An Empirical Study of Oversampling and Undersampling Methods for LCMine an Emerging Pattern Based Classifier

Octavio Loyola-González[1,2], Milton García-Borroto[1], Miguel Angel Medina-Pérez[2], José Fco. Martínez-Trinidad[2], Jesús Ariel Carrasco-Ochoa[2], and Guillermo De Ita[3]

[1] Centro de Bioplantas, Universidad de Ciego de Ávila. Carretera a Morón km 9, Ciego de Ávila, Cuba, C.P. 69450
{octavioloyola,mil}@bioplantas.cu
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica. Luis Enrique Erro No. 1, Sta. María Tonanzintla, Puebla, México, C.P. 72840
{migue,fmartine,ariel}@ccc.inaoep.mx
[3] Benemérita Universidad Autónoma de Puebla, Faculty of Computer Science. Av. San Claudio y 14 sur, Puebla, México
deita@cs.buap.mx

Abstract. Classifiers based on emerging patterns are usually more understandable for humans than those based on more complex mathematical models. However, most of the classifiers based on emerging patterns get low accuracy in those problems with imbalanced databases. This problem has been tackled through oversampling or undersampling methods, nevertheless, to the best of our knowledge these methods have not been tested for classifiers based on emerging patterns. Therefore, in this paper, we present an empirical study about the use of oversampling and undersampling methods to improve the accuracy of a classifier based on emerging patterns. We apply the most popular oversampling and undersampling methods over 30 databases from the UCI Repository of Machine Learning. Our experimental results show that using oversampling and undersampling methods significantly improves the accuracy of the classifier for the minority class.

Keywords: supervised classification, emerging patterns, imbalanced databases, oversampling, undersampling.

1 Introduction

Supervised classification is a branch of Pattern Recognition that finds relations between unseen objects and a set of objects previously classified, in order to predict the class of those unseen objects. Due to the high diversity in pattern recognition problems, there is a large collection of techniques (classifiers) to find out these relations. Commonly, for a given problem, the user has to test different classifiers to select the most accurate. Nevertheless, for many learning tasks [12],

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2013, LNCS 7914, pp. 264–273, 2013.

a high accuracy is not the only goal; the result of the classifier should also be understandable by humans [11].

An important family of both understandable and accurate classifiers is the one based on emerging patterns [7]. A pattern is an expression, defined in a language, which describes a collection of objects [9]. An emerging pattern is a pattern that frequently appears in objects of a single class, but it barely appears in objects belonging to other classes. This way, emerging patterns can be used to predict the class of unseen objects. Classifiers based on emerging patterns are valuable tools that have been used to solve real-world problems in fields like Bioinformatics, streaming data analysis, intruder detection, human activity recognition, anomaly detection in network connection data, rare event forecasting, and privacy preserving data mining; among others [12].

Like most classifiers, those based on emerging patterns do not have a good behavior when they are trained with imbalanced datasets, where objects are not equally distributed into the classes, and therefore, classifiers get results which are biased by the class with more objects. These classifiers generate many emerging patterns for the majority class and only a few patterns (or none) for the minority class. This fact leads to low accuracy for the minority class. Imbalanced databases often appear in fields like finance [2], biology and medicine [15].

Currently, applying oversampling or undersampling methods [1,8,4] is the most common approach to deal with databases containing imbalanced classes. However, to the best of our knowledge, there is not any study about the impact of these methods for emerging pattern based classifiers.

In this paper, we present a study of applying oversampling and undersampling methods for an emerging pattern based classifier, over 30 imbalanced databases. We show that the accuracy is significantly improved (according to the Friedman test [6] and the Bergmann-Hommel dynamic post-hoc procedure [10]) for the minority class.

The rest of the paper has the following structure. Section 2 provides a brief introduction to emerging patterns. Section 3 reviews the most popular oversampling and undersampling methods. Section 4 presents the empirical study developed with the methods presented in section 3, it includes a description of the setup, the way we evaluate the results and some concluding remarks that arise from these results. Finally, section 5 provides conclusions and future work.

