Pattern Recognition Letters 34 (2013) 1146–1151

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Novel classifier scheme for imbalanced problems

Matías Di Martino *, Alicia Fernández [1], Pablo Iturralde [2], Federico Lecumberry [3]

Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

ARTICLE INFO

Article history:
Available online 1 April 2013

Communicated by Eckart Michaelsen

Keywords:
Class imbalance
One class SVM
F-measure
Recall
Precision
Fraud detection

ABSTRACT

There is an increasing interest in the design of classifiers for imbalanced problems due to their relevance in many fields, such as fraud detection and medical diagnosis. In this work we present a new classifier developed specially for imbalanced problems, where maximum F-measure instead of maximum accuracy guide the classifier design. Theoretical basis, algorithm description and real experiments are presented. The algorithm proposed shows suitability and a very good performance in imbalance scenarios and high overlapping between classes.

1. Introduction

In past years a lot of effort has been done to give better solutions to the class imbalance problems (see Sun et al., 2009; García et al., 2007; Guo et al., 2008 and references therein). A two-class data set is said to be imbalanced when the instances of a class (the majority one) heavily outnumber the instances of the other (the minority) class. This problem is particularly important in those applications where it is costly to misclassify samples from the minority class, for example in information retrieval (Manning et al., 2008) and nontechnical losses in power utilities (Di Martino et al., 2012; Muniz et al., 2009; Nagi and Mohamad, 2010).

In a Bayesian decision framework formulation, looking for the optimal decision rule implies to minimize the overall risk, taking into account the different misclassification cost (Duda et al., 2001). In an equal misclassification cost problem we can find the optimal solution, with maximum accuracy, selecting the class that has the maximum a posteriori probability. Finding a decision rule that looks for minimum error rate or maximum accuracy in an imbalanced domain gives solutions strongly biased to favor the majority class (the less important in most applications), getting poor performance.

In almost all the approaches that deal with an imbalanced problem, the idea is to adapt the classifiers that have good accuracy in balanced domains. Many solutions have been proposed to deal with this problem (García et al., 2007; Guo et al., 2008): changing class distributions (Chawla et al., 2002; Chawla et al., 2003; Kolez et al., 2003), incorporating costs[4] in decision making (Batista et al., 2004; Barandela et al., 2003), and using alternative performance metrics instead of accuracy (García et al., 2012) in the learning process with standard algorithms. In López et al. (2012) a comparative analysis of the two former methodologies is done, showing that both have similar performance and that they could be improved by hybrid procedures that combine the best of both methodologies.

In this work we propose a different approach to this problem, designing a classifier based on an optimal decision rule that maximizes the F-measure (van Rijsbergen, 1979) instead of the accuracy. In contrast with common approaches, the proposed algorithm does not need to change original distributions or arbitrarily assign misclassification costs in the algorithm to find an appropriate decision rule.

In Section 2 a theoretical analysis is performed. In Section 3 the proposed classifier is presented. In Section 4 the experimental results are shown and in the last section we share conclusions and future work.

2. Theory

2.1. Class imbalance problem

Identifying rare events is a challenging issue with great impact regarding many problems in pattern recognition and data mining. The main difficulty in finding discriminative rules is that we have to deal with small data sets, with skewed data distributions and

* Corresponding author. Tel.: +598 2711 5444.
E-mail addresses: matiasdm@fing.edu.uy (M.D. Martino), alicia@fing.edu.uy (A. Fernández), iturral@fing.edu.uy (P. Iturralde), fefo@fing.edu.uy (F. Lecumberry).
[1] Tel.: +598 2711 0974.
[2] Tel.: +598 2711 5445.
[3] Tel.: +598 2711 0974.

