The Application of Emerging Patterns for Improving the Quality of Rare-Class Classification

Hamad Alhammady and Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering
The University of Melbourne
Parkville, Victoria 3010, Australia
{hhammady, rao}@cs.mu.oz.au

Abstract. The classification of rare cases is a challenging problem in many real life applications. The scarcity of the rare cases makes it difficult for traditional classifiers to classify them correctly. In this paper, we propose a new approach to use emerging patterns (EPs) [3] in rare-class classification (EPRC). Traditional EP-based classifiers [2] fail to achieve accepted results when dealing with rare cases. EPRC overcomes this problem by applying three improving stages: generating new undiscovered EPs for the rare class, pruning low-support EPs, and increasing the supports of the rare-class EPs. An experimental evaluation carried out on a number of rare-class databases shows that EPRC outperforms EP-based classifiers as well as other classification methods such as PNrule [1], Metacost [6], and C4.5 [7].

1 Introduction

Classification of rare cases is an important problem in data mining. This problem is identified as distinguishing rarely-occurring samples from other overwhelming samples in a significantly imbalanced dataset [1]. In this paper, we investigate how to employ emerging patterns (EPs) in rare-case classification. EPs are a new kind of patterns that introduced recently [3]. EPs are defined as itemsets whose supports increase significantly from one class to another. The power of EPs can be used to build high-performance classifiers [2]. Usually, these classifiers achieve higher accuracies than other state-of-the-art classifiers. However, simple EP-based classifiers do not retain their high performance when dealing with datasets which have rare cases. The reason for this failure is that the number of the rare-class EPs is very small, and their supports are very low. Hence, they fail to distinguish rare cases from a vast majority of other cases.

In this paper we propose a new approach to use the advantage of EPs to classify rare cases in imbalanced datasets. The aim of our approach (called EPRC) is to improve the discriminating power of EPs so that they achieve better results when dealing with rare cases. This is achieved through three improving stages; 1) generating new undiscovered EPs for the rare class, 2) pruning the low-support EPs, and 3) increasing the support of rare-class EPs. These stages are detailed in section 3.

H. Dai, R. Srikant, and C. Zhang (Eds.): PAKDD 2004, LNAI 3056, pp. 207–211, 2004.

In this paper we adopt the *weighted accuracy* [8], and the *f-measure* [9] as they are well-known metrics for measuring the performance of rare-case classification. The f-measure depends on the *recall* and *precision* of the rare class.

2 Emerging Patterns

Let $obj = \{a_1, a_2, a_3, \ldots a_n\}$ is a data object following the schema $\{A_1, A_2, A_3, \ldots A_n\}$. A_1, A_2, A_3,.... A_n are called *attributes*, and a_1, a_2, a_3, ... a_n are *values* related to these attributes. We call each pair (attribute, value) an *item*.

Let I denote the set of all items in an encoding dataset D. *Itemsets* are subsets of I. We say an instance Y contains an itemset X, if $X \subseteq Y$.

Definition 1. Given a dataset D, and an itemset X, the support of X in D is defined as the percentage of the instances in D that contain X.

Definition 2. Given two different classes of datasets *D1* and *D2*. The growth rate of an itemset X from *D1* to *D2* is defined as the ratio between the support of X in *D2* and its support in *D1*.

Definition 3. Given a growth rate threshold $p>1$, an itemset X is said to be a p-emerging pattern (p-EP or simply EP) from *D1* to *D2* if $GrowthRate_{D_1 \to D_2}(X) \geq p$.

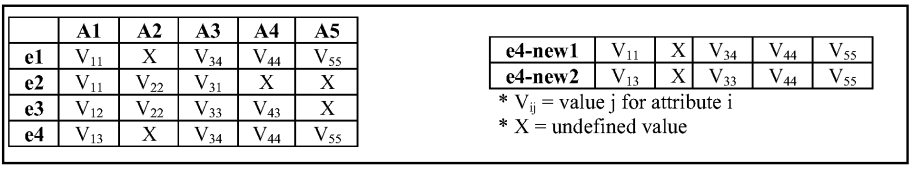
3 Improving Emerging Patterns

As described earlier, our approach aims at using the discriminating power of EPs in rare-case classification. We introduce the idea of generating new EPs for the rare class. Moreover, we adopt eliminating low-support EPs in both the major and rare classes, and increasing the support of rare-class EPs.