2 Emerging Patterns

A *pattern* is an expression, defined in a language, which describes a collection of objects. The objects described, or covered, by a pattern are named the pattern *support*. In a supervised classification problem, we say that a pattern is an *emerging pattern* if its support increases significantly from one class to the others [12]. Emerging patterns are usually expressed as combinations of feature values, like $(Eyes = blue, Sex = male, Age = 30)$ or as logical properties, for example $[Eyes = blue] \wedge [Sex = male] \wedge [Age > 30]$.

Extracting emerging patterns from a training sample is a challenge because the number of candidates grows exponentially with respect to the number of features. Moreover, the downward closure property, one of the most effective pruning strategies, does not hold for emerging patterns [12].

In the literature, there are several algorithms for mining emerging patterns. Special attention deserve those algorithms based on decision trees, which usually do not find all the emerging patterns, but obtain a good collection of high quality patterns [12]. In this paper, we use LCMine [11] because it is an efficient algorithm for finding discriminative regularities (patterns) for supervised classification in problems with mixed and incomplete data. LCMine induces diverse decision trees, extracts patterns from these trees, and in a filtering post-processing stage, LCMine finds a reduced set of high quality discriminative properties (emerging patterns) for each class. In [11] the authors propose a classifier (LCMine classifier), which uses several strategies to avoid over-fitting [11].

As far as we know, this is the first paper that studies the use of oversampling and undersampling methods for a classifier based on emerging patterns (LCMine classifier) in order to solve the imbalance in databases.

3 Oversampling and Undersampling Methods

Most supervised classifiers work with databases containing balanced classes. However, there are application domains that contain high imbalance among classes. Imbalanced classes bias the classifiers which tend to classify all objects into the majority class. One way to deal with this problem is applying oversampling and undersampling methods.

Oversampling methods increase the amount of objects in the minority class in order to balance the classes. On the contrary, undersampling methods adjust the class distribution by removing objects from the majority class. In the literature there are also hybrid methods which combine oversampling and undersampling. However, there is no consensus in the literature about what type of method is the best [5].

In this paper, we perform several experiments to study the impact of oversampling and undersampling methods on the LCMine classifier. The following are the methods that we use in our study:

1. Spread Subsample: This undersampling method generates a random subsample of a database. This method adjusts the class distribution through a random elimination of objects from the majority class. This distribution is computed in dependence of a *Spread* value determined by the user. The *Spread* value (a parameter) specifies the maximum ratio between the majority and minority classes.
2. Synthetic Minority Over-sampling Technique (SMOTE) [4]: This is an oversampling method that generates synthetic objects based on the nearest neighbor of each sample in the minority class. Synthetic samples are generated computing the difference between the feature vector (sample) under

consideration and its nearest neighbor, then this difference is multiplied by a random number between 0 and 1, and the result is added to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features values. This approach effectively forces the decision region of the minority class to become more general.

3. SMOTE_NEW: This method is similar to SMOTE but it determines dynamically for each class the percent of new objects to be generated depending on the ratio between classes. Since this percent depends on the ratio between classes, the higher the imbalance, the higher will be the percent to be used. In short, the goal of this algorithm is to achieve uniform distribution among classes, increasing the amount of objects in the minority class but without exceeding the majority class as occurs in SMOTE.
4. Resample: This is a hybrid method that produces a random subsample of the majority class and applies oversampling the minority class in order to obtain uniform class distribution. This method can use sampling with replacement or without replacement. The parameter B specifies the level of balance between classes; values close to one will produce more uniform class distribution.

4 Experimental Results

This section presents the empirical study developed in this paper.

4.1 Experimental Setup

For our experiments, we use 30 databases taken from the UCI Repository of Machine Learning [3]. Table 1 describes the used databases. These databases have different characteristics according to size, class distribution, feature types and percentage of objects with missing values.

Similar to other authors [17,14] we modify the databases *hypothyroid_M*, *page-blocks_M* and *postoperative_M*. In these databases, we merge into a single class (named minority class) those objects belonging to the complement of the majority class. The *iris_M* database is a modification of the original iris database where we join the two classes with higher overlapping.

For each database and each oversampling and undersampling method, we perform 10 fold cross validation averaging the classification accuracy for the minority and majority classes.

