An optimal multiclass classifier design

Marcelo Fiori, Matías Di Martino, and Alicia Fernández
Facultad de Ingeniería - Universidad de la República, Uruguay.

***Abstract*—The use of different evaluation measures for classification tasks is gaining attention, specially for multiclass imbalanced problems. However, the optimization of classifiers with respect to these measures is still heuristic, using ad-hoc rules to classical accuracy-optimized classifiers. We propose a classifier designed specifically to optimize one of the possible measures, namely, the so-called G-mean. Nevertheless, the technique is general, and it can be used to optimize generic evaluation measures. The optimization algorithm to train the classifier is described, and the numerical scheme is tested showing its usability and robustness. The code is publicly available, as well as the datasets used along this paper.**

I. Introduction

Evaluation measures have a crucial role in binary and multiclass classifier analysis and design. There are several proposed measures both for binary classification (Accuracy, Recall, Precision, F-measure, Kappa, ACU [1], Informedness and Markedness [2]) and multiclass classification, like Average G-mean [3], MAUC, Average Accuracy [4] micro and macro Recall, Precision and F-measure [5], [6]. Depending on the problem and field of application one measure could be more suitable than another. While in the Behavioral Sciences, Specificity and Sensitivity are commonly used, in Medical Sciences, ROC analysis is a standard for evaluation. On the other hand, in the Information Retrieval community and fraud detection, Recall, Precision and F-measure are considered appropriate measures for testing effectiveness.

In this sense, obtaining an optimal classifier for a given measure is a very important and challenging problem.

In [7] we proposed a general framework to design an optimal binary classifier which maximizes a selected performance measure. One of the main motivations was to find classifiers adjusted to imbalanced problems, for which the accuracy-based algorithms perform poorly. The main difficulties in finding discriminatory rules for these applications consist on dealing with skewed data distributions and severe overlapping between classes.

Compared with binary classification, the multiclass classification problem is more complex and less studied [8], [9]. In particular, mutliclass imbalance problems pose new challenges that are not observed in two-class problems; for example, it is harder to deal with different misclassification costs, and multiclass also makes the imbalance problem harder [10]. Several solutions are based on considering a multiclass problem in a set of two-class sub-problems, being desirable to develop more general and effective strategies (see for example [10] and references therein)

In [4], the authors provide an experimental analysis to determine the behavior of different approaches: binarization schemes, one versus one and one versus all in order to applied imbalance techniques for binary classification problems, and compare with methods like Adaboost.NC proposed by [10] and others ad hoc methods.

In this work we propose a different approach to this problem, generalizing the framework presented in [7] and [11] to multiclass problems. The result is a multiclass classifier based on an optimal decision rule that maximizes a chosen evaluation measure, in this case the G-mean. We chose this measure because is simple and suitable for imbalance problems [3], [12] but the framework is general and could be extended to other measures as micro or macro F-measure or Average Accuracy. In contrast with common solutions, the proposed algorithm does not need to change original distributions [13]–[15], tune thresholds of classifiers' outputs [16], or arbitrarily assign misclassification costs to find an appropriate decision rule for severe imbalanced problems.

The main contributions of the present paper are:

i) We extend our previous works [11] and [7] for problems with multiple classes, and we show how to find a partition of the feature space guided by the G-mean measure.
ii) We formulate the classification problem using level sets of auxiliary functions which may inspire novel approaches to deal with imbalanced multi-class problems.
iii) We propose a practical implementation of the proposed theory and we evaluate it on synthetic and real data.

The rest of the paper is organized as follows. In Section II the optimal multi-class classifier for the G-measure is proposed, and a numerical scheme is presented. Experimental results are shown in Section III, and we conclude in Section IV.