[4] The misclassification cost can be set by experts or learned (Sun et al., 2009).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.03.012

overlapping classes. In this context a range of classifiers that work successfully for others applications (decision trees, neural networks, support vector machines, etc.) get a poor performance. For example, in a decision tree the pruning criterion is usually the classification error, this can remove branches related with the minority class. In back-propagation neural networks, the expected gradient vector length is proportional to the class size, and so the gradient vector is dominated by the prevalent class and consequently the weights are determined by this class. SVMs are thought to be more robust to the class imbalance problem since they use only a few support vectors to calculate region boundaries. However, in a two class problem, the boundaries are determined by the prevalent class as the algorithm tries to find the largest margin and the minimum error (Sun et al., 2009). A different approach is taken in one-class learning, for example one class SVM, where the model is created based on the samples of only one of the classes. In Raskutti and Kowalczyk (2004) the optimality of one-class SVMs over two-class SVM classifiers is demonstrated for some important imbalance problems.

Evaluation measures have a crucial role in classifier design. Most of the previously seen classifiers use accuracy (or minimum error) measure, getting results that could be meaningless if the rare events (minority class) are the relevant samples. For example in a problem where the rare samples represent 1% of the training data set, using accuracy as an objective measure allows for a 99% accuracy with a decision rule that always chooses the majority class.

Assuming there are two classes, one called the **negative** class ω_-, representing the majority class, usually associated to the normal scenario, and the other called the **positive** class ω_+ representing the minority class (with very few training samples but high identification importance). We define $\Omega = \{\omega_+, \omega_-\}$ as the set of possible classes, being TP (true positive) the number of $x \in \omega_+$ correctly classified (in other words, TP is the number of samples that belong to the positive class classified as positive), TN (true negative) the number of $x \in \omega_-$ correctly classified, FP (false positive) and FN (false negative) the number of $x \in \omega_-$ and $x \in \omega_+$ misclassified respectively. Let us also recall some related well know definitions:

$$\text{Accuracy}: \quad \mathcal{A} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$\text{Recall}: \quad \mathcal{R} = \frac{TP}{TP + FN}, \tag{2}$$

$$\text{Precision}: \quad \mathcal{P} = \frac{TP}{TP + FP}, \tag{3}$$

$$\text{F} - \text{measure}: \quad F_m = \frac{(1+\beta^2)\mathcal{R}\mathcal{P}}{\beta^2\mathcal{P} + \mathcal{R}}. \tag{4}$$

Precision and Recall are two important measures to evaluate the performance of a given classifier in an unbalance scenario. The Recall measures the True Positive Rate, while the Precision measures the Positive Predictive Value. The F-measure combines them with a positive parameter β. With $\beta = 1$, F_m is the harmonic mean between Recall and Precision, meanwhile with $\beta \gg 1$ or $\beta \ll 1$, the F-measure approaches the Recall or the Precision respectively. A high value of F_m ensures that both Recall and Precision are reasonably high, which is a desirable property since indicates reasonably values of both true positive and false positive rates.The best β value for a specific application depends on which is the adequate relation between Recall and Precision for each particular problem (Manning et al., 2008).

In this paper we propose to use the F-measure instead of the Accuracy to guide the classifier design.

2.1.1. F-measure optimum threshold determination

Eq. (4) can be easily transformed as

$$\frac{1}{F_m} = \frac{\beta^2\mathcal{P} + \mathcal{R}}{(1+\beta^2)\mathcal{R}\mathcal{P}} = \frac{\beta^2 \frac{1}{\mathcal{R}} + \frac{1}{\mathcal{P}}}{1+\beta^2} = 1 + \frac{\beta^2 FN + FP}{(1+\beta^2)TP}.$$

Therefore, the problem of maximizing the F-measure is equivalent to minimizing the following equation

$$\varepsilon = \frac{\beta^2 FN + FP}{TP}. \tag{5}$$

Defining the region R_+ as the portion of the feature space where samples are labeled as positive and $R_- = R_+^C$ where samples are labeled as negative (or normal). Given this, Eq. (5) can be written as

$$\varepsilon = \frac{\beta^2 \int_{R_-} p(\omega_+|x)p(x)dx + \int_{R_+} p(\omega_-|x)p(x)dx}{\int_{R_+} p(\omega_+|x)p(x)dx}. \tag{6}$$