The first step in our approach involves generating new rare-class EPs. Given a training dataset and a set of the discovered EPs, the values that have the highest growth rates from the major class to the rare class are found. The new EPs are generated by replacing different attribute values (in the original rare-class EPs) with the highest-growth-rate values. After that, the new EPs that already exist in the original set of EPs are filtered out. Figure 1 shows an example of this process. The left table shows four rare-class EPs. Suppose that the values that have the highest growth rates for attributes A1 and A3 are V_{11} and V_{33} respectively. Using these two values and EP e4, $\{V_{13}, X, V_{34}, V_{44}, V_{55}\}$, we can generate 2 more EPs (in the right table). The first EP is $\{V_{11}, X, V_{34}, V_{44}, V_{55}\}$ (by replacing V_{13} with V_{11}). The second EP is $\{V_{13}, X, V_{33}, V_{44}, V_{55}\}$ (by replacing V_{34} with V_{33}). However, the first new EP already exists in the original set of EPs (e1). This EP is filtered out. We argue that these new generated EPs have a strong power to discriminate rare-class instances from major-class instances. There are two reasons for this argument. The first reason

is that these new EPs are inherited from the original rare-class EPs which themselves have a discriminating power to classify rare cases. The second reason is that they contain the most discriminating attribute values (attribute values with the highest growth rates) obtained from the training dataset.

Fig. 1. Example of generating new rare-class Eps



	A1	A2	A3	A4	A5
e1	V_{11}	X	V_{34}	V_{44}	V_{55}
e2	V_{11}	V_{22}	V_{31}	X	X
e3	V_{12}	V_{22}	V_{33}	V_{43}	X
e4	V_{13}	X	V_{34}	V_{44}	V_{55}

e4-new1	V_{11}	X	V_{34}	V_{44}	V_{55}
e4-new2	V_{13}	X	V_{33}	V_{44}	V_{55}

* V_{ij} = value j for attribute i
* X = undefined value

Based on the above explanation, we have algorithm 1 to generate new rare-class EPs.

```
Algorithm 1 (Generating new rare-class EPs)
Input: the training dataset D, and discovered EPs E.
Output: a set of new rare-class EPs.
Method:
For each attribute i in D
  A_i = value with the highest growth rate of attribute i.
For each rare-class EP e
  For each attribute value k related to i
    If k != A_i
      Generate a new EP  e_new = e
      Replace k by A_i in e_new
      If e_new does not exist in E
```

The second step involves pruning the low-support EPs. This is performed for both the major and rare classes. Given a pruning threshold, the average growth rate of the attribute values in each EP is found. The EPs whose average growth rates are less than the given threshold are eliminated. We argue that these eliminated EPs have the least discriminating power as they contain many least-occurring values in the dataset.

The third step involves increasing the support of rare-class EPs by a given percentage. The postulate behind this point is that this increase compensates the effect of the large number of major-class EPs. That is, the overwhelming major-class EPs make many rare-class instances classified as major-class.

4 Experimental Evaluation

In order to investigate the performance of our approach, we carry out a number of experiments. We use three challenging rare-class databases with different distributions of data between the major and rare classes. These datasets are the insurance dataset [5], the disease dataset [4], and the sick dataset [4]. We compare our

approach to other methods such as PNrule, Metacost, C4.5, and CEP. The comparison we present is based on the weighted accuracy, traditional accuracy, major-class accuracy, recall (rare-class accuracy), precision, and F-measure.

4.1 Tuning

Our approach uses three parameters. These parameters are the threshold of pruning the major-class EPs, the threshold of pruning the rare-class EPs, and the percentage of increasing the support of rare-class EPs. The parameters of our approach need to be tuned to achieve the best results. To achieve this aim, we split the training set into 2 partitions. The first partition (70%) is used to train the classifier. The second partition (30%) is used to tune the parameters.

4.2 Comparative Results

After tuning the insurance dataset, the parameters of our approach are fixed to deal with this dataset. We run different methods on the test set of this dataset. These methods include EPRC (our approach), PNrule [1], C4.5 [7], Metacost [6], and CEP (EP-based classifier) [2]. The results of these methods are presented in table 1. Our approach outperforms all other methods in terms of the weighted accuracy and f-measure. This performance is achieved by balancing both the recall and the precision.

Table 1. The results of the insurance dataset

Classifier	Weighted accuracy	Traditional accuracy	Major-class accuracy	Recall (rare-class accuracy)	Precision	F-measure
EPRC	**63.89%**	80.57%	82.82%	44.95%	14.20%	**21.59%**
PNrule	58.91%	87.12%	90.03%	26.89%	15.80%	19.90%
C4.5	52.87%	90.95%	96.09%	9.66%	13.52%	11.27%
Metacost	49.80%	5.95%	0.02%	**99.57%**	5.92%	11.18%
CEP	50.85%	**93.8%**	**99.60%**	2.10%	**25.00%**	3.87%

4.3 The Effects on the EP-Based Classifier

Our basic aim behind the work presented in this paper is to improve the performance of EP-based classifiers in rare-case classification. In this experiment we compare the results obtained for our three datasets using CEP (EP-based classifier) and EPRC. As stated in section 2, EPRC uses CEP as its basic EP-based classifier. The three datasets were tuned using 30% of the training set. Table 2 shows how our approach enhances the performance of the EP-based classifier. There are significant increases in the weighted accuracy and f-measure from CEP to EPRC.

Table 2. The effect on the EP-based classifier

Experiment	Weighted accuracy	F-measure
Insurance dataset (CEP)	50.85%	3.87%
Insurance dataset (EPRC)	63.89%	21.59%
Disease dataset (CEP)	49.94%	Undefined
Disease dataset (EPRC)	65.07%	34.78%
Sick dataset (CEP)	78.89%	70%
Sick dataset (EPRC)	94.57%	79.71%

5 Conclusions and Future Research

In this paper, we propose a new EP-based approach to classify rare classes. Our approach, called EPRC, introduces the idea of generating new rare-class EPs. Moreover, it improves EPs by adopting pruning low-support EPs, and increasing the support of rare-class EPs. We empirically demonstrate how improving EPs enhances the performance of EP-based classifiers in rare-case classification problems. Moreover, our approach helps EP-based classifiers outperform other classifiers in such problems. The proposed approach opens many doors for further research. One possibility is improving the performance of EP-based classifiers further by adding further improving stages to increase the discriminating power of EPs.

References

1. M. V. Joshi, R. Agarwal, and V. Kumar. Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction. In Proceedings of ACM (SIGMOD '01), Santa Barbara, California, USA.
2. J. Bailey, T. Manoukian, and K. Ramamohanarao. Classification Using Constrained Emerging Patterns. In Proceedings of the Fourth International Conference on Web-Age Information Management (WAIM '03), Chengdu, China.
3. G. Dong, and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In Proceedings of 1999 International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, USA.
4. C. Blake, E. Keogh, and C. J. Merz. UCI repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine, CA, 1999. http://www.ics.uci.edu/~mlearn/MLRepository.html.
5. P. van der Putten, M. de Ruiter, and M. van Someren. The CoIL Challenge 2000 report. http://www.liacs.nl/~putten/library/cc2000, June 2000.
6. P. Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In Proceedings of 1999 International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, USA.
7. I. H. Witten, E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, CA., 1999.
8. J. Cheng, C. Hatzis, H. Hayashi, M. Krogel, S. Morishita, D. Page, and J. Sese. KDD Cup 2001 report. ACM SIGKDD Explorations, January, 2002.
9. C. J. van Rijsbergan. Information Retrieval. Butterworths, London, 1979.