A New Emerging Pattern Mining Algorithm and Its Application in Supervised Classification

Milton García-Borroto[1,2], José Francisco Martínez-Trinidad[2], and Jesús Ariel Carrasco-Ochoa[2]

[1] Centro de Bioplantas. Carretera a Moron km 9, Ciego de Avila, Cuba
mil@bioplantas.cu
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica. Luis Enrique Erro No. 1, Sta. María Tonanzintla, Puebla, México, C.P. 72840
{ariel,fmartine}@ccc.inaoep.mx

Abstract. Obtaining an accurate class prediction of a query object is an important component of supervised classification. However, it could be important to understand the classification in terms of the application domain, mostly if the prediction disagrees with the expected results. Many accurate classifiers are unable to explain their classification results in terms understandable by an application expert. Emerging Pattern classifiers, on the other hand, are accurate and easy to understand. However, they have two characteristics that could degrade their accuracy: global discretization of numerical attributes and high sensitivity to the support threshold value. In this paper, we introduce a novel algorithm to find emerging patterns without global discretization, which uses an accurate estimation of the support threshold. Experimental results show that our classifier attains higher accuracy than other understandable classifiers, while being competitive with Nearest Neighbors and Support Vector Machines classifiers.

Keywords: Emerging pattern mining, Understandable classifiers, Emerging pattern classifiers.

1 Introduction

The main goal of a supervised classification algorithm is to build a model based on a representative sample of the problem classes [1]. This model can be used to predict the class of new objects or to gain understanding of the problem domain. In many cases, the result of the classification is not enough; the user could need to understand the classification model and the classification results, mostly if the classification disagrees with the expected results.

Many accurate classifiers, like Neural Networks [2] or Support Vector Machines [3], are unable to explain their classification results in terms understandable by an application expert. Emerging Pattern classifiers, on the other hand, build accurate and easy to understand models. An emerging pattern is a combination of feature values that appears mostly in a single class. This way, an

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 150–157, 2010.

emerging pattern can capture useful contrasts among the problem classes [4], which can be used to predict the class of unseen objects.

Emerging pattern classifiers are very valuable tools to solve real problems in fields like Bioinformatics [5], streaming data analysis [6], and intruder detection [7].

Current methods for finding Emerging Patterns in a database have two main drawbacks:

- Global discretization of numerical attributes, which could seriously degrade the classification accuracy
- High sensitivity to the support threshold value, which makes very hard for the user to select a good value

In this paper, we introduce the Crisp Emerging Pattern Mining (CEPM), a novel algorithm to find emerging patterns, which does not apply global discretization of numerical attributes. CEPM extracts patterns using a special procedure, from a collection of C4.5 decision trees. To find a representative collection of patterns, our algorithm uses a novel object weighting scheme. CEPM applies local discretization, using only such attribute values appearing in the objects on each tree node. Additionally, CEPM finds an accurate estimation of the minimal support threshold, testing different values decrementally. It starts from a high enough value and ends when the threshold attains the expected abstention ratio. CEPM returns a set of emerging patterns with the highest support value associated with the lowest expected abstention ratio.

The rest of the paper is organized as follows: Section 2 presents a brief revision about classification using emerging patterns, Section 3 introduces the new algorithm for mining emerging patterns without global discretization, Section 4 presents the algorithm to estimate the minimal support threshold, Section 5 shows the experimental results, and Section 6 presents our conclusions.

2 Classification Using Emerging Patterns

A *pattern* is an expression, defined in a language, which describes a collection of objects; the amount of objects described by a pattern is the pattern *support*. In a supervised classification problem, we say that a pattern is *emerging* if its support increases significantly from one class to the others [4]. Emerging patterns (EPs) are usually expressed as combinations of feature values, like $(Color = green, Sex = male, Age = 23)$ or as logical properties, like $[Color = green] \wedge [Sex = male] \wedge [Age > 23]$.

Most algorithms for emerging pattern mining have the goal of finding the patterns that satisfy a desired property: being supported by a single class, minimality over subset inclusion, or tolerance to noisy objects. These algorithms have the following general steps:

1. Selection of the minimal support threshold μ
2. Global discretization of numerical attributes

3. Representation of the transformed objects using a particular structure
4. Traversing the structure to find emerging patterns
5. Pattern filtering

Using this traditional algorithm has two important drawbacks:

1. Global discretization of numerical attributes could drastically degrade the classifier accuracy, since an emerging pattern relates a combination of feature values with a class. Therefore, discretizing a numerical attribute without considering the values of other features could hide important relations.
   In Table 1, we can see that SJEP [8], one of the most accurate emerging pattern classifiers, obtains very poor accuracies in databases like *Iris*, while all other classifiers attain accuracies above 93%. In some other databases, SJEP is unable to extract even a single pattern, because most numerical features are discretized into a single categorical value. This behavior is mainly due to using the Entropy discretization method [9], but other discretization methods obtain similar results, maybe in different databases.
2. High sensitivity to the support threshold value. The accuracy of the classifier could have a serious degradation on small variations of the minimal support value. For example, in *chess* and *census* databases, the accuracy drops 3% with a variation of 2 in the threshold value [10].