Considering the special case of a one dimensional feature descriptor $x \in \mathbb{R}$ and that both regions can be separated by a single threshold λ, the regions R_+ and R_- can be written as $R_+ = [\lambda, +\infty)$ and $R_- = (-\infty, \lambda)$, and Eq. (6) becomes

$$\varepsilon(\lambda) = \frac{\beta^2 \int_{-\infty}^{\lambda} p(\omega_+|x)p(x)dx + \int_{\lambda}^{+\infty} p(\omega_-|x)p(x)dx}{\int_{\lambda}^{+\infty} p(\omega_+|x)p(x)dx}. \tag{7}$$

Being λ^* the minimizer of Eq. (7), it should verify

$$\begin{aligned} &p(\lambda^*|\omega_-)\int_{\lambda^*}^{+\infty} p(x|\omega_+)dx + p(\lambda^*|\omega_+)\int_{-\infty}^{\lambda^*} p(x|\omega_-)dx \\ &- p(\lambda^*|\omega_+)\left[1 + \beta^2\frac{p(\omega_+)}{p(\omega_-)}\right] \\ &= 0. \end{aligned} \tag{8}$$

This is a necessary but not sufficient condition for the existence of an optimal threshold in the F-measure sense. Other points might satisfy this condition, for example a minimum or an inflexion point. In these cases a consistent strategy for label assignation should be taken. This is further explained in Section 3 where we introduce the classification algorithm based in the F-measure approach.

Eq. (8) involves two kind of terms, one kind evaluates the probability in selected points and the other considers the integrals of the probabilities. Therefore, in order to find the optimal threshold, the decision rule takes into account local and global properties of

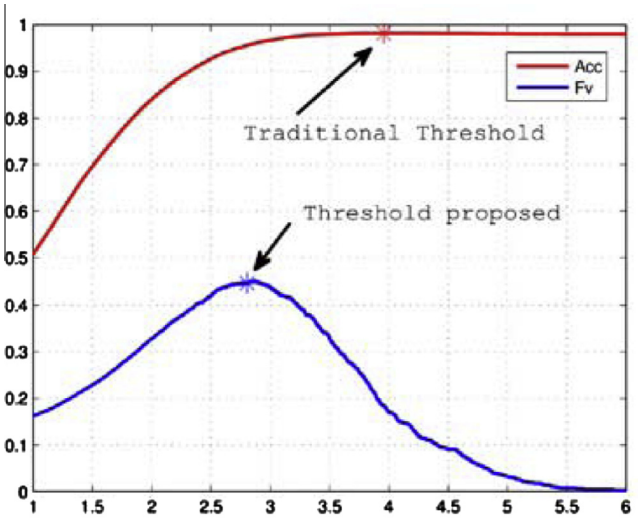

Fig. 1. F-measure and Accuracy obtained for different values of λ. The marked points are the optimal thresholds for each case.

	Labeled as:	
	Positive	Negative
Positive	156	844
Negative	125	49875

Fig. 2. Confusion matrix for the value of λ that maximizes accuracy.

	Labeled as:	
	Positive	Negative
Positive	573	427
Negative	1868	48132

Fig. 3. Confusion matrix for the value of λ that verifies (8).

