A Rough Set Based Minority Class Oriented Learning Algorithm for Highly Unbalanced Data Sets

Dongyi Ye, Zhaojiong Chen
College of Mathematics and Computer
Fuzhou University
Fuzhou 350002, P.R.China
yiedy@fzu.edu.cn

Abstract

Highly unbalanced data sets occur frequently in many practical applications and quite often the class of interest in such data sets is just a minority class. Like most standard machine learning methods, traditional rough sets based rule learning algorithms do not usually work well on highly unbalanced data sets. In this paper, we present a minority class rule learning algorithm for a highly unbalanced inconsistent data set where the class of interest is the minority one. The proposed algorithm pivots on discovery of the main features that discriminate the minority class from majority classes by finding the so called dominant minority subset. An illustrative example and a real application to customer churning prediction in Telecom are given to show the effectiveness of the proposed algorithm.

1. Introduction

Rough set theory has proved to be an effective tool for learning descriptive knowledge from data sets under inconsistency [1-3]. One typical rough set based decision or classification rule generation approach contains three phrases [3] : (1) find attribute reducts; (2) find value reduction based on the reducts found and yield a reduced table; (3) extract rules from the reduced table. An alternative is to directly generate rules without computing reducts by using discernibility matrices or by gradually eliminating redundant attributes and their values under rule consistency restriction [4-6]. The approaches usually perform well when the training examples are basically evenly distributed among different decision classes. However, when dealing with unbalanced data sets which often appear in many practical applications as indicated in [7], the approach might cause poor performance since the resulting possible rules would generally be governed by the majority classes. This is obviously not reasonable nor desirable in the case of unbalanced data sets where the class of interest is a minority one. Hence, it is of significance to develop learning algorithms that can suitably cope with such data imbalance. Actually, dealing with unbalanced data sets has been a challenging task [8] and there have already been many attempts at dealing with classification of unbalanced data sets [9-13]. However, little has been done in developing rough set based minority class oriented rule learning methods for highly unbalanced data sets. This motivated our study. The purpose of this paper is to develop in the context of rough sets a simple and effective minority class rule learning algorithm for unbalanced data sets. The concept of minority dominant subsets is introduced to describe the knowledge granularity that fits the minority class well. The proposed algorithm searches for the attributes that best characterize the minority class by finding those dominant subsets and generates decision rules (certain and possible) for the minority class by taking into account the information of majority classes.

The paper is organized as follows. In section 2, we review some basic concepts of rough sets and make some comments on highly unbalanced data sets. In section 3, we describe a minority class priority strategy for learning rules from a given unbalanced data set and then present a rule generation algorithm. In section 4, an example is given to illustrate the proposed algorithm. A real world application of the proposed algorithm to the customer churning prediction problem is also presented. Section 5 concludes.

2. Some Basic Concepts and Related Work

Formally, a decision table can be represented as a quadruple $L = \{U, A, V, F\}$ [1], where $U = \{x_1, \cdots, x_n\}$ is a non-empty finite set of objects called universe of discourse, A is a union of condition attributes set C and decision attributes set D, V is the domains of attributes belonging to A, and $F : U \times A \longmapsto V$ is an information function assigning attribute values to objects belonging to U. We assume that C contains m condition attributes $c_1, c_2, ..., c_m$ and without loss of generality that D contains only one decision attribute which takes $k(> 1)$ distinct values. For a subset $P \subseteq A$, $IND(P)$ represents the indiscernible relation induced by the attributes belonging to P and there should be no confusion if we use U to represent either a set of attributes or the relation $IND(P)$. A subset $X \subseteq U$ represents a concept and the partition induced by $IND(P)$ is called a knowledge base and denoted by $U/IND(P)$. In particular, $U/IND(D) = \{Y_1, \cdots, Y_k\}$ is the knowledge base of decision classes. Without loss of generality, assume that $Y_1 \subset U$ is the class of interest which is a minority class.