II. Proposed Classifier Formulation

Let us consider a classification problem with N classes, and let us define $C = \{\omega_i\}_{i=1\dots N}$ as the set of possible classes. Given a classifier, for each individual class ω_i consider the following quantities: TP_i (true positives) which denotes the number of samples $x \in \omega_i$ correctly classified, FP_i (false positives) which denotes the number of $x \notin \omega_i$ classified as belonging to class ω_i, and FN_i (false negatives) which denotes the number of samples in ω_i classified as other class ω_j, $j \neq i$. Different multiclass measures can be defined based on these basic quantities. For instance, G-mean is a suitable measure for imbalance scenarios, since it is the geometric mean of all class accuracies:

$$\text{G-mean} = \left(\prod_{i=1}^{i=N} \frac{TP_i}{TP_i + FN_i} \right)^{1/N} .$$

If Ω denotes the feature domain, a multiclass classifier can be characterized by the regions $\Omega_i \subset \Omega$ such that if $x \in \Omega_i$, then it is labeled as belonging to class ω_i. For the problems we are restricting to, the partition $\{\Omega_1 \ldots \Omega_N\}$ must also satisfy $\bigcup \Omega_i = \Omega$ (all samples have to be labeled) and $\Omega_i \cap \Omega_j = \phi \;\; \forall \; i \neq j$ (it is not allowed to assign more than one label to the same sample).

In order to find the classifier that maximizes a given performance measure, we should be able to express the basic quantities $\{TP_i, FP_i, FN_i\}_{i=1\ldots N}$ in terms of Ω_i.

Specifically, let us suppose that we have estimated the probability density function of each class ω_i ($f_i(x)$). Then we have the following approximation for the number of elements of class ω_i labeled as belonging to class ω_j (from now on A_{ij}):

$$A_{ij} = \int_{\Omega} f_i(x) \mathbb{1}_{\Omega_j}(x) dx, \tag{1}$$

where $\mathbb{1}_{\Omega_j}(x)$ is the Ω_j characteristic function of Ω_j:

$$\mathbb{1}_{\Omega_j}(x) = \begin{cases} 1 & x \in \Omega_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In a general formulation, we have to find the regions Ω_i which minimize a certain cost function $\mathcal{L}(A)$. This is an extremely difficult problem, at least with the present formulation. Let us express the problem in terms of N auxiliary functions $u_k(x) \;\; k = 1 \ldots N$ such that $\mathbb{1}_{\Omega_k} = H(u_k)$, where H is the Heaviside step function. This is, each region Ω_k is determined as the set where the function $u_k(x)$ is positive. This trick is known as the Level Set Method [17], and it allows us to formulate the problem in terms of these functions u_k instead of the regions or boundaries of Ω_k. In this sense, the optimality conditions for $\{u_k(x)\}_k$ to maximize/minimize $\mathcal{L}$ are:

$$\sum_{ij} \frac{\partial \mathcal{L}}{\partial A_{ij}} \frac{\delta A_{ij}}{\delta u_k(x)} = 0 \tag{3}$$

with

$$\frac{\delta A_{ij}}{\delta u_k(x)} = f_i(x) \frac{\partial \mathbb{1}_{\Omega_j}}{\partial u_k}(x) \quad \text{for } k = 1 \ldots N \tag{4}$$

Assuming that the functions $f_i(x)$ are known, the task of finding the optimal classifier consists in finding functions u_k determining regions Ω_i that maximize the chosen measure, in this case the G-mean.

The G-mean measure is defined as the geometric mean of the accuracies $p_k = \frac{TP_k}{TP_k + FN_k}$ of each class; so the (equivalent after a monotone operator) function to maximize is $L_G = \Pi_{k=1}^{N} p_k$. Of course we have to add the constraint that the support of the functions u_k form a partition of the feature space Ω. Let us call $S_{ij} = \int_{\Omega} H(u_i(x))H(u_j(x))$ the overlap between the supports of u_i and u_j. The optimization problem can be then written as:

$$\begin{aligned} &\max_{u_i : i=1\ldots N} L_G \\ &s.t. \; S_{ij} = 0 \quad i \neq j \end{aligned} \tag{5}$$

Note that the condition of $\bigcup_i \Omega_i = \Omega$ is not necessary, since the G-mean can only increase when any of the Ω_i is expanded to an empty portion of the space Ω.