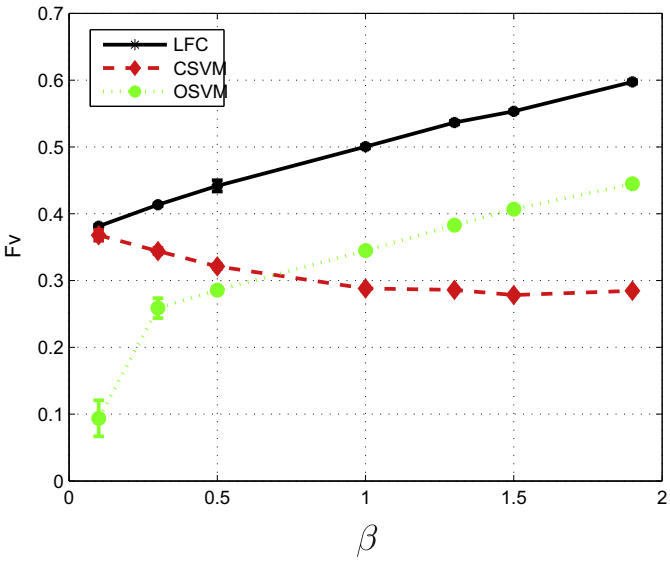

Fig. 5. F-measure obtained with LFC (solid black line), one class SVM (dotted green line) and cost sensitive SVM (dashed red line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the data. Besides this condition was deduced assuming one threshold between positive and negative classes, all the points that verify it are candidates of local optima and should be considered in the design of a classifier, as is presented in Section 3.

2.1.2. One dimensional example

We first evaluate this result with a simplified example though illustrative, with two one dimensional Gaussian distribution with same variance but different mean value. The negative class is centered in $x = 1$ while the positive class is centered in $x = 3$, with 50,000 and 1000 samples drawn from each distribution respectively. For a given threshold λ defining the R_- and R_+ regions, we evaluate the Accuracy and F-measure obtained. These values are plotted in Fig. 1 for $\lambda \in [1, 6]$. The best performance (in the F-measure sense) is obtained for $\lambda = \lambda^*$ given by Eq. (8), while the best Accuracy is obtained by the traditional threshold obtained by minimizing the overall error (Duda et al., 2001) (see Figs. 2 and 3).

3. Proposed classifier

Based on the previous development we proposed a general algorithm that, under the hypothesis assumed in the previous section, follows the optimal F-measure classification. Despite that, the proposed algorithm runs in more general problems (multidimensional spaces and allowing multi-modal distributions) and empirically shows to be robust and with very reasonable performance.

3.1. Description of the classifier training algorithm

The main steps for the classifier training are:

1. Estimate the probability density functions for each class.
2. For each dimension, taking the marginal distribution, find the set of points (region's thresholds) verifying the *optimality condition* (Eq. 8).
3. Assign a label (positive or negative) for each region of the space obtained in the previous step.

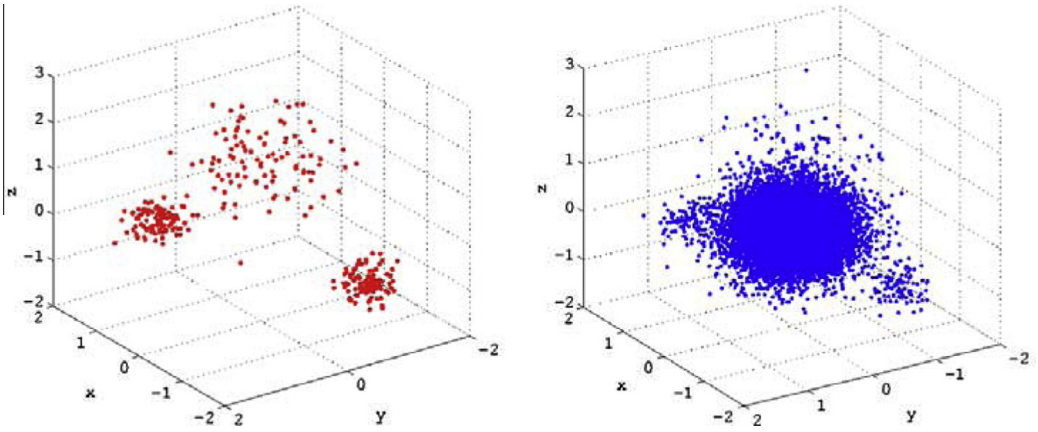

Fig. 4. Data used for experimental validation. Left: samples of the positive class. Right: samples of the negative class. Note that samples of the negative and positive classes are plot separately for a better visualization of each distribution, besides that, classes are highly overlapped.

