A New Performance Evaluation Method for Two-Class Imbalanced Problems

Vicente García[1,2], Ramón A. Mollineda[2], and J. Salvador Sánchez[2]

[1] Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca
Av. Tecnológico s/n, 52140 Metepec (México)
[2] Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)

Abstract. In this paper, we introduce a new approach to evaluate and visualize the classifier performance in two-class imbalanced domains. This method defines a two-dimensional space by combining the geometric mean of class accuracies and a new metric that gives an indication of how balanced they are. A given point in this space represents a certain trade-off between those two measures, which will be expressed as a trapezoidal function. Besides, this evaluation function has the interesting property that it allows to emphasize the correct predictions on the minority class, which is often considered as the most important class. Experiments demonstrate the consistency and validity of the evaluation method here proposed.

Keywords: Imbalance, performance measure, learning.

1 Introduction

In the last years, the class imbalance problem has received considerable attention in areas such as Machine Learning and Pattern Recognition. A two-class data set is said to be imbalanced when one of the classes (the minority one) is heavily under-represented in comparison to the other class (the majority one). This issue is particularly important in real-world applications where it is costly to misclassify examples from the minority class, such as diagnosis of rare diseases, detection of fraudulent telephone calls, text categorization, information retrieval and filtering tasks. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases, respectively, they are also known as positive and negative examples.

Most research on this topic has traditionally focused on a number of solutions both at the data and algorithmic levels. Data level methods for balancing the classes consists of resampling the original data set, either by over-sampling the minority class or by under-sampling and/or under-sampling the majority class, until the classes are approximately equally represented [2, 4, 10, 14, 23]. Solutions at the algorithmic level consist of adapting existing algorithms and techniques to the particular characteristics of the imbalanced data sets [5, 7, 12, 15, 16, 20].

Recently, some authors have also drawn attention to the fact that there does not exist a direct correlation between class imbalance and the loss of performance [3, 9, 12, 17, 18], suggesting that the class imbalance is not a problem by itself. From this point of view,

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 917–925, 2008.

the degradation of performance is also related to other data characteristics, such as the size of the data set, distribution of the data within each class, small disjuncts, data duplication, density and overlap complexity.

Within the context of class imbalance, a third area to investigate refers to measuring the classifier performance. Common metrics estimate classification accuracy and/or error rates. However, when the prior class probabilities are very different and the error costs are unequal, the use of these measures might produce misleading conclusions since they do not take into account misclassification costs, are strongly biased to favor the majority class, and are sensitive to class skews [19, 11, 6]. Under these circumstances, it can be more appropriate to employ the ROC curve or other alternative metrics, such as the geometric mean (g-mean) or the area under the ROC curve (AUC) [2, 7, 8].

The geometric mean, AUC and other measures give an estimate of the overall classifier performance, but they cannot reflect the contribution of each class to it. In some cases, it can be even more important to know whether the accuracies on each class are balanced and if not, to find out which is the 'dominant class' (the class with the highest accuracy rate). For this purpose we will here suggest a simple metric to reflect the 'dominance' relationship between the majority class and the minority class. After this, we will also introduce a new method to evaluate and visualize the classifier performance in class imbalance problems.

2 Evaluation of Classifier Performance in Imbalanced Domains

Typical metrics for measuring the performance of learning systems are classification accuracy and error rates, which can be easily derived from a 2×2 confusion matrix as that given in Table 1 (for a two-class problem).

Empirical evidence shows that most of these commonly used measures are biased with respect to the data imbalance and proportions of correct and incorrect classifications. Shortcomings of these evaluators have motivated search for new measures, such as the geometric mean of class accuracies or the AUC [2, 7, 8].

Given the True Positive rate, $TPrate = TP/(TP + FN)$, and the True Negative rate, $TNrate = TN/(TN + FP)$, the *geometric mean* of $TPrate$ and $TNrate$ is computed as $\sqrt{TPrate \cdot TNrate}$. This measure can be seen as a sort of correlation between both rates, because a high value occurs when they both are also high, while a low value is related to at least one low rate. An alternative is the ROC space, in which the classifier performance is characterized by a point that represents a trade-off between the $TPrate$ and the False Positive rate, $FPrate = FP/(FP + TN)$ (also known as

Table 1. Confusion matrix for a two-class problem

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

false alarm rate). The performance can be finally computed by means of the AUC, which is a combined measure of $TPrate$ and specificity $(1 - FPrate)$.