Let us re-write the constrained problem (5) as the following unconstrained problem with increasing λ:

$$\{u_i\}_{i=1\ldots N} = \lim_{\lambda \to \infty} \arg \max_{u_i : i=1\ldots N} L_G - \lambda \sum_{ij} S_{ij} \tag{6}$$

In order to solve this problem, we use a gradient ascent methodology, while increasing λ simultaneously if the condition $\sum_{ij} S_{ij} = 0$ does not hold. This can be seen as a penalty-like optimization approach [18].

The differential of the functional with respect to each u_k is the following:

$$\frac{\delta L}{\delta u_k(x)} = \prod_{i \neq k} (A_{ii}) f_k(x) \delta(u_k(x)) - 2\lambda \sum_{i \neq k} H(u_i(x)) \delta(u_k(x))$$

so the resulting numerical scheme to solve the optimization is:

$$u_k^{n+1} = u_k^n + \delta_t \left[\left(\prod_{j \neq k} \int f_j H(u_j^n) \right) f_k \delta(u_k^n) - 2\lambda^n \sum_{j \neq k} \delta(u_k^n) H(u_j) \right] k$$

$$\lambda^{n+1} = \lambda^n + \delta_t' \left[\sum_{i \neq j} \int H(u_i^n) H(u_j^n) \right]$$

where δ_t and δ_t' are time steps. Both parameters were set to 10^{-2} along the experiments in this paper; higher values can speedup convergence but can also turn the scheme unstable. This iterative algorithm is repeated until convergence (i.e. the difference between $[u_1^n \ldots u_N^n]$ and $[u_1^{n+1} \ldots u_N^{n+1}]$ is small and $S_{ij} = 0$ for $i \neq j$).

In Figure 1, the evolution of $u_{1,2,3}$ is shown for a two dimensional example with three classes. Note that in this example, the data is complex, multi-modal and overlapped with imbalanced classes. In order to visualize the evolution of the functions, we set on Figure 1 (superposed with training samples) in red, green and blue channels $H(u_1^n)$, $H(u_2^n)$ and $H(u_3^n)$ respectively. As we can see, at the beginning there is some overlapping between $\Omega_{1,2,3}$ (so the combination of colors instead of pure red, green or blue can be seen). Then λ is increased along iterations until $\Omega_{1,2,3}$ are disjoint. It is important to pay special attention to the behavior of the algorithm in those areas in which there are several samples from different classes (areas with high overlapping), since these are the more challenging and important portions of the domain (in contrast to those regions where samples are very unlikely).

III. Experimental Results

In this section we present both simulations and experiments with real data, which illustrate the benefits of designing the classifier to maximize the G-mean. In particular, we show that this strategy is better than training a big number of SVM classifiers (varying the parameters for a large grid), and then choosing the best of them in terms of the G-mean.

Since it is very important to observe how the proposed classifier performs for different dataset properties (like the number of features, classes and imbalance ratio), in the following section we run experiments with synthetic data trying to cover a wide range of data characteristics. In Section III-B, we present the results of experiments with two very challenging multi-class datasets, showing the applicability of the proposed technique for real problems.

A. Experiments with synthetic data

For experimental validation we used 16 datasets with different shapes, number of classes and features, overlapping degree, and imbalance ratio between classes. Table I summarizes some of the main characteristics of the used datasets, which are available on-line.[1]

TABLE I
Description of the used datasest. IR stands for *imbalance rate*, defined as the number of samples of the class with more instances divided by the number of samples of the class with less instances

Id	Num. Classes	Num. Features	Num. Samples	IR
22	2	2	$5e+03$	3
23	2	3	$4.1e+03$	2.3
24	2	4	$4.6e+03$	2.7
25	2	5	$3.6e+03$	1.9
32	3	2	$6e+03$	3.8
33	3	3	$6.5e+03$	2.3
34	3	4	$5e+03$	8.2
35	3	5	$4.8e+03$	2
42	4	2	$7.1e+03$	3.8
43	4	3	$9.9e+03$	2.8
44	4	4	$6.2e+03$	8.2
45	4	5	$6.3e+03$	2
52	5	2	$1e+04$	3.8
53	5	3	$1.1e+04$	3
54	5	4	$1e+04$	10
55	5	5	$7.1e+03$	2.9

For these experiments, a classical kernel density estimation technique was used to infer the densities of the different classes [19].