3.1.1. Probability estimation

In this work, the probability density functions are estimated by convolving the training instances of each class with a Gaussian kernel. Any other density estimation algorithm could be used for the task without changes to the subsequent steps of the classifier training algorithm.

3.1.2. Find a suitable partition of the space

A rectangular partition of the space is found by considering independently probability distributions in each dimension. For each of these marginal distributions, the thresholds that verifies Eq. (8) are used as partition boundaries in that dimension. Following this procedure in all the dimensions, one at a time, a set of hyper-rectangles are defined. The idea is similar to the procedure followed when constructing a tree, where the n-dimensional space is sequentially divided one dimension at a time. The resulting frontiers are piecewise linear, like a tree does. If the feature are correlated we can make a feature extraction previous step to get a more adequate space.

3.1.3. Assign a label for each region

As we mention in Section 2, the optimality condition is necessary but not sufficient, also minima and inflexion points will verify Eq. (8). The existence of a minimum instead of a maximum implies that we must label the space in the opposite way, as it is easy to see that if labeling $R_+ = [\lambda, +\infty)$ and $R_- = (-\infty, \lambda]$ provides a minimum F-measure, the same λ with $R_- = [\lambda, +\infty)$ and $R_+ = (-\infty, \lambda]$ provides a maximum F-measure. Finally if the value of λ solving Eq. (8) occurs in an inflexion point we must assign the same label to both sides of the threshold.

Taking the previous comments into account, the last step is to assign labels for each given region in the partition. In this first version of the algorithm we implement a simple way to assign the partition labels. First, we calculate the ratio of positive samples in the training set (r_0), then for each region R_i we calculate this ratio (r_{R_i}). Then if $r_{R_i} > r_0$ we assign to that hyper-rectangles the positive label, and the negative label otherwise.

3.2. Considerations on the resulting classifier

In Section 2 the optimality condition was presented in order to find a single threshold for a one dimensional problem with two regions, assuming that just one partition would be performed. It is easy to see that in these conditions the proposed algorithm would result in an optimal classifier regarding the F-measure. In more general conditions, for example when multidimensional features are considered or when there are several thresholds per dimension (which is the case when multimodal distributions are considered), the algorithm will result in a classifier that not need to be optimal in the same sense. However, in a more general case, as will be shown in the experimental section, the proposed algorithm