For our experiments we use the Friedman test [6] and the Bergmann-Hommel dynamic post-hoc procedure [10] to compare the accuracy results. We also use CD diagrams to show the post-hoc results. In a CD diagram, the top line is the axis where the average rank of the classifiers is plotted, the rightmost classifier is the best classifier, while two classifiers sharing a thick line have statistically similar behavior [6].

We use the implementations of Resample, Spread Subsample and SMOTE taken from Weka [13]. We modify the parameter values, as it is shown in the Table 2, to ensure an uniform distribution of classes.

Table 1. Databases used in the experiments. #Obj: number of objects; Class Distrib: objects per class; #Features: number of features; Missing values: percentage of objects with missing values; Ratio: the ratio between the majority class and its complement.

			# Features			
Database	#Obj	Class Distrib (%)	Numeric	Non-Numeric	Missing values	Ratio
sick	3772	6/94	7	22	5.54%	15.3
hypothyroid_M	3772	8/92	7	22	5.54%	12.0
page-blocks_M	5473	10/90	10	0	-	8.8
wdbc	569	37/63	30	0	-	3.2
haberman	306	26/74	2	1	-	2.8
postoperative_M	90	30/70	0	8	$< 1\%$	2.5
breast-cancer	286	30/70	0	9	$< 1\%$	2.4
credit-g	1000	30/70	7	13	-	2.3
iris_M	150	34/76	4	0	-	2.0
breast-w	699	35/65	9	0	$< 1\%$	1.9
tic-tac-toe	958	35/65	0	9	-	1.9
diabetes	768	35/65	8	0	-	1.9
labor	57	35/65	8	8	35.75%	1.9
ionosphere	351	36/64	34	0	-	1.8
heart-h	294	36/64	6	7	20.46%	1.8
colic	368	37/63	7	15	23.80%	1.7
colic.ORIG	368	37/63	7	20	19.39%	1.7
wpbc	198	24/76	33	0	$< 1\%$	1.7
vote	435	39/61	0	16	5.63%	1.6
spambase	4601	39/61	57	0	-	1.5
shuttle-landing	15	40/60	0	6	28.89%	1.5
liver-disorders	345	42/58	6	0	-	1.4
cylinder-bands	540	43/57	18	21	4.74%	1.4
heart-statlog	270	44/56	13	0	-	1.3
credit-a	690	45/55	6	9	$< 1\%$	1.2
crx	690	45/55	6	9	$< 1\%$	1.2
cleveland	303	46/54	6	7	$< 1\%$	1.2
sonar	208	46/54	60	0	-	1.1
kr-vs-kp	3196	48/52	0	36	-	1.1
mushroom	8124	48/52	0	22	1.39%	1.1

Table 2. Description of the oversampling an undersampling methods and the parameters in our experiments

Path in Weka	Parameters
weka.filters.supervised.instance.Resample	-B 1.0 -S 1 -Z 100.0
weka.filters.supervised.instance.SpreadSubsample	-M 1.2 -X 0.0 -S 1
weka.filters.supervised.instance.SMOTE	-C 0 -K 5 -P 100.0 -S 1

4.2 Accuracy Analysis

In this section, we analyze the global accuracy and the accuracy in the minority and majority classes obtained by oversampling and undersampling methods over the tested databases. We also include the plain results for LCMine.

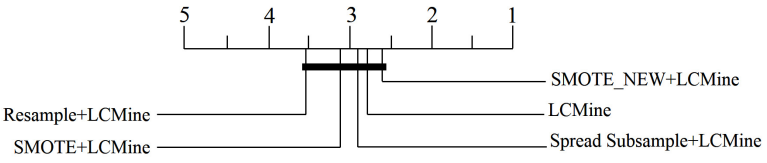

Fig. 1. CD diagram with a statistical comparison of the global accuracy of the LCMine classifier before and after using oversampling and undersampling methods over all the tested databases

Figure 1 shows that the global accuracy of SMOTE_NEW+LCMine classifier is the best. Nevertheless, there is not significant statistical difference among using or not using oversampling or undersampling methods.