3 Crisp Emerging Pattern Mining (CEPM)

In this section, we introduce CEPM, a new emerging pattern mining algorithm with local discretization of numerical features. CEPM extracts patterns from a collection of C4.5 decision trees [11], using a special pattern mining procedure during the tree induction. To guarantee that CEPM finds a representative collection of patterns, it uses a novel object weighting scheme.

The tree induction procedure has the following characteristics:

- Candidate splits are binary. Nominal attributes use properties like $[Feature = a]$ and $[Feature \neq a]$ for each one of their values; numerical attributes use properties like $[Feature > n]$ and $[Feature \leq n]$ for all candidate cut points
- If a node has less than μ objects, it is not further split because it cannot generate emerging patterns
- To select the best split, our algorithm evaluates the *weighted information gain*. The weighted information gain is similar to the classical information gain but there is a weight associated to each object. This way, P_{Class} and P_{child} are calculated using (1). Note that objects with weight close to 0 have low influence in the determination of the best split.

$$P_{Class} = \frac{\sum_{o \in Class} w_o}{\sum w_o}, \quad P_{child} = \frac{\sum_{o \in child} w_o}{\sum w_o} \ . \tag{1}$$

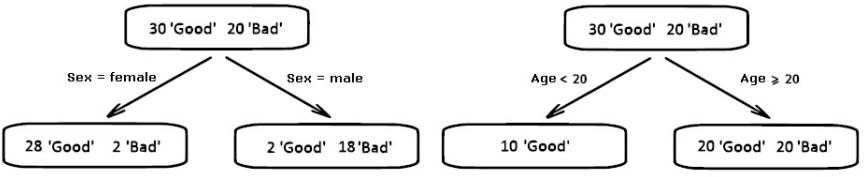

Fig. 1. Example of an emerging pattern appearing in a non-optimal candidate split

During the tree induction, every tree node that (A) has at least μ objects in a class, and (B) has at most one object in the complement of that class, generates a new emerging pattern. Each pattern consists in the conjunction of the properties from the node to the root.

Additionally, CEPM extracts patterns while evaluating the splits, even if a split does not have the highest gain; any tree node that fulfills (A) and (B) generates an emerging pattern. For example, Fig. 1 shows two candidate splits, using different properties. Although the first one has the highest information gain, the second contains the emerging pattern ($Age < 20$). So, this pattern is extracted although the split is discarded.

CEPM iteratively induces several decision trees, updating the object weights after each induction. The algorithm updates the weights using (2).

$$w_o = \frac{\operatorname{arccot}\left(5 \cdot \frac{Support_o}{averageSupport}\right)}{\pi} \quad . \tag{2}$$

where

- $Support_o$ is the sum of the support of such patterns contained in o. If a pattern belongs to a different class than o, its support is multiplied by -1
- $averageSupport$ is the average support of the patterns in the database, which is estimated based on the patterns found in the first built tree.

We can describe CEPM using the following pseudocode:

1. Initialize object weights to 1
2. Induce the first decision tree with the initial weights and extract the first emerging patterns
3. Calculate the average support, used in weight recalculation
4. Repeat while a new pattern is added in the last iteration
   (a) Recalculate object weights
   (b) Induce the new decision tree with current weights and extract the emerging patterns
   (c) Add the new patterns to the pattern collection
5. Return mined patterns

It is worth to mention that CEPM returns a set of the most general emerging patterns with support greater or equal to μ. A pattern P is more general than

a pattern Q if the set of objects described by Q is strictly contained in those described by P, considering all the objects in the universe. Additionally, CEPM returns the abstention ratio, which is the ratio of objects that are not covered by the resultant patterns.

The CEPM based classifier, named CEPMC, uses the following decision rule: to assign an object to the class with maximum value of the total votes given by the patterns contained in the object. Every pattern contained in the object votes for its own class with its total support. If no pattern supports the object or there is a tie in the votes, the classifier refuses to classify the object.