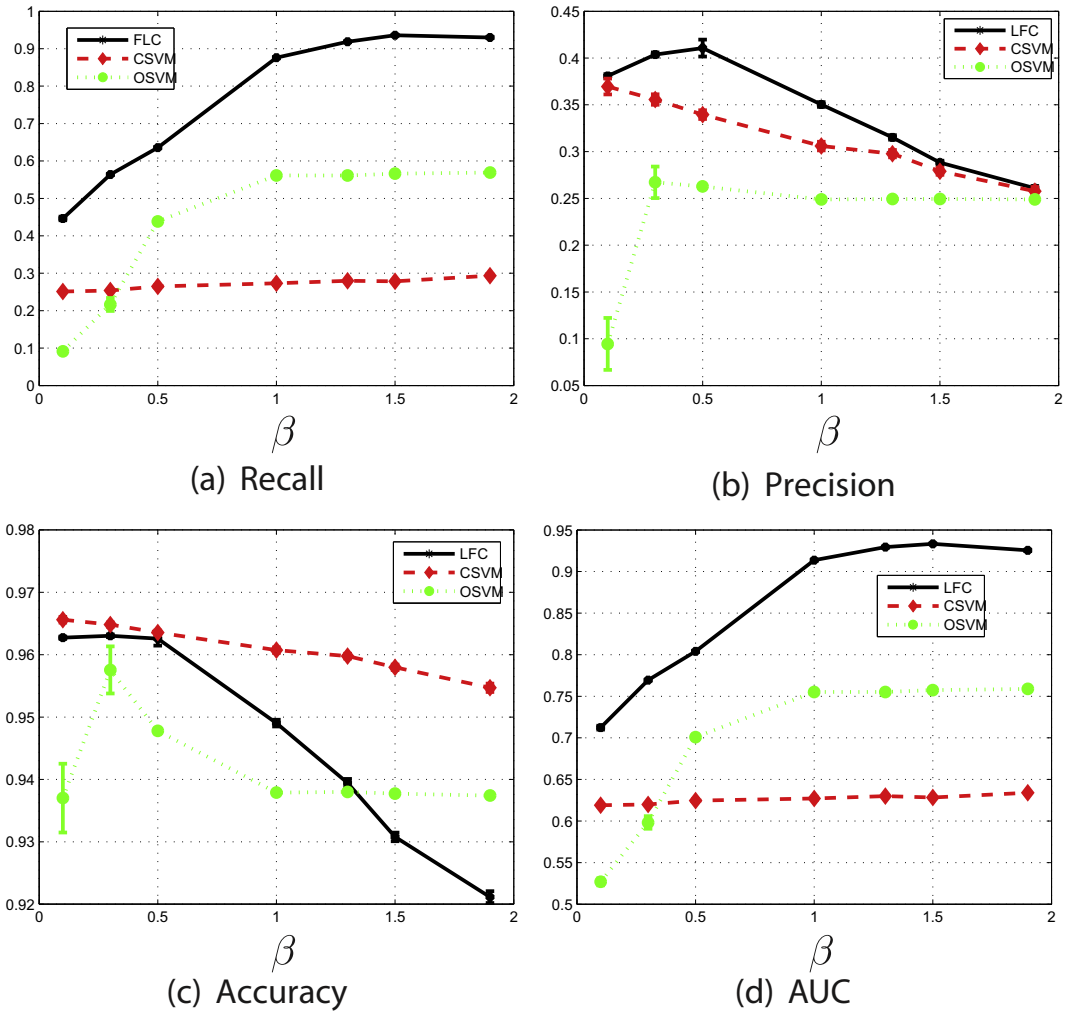

Fig. 6. Other performance measurements.

determines a set of thresholds that still obtain a good performance in the classification.

At this point we need to stress our consideration that several techniques are proposed in the literature to get more adequate feature space (PCA, ICA, Kernel PCA, among others), however, all these methods can be considered as pre-processing techniques that use the base classifiers as black boxes. The main point in this paper is to show that the algorithm is suitable for imbalance problems and to compare base classifiers' performance by themselves.

4. Experimental results and performance analysis

4.1. Data description

For the experimental validation, we arbitrary used 3 different datasets. Dataset 1 consist on 3D samples with 10,000 samples of negative instances and 300 of positive instances. Fig. 4 plots samples of the positive (left plot) and negative (right plot) samples. In both plots the same axes were used, as we can see both classes present multi-modal distributions and they are highly overlapped and unbalanced.

Dataset 2 (Haberman, 1976) and Dataset 3 (Skin segmentation data Bhatt and Dhall, 2010) are public available datasets from the UCI Machine Learning Repository. Both datasets represent real imbalanced problems. Base 2 presents also high overlapping between negative and positive classes.

Database (Haberman, 1976) (Dataset 2) was used in Zhang and Street (2002) where cost sensitive approaches were used to deal with unbalance and overlapping between classes.

Skin dataset (Dataset 3) (Bhatt and Dhall, 2010) is collected by randomly sampling B, G, R values from face images of various age groups (young, middle, and old), race groups (white, black, and asian), and genders obtained from FERET database and PAL database. Total learning sample size is 245,057; out of which 50,859 is the skin samples and 194,198 is non-skin samples.