These measures are sensitive to skewed distributions of positive and negative instances, but they cannot explain the contribution of each class to it, nor which is the prevalent class. For this reason, in this paper we propose a simple performance metric, here called *dominance*, which will be able to reflect the dominance or prevalence relationship between the positive class and the negative class. The dominance can be computed as follows:

$$Dominance = TPrate - TNrate \tag{1}$$

This measure can take on any value between -1 and $+1$, since both the $TPrate$ and the $TNrate$ are in the range $[0, +1]$. A dominance value of $+1$ represents a situation of perfect accuracy on the positive class, but failing on all negative cases; a value of -1 corresponds to the opposite situation. The closer the dominance is to 0, the more balanced both individual rates. In practice, the dominance can be interpreted as an indicator of how balanced the $TPrate$ and the $TNrate$ are. Obviously, any particular value of dominance could represent very different scenarios (for example, a dominance value of 0 could correspond to an optimal case with $TPrate = TNrate = +1$, but also to a dramatic situation with $TPrate = TNrate = 0$). In order to estimate the performance of a classifier, one should use the dominance together with other measures, such as the geometric mean or the accuracy.

2.1 The Accuracy-Dominance Space

The performance of a classification model on imbalance scenarios could be better evaluated by combining a suitable measure of its overall accuracy (g-mean, AUC or others) and the dominance.

This section introduces the $Accuracy - Dominance$ (AD) space as a tool to visualize and measure the behavior of a classifier from the joint perspective of global accuracy and dominance. Fig. 1 illustrates the AD space as a two-dimensional coordinate system where the X axis corresponds to dominance and the g-mean is plotted on the Y axis. An AD graph depicts relative trade-offs between dominance and g-mean. The performance of a classifier measured as a pair ($dominance$, g-mean) corresponds to a single point in this AD space. The upper central point $(0, +1)$ represents a perfect classifier where $TPrate = TNrate = 1$, while points $(-1, 0)$ and $(+1, 0)$ match with the useless cases of $TPrate = 0, TNrate = 1$ and $TPrate = 1, TNrate = 0$, respectively.

The upper left $(-1, +1)$ and upper right $(+1, +1)$ points correspond to the 'unfeasible cases' because when dominance is -1 or $+1$, one of the two class rates is 0 what makes impossible for the g-mean to achieve values greater than 0. Actually, there is an infinite number of points in the AD graph which represents unfeasible cases. Fig. 1(left) shows the bell-shaped region of probable cases, while the part out of this region corresponds to those unfeasible cases. The rationale behind this effect is that g-mean decreases when the absolute value of dominance increases.

Given a point (d, g) in an AD graph, it would be interesting to quantify the trade-off between dominance and g-mean represented by that point. A plain solution could

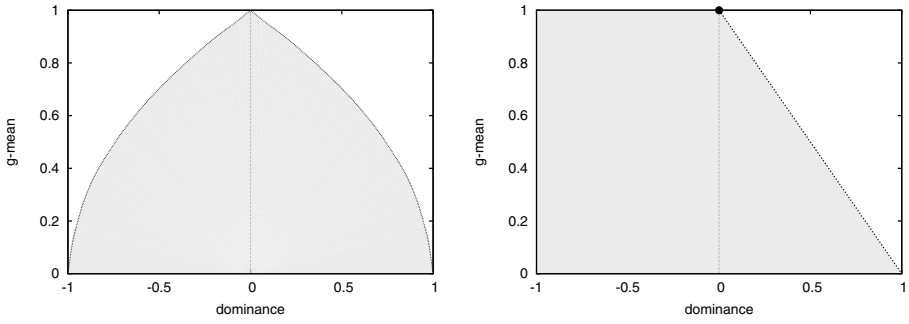

Fig. 1. Accuracy-Dominance Space. Left plot corresponds to the region of feasible values. Right plot represents the optimal point (maximum dominance and maximum g-mean), producing the maximum area.