We compare the proposed algorithm, from now on called OMG (acronym for Optimal Multiclass G-mean), with Multiclass RBF-Kernel Supports Vector Machine algorithm (SVM). Parameters for each algorithm were chosen to maximize G-mean (performing 10-fold cross validation over train dataset). Each set was equally split on train and test subsets. The LibSVM [20] library was used for the SVM classification.

Results for accuracy and G-mean are shown in Figures 2 and 3 for training and testing sets. Results were sort by accuracy. Note that for those datasets where high accuracy can be achieved (right part of the charts), results for OMG and SVM approaches are very similar. This can be explained as follows: high accuracy is only possible for problems where different classes present small overlapping, i.e. classes are almost separable. If that is the case, then all the approaches will converge to the decision boundaries which separate the classes.

On the other hand, for those cases in which classes present high overlapping (left part of the charts), finding decision boundaries is more challenging and differences between algorithms become more significant.

B. Experiments with real data

On a second round of experiments, we evaluated the proposed approach with real and publicly available multi-class and imbalanced data [21]. Tables II and III shows the Accuracy, G-mean and the accuracy per class for the Page-Blocks and Yeast databases of the *UCI* machine learning repository. As it can be observed, even for the parameters (cost and γ when we consider a Gaussian kernel) that lead to maximum G-mean, the performance of SVM approaches is outperformed (in the G-mean sense) by the proposed technique. This can be explained as follows: even if we look for those parameters that maximize alternative measures -in this case G-mean- the intrinsic optimization of SVM tries to achieve maximum accuracy. In contrast, the proposed technique is guided to maximize the geometrical mean of each class accuracy and hence we do not need to tune parameters, move to high-dimensional spaces or modify samples distributions to handle imbalanced and multi class databases.

IV. Conclusions and Future Work

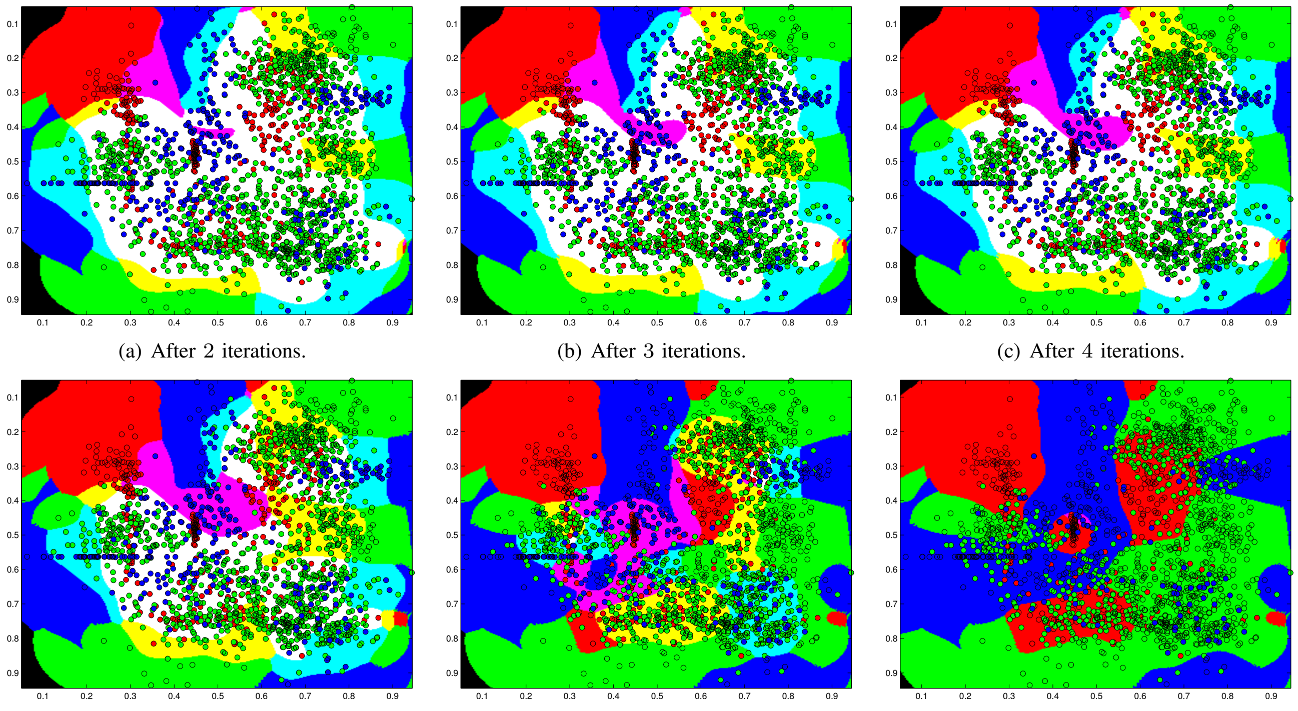
We proposed a framework generalization to multi-class classification problems in imbalanced domains. We presented the optimality conditions for the decision frontier to maximize the G-mean, and a numerical scheme to solve it. The technique is general, in the sense that it can be used to obtain optimal multi-class classifiers with respect to other evaluation measures. The analysis is supported by experimental results, which show the potential of the proposed scheme.