4.2. Results

For performance analysis, we compare the proposed algorithm, from now on called LFC (acronym of linear F-measure classification), with one class SVM because is the most similar algorithm (in the sense that is a classifier specifically designed for imbalanced problems) and cost sensitive SVM (where different weight or cost are assign to different classes in order to compensate the high unbalance present between classes). We use 5-fold cross validation when training the LFC, O-SVM and CS-SVM, for tuning all the parameters (involved in O-SVM and CS-SVM training) we look for those who maximize the F-measure.

As first performance evaluation, we compare algorithms (LFC O-SVM and CS-SVM) using Database 1 and different definitions of the F-measure (different values of β), to see how the performance is affected in different scenarios. Figs. 5 and 6 show the mean value for the performance achieved after running each algorithm 10 times and also the bar lines in each point show the variance of the results over the 10 runs.

Fig. 5 plots the F-measure for different values of β. LFC shows the best performance (in the F-measure sense) for all the values of β tested. Another interesting feature of LFC is how it is capable to find different solutions for the problem, depending on the β value. In each case it gives either better Recall or Precision according to the weight of each one in the F-measure definition, as we can see in Fig. 6 (a) and (b). For completeness we include in Fig. 6(c) and (d) the Accuracy and the area under the ROC curve for each test realized.

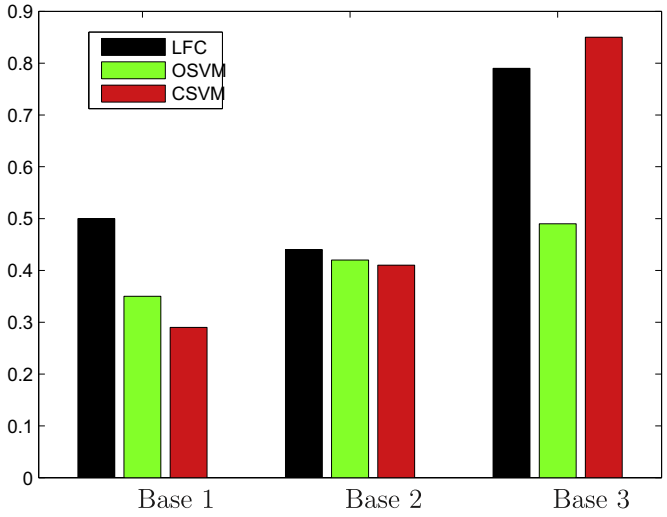

Fig. 7. LFC (first black columns), OSVM (second green columns) and CSVM (third red columns) F-measure comparison for 3 different database. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To conclude this section we present an additional experiments with two UCI Machine Learning Repository datasets (previously describe and labeled as Dataset 2 and 3). Both datasets are highly imbalanced, but differ in the amount of overlapping between classes. These should be considered in the interpretation of the results, as follows.

Fig. 7 presents the F-measure for the LFC, OSVM and CSVM classifiers using databases 1–3 for $\beta = 1$. In all the cases LFC has a good performance. When the overlapping between classes is not important, algorithms that minimize error instead of F-measure can give also good solutions (when classes are separable maximize the Accuracy implies maximize F-measure), in that cases CS-SVM slightly outperformed LFC. Nevertheless, LFC still obtains a better performance than its similar class algorithm, OSVM. In the Haberman datasets, where the overlapping is more important and traditional (minimal error approaches tend to fail) CS-SVM performance degrades, while LFC and OSVM present better results.

From the experimental results we can conclude that LFC has good performance in imbalance scenarios with high overlapping between classes.

5. Conclusions

A new algorithm for classification in imbalanced problems was proposed. A theoretical analysis was presented for determine an optimal decision rule under some simple hypothesis, showing that the proposed algorithm gives the best possible decision rule in those conditions. Based on this result, a more general algorithm was designed, and experimentally show that it performs adequately although it need not be optimal. A comparison with one class and cost sensitive SVM was done, showing that LFC is a suitable algorithm for imbalance scenarios with overlapped distributions.