Let $X \subseteq U$ and $R \subseteq C$. The R–lower approximation of X is defined as $\underline{R}X = \{x \in U : [x]_R \subseteq X\}$, where $[x]_R$ refers to an equivalence class of $IND(R)$ determined by element x. The membership function of object x to concept X with respect to the equivalence class $IND(R)$ is given by

$$\alpha_R^X(x) = \frac{|[x]_R \bigcap X|}{|[x]_R|}.$$

In other words, the value of $\alpha_R^X(x)$ gives the accuracy of a decision rule induced from object x concerning concept X under $IND(R)$. In addition, it is of interest to define the membership in all classes determined by the decision attribute D. Such membership, also called decision class membership distribution, is defined as a mapping $\mu_R : U \to [0,1]^k$, where

$$\mu_R(x) = (\alpha_R^{Y_1}(x), \cdots, \alpha_R^{Y_k}(x)).$$

The largest membership function is given by

$$\lambda_R(x) = \max\{\alpha_R^{Y_1}(x), \cdots, \alpha_R^{Y_k}(x)\}.$$

A decision rule is in the form of "If-Then" associated with a degree of membership to the most possible decision class, also known as the rule's confidence degree or accuracy. More precisely, when inducing a rule from object x under $IND(R)$ where R is an attribute reduct, the value of $\lambda_R(x)$ is usually assigned to the rule as the rule's confidence degree or accuracy. And the rule implies that the object with $IND(R)$ as the knowledge granularity most likely belongs to the decision class Y_k with $\alpha_R^{Y_k}(x) = \lambda_R(x)$. Statistically, a majority class often prevails over a minority class (often the class of interest in practice) in this type of membership assigning. This is not reasonable nor desirable in practice. In the next section, we shall present a minority oriented learning algorithm that lends itself to rule generation for the minority class of a highly unbalanced data set.

Let $M = (m_{ij})$ be the Skowron's discernibility matrix of the decision table [3]. Set $PCore = \{\cup m_{ij} : m_{ij}\ \texttt{is a singleton}, x_i \in Y_1, x_j \in (U - Y_1)\}$. Obviously, the attributes in $Pcore$ are indispensable for discerning objects from those in majority classes. Hence, we call $Pcore$ the partial core with respect to the minority class.

3. Minority Class Priority Strategy and Learning Algorithm

The idea behind our approach is to use a minority class priority strategy. We are not looking for an attribute reduct. Instead, we focus on finding those attributes that best characterize the minority class. In order to better explain our idea, suppose that $Y_1 \subset U$ is the class of interest which is the minority class. For some $x \in Y_1$ and any $P \in C$, let $\delta_P(x) = \frac{\alpha_P^{Y_1}(x)}{\lambda_P(x)}$. The point is to find a subset R of attributes that maximizes the value of $\delta_P(x)$ among all possible subsets P. The subset R is called the dominant subset that best features the object x of the minority class. We use a greedy approach to find such a dominant subset. We start from the partial core, namely, set $R = PCore$. If $\delta_R(x) = 1$, then $\alpha_R^{Y_1}(x) = 1$(i.e.,$[x]_R \in Y_1$) or $\alpha_R^{Y_1}(x) = \lambda_R(x)$. In either case, R is a dominant subset for object x of the minority class. Thus, the decision rule induced from the equivalence class of object x, namely, $[x]_R$, is reasonably governed by the minority class under IND(R) and reads as follows:

$$d_x : [x]_R \Rightarrow d = 1(cf = 1)$$

Here, whether $\alpha_R^{Y_1}(x) = 1$ or not, we assign 1 to the rule as its confidence degree to strengthen the bias towards the minority class.

If $threshold < \delta_R(x) < 1$, it indicates that it is acceptable and reasonable to bias the rule towards the minority class under the minority priority policy. So, the value $\lambda_R(x)$ instead of $\alpha_R^{Y_1}(x)$ is assigned to the rule induced from $[x]_R$ as follows:

$$d_x : [x]_R \Rightarrow d = 1(cf = \lambda_R(x))$$

Finally, if $\delta_R(x) < threshold$, then we proceed to find the dominant subset for object x. We use a greedy policy. Identify a new subset R' of attributes by adding a new attribute to R such that the value of $\delta_{R'}(x)$ attains its maximum among all possible new subsets of attributes thus obtained. The process is then repeated until a rule is generated from x or $R' = C$. In the latter case, x is regarded as noise data and no rule will be generated from it. We then move to operate on next object. The above process can be formally summarized in Fig. 1.