The proposed framework allows to face multi-class imbalance problems with a specific classifier adequate to the problem, which is theoretically simple and direct, in contrast with most of the approaches that combine two-class classifiers with strategies like one versus all or assign ad-hoc weights to classifiers outputs that have a very difficult theoretical interpretation.

Acknowledgments

Work partially supported by ANII and CSIC (Uruguay). The authors want to thank Msc.Eng. Guzman Hernandez for significant contributions to this work and Dr. Pablo Belzarena for fruitful discussions.

[1] *www.fing.edu.uy/˜matiasdm/Archivos/CC/Data.zip*



(a) After 2 iterations.

(b) After 3 iterations.

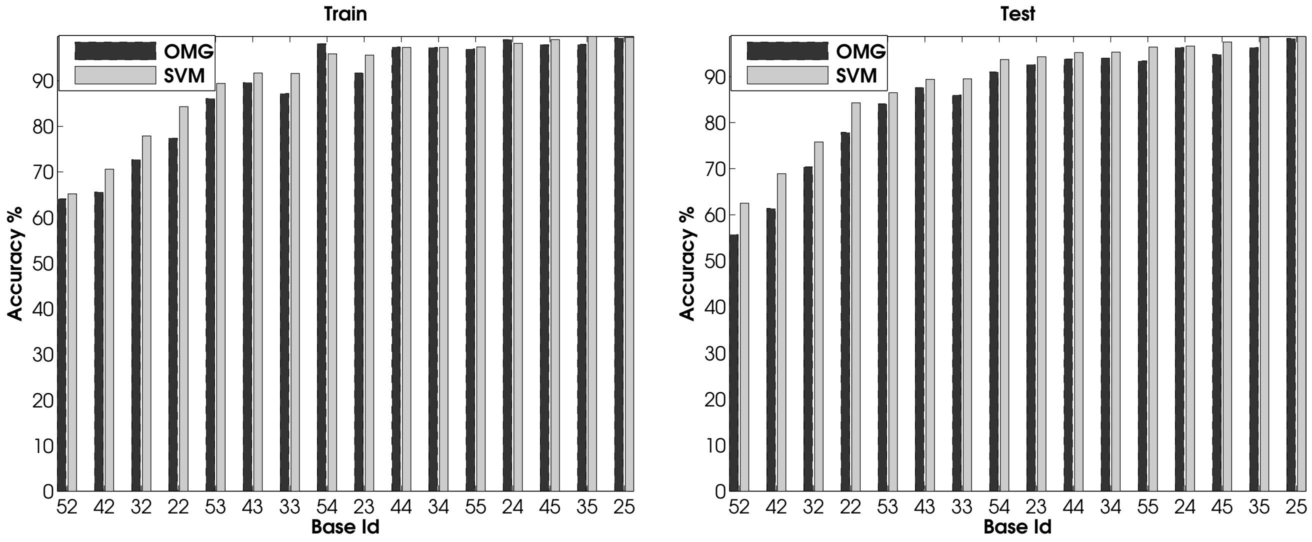
(c) After 4 iterations.