4 Estimating the Minimal Support Threshold for CEPM

Selecting the minimal support threshold for an emerging pattern classifier is a difficult task; a classifier using patterns with higher μ values, is a more accurate classifier, but could reject to classify more objects. On the contrary, a classifier using patterns with lower μ values might contain many useless patterns, which could degrade the classification accuracy.

The algorithm proposed for calculating the minimal support value infers the initial support ($MaxSupport$) and the minimal expected abstention rate ($MinAbstRate$). Then, it tests support values, starting from $\mu = MaxSupport$, until a μ value produces an abstention rate lower than $MinAbstRate$. The value of μ is decremented using a calculated $Step = MaxSupport/10$, because if $MaxSupport$ is high, decrementing μ by 1 could be too costly.

Some important remarks:

1. $MaxSupport$ is inferred based on two criteria. If it is higher than the optimum, the algorithm makes unnecessary iterations; if it is lower, better models (with higher μ) are disregarded.
2. $MinAbstRate$ is inferred using $\mu = 2$, so it measures the minimal expected abstention ratio of a pattern based classifier using CEPM. The algorithm searches for more accurate classifiers (having patterns with higher μ values) with the same abstention level.

5 Experimental Results

To compare the performance of the CEPMC classifier, we carried out some experiments over 22 databases from the UCI Repository of Machine Learning [12]. We selected six state-of-the-art classifiers: Nearest Neighbors [13], Bagging and Boosting [14], Random Forest [15], C4.5 [11] and Support Vector Machines [3]. For each classifier, we used the Weka 3.6.1 implementation [16] with its default parameters. We also tested SJEP [8], which is one of the most accurate emerging pattern based classifiers, using the minimal support threshold suggested by their authors.

We performed 10-fold cross validation, averaging the results. In both SJEP and CEPMC we reported abstentions as errors. In these objects, the classifier

Table 1. Accuracy results of the classifiers in the selected databases. The highest accuracy per database is bolded.

DBName	3NN	AdaBoost	Bagging	C4.5	RandFor	SVM	SJEP	CEPMC
balance-scale	85.4	71.7	82.6	77.6	79.4	**87.5**	16.0	79.5
breast-cancer	70.3	72.4	71.0	**73.4**	65.8	70.7	44.5	72.0
breast-w	**96.7**	94.9	96.0	94.9	95.1	**96.7**	96.1	96.0
cleveland	82.5	84.2	79.9	78.2	78.6	**84.5**	77.9	81.2
haberman	70.6	70.9	**72.5**	68.0	67.6	**72.5**	0.0	68.6
hayes-roth	71.4	53.6	75.0	**89.3**	85.7	53.6	0.0	78.6
heart-c	81.2	**83.2**	81.9	76.2	80.9	82.8	78.6	81.2
heart-h	**83.6**	81.6	79.9	79.6	79.3	82.6	46.3	81.0
heart-statlog	79.3	80.7	79.3	79.3	79.3	**83.0**	64.8	80.0
hepatitis	82.0	81.2	82.0	78.8	81.3	**86.5**	77.5	82.5
iris	96.0	**96.7**	93.3	94.0	94.7	96.0	66.7	95.3
labor	**90.7**	87.0	84.0	80.0	86.7	**90.7**	82.0	89.0
liver-disorders	65.5	66.1	68.7	68.7	**70.7**	57.7	0.0	69.3
lymph	85.9	75.7	77.7	78.5	79.9	**87.9**	51.5	83.7
mp1	50.0	50.0	50.0	50.0	50.0	50.0	57.9	**100.0**
mp2	51.4	50.0	55.1	59.7	58.6	50.5	34.0	**83.8**
mp3	50.0	50.0	50.0	50.0	50.0	50.0	63.7	**97.5**
shuttle	60.0	**65.0**	60.0	60.0	55.0	45.0	0.0	50.0
spect	64.7	66.8	61.5	66.8	62.0	67.9	0.0	**78.1**
tic-tac-toe	**98.5**	73.5	91.0	83.8	91.9	98.3	91.3	96.5
vote	92.0	94.7	95.2	**96.1**	**96.1**	95.9	91.1	94.0
wine	96.1	87.5	94.3	92.7	97.2	**98.9**	55.1	93.3
Average	77.4	74.4	76.4	76.2	76.6	76.8	49.8	**83.2**

is unable to assign a class; returning the majority or a random class could hide these undesirable cases. In Table 1, we can find the accuracy results, in percent.

Experimental results show that SJEP has low accuracy values in some databases, compared to other classifiers. In those databases, most numerical attributes were discretized into a categorical attribute with a single value, so they were useless for mining patterns. CEPMC has higher accuracies than SJEP in most databases. It also has the highest average accuracy from all tested classifiers.