Notice that the algorithm generates rules only for the minority class. And the threshold value plays an important role in the algorithm since it determines the extent of bias in favor of the minority class. The smaller the threshold, the more likely a minority class example(referred to as positive example) is to be correctly labeled and at the same time the more likely a majority class example (referred to as negative example) is to be classified as *positive*.

```
**Algorithm**RSMLA(Data: $U$, Attributes: $C \cup D$, Minority class: $MinY$, Threshold: $\varepsilon$, Rule Set: $Rset$)
**begin**
    Compute $IND(U)/\{D\}$, $MinY$ and $PCore$;
    $Rset=null$;
    **while** not($MinY$= Empty) **do begin**
        $R$= $PCore$ and $Q=C-R$;
        Take an object $x \in MinY$ and compute $\delta_R(x)$;
        **while** $\delta_R(x) \leq \varepsilon$ **do begin**
            **If** $R = C$, **then break**
            **else begin**
                find an attribute $a*$ that maximizes $\delta_{R\cup\{a\}}(x)$ over $Q$;
                $R= R \cup \{a*\}$;
                $Q=Q-\{a*\}$
            **end**;
        **end**;
        $MinY$=$MinY$-$\{[x]_R\}$;
        **If not**($R = C$), **then begin**
            **If** $\delta_R(x) = 1$, **then** $Rset$= $Rset \cup \{d_x : [x]_R \Rightarrow d = 1(cf = 1)\}$
            **else** $Rset$= $Rset \cup \{d_x : [x]_R \Rightarrow d = 1(cf = \lambda_R(x))\}$;
        **end**
    **end**;
    **return**($RSet$);
**end**
```

Figure 1. Minority Oriented Algorithm

4. An Illustrative Example and a Real Application Example

The data set as shown in table 1 contains 16 objects of two classes with 5 condition attributes. Obviously, we are dealing with an unbalanced data set with only three objects belonging to the minority class.

Table 1. an unbalanced data set

a1	a2	a3	a4	a5	d
0	1	1	0	1	0
1	0	2	0	2	0
1	1	1	0	2	0
1	1	1	0	1	1
1	1	1	0	2	1
0	0	0	1	2	1
1	0	1	0	1	1
1	0	2	1	1	1
0	2	1	1	0	1
1	0	2	1	1	1
1	1	1	1	1	1
0	0	2	0	0	1
1	1	1	1	0	1
0	1	0	0	0	1
1	0	0	0	0	1
0	2	2	0	2	1

If we use a classic attribute reduction method, we will have in total two reducts, namely, $Q1 = \{a1, a3, a5\}$ and $Q2 = \{a1, a2, a3, a4\}$. Let us choose the smaller one to extract rules in the traditional way. Then, the decision rules concerning the minority class read as follows:
$r1 : (a_1 = 0) \wedge (a_3 = 1) \wedge (a_5 = 0) \Rightarrow d = 0(cf = 1)$;
$r2 : (a_1 = 1) \wedge (a_3 = 2) \wedge (a_5 = 2) \Rightarrow d = 0(cf = 1)$;
$r3 : (a_1 = 1) \wedge (a_3 = 1) \wedge (a_5 = 2) \Rightarrow d = 0(cf = 0.5)$;
Now, let us apply the RSMAL algorithm to generate rules for the minority class. Let the threshold be 2/3 and set $R = Pcore = \{a1\}$ and pick x_1 from the minority class MinY. We have $\delta_R(x_1) = \frac{1}{5}$, so add a new attribute to R. As $\delta_{R\cup\{a5\}}(x_1) = 1$, we generate a rule for $[x_1]_{\{a1,a5\}}$. We then continue the searching process on object x_2 and reset R to be the partial core $\{a1\}$. As $\delta_R(x_2) = \frac{1}{4} < \varepsilon$, an appropriate attribute should be added to R. Since $\delta_{R\cup\{a5\}}(x_2) = \frac{2}{3} \geq \varepsilon$, we get a rule for $[x_2]_{\{a1,a5\}} = \{x_2, x_3\}$; Now MinY is empty, the algorithm is thus terminated. The resulting minority class rule set contains two rules as follows:
$r1 : (a_1 = 0) \wedge (a_5 = 1) \Rightarrow d = 0(conf = 1)$;
$r2 : (a_1 = 1) \wedge (a_5 = 2) \Rightarrow d = 0(conf = \frac{2}{3})$;
Notice that the subset $R = \{a1, a5\}$, though not a reduction in Pawlak's sense, gives a more concise and reasonably biased description of the minority class. We have thus found an appropriate granularity for describing the class of interest.