be to measure the area of the triangle defined by $(-1, 0)$, (d, g) and $(+1, 0)$, which is maximized for $d = 0$, $g = +1$ representing a perfect classification. Nevertheless, the triangular function only depends on the value of the g-mean (the height of a triangle with base equal to 2), thus resulting in a constant area for an infinite number of pairs $(TPrate, TNrate)$ that produce the same g-mean value. Consequently, no information about the prevalence of a class accuracy over the other would be given by the triangular function. Instead, as we are seeking a function dependent on both dominance and g-mean, we will employ the area of a rectangular trapezium whose vertices are at points $(-1, 0)$, $(-1, g)$, (d, g) and $(+1, 0)$ (see Fig. 1(right)).

The area of such a trapezium will conveniently reflect the trade-off between dominance and g-mean. Besides, it is able to capture the fact that the minority class is usually deemed as the class of most importance and with a greater misclassification cost: for a given g-mean, the value of the trapezoidal function increases with the dominance. For two models A and B producing g-means $g_A = g_B$ and dominance values $d_A > d_B$, it can be easily proved that the trapezoidal function of model A will be greater than that of model B.

The trapezoidal function can take on any value between 0 and $+1.5$ (the area of the greatest trapezium, that is, the one corresponding to a point with dominance equal to 0 and g-mean equal to +1, which represents a perfect classification). Informally, one point in AD space is better than another if that is to the northeast (g-mean is higher, dominance is higher, or both) of the second.

3 Experimental Data Sets and Results

A number of experiments have been conducted to illustrate the performance evaluation method introduced in the previous section. Ten real data sets taken from the UCI Machine Learning Database Repository (`http://archive.ics.uci.edu/ml/`) have been employed in the experiments (see a summary in Table 2). All data sets were transformed into two-class problems. For example, in Vehicle database the objects of

Table 2. Data sets used in the experiment

	Positive Examples	Negative Examples	Classes	Majority Class
Breast	81	196	2	1
Ecoli	35	301	8	1,2,3,5,6,7,8
German	300	700	2	1
Glass	17	197	9	1,2,4,5,6,7,8,9
Haberman	81	225	2	1
Phoneme	1586	3818	2	1
Pima	268	500	2	1
Satimage	626	5809	7	1,2,3,5,6,7
Vehicle	212	634	4	2,3,4
Yeast	429	1055	10	1,3,4,5,6,7,8,9,10

classes 2, 3, and 4 were joined to form a unique majority class and the original class 1 was left as the minority class.

For each database, we have estimated the geometric mean, the dominance and the area by 5×10–fold cross–validation: each data set is divided into ten blocks of size $n/10$ (n is the total number of objects), using nine folds as the training set and the remaining block as the test set. This is repeated 5 times. Values in Table 3 correspond to the average results with the nearest neighbor (1-NN) classifier, a support vector machine (SVM) and a multi-layer perceptron (MLP). Each classifier has been applied to the original training set and also to sets whose training partitions have been preprocessed by SMOTE (Synthetic Minority Oversampling Technique) and by random under-sampling. These two techniques have been included in order for checking validity and consistency of the new performance measure: theory dictates that this sort of preprocessing techniques should balance the class distributions and correspondingly, the value of the area for the preprocessed sets should result greater than that for the original training set. All hyper–parameters of the classifiers have been set to the default values given in the Weka toolkit [21].

SMOTE and random under-sampling are two well-known data-driven strategies to tackle the class imbalance problem. These consists of resampling the training data by expanding the minority class (SMOTE) or by shrinking the majority class (under-sampling), thus obtaining an altered class distribution. SMOTE [4] adds artificial examples of the minority class by interpolating existing instances that lie close together. This algorithm first finds the k nearest neighbors belonging to the minority class for each positive example and then, the synthetic examples are generated in the direction of some or all of the nearest neighbors, depending on the amount of over-sampling required. In the case of random under-sampling, instances of the majority class are randomly discarded from the data set until the ratio between the minority and majority classes is at the desired level.