(d) After 10 iterations.

(e) After 100 iterations.

(f) After 300 iterations.

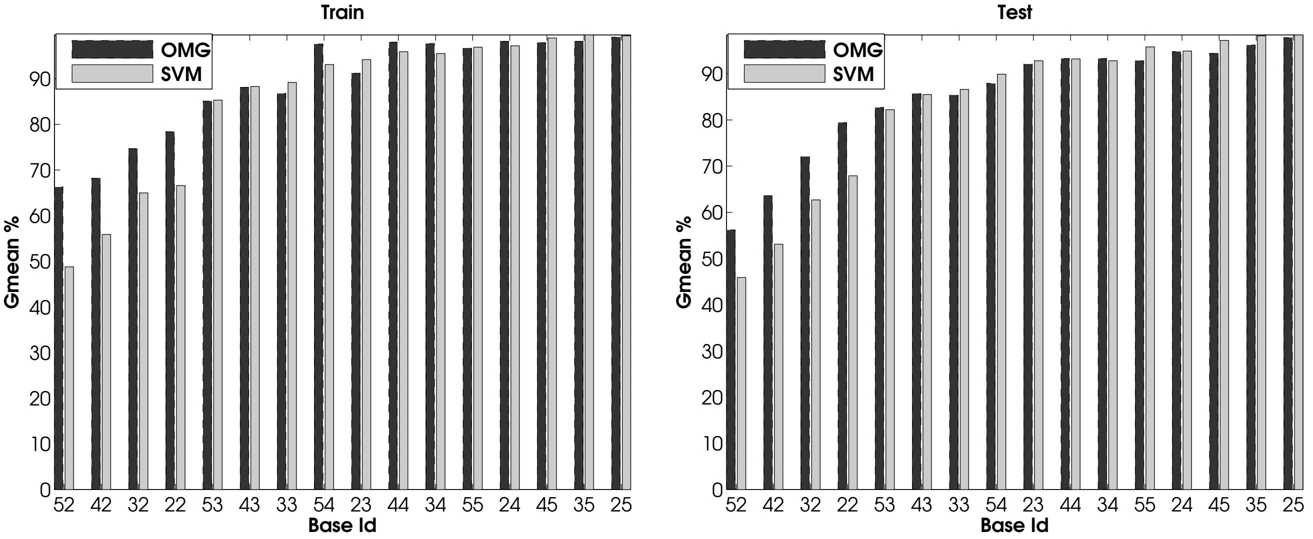
Fig. 1. Evolution of the zero level sets of u_k for $k = 1 \ldots 3$ (decision frontiers).



(a) Accuracy for Training set

(b) Accuracy for Testing set

Fig. 2. Accuracy for OMG (dark-dashed) and SVM (clear-solid) comparison



(a) G-mean for Training set

(b) G-mean for Testing set

Fig. 3. G-mean for OMG (dark-dashed) and SVM (clear-solid) comparison

TABLE II
RESULTS FOR *page-blocks* DATABASE.

Test Set	G-mean (%)	Acc. (%)	A1 (%)	A2 (%)	A3 (%)	A4 (%)	A5 (%)
OMG	**74.2**	79.7	80.9	61.7	63.6	90.2	78.3
SVM	64.4	95.2	99.1	65.3	63.6	73.2	36.7
SVM-RBF	73.5	**96.1**	98.8	79.6	72.7	68.3	55.0
Train Set	G-mean (%)	Acc. (%)	A1 (%)	A2 (%)	A3 (%)	A4 (%)	A5 (%)
OMG	87.3	81.7	81.9	67.9	100	100	90.9
SVM	67.5	95.7	99.4	61.1	58.8	93.6	41.8
SVM-RBF	92.2	98.3	99.4	87.7	100	93.6	81.8

TABLE III
RESULTS FOR *yeast* DATABASE.