Next, we discuss the application of the proposed algorithm to a customer churn prediction problem. The database of our application came from a local Telecom company. In addition to phone numbers and customers, each transaction has ten other features, including monthly bill, credit evaluation, SMS, toll ticket(long distance call average), IP call, integral, appeal, and service status. The last feature indicates if the customer involved is churning or not. We create an example for each transaction. Each example is a vector of 53 attribute values. An example is marked as *bad* or positive, if the corresponding transaction manifested an active turnover of the associated customer. Otherwise, it is marked as *good* or negative. The distribution of examples among these two classes, bad and good, is highly unbalanced with about 1 % of all examples belonging to class bad. We also used a simple neighboring interval merging based discretization algorithm to handle numeric attributes. The database provided by the company contains more than thirty thousands examples. We randomly chose about 70 % of examples to build the training set(i.e., the underlying decision table) and the remaining 30 % were used for testing.

Table 2. Prediction of the test data

threshold	TPR	FPR
0.05	0.92	0.022
0.1	0.92	0.015
0.15	0.87	0.013
0.2	0.83	0.012
0.25	0.80	0. 009

We tried several threshold values to select a suitable set of rules. Note that the algorithm generates rules only for the minority(*bad* or *positive*) class. Thus, only the true- positive rate and the false-positive rate could be available. The churning prediction results on the test data set are shown in Table 2, where TPR is the true-positive rate and FPR is the false-positive rate.

We see from Table 2 that the algorithm performs well in terms of correct prediction of the minority class. Among all the listed candidate thresholds, the value 0.1 seems to be an appropriate choice that yields reasonably high TPR and reasonably low FPR.

5. Conclusion

Dealing with unbalanced data sets is one of the main recent issues in the data mining community. In this paper, we presented in the context of rough sets a minority class oriented learning algorithm for a highly unbalanced data set where the minority class is the class of interest. The proposed minority biased strategy that finds dominant attributes in favor of the minority class proved to be effective in retaining the decision information of the minority class.

Acknowledgement: This work was partly funded by Natural Science Foundation of China (No. 60602052), Natural Science Foundation of Fujian Province of China (No. A060014) and the Key Lab of Spatial Data Mining and Information Sharing of Ministry of Eduction of China.

References

[1] Pawlak Z., Slowinski R., Rough set approach to multiattribute decision analysis. European Journal of Operational Research, **72** (1994)443–459

[2] Liu Q., Rough Sets and Rough Reasoning, Science Press, Beijing, 2001

[3] Wang G.Y., Rough Set Theory and Knowledge Acquisition, Xian Jiaotong University Press, Xian, 2001

[4] Wang G.Y., Fisher P.S., Rule generation based on rough set theory,In: Proceedings of SPIE 4057, 2000, 181–189

[5] Nakayama, H.; Hattori, Y.; Ishii, R.Rule extraction based on rough set theory and its application to medical data analysis, In: Proceedingsof IEEE International Conference on SMC, Volume 5, 1999 924 – 929

[6] Yuxia Qiu , Keming Xie and Gang Xie, Decision Rules Extraction Strategy Based on Bit Coded Discernibility Matrix, LNAI 4062, 2006, Springer-Verlag

[7] Japkowicz N. and Stephen, S. The class imbalance problem: A systematic study. Intelligent Data Analysis. 2002,6(5).

[8] Yang Qiang, Wu Xindong, Ten Challenging problems in data mining, IEEE Transactions on Knowledge and data Engineering, **18**(2007)43–56

[9] Kubat, M. Matwin, S. (1997). Addressing the curse of imbalanced data sets: One-sided sampleing. Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179-186. Morgan Kaufmann.

[10] Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 155-164, San Diego, CA.

[11] Jianping Zhang, Eric Bloedorn, Lowell Rosen, et al, Learning Rules from Highly Unbalanced Data Sets, Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM04),2004

[12] YANG Ming, YANG Ping, An algorithm for computation of a core for unbalanced classification data, Control and Decision, 2007,22(06)

[13] Yang Qiang, Yin Jie, Ling Charles,and Pan Rong,Extracting actionable knowledge from decision trees, IEEE Transactions on Knowledge and data Engineering, **18**(2007)43–56