As already mentioned in Section 2, the purpose of the new performance evaluation method is to reflect a certain trade-off between the geometric mean of accuracies and the dominance, but emphasizing the true predictions on the minority class. This means that when comparing two classifiers with equal geometric means, the model with the

Table 3. Results for the three classifiers

	1-NN								
	Original			SMOTE			Random under-sampling		
	g-mean	dominance	area	g-mean	dominance	area	g-mean	dominance	area
Breast	0.59	-0.31	0.79	0.60	-0.24	0.83	0.60	0.05	0.92
Ecoli	0.70	-0.43	0.90	0.83	-0.14	1.19	0.85	0.04	1.30
German	0.59	-0.31	0.79	0.61	-0.16	0.86	0.61	0.02	0.93
Glass	0.55	-0.63	0.65	0.71	-0.32	0.95	0.69	0.12	1.08
Haberman	0.50	-0.44	0.64	0.55	-0.24	0.77	0.55	-0.07	0.81
Phoneme	0.87	-0.14	1.25	0.88	-0.09	1.28	0.87	0.00	1.30
Pima	0.64	-0.27	0.88	0.68	-0.11	0.98	0.68	-0.05	1.00
Satimage	0.82	-0.26	1.13	0.90	-0.03	1.33	0.87	0.06	1.34
Vehicle	0.62	-0.39	0.81	0.66	-0.19	0.94	0.71	-0.01	1.06
Yeast	0.63	-0.34	0.83	0.65	-0.20	0.91	0.66	-0.02	0.98
	SVM								
	Original			SMOTE			Random under-sampling		
	g-mean	dominance	area	g-mean	dominance	area	g-mean	dominance	area
Breast	0.55	-0.52	0.69	0.65	-0.02	0.97	0.64	-0.04	0.95
Ecoli	0.00	-1.00	0.00	0.88	0.04	1.33	0.85	0.19	1.35
German	0.65	-0.41	0.85	0.72	0.01	1.09	0.71	0.02	1.07
Glass	0.00	-1.00	0.00	0.55	0.60	1.00	0.55	0.59	0.98
Haberman	0.00	-1.00	0.00	0.56	-0.56	0.69	0.56	-0.49	0.70
Phoneme	0.70	-0.28	0.95	0.75	0.10	1.16	0.75	0.09	1.15
Pima	0.94	-0.36	1.24	0.74	-0.06	1.09	0.74	-0.07	1.09
Satimage	0.00	-1.00	0.00	0.70	0.43	1.20	0.68	0.47	1.19
Vehicle	0.00	-1.00	0.00	0.74	0.08	1.15	0.72	0.07	1.10
Yeast	0.42	-0.79	0.47	0.71	0.05	1.08	0.71	0.06	1.08
	MLP								
	Original			SMOTE			Random under-sampling		
	g-mean	dominance	area	g-mean	dominance	area	g-mean	dominance	area
Breast	0.56	-0.48	0.70	0.57	-0.30	0.77	0.58	-0.02	0.86
Ecoli	0.83	-0.24	1.14	0.88	-0.09	1.29	0.87	0.05	1.32
German	0.64	-0.31	0.86	0.66	-0.12	0.95	0.67	0.00	1.00
Glass	0.37	-0.83	0.40	0.74	-0.32	1.00	0.75	0.07	1.15
Haberman	0.49	-0.65	0.57	0.63	-0.01	0.95	0.63	-0.03	0.93
Phoneme	0.78	-0.13	1.12	0.81	0.10	1.26	0.81	0.11	1.26
Pima	0.68	-0.31	0.92	0.73	0.03	1.10	0.72	-0.04	1.07
Satimage	0.78	-0.33	1.05	0.86	-0.13	1.24	0.87	0.04	1.32
Vehicle	0.75	-0.29	1.01	0.78	-0.10	1.13	0.78	0.00	1.17
Yeast	0.67	-0.35	0.88	0.71	-0.04	1.05	0.69	-0.03	1.03

highest value of dominance will be selected as the "best", which in practice corresponds to the trapezium with largest area.

Table 3 reports the g-mean, the dominance and the trapezoidal area for the three classifiers in order to illustrate the behavior of the evaluation method here introduced.

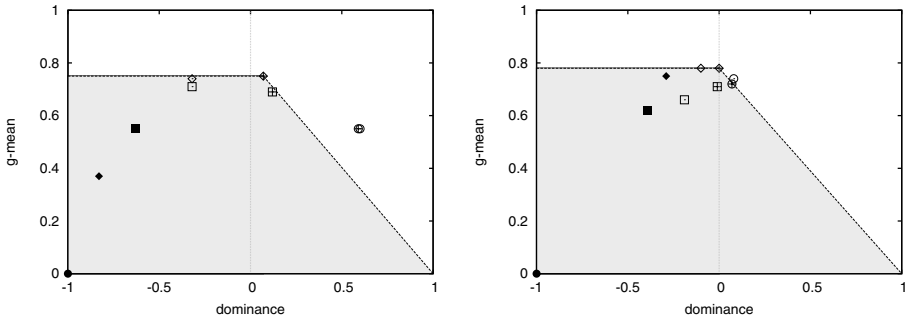

Fig. 2. Scatterplot of the results for Glass (left) and Vehicle (right) databases. Shaded is the area of the best approach. Squares are for 1-NN, circles represent SVM, and diamonds are for MLP. Black figures correspond to the results with the original training set, white are for SMOTE, and crossed figures are for random under-sampling.

The main question is whether the area can be considered as a consistent measure. To answer this question one can observe that in all cases, the area of the trapezium for the sets processed with SMOTE and under-sampling are greater than the value for the original training set.

In order to fully understand the meaning of the trapezoidal function, we can focus on the results for Pima domain. In this case, the geometric mean of both SMOTE and under-sampling with 1-NN is equal to 0.68, making it impossible to consider one technique better than the other. Nevertheless, it is clear that the cost of misclassifications on the minority class (presence of diabetes) is higher than that on the majority class (absence) and therefore, the under-sampling model should be preferable over SMOTE because it has a higher dominance. What is important to note here is that the value of the area for under-sampling (1.00) is also higher than that for SMOTE (0.98). Correspondingly, one could select the best model by looking at the value of the trapezoidal function.

It is also interesting to note the results for Glass database in Table 3 with the 1-NN classifier. In this case, SMOTE obtains a higher geometric mean than the under-sampling technique. By using only this metric, one should select SMOTE as the best model. Nevertheless, the value of the area gives the under-sampling as the best because it also looks at the true positive rate and the true negative rate by means of the dominance. In some way, the evaluation method here proposed prefers some loss in the geometric mean if it produces a higher $TPrate$.

Results on several databases (Ecoli, Glass, Haberman, Satimage, and Vehicle) with the SVM are worth to be mentioned. Here the g-mean for the original training set is equal to 0 and the dominance is -1, thus meaning that the $TPrate = 0$. Obviously, this corresponds to a very undesirable case because all predictions on the minority class are wrong. Such a situation has been conveniently reflected by the trapezoidal function, which is equal to 0. Futhermore, the behavior of the SVM when it ignores the positive instances, it suggests they have been treated as noise [1].

Figure 2 shows a scatterplot of the nine methods (combining the original training set and the two preprocessing techniques with the three classifiers) for Glass and Vehicle

databases. The (greatest) area corresponding to the best pair ($dominance$, g-mean) is also represented. This graphical model allows to visualize in an easy way the behavior of each technique, taking into account a trade-off between the geometric mean and the dominance. For example, in Vehicle (on the right side) one can observe that the best method corresponds to under-sampling with the MLP classifier. Although SMOTE with the MLP obtains the same geometric mean, the dominance is worst (-0.10) than that of under-sampling (0.00).

4 Conclusions

In this paper, we have introduced a new method to evaluate and visualize the performance of learning algorithms in two-class imbalanced domains. This is based on the use of two metrics: the geometric mean of accuracies and the dominance (which is here defined as the difference between the true positive rate and true negative rate). The way in which we have defined the evaluation method allows to stress the importance of the true predictions on the minority class because in many practical problems, the minority class constitutes the class of primary interest. For these situations, the cost of misclassifications on the minority class is higher than that on the majority class.