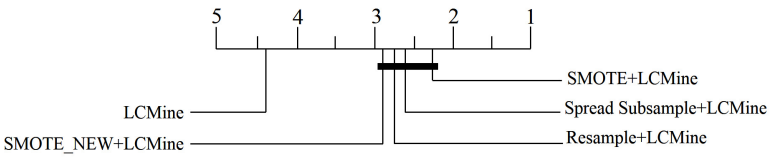

Fig. 2. CD diagram with a statistical comparison of the accuracy in the minority class of the LCMine classifier before and after using the oversampling and undersampling methods over all the tested databases

Figure 2 shows that applying oversampling or undersampling methods improves the accuracy of the LCMine classifier in the minority class. SMOTE+LCMine achieves the best results, nevertheless, notice that there is not statistical significant difference among the different results obtained by oversampling and undersampling methods.

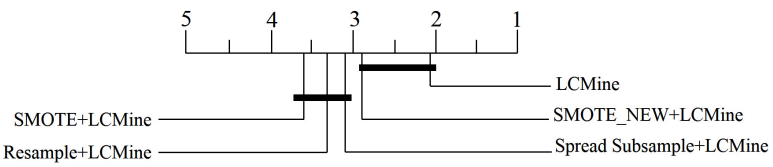

Fig. 3. CD diagram with a statistical comparison of the accuracy in the majority class of the LCMine classifier before and after using the oversampling and undersampling methods over all the tested databases

Figure 3, we can see that LCMine classifier obtained the best results in the majority class. Nevertheless, notice that there is not significant statistical difference between the results of LCMine and SMOTE_NEW+LCMine .

Table 3. Accuracy results of the compared oversampling and undersampling methods on the tested databases. We show the accuracy for the minority (*min*) and majority (*maj*) classes. The best results for each database in the minority and majority classes appear bolded.

Database	Resample		SMOTE NEW		SMOTE		No resampling		Spread Subsample	
	min	*maj*	*min*	*maj*	*min*	*maj*	*min*	*maj*	*min*	*maj*
sick	87.01	73.85	83.98	**75.01**	82.68	71.14	83.55	67.27	**94.37**	72.86
hypothyroid_M	84.88	**88.25**	31.62	68.86	95.53	75.98	86.94	14.22	**98.63**	87.04
page-blocks_M	84.11	**97.44**	54.11	83.72	84.82	91.33	83.21	84.86	**93.93**	95.91
wdbc	**72.34**	64.24	59.57	72.19	48.94	81.46	34.04	**93.38**	70.21	62.91
haberman	58.02	69.78	38.27	79.56	41.98	80.44	28.40	**83.11**	**59.26**	67.11
postoperative_M	26.92	53.13	7.69	65.63	7.69	64.06	3.85	**84.38**	**38.46**	64.06
breast-cancer	**57.65**	65.67	40.00	77.61	41.18	78.61	34.12	**86.57**	56.47	65.17
credit-g	**66.33**	71.29	54.67	84.43	53.00	83.43	41.00	**90.29**	65.67	74.57
iris_M	**100**	99.00	**100**	**100**	**100**	**100**	96.00	**100**	**100**	99.00
breast-w	**93.36**	96.51	92.95	96.51	92.95	96.29	92.53	**97.16**	92.53	96.51
tic-tac-toe	94.88	96.49	**96.08**	99.52	94.88	99.52	92.77	**100**	95.78	99.20
diabetes	74.63	75.00	70.90	76.20	73.13	75.00	59.33	**83.60**	**74.63**	76.60
labor	85.00	62.16	90.00	70.27	**100**	59.46	80.00	**78.38**	90.00	67.57
ionosphere	82.54	96.89	**83.33**	97.78	80.95	96.44	76.98	**99.11**	76.98	97.78
heart-h	86.79	69.15	86.79	62.23	**87.74**	52.13	76.42	**81.38**	84.91	70.21
colic	71.32	88.79	77.21	86.21	**80.15**	82.33	72.06	**90.95**	74.26	87.93
colic.ORIG	75.74	87.93	**76.47**	86.64	72.79	86.64	69.12	**92.24**	75.74	88.79
wpbc	91.98	96.92	**93.87**	96.08	93.40	95.24	91.51	**97.48**	93.40	96.36
vote	**94.05**	92.88	91.07	**94.01**	92.86	**94.01**	89.88	94.01	91.67	93.26
spambase	**93.33**	**90.32**	78.27	83.21	47.10	83.39	92.83	78.08	91.06	81.71
shuttle-landing	0.00	77.78	0.00	**100**	**50.00**	77.78	0.00	**100**	0.00	**100**
liver-disorders	60.69	78.00	60.69	78.50	**68.28**	68.00	60.00	**80.50**	62.76	76.00
cylinder-bands	44.30	78.53	49.56	78.53	**54.39**	73.08	32.46	**85.58**	42.11	80.77
heart-statlog	77.50	84.67	74.17	**87.33**	**80.00**	84.67	76.67	84.00	77.50	85.33
credit-a	83.71	85.38	87.95	84.07	**88.60**	**85.90**	85.34	85.12	86.64	85.12
crx	85.34	**84.60**	86.64	83.81	**87.30**	83.81	84.36	84.33	85.02	78.50
cleveland	78.42	77.44	75.54	**88.41**	**79.14**	81.71	76.98	86.59	77.70	86.59
sonar	61.86	**90.99**	75.26	45.59	**82.47**	71.17	74.23	84.68	74.23	84.68
kr-vs-kp	98.82	99.34	99.41	**99.46**	**99.61**	98.74	99.41	**99.46**	99.41	**99.46**
mushroom	99.13	**100**	99.80	**100**	**100**	**100**	99.18	**100**	99.18	**100**
Average	75.69	83.08	70.53	83.38	75.38	82.39	69.11	**86.22**	**77.42**	84.03