In order to determine if the differences in accuracy are statistically significant, we performed a pairwise comparison between our classifier and the others. Each cell in Table 2 contains the number of databases where our classifier Win/Lose/Tie to each other classifier. We detect ties using a two-tailed T-Test [17] with significance of 0.05. The pairwise comparison shows that, in the tested databases, CEPMC is more accurate than other understandable classifiers, while being competitive with Nearest Neighbors and Support Vector Machines classifiers.

The model built by CEPMC is very easy to understand in terms of the problem domain, unlike Nearest Neighbors and Support Vector Machines models. Each

Table 2. Pairwise comparison between our classifier and the others. Each cell shows the number of Win/Loss/Tie of CEPMC with respect to the corresponding classifier over the selected 22 databases.

	3NN	7NN	AdaBoost	Bagging	C4.5	RandFor	SVM	SJEP
CEPMC	6/5/11	6/6/10	10/3/9	8/3/11	10/3/9	9/4/9	7/7/8	21/0/1

Table 3. Classifier model built by CEPMC for one of the folds in database Iris

iris-setosa
$[PetalLength \leq 1.90]$
$[PetalWidth \leq 0.60]$
iris-versicolor
$[PetalLength > 1.90] \wedge [PetalLength \leq 4.90] \wedge [PetalWidth \leq 1.60]$
iris-virginica
$[PetalLength > 1.90] \wedge [PetalWidth > 1.60]$
$[PetalLength > 4.90]$

class is described as a collection of discriminative properties, as you can see in the example appearing in Table 3.

6 Conclusions

In this paper, we introduced CEPM, a new algorithm for mining Emerging Patterns. It uses local discretization of numerical values for solving the global discretization drawback of previous emerging pattern classifiers. CEPM extracts patterns from a collection of decision trees, using a special extraction procedure during the tree induction. To obtain a collection of representative patterns, CEPM uses a novel object weighting scheme. Furthermore, this paper proposes an algorithm for accurately estimate the minimal support threshold.

Experimental results show that CEPMC, a classifier based on CEPM, is more accurate than one of the most accurate emerging pattern classifiers in the majority of tested databases. A pairwise comparison reveals that CEPMC is more accurate than other understandable classifiers, and as accurate as Nearest Neighbors and Support Vector Machines, while the model built by CEPMC for classification is easy to understand in terms of the problem domain.

In the future, we will work on speeding up the algorithm to estimate the minimal support threshold, which is the slowest component of the current algorithm.

Acknowledgments

This work is partly supported by the National Council of Science and Technology of México under the project CB-2008-01-106443 and grant 25275.

References

1. Berzal, F., Cubero, J.C., Sánchez, D., Serrano, J.M.: Art: A hybrid classification model. Machine Learning 54, 67–92 (2004)
2. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Englewood Cliffs (1998)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
4. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, United States, pp. 43–52. ACM, New York (1999)
5. Quackenbush, J.: Computational approaches to analysis of dna microarray data. Methods Inf. Med. 45(1), 91–103 (2006)
6. Alhammady, H.: Mining streaming emerging patterns from streaming data. In: IEEE/ACS International Conference on Computer Systems and Applications, Amman, pp. 432–436 (2007)
7. Chen, L., Dong, G.: Masquerader detection using oclep: One-class classification using length statistics of emerging patterns. In: WAIMW 2006: Proceedings of the Seventh International Conference on Web-Age Information Management Workshops, Washington, DC, USA, vol. 5, IEEE Computer Society, Los Alamitos (2006)
8. Fan, H., Ramamohanarao, K.: Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. IEEE Trans. on Knowl. and Data Eng. 18(6), 721–737 (2006)
9. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: 13th Int'l Joint Conf. Artificial Intelligence (IJCAI), pp. 1022–1029 (1993)
10. Bailey, J., Manoukian, T., Ramamohanarao, K.: Fast algorithms for mining emerging patterns. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 39–208. Springer, Heidelberg (2002)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
12. Merz, C., Murphy, P.: Uci repository of machine learning databases. Technical report, University of California at Irvine, Department of Information and Computer Science (1998)
13. Dasarathy, B.D.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
14. Kuncheva, L.I.: Combining Pattern Classifiers. In: Methods and Algorithms. Wiley-Interscience, Hoboken (2004)
15. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
16. Frank, E., Hall, M.A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H.: Weka: A machine learning workbench for data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, pp. 1305–1314. Springer, Berlin (2005)
17. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10(7), 1895–1923 (1998)