6. Future work

This work presents several lines of future work but are out of the scope of this article. In particular we are interested in the study of the impact of using Kernel PCA or other method in order to modify the original space, and then apply the algorithm in the new (more suitable) space. Also, we will study how the generalization

could be performed to obtain the optimal hyperplane (in the F-measure sense) in a multidimensional case. We also want to analyze how the performance of the algorithm changes if it is combined with other techniques used for the improvement of *traditional classifiers* in imbalance scenarios, such as smoote, boosting or adaboost. Finally, there is room to improve the way labels are assigned for each partition, for example, using a Genetic Algorithm or another optimization method to find the optimal labels.

Acknowledgments

Authors thanks Dr. Pablo Muse for fruitful discussions and Fernanda Rodriguez for contributions on experiment simulation. Authors also want to thanks anonymous reviewers whose comments where very useful for the improvement of the paper. Work partially supported by ANII and CSIC (Uruguay).

References

Barandela, R., Sánchez, J.S., García, V., Rangel, E., 2003. Strategies for learning in class imbalance problems. Pattern Recogn. 36 (3), 849–851.

Batista, G.E.A.P.A., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. 6 (1), 20–29.

Bhatt, R., Dhall, A., 2010. Skin Segmentation Dataset. UCI Machine Learning Repository.

Chawla, N., Lazarevic, A., Hall, L., 2003. Smoteboost: improving prediction of the minority class in boosting. In: European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 107–119.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. J. Artif. Int. Res. 16 (1), 321–357.

Di Martino, J., Decia, F., Molinelli, J., Fernández, A., 2012. Improving electric fraud detection using class imbalance strategies. In: First International Conference in Pattern Recognition Applications and Methods, vol. 2, pp. 135–141.

Duda, R., Hart, P., Stork, D., 2001. Pattern Classification, second ed. Wiley, New York.

García, V., Sanchez, J., Mollineda, R., Alejo, R., Sotoca, J., 2007. The class imbalance problem in pattern classification and learning. In: Congreso Español de Informtica.

Guo, X., Yin, Y., Dong, C., Yang, G., Zhou, G., 2008. On the class imbalance problem. In: Fourth International Conference on Natural Computation (ICNC08), pp. 192–201.

Haberman, S.J., 1976. Generalized residuals for log-linear models. In: Proceedings of the 9th International Biometric Conference, vol. 1. Biometric Society, Raleigh, NC, pp. 104–122.

Kolez, A., Chowdhury, A., Alspector, J., 2003. Data duplication: an imbalance problem? In: Proceedings of the International Conference on Machine Learning, Workshop on Learning with Imbalanced Data Sets II.

Manning, C., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.

Muniz, C., Vellasco, M.M.B.R., Tanscheit, R., Figueiredo, K., 2009. A neuro-fuzzy system for fraud detection in electricity distribution. In: Proceedings of the Joint 2009 International Fuzzy Systems. Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, pp. 1096–1101.

Nagi, J., Mohamad, M., 2010. Nontechnical loss detection for metered customers in power utility using support vector machines. IEEE Trans. Power Delivery 25 (2).

Raskutti, B., Kowalczyk, A., 2004. Extreme rebalancing for svms: a case study. SIGKDD Explor. 6 (1), 60–69.

Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification if imbalanced data: a review. Int. J. Pattern Recogn. Artif. Intell. 23 (4), 687–719.

van Rijsbergen, C.J., 1979. Information Retrieval. Butterworth.

Vicente García, J.S.S. Mollineda, R.A., 2012. On the suitability of numerical performance measures for class imbalance problems. In: First International Conference in Pattern Recognition Applications and Methods, vol. 2, pp. 310–313.

Victoria López, Alberto Fernández, M.J.D.J., Herrera, F., 2012. Cost sensitive and preprocessing for classification with imbalanced data-sets: similar behavior and potential hybridizations. In: First International Conference in Pattern Recognition Applications and Methods, vol. 1, pp. 98–107.

Zhang, Y., Street, W.N., 2002. Bagging with adaptive costs. Trans. Knowl. Data Eng. 1 (1).