In the Table 3 we can see the global accuracy. This table shows that, in most of the databases, the use of oversampling and undersampling methods jointly with LCMine achieved the best average accuracy in the minority class (*min*). In this table we can also see that Spread Subsample gets the best average accuracy in the minority class; nevertheless, the original LCMine classifier gets the best average accuracy in the majority class (*maj*).

4.3 Accuracy in the Minority Class Regarding the Imbalance Ratio

In this section, we show the accuracy results in the minority class regarding the imbalance ratio. For this analysis we divide the databases in two groups depending if their imbalance ratio is lower than 2 or greater than or equal to 2. The goal of this experiment is to show the behavior of oversampling and undersampling methods with respect to imbalance ratio.

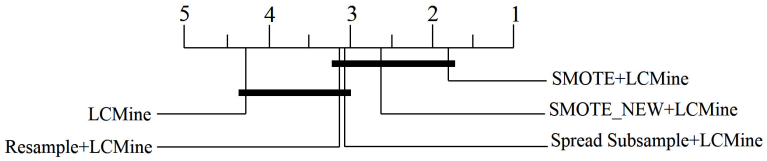

Fig. 4. CD diagram with a statistical comparison of the LCMine classifier before and after using the oversampling and undersampling methods over databases with imbalance ratio lower than 2 (see Table 1)

Figure 4 shows that, when the imbalance ratio is lower than 2, there is not statistical significant difference among the results obtained by LCMiner and those results obtained by Resample+LCMine and Spread Subsample+LCMine. While, applying SMOTE or SMOTE_NEW jointly with LCMine is statistically better than using the original LCMine classifier.

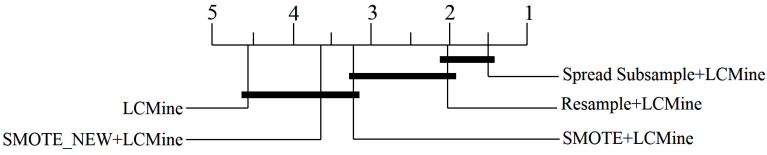

Fig. 5. CD diagram with a statistical comparison of the LCMine classifier before and after using the resampling methods over databases with imbalance ratio greater than or equal to 2 (see Table 1)

Figure 5 shows that, when the imbalance ratio is greater than or equal to 2, there is not statistical significant difference among the results obtained by the original LCMine classifier and the results obtained by SMOTE+LCMine and SMOTE_NEW+LCMine. While, applying Resample or Spread Subsample jointly with LCMine is statistically better than using the original LCMine classifier.

4.4 General Concluding Remarks

The results shown in the previous section lead us to conclude that oversampling and undersampling methods improve the accuracy of the LCMine classifier in the minority class without significantly reducing the accuracy in the majority class. Moreover, if the imbalance ratio is lower than 2 then it is better to use SMOTE; else, it is better to use Spread Subsample.