Test Set	G-mean (%)	Acc. (%)	A1 (%)	A2 (%)	A3 (%)	A4 (%)	A5 (%)	A6 (%)	A7 (%)	A8 (%)	A9 (%)	A10 (%)
OMG	**53.1**	44.7	50.4	34.6	41.5	85.0	37.0	60.5	68.4	25.0	75.0	100
SVM	0	30.9	0	0	100	0	0	0	0	0	0	0
SVM-RBF	0	**58.8**	54.5	43.4	71.6	85.0	37.0	86.8	52.6	0	25.0	100
Train Set	G-mean (%)	Acc. (%)	A1 (%)	A2 (%)	A3 (%)	A4 (%)	A5 (%)	A6 (%)	A7 (%)	A8 (%)	A9 (%)	A10 (%)
OMG	65.1	50.7	53.7	41.3	42.3	70.8	58.3	64.4	93.8	77.8	75.0	100
SVM	0	30.9	0	0	100	0	0	0	0	0	0	0
SVM-RBF	62.0	69.4	62.6	57.9	76.5	100	62.5	89.7	62.5	16.7	50.0	100

REFERENCES

[1] V. García, J. Sánchez, and R. Mollineda, "On the suitability of numerical performance measures for class imbalance problems," *International Conference in Pattern Recognition Aplications and Methods*, pp. 310–313, 2012.

[2] D. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, pp. 37–63, 2011.

[3] P. Piyaphol, Z. Yanqing, Z. Yichuan, and S. Bismita, "Multiclass SVM with ramp loss for imbalanced data classification," in *IEEE International Conference on Granular Computing (GrC)*, 2012, pp. 376–381.

[4] A. Fernádez, V. López, M. Galar, M. J. del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowledge-Based Systems*, vol. 42, pp. 97 – 110, 2013.

[5] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427 – 437, 2009.

[6] I. Pillai, G. Fumera, and F. Roli, "F-measure optimisation in multi-label classifiers," *Proceedings - International Conference on Pattern Recognition*, pp. 2424–2427, 2012.

[7] M. Di Martino, G. Hernández, M. Fiori, and A. Fernández, "A new framework for optimal classifier design," *Pattern Recognition*, vol. 46, no. 8, pp. 2249–2255, 2013.

[8] G. Ou and Y. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognition*, vol. 40, no. 1, pp. 4–18, 2007, cited By (since 1996)79.

[9] Y. Sun, M. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalances class distribution," *IEEE International Conference on Data Mining*, pp. 592–602, 2006.

[10] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 4, pp. 1119–1130, 2012.

[11] M. Di Martino, A. Fernández, P. Iturralde, and F. Lecumberry, "Novel classifier scheme for imbalanced problems," *Pattern Recognition Letters*, vol. 34, no. 10, pp. 1146–1151, 2013.

[12] P. Phoungphol, Y. Q. Zhang, and Y. Zhao, “Robust multiclass classification for learning from imbalanced biomedical data,” *Special Issue of Tsinghua Science and Technology on Bioinformatics and Computational Biology*, 2012.

[13] M. A. Tahir, J. Kittler, and F. Yan, “Inverse random under sampling for class imbalance problem and its application to multi-label classification,” *Pattern Recognition*, vol. 45, no. 10, pp. 3738 – 3750, 2012.

[14] R. Barandela, J. Sánchez, V. García, and E. Rangel, “Strategies for learning in class imbalance problems,” *Pattern Recognition*, vol. 36, no. 3, pp. 849 – 851, 2003.

[15] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, “Cost-sensitive boosting for classification of imbalanced data,” *Pattern Recognition*, vol. 40, no. 12, pp. 3358 – 3378, 2007.

[16] I. Pillai, G. Fumera, and F. Roli, “Threshold optimisation for multi-label classifiers,” *Pattern Recognition*, vol. 46, no. 7, pp. 2055 – 2065, 2013.

[17] S. Osher and J. A. Sethian, “Fronts propagating with curvature- dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[18] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[19] M. P. Wand and M. C. Jones, *Kernel Smoothing (Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, Dec. 1994.

[20] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[21] K. Bache and M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml