

$$S_{on}(D) = \{BV : D(BV) = 1\} \tag{4}$$

$$S_{off}(D) = \{BV : D(BV) = 0\} \tag{5}$$

By applying the formulas (4) and (5) to the Table 2, the following sets are obtained.

$$S_{on}(D) = \{BV_2, BV_4, BV_5, BV_6, BV_7\}$$
$$= \{0111, 1001, 1100, 1111, 0011\}$$
$$S_{off}(D) = \{BV_1, BV_3\} = \{0000, 0100\}$$

For creating DF, the "XOR" bitwise operator is used for indicating the differences between the sets $S_{on}(D)$ and $S_{off}(D)$. This process is done through algorithm Create_DF explained next.

Algorithm Create_DF ($S_{on}(D)$, $S_{off}(D)$)

(i) Initialize DF =NULL

(ii) For each j, $1 \leq \mathrm{j} \leq |S_{on}(D)|$ {

For each k, $1 \leq \mathrm{k} \leq |S_{off}(D)|$ {

Add $S_{on}(D_j) \wedge S_{off}(D_k)$ to DF}}

(iii) Return DF

We can create DF according to algorithm *Create_DF*. However DF includes redundant terms which increase the used memory and slow down the proposed L.FB.F.S method. For deleting these redundant terms, we used "*AND*" boolean operator between all elements of DF[21]. This process is done through algorithm *Absorbe_Redundants* explained next. This algorithm derives DF_{min} from DF.

Algorithm *Absorbe_Redundants (DF)*

(i) For each l, $1 \leq \mathrm{l} \leq |DF|$ -1

For each m, m+1 $\leq$ m$\leq$ $|DF|$

If $DF_l \,\&\, DF_m = DF_l$

Then

{Remove DF_m from DF}

Else

{If $DF_l \,\&\, DF_m = DF_m$

Then

{Remove DF_l from DF}}

(ii) DF=DF$_{min}$

(iii) Return DF$_{min}$

For example, we can create the DFmin according to sets $S_{on}(D)$, $S_{off}(D)$ and the algorithms Create_DF, Absorbe_Redundants as following:

Step1. DF=∅,$S_{on}(D_1)$={0111}, $S_{off}(D)$={0000,0100}
Step2. DF=DF ∪ ($S_{on}(D_1)$ ^ $S_{off}(D)$) = DF ∪ (0111^0000) ∪ (0111^0100)
Step3. DF={0111,0011}, $S_{on}(D_2)$={1001}
Step4. DF=DF ∪ ($S_{on}(D_2)$ ^ $S_{off}(D)$) = DF ∪ (1001^0000) ∪ (1001^0100)
Step5. DF={0111,0011,1001,1101}, $S_{on}(D_3)$={1100}
Step6. DF=DF ∪ ($S_{on}(D_3)$ ^ $S_{off}(D)$) = DF ∪ (1100^0000) ∪ (1100^0100)
Step7. DF={0111,0011,1001,1101,1100,1000}, $S_{on}(D_4)$={1111}
Step8. DF=DF ∪ ($S_{on}(D_4)$ ^ $S_{off}(D)$) = DF ∪ (1111^0000) ∪ (1111^0100)
Step9. DF={0111,0011,1001,1101,1100,1000,1111 ,1011}, $S_{on}(D_5)$={0011}
Step10. DF=DF ∪ ($S_{on}(D_5)$ ^ $S_{off}(D)$) = DF ∪ (0011^0000) ∪ (0011^0100)
Step11. DF={0111,0011,1001,1101,1100,1000,11 11,1011,0011,0111}

DF_{min}= *Absorbe_Redundants* (DF)

Step1. DF_{min}={~~0111~~,0011,~~1001~~,~~1101~~,~~1100~~,1000,~~11 11~~,~~1011~~,~~0011~~,~~0111~~}

According to Absorbe_Redundants algorithm the left side bit vectors are absorbed by right side bit vector as the following:

{0111, 0011, 0111, 1111, 1011} absorbed by {0011}

{1001, 1101, 1100} absorbed by {1000}

2.3 Convert DF_{min} to DNF Phase

We derived the DF_{min} from the IS is shown Table 1 by using the bitwise operations. We can represent the N-bit string (*Definition 1)* as the following:

$$BV = b_1, b_2, \ldots, b_n = \{\vee A_i \mid 1 \le i \le n, b_i = 1\} \quad (6)$$

According to formula (6), we can present the first term of the $DF(BV_1) = \{0011\}$ as $DF(BV_1) = \{A_3 \vee A_4\}$ and $DF(BV_2) = \{1000\}$ as $DF(BV_2) = \{A_1\}$. Furthermore we can represent the DF_{min} with bit vectors as the following:

$$DF_{\min} = \{BV_1, BV_2, \ldots, BV_n\} = \{BV_1 \wedge BV_2 \wedge \ldots \wedge BV_n\} \quad (7)$$