A possible explanation for this behavior is that, when the imbalance ratio is greater than or equal to 2, the oversampling methods create as many false objects as the real objects in the oversampled class. This way, the classifier cannot correctly classify new objects in the minority class if its knowledge in this class is 50% artificial.

5 Conclusions and Future Work

The classifiers based on emerging patterns are sensitive to databases containing imbalanced classes. These classifiers generate many emerging patterns for the majority class and only a few patterns (or none) for the minority class. This fact affects this type of classifiers, leading them to obtain low accuracy for the minority class.

The main contribution of this paper is an empirical study of the behavior of a classifier based on emerging patterns when using oversampling and undersampling methods in databases containing imbalanced classes. The experimental results show that there is not a method which clearly outperforms the others, but applying any oversampling or undersampling method improves the LCMine classifier accuracy.

From our experimental study we can conclude that if the classes in the database have an imbalance ratio greater than or equal to 2 (1:2) the best option is to use undersampling methods; otherwise, if the ratio is lower than 2 the best option is to use an oversampling method.

As future work, we plan to build a cascade classifier, based on emerging patterns, capable of accurately classifying imbalanced databases with more than two classes. The idea is to apply an oversampling or undersampling method in the complement of the majority class, and then applying this procedure recursively for the other classes, in order to build a cascade classifier. On the other hand, since some studies in the literature propose decision trees that are robust to imbalanced databases [16]; another future work would be studying how to modify the LCMine classifier following these ideas in order to improve its results for imbalanced databases without using oversampling or undersampling methods.

Acknowledgment. This work was partly supported by the National Council of Science and Technology of Mexico under the project grants CB2008-106443 and CB2008-106366.

References

1. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explor. Newsl. 6(1), 20–29 (2004)
2. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: A comparative study. Decision Support Systems 50(3), 602–613 (2011)
3. Blake, C., Merz, C.J.: {UCI} Repository of machine learning databases. Tech. rep., University of California, Irvine, School of Information and Computer Sciences (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16(1), 321–357 (2002)
5. Chawla, N.: Data Mining for Imbalanced Datasets: An Overview. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 875–886. Springer, US (2010)
6. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 1–30 (2006)
7. Dong, G.: Preliminaries. In: Dong, G., Bailey, J. (eds.) Contrast Data Mining: Concepts, Algorithms, and Applications, ch. 1. Data Mining and Knowledge Discovery Series, pp. 3–12. Chapman & Hall/CRC, United States of America (2012)
8. Estabrooks, A., Jo, T., Japkowicz, N.: A Multiple Resampling Method For Learning From Imbalanced Data Sets. Computational Intelligence 20(1), 18–36 (2004)
9. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge Discovery in Databases: An Overview. AI Magazine 13(3), 57–70 (1992)
10. García, S., Herrera, F.: An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. Journal of Machine Learning Research 9, 2677–2694 (2008)
11. García-Borroto, M., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Medina-Pérez, M.A., Ruiz-Shulcloper, J.: LCMine: An efficient algorithm for mining discriminative regularities and its application in supervised classification. Pattern Recognition 43(9), 3025–3034 (2010)
12. García-Borroto, M., Martínez-Trinidad, J., Carrasco-Ochoa, J.: A survey of emerging patterns for supervised classification. Artificial Intelligence Review 1–17 (2012)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations 11(1), 10–18 (2009)
14. Lenca, P., Lallich, S., Do, T.-N., Pham, N.-K.: A comparison of different off-centered entropies to deal with class imbalance for decision trees. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 634–643. Springer, Heidelberg (2008)
15. Li, D.C., Liu, C.W., Hu, S.C.: A learning method for the class imbalance problem with medical data sets. Computers in Biology and Medicine 40(5), 509–518 (2010)
16. Liu, W., Chawla, S., Cieslak, D.A., Chawla, N.V.: A Robust Decision Tree Algorithm for Imbalanced Data Sets. In: SDM 2010, pp. 766–777 (2010)
17. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: A Study with Class Imbalance and Random Sampling for a Decision Tree Learning System. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice II, vol. 276, pp. 131–140. Springer, Boston (2008)