The experiments carried out have been thought as a way of illustrating the behavior of this new evaluation method. To this end, we have used three classifiers and two well-known resampling techniques to balance the class distributions. We have empirically shown that the computation of the trapezoidal function allows to represent the best model in terms of trade-off between the geometric mean and the dominance. The new evaluation method has been demonstrated to be valid and consistent in all cases.

Future work will be addressed to relate the value of the trapezoidal area with the AUC, trying to complement the particular features of each one. Also, defining a three-dimensional space by the inclusion of some other metric could provide more accurate estimates of the classifier performance, especially in class imbalance domains with unequal misclassification costs.

Acknowledgment

This work has been partially supported by grants DPI2006–15542 and CSD2007–00018 from the Spanish Ministry of Education and Science, GV–2007–105 from Generalitat Valenciana and SEP-2003-C02-44225 from the Mexican CONACyT.

References

1. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Proc. XVth European Conference on Machine Learning (ECML 2004), Pisa, Italy, pp. 39–50 (2004)
2. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36, 849–851 (2003)
3. Batista, G.E., Prati, R.C., Monard, M.C.: Balancing Strategies and Class Overlapping. In: Proc. 6th Intl. Symposium on Intelligent Data Analysis, Madrid, Spain, pp. 24–35 (2005)

4. Chawla, N.V., Bowyer, K.W., Hall, L., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
5. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In: Proc. 7th European Conf. on Principles and Practice of Knowledge Discovery in Databases, Cavtat–Dubrovnik, Croatia, pp. 107–119 (2003)
6. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of classifiers for an uneven class distribution problem. Applied Artificial Intelligence 20, 381–417 (2006)
7. Domingos, P.: Metacost: a general method for making classifiers cost-sensitive. In: Proc. 5th Intl. Conf. on Knowledge Discovery and Data Mining, San Diego, CA, pp. 155–164 (1999)
8. Drummond, C., Holte, R.C.: Explicitly representing expected cost: an alternative to ROC representation. In: Proc. 6th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, Boston, MA, pp. 198–207 (2000)
9. García, V., Mollineda, R.A., Sánchez, J.S., Alejo, R., Sotoca, J.M.: When overlapping unexpectedly alters the class imbalance effects. In: Pattern Recognition and Image Analysis, pp. 499–506 (2007)
10. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Proc. Intl. Conf. on Intelligent Computing, Hefei, China, pp. 878–887 (2005)
11. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. on Knowledge and Data Engineering 17, 299–310 (2005)
12. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6, 40–49 (2002)
13. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explorations 6, 40–49 (2004)
14. Kubat, M., Matwin, S.: Adressing the curse of imbalanced training sets: one-sided selection. In: Proc. 14th Intl. Conf. on Machine Learning, Nashville, TN, pp. 179–186 (1997)
15. Liu, X.Y., Zhou, Z.H.: The influence of class imbalance on cost-sensitive learning: an empirical study. In: Proc. 6th Intl. Conf. on Data Mining, pp. 970–974 (2006)
16. Maloof, M.A.: Learning when data sets are imbalanced and when costs are unequal and unknow. In: ICML–2003 Workshop on Learning from Imbalanced Data Sets II (2003)
17. Okamoto, S., Yugami, N.: Effects of domain characteristics on instance-based learning algorithms. Theoretical Computer Science 298, 207–233 (2003)
18. Prati, R.C., Batista, G.E., Monard, M.C.: Class imbalance versus class overlapping: an analysis of a learning system behavior. In: Proc. 3rd Mexican Intl. Conf. on Artificial Intelligence, Mexico City, Mexico, pp. 312–321 (2004)
19. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. 3rd Intl. Conf. on Knowledge Discovery and Data Mining, Newport Beach, CA, pp. 43–48 (1997)
20. Raskutti, B., Kowalczyk, A.: Extreme rebalancing for svms: a case study. SIGKDD Explorations 6, 60–69 (2004)
21. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
22. Yen, S.H., Lee, Y.S., Lin, C.H., Ying, J.C.: Investigating the effect of sampling methods for imbalanced data distributions. In: Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics, Taipei, Taiwan, pp. 4163–4168 (2006)
23. Zhang, J., Srihari, R.K.: kNN approach to unbalanced data distributions: a case study involving information extraction. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II (2003)