Eventually, according to formulas (6) and (7), DF_{min} is shown next.

$$DF_{\min} = \{0011,1000\} \rightarrow \{(A_3 \vee A_4) \wedge (A_1)\}.$$

This is the CNF which used in rough set theory. Furthermore, in the DM based rough set theory for this example generates DF with 10 clauses, while the size of DF_{min} is 2. In our proposed method, we indicate CNF as DF_{min} which is reduced from redundant bit vectors. In order to generating the MSAs, we need to convert DF_{min} from CNF to DNF. Therefore, we should expand the bit vector into a structure that preserves all information contained in it because each 1 in the bit vector associated with one feature in the Table 1. We can expand the bit vectors as the algorithm defined below:

Algorithm *Expand_BV (DF_{min})*

(i) For each j, $1 \le j \le |DF_{\min}|$ {
E(BV_j)=NULL
Temp= NULL
For each k, $1 \le k \le n$ {
If $BV_j[k]$=0 temp=temp+0
If $BV_j[k]$=1 {
Temp=temp+1
For each l, $k+1 \le l \le n$
Temp=temp+1
Exit }}
Add Temp to E(BV_j) }
(ii) Return E(BV)

Step1. E(BV_1)=∅, BV_1=0011
Step2. E(BV_1)={0010, 0010}
Step3. E(BV_2)=∅, BV_2=1000
Step4. E(BV_2)={1000}

For creating DNF, we used *"OR"* boolean operator between the sets of E(BV) is defined below formula:

$$DNF = \mathop{\dagger}_{q=1}^{Q} E(BV_q), Q \text{ is the number of E(BV)} \quad (8)$$

For example, we can obtain the DNF according to formula (8) as:

DNF = E(BV_1) ¦ E(BV_2) = {0010,0001} ¦ {1000} = {1010,1001}

ISBN: 978-1-61804-126-5

DNF has the MSAs of the IS which is given in Table 1. The size of all MASs is 2 and so that all MSAs are reduct (MSA have least cardinality). Eventually we use all of the MSAs at the classification phase for this IS. In this DNF, we can associate the 1's positions with the condition features indicated in the Table 1 as the following:

$$DNF = \{1010,1001\} \rightarrow \{(A_1, A_3), (A_1, A_4)\}$$

3 Experimental Results

To estimate the performance of the proposed method, we compared the results generated by selected MSA with the results generated by original sets of attributes for a lot of datasets. In the experiments, we used a target machine with an Intel Core2Quad@2.83 GHz processor and 4 GB memory, running on Microsoft Windows 7 OS. For experiments we chose datasets with different characteristics such as: the number of attributes, the number of classes, the number of distinct values of the attributes and the number of examples. As classification algorithms, we used the algorithms C4.5 [21] without pruning, K-NN [22] with 7 neighbors and Naive Bayes [23]. For estimating the classification accuracy of the algorithms, we used the most widely used cross-validation method [24]. More specifically, we used ten-fold cross-validation in which the dataset to be processed is permuted and partitioned equally into ten disjoint sets D_1, $D_2, \ldots, D_{10}$. In each phase of a cross-validation, one of the yet unprocessed sets was tested, while the union of all remaining sets was used as training set for classification by the algorithms C4.5, K-NN and Naive Bayes. The characteristic of the used datasets are shown in Table 3.

Table3 The characteristic of the used datasets

Dataset	Dataset Type	Number of instances /features /class values	Number of MSAs /max Size MSA /min size MSA	Selected MSA size
Monk1	Categorical	556/6/2	1/3/3	3
Diabet	Mixed	768/8/2	28/4/3	4
Statlog	Mixed	690/14/2	44/7/3	7
Heart	Mixed	270/13/2	109/8/3	8
Breast Cancer	Categorical	699/9/2	19/6/4	6
Chess	Categorical	3196/36/2	4/29/29	29
Vote	Categorical	435/16/2	3/13/9	10

The experimental results of classification algorithms with selected MSA are shown in Table 4. The better results are shown in bold. As a result we obtained better results at the 18 of the 24 experiments.

Table4 The classification accuracy for the datasets provided by original attribute sets and by selected MSA

Dataset	Number of instances /features	Classification Accuracy with original features			Number of selected MSAs	Max. Classification accuracy with selected MSAs from our proposed method		
		C4.5	Naïve Bayes	k-NN		C4.5	Naïve Bayes	k-NN
Monk1	124/6	0.7110	**0.7580**	0.8387	1	**0.8560**	0.7250	**0.9760**
Diabet	768/8	0.7344	0.7604	0.7175	28	**0.7448**	**0.7773**	**0.7265**
Iris	150/4	0.9533	0.9200	0.9533	4	0.9533	**0.9533**	0.9533
Vote	435/16	**0.9540**	0.9103	0.9241	3	0.9517	**0.9333**	**0.9471**
Heart	270/13	0.7704	0.8000	0.7741	109	**0.8259**	**0.8444**	**0.7889**
Chess	3196/36	**0.9921**	0.8546	0.9456	4	0.9882	**0.8893**	**0.9607**
Tictictoe	958/9	0,8257	**0,9154**	0,6983	9	**0,8319**	0,9040	**0,7213**
Australian	690/14	0,8522	0,5623	0,8652	44	**0,8652**	**0,6507**	**0,8664**

ISBN: 978-1-61804-126-5

4 Conclusion

In this paper a logic based feature selection method is proposed for finding the minimal subset of attributes with the best classification accuracy. Usually a dataset has a lot of reducts from which the only one can provide the best classification of this dataset. Therefore, we propose a two-phase approach for obtaining the reduct with the best classification accuracy. In the first phase, we obtain the all of the minimal subset of attributes. Second phase, we select the MSA with the best classification accuracy with using the data mining classification algorithms. Consequently, significantly better results are obtained.

References:

[1] J. A. Starzyk, D. E. Nelson, K. Sturtz, "A Mathematical Foundation for Improved Reduct Generation in Information Systems", Journal of Knowledge and Information Systems, Vol. 2, No. 2, pp.131-146, 2000.

[2] J. Komorowski, L. Polkowski, A. Skowron, "Rough Set: A Tutorial", http://folli.loria.fr/cds/1999/library/pdf/skowron.pdf, 112 pages, 1999.

[3] R. Jensen, Q. Shen, "Semantics-preserving Dimensionality Reduction: Rough and Fuzzy-rough Based Approaches", IEEE Trans. on Knowledge and Data Engineering, Vol. No.12, pp. 1457–1471, 2004.

[4] Fayyad UM, Piatetsky-Shapiro G, Smyth P. Advances in knowledge discovery and data mining. Cambridge: AAAI Press/MIT Press, 1996.

[5] Nikolaos Mastrogiannis, Basilis Boutsinas, Ioannis Giannikos, A method for improving the accuracy of data mining classification algorithms, Computers & Operations Research, Vol:36, pp:2829-2839, 2009

[6] Quinlan J., C4.5: programs for machine learning, California: Morgan Kaufmann; 1993.

[7] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. Parallel distributed processing: explorations in the microstructure of cognition. Cambridge: MIT Press, pp. 318–63, 1986.

[8] Liu, H., Motoda, H., Feature selection for knowledge discovery and data mining, Kluwer, Boston, 1998.

[9] Mark Last, Abraham Kandel, Oded Maimon, Information-theoretic algorithm for feature selection, Pattern Recognition Letters, Vol:22, pp:799-811, 2001.

[10] K. Thangavel, A. Pethalakshmi, Dimensionality reduction based on rough set theory: A review, Applied Soft Computing, Vol:9 Pp:1-12, 2009.

[11] X. Wang, Y. Jie, X.Teng, W. Xia, R. Jensen, Feature Selection Based on Rough Sets and Particle Swarm Optimization, Pattern Recognition Letters, Vol. 28, pp. 459–471, 2007.

[12] Shang WQ, Huang HK, Zhu HB, et al., "A novel feature selection algorithm for text categorization", Expert Systems with Applications, Vol. 33, pp.1-5, 2007.

[13] Matsumoto Y. and Watada J., Knowledge acquisition from time series data through rough sets analysis, International Journal of Innovative Computing, Information and Control, vol. 5/12(B), pp. 4885-4897, 2009.

[14] T.Y. Lin, N. Cercone, Rough sets and Data Mining: Analysis of Imprecise Data, Kluwer Academic Publishers, 1997.

[15] Z. Pawlak, "Rough Sets," Information Journal of Computer and Information Science, vol. 11, no. 5, pp. 341–356, 1982.

[16] X. Hu, Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications, Proceedings of ICDM pp. 233–240, 2001.

[17] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24, pp. 833–849, 2003.

[18] X. Hu, T.Y. Lin, J. Jianchao, A new rough sets model based on database systems, Fundamenta Informaticae, pp. 1–18, 2004.

[19] A. Skowron, The rough sets theory and evidence theory, Fundamenta Informaticae, Vol.13, pp.245-262, 1990.

[20] R. Jensen, Q. Shen, Rough set based attribute selection: A review, htp//cadair.aber.ac.uk /dspace/handle /2160/490, 52 pages, 2007.

[21] S. Kahramanli, M. Hacibeyoglu, A. Arslan, A Boolean Function Approach To Feature Selection In Consistent Decision Information Systems, Expert Systems with Application 38(7) (2011) pp:8229-8239, 2011.