Optimization of Cost Sensitive Models to Improve Prediction of Molecular Functions

Sebastián García-López[1], Jorge Alberto Jaramillo-Garzón[1,2](✉), and German Castellanos-Dominguez[1]

[1] Signal Processing and Recognition Group, Universidad Nacional de Colombia, Campus la Nubia, Km 7 vía al Magdalena, Manizales, Colombia
[2] Grupo de Automática y Electrónica, Instituto Tecnológico Metropolitano, Cll 54A No 30-01, Medellín, Colombia
{sgarcialop,jajaramillog,cgcastellanosd}@unal.edu.co

Abstract. The prediction of unknown protein functions is one of the main concerns at field of computational biology. This fact is reflected specifically in the prediction of molecular functions such as catalytic and binding activities. This, along with the massive amount of information has made that tools based on machine learning techniques have increase their popularity in the last years. However, these tools are confronted to several problems associated to the treated data, one of them is the learning with large imbalance between their categories. There exist several techniques to overcomes the class imbalance, but most of them present many weakness that difficult the obtaining of reliable results. Moreover, models based on cost sensitive learning seems to be a good choice to deal with imbalance data, yet, the obtaining of a optimal cost matrix still remains an open issue. In this paper, a methodology to calculate a optimal cost matrix for models based on cost sensitive learning is proposed. The results show the superiority of this approach compared with several techniques in the state of the art regarding to class imbalance. Tests were applied to prediction of molecular functions in Embryophyta plants.

Keywords: Molecular functions prediction · Proteins · Cuckoo search · Cost sensitive learning · Class imbalance

1 Introduction

Nowadays, modern biology has seen an increasing use of computational techniques for large scale and complex biological data analysis. Several computational machine learning techniques are applied [16]. For example, to classify different types of samples in gene expression of microarrays data [2] or mass spectrometry based proteomics data [1]. In this context, there is a vast number of problems associated with nature of data. In particular, given that same protein can be associated to several functional classes, classification problem with multiple labels arises. A straightforward way to cope with this issue is the "one-against-all" strategy, by which a binary classifier is trained per class to take independent decisions about protein membership. Yet, this approach leads to a high imbalance between sample number per each class, magnifying an already present disparity of their sizes, and thereby producing a large bias towards category having more information [21].

There are several ways to address class imbalance problems, being the most representative sampling, boosting, and cost sensitive strategies. Sampling techniques can be divided into oversampling and subsampling. Former techniques reproduces samples of minority class until they reach the same size as the majority class, but it induces two major problems: (*i*) over-training and (*ii*) noise addition in training set, affecting reliability of protein localization [12]. Although, subsampling eliminates samples of majority class until reaching the same size of category having fewer samples (i.e., minority class), this technique may eliminate useful information if sample selection criteria are not properly selected [12]. Boosting strategies, in turn, are designed to train a set of individually trained classifiers in an iterative way, such that every new classifier emphasizes on incorrectly learned instances by previous trained classifier [7]. Boosting methods, however, are prone to fail if there is not enough data [20] or if training data holds too much noise [6]. Finally, models based on cost sensitive learning assume different costs (or penalties) whenever examples are misclassified. This process is modelled by a cost matrix that is a numerical representation of penalty of misclassifying examples from one category to another. Conventionally, such models assume that costs are fixed; but since this condition is far for being matched in real-world applications it posses as an open problem [18].

In this paper, an efficient methodology of obtaining the optimal cost matrix for a cost sensitive learning model is proposed. This methodology is applied to the prediction of molecular functions in *Embryophyta* plants and is compared with a broad spectrum of class-balance strategies to obtain a comprehensive analysis of the problem. Results show that cost sensitive models are highly reliable and can outperform many commonly used balance strategies in the prediction of molecular functions.

2 Class-Balance Strategies

Generally, class-balance strategies are divided into following explained below strategies: sampling, boosting, and cost sensitive.

2.1 Sampling Strategies

Synthetic Minority Oversampling Technique (`SMOTE`). In this case, new synthetic samples are generated that are addressed to minority class [5]. Further, these samples are computed by interpolation among several closely spaced real samples. In this way, the decision boundary of the minority class becomes more general [11]. Synthetic samples are generated as follows: for each real sample under consideration, represented as a feature vector, distance between it and its nearest neighbors is taken. The result is multiplied by a random number ranging within interval $(0, 1)$ with uniform probability, and this output is added to original feature vector. This procedure causes selection of a random point along the line segment between two neighboring samples.

Subsampling Based on Particle Swarm Optimization (`PSO`). This technique is based on search of an optimal sample subset from a given majority class that maximizes generalization capability of classifier. To this end, Metaheuristic optimization strategy is used. PSO algorithm creates several subsets of majority class and evaluates its classification performance. When completion criterion is accomplished, samples are ranked by their frequency selection. After the frequency listing of selected samples is obtained, a balanced dataset is constructed by combination of samples, which belong to majority class with major frequency indexes, and minority class [24], as summarized in Fig. 1.

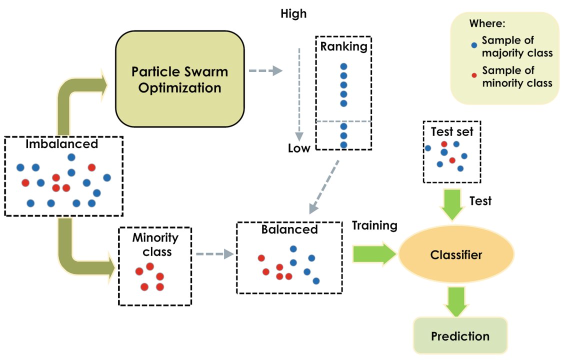


Fig. 1. Schematic representation of PSO subsampling based on algorithm [24].

2.2 Boosting Strategies

Boosting Algorithm (`AdaBoost`). Boosting algorithms place iteratively different weights on training data at each iteration, in such a way that boosting increases those weights associated to incorrectly classified examples, but decreases weights related to correctly classified examples. Thus, training system is forced to focus on rare items. AdaBoost is the most representative boosting technique, which generates a set of classifiers to be further combined using the weighted majority voting [19]. Basically, a weak classifier is trained using instances drawn from an iteratively updated distribution of training data. Introduced distribution update ensures that instances misclassified by previous classifier are more likely to be included in training data of the next classifier. To make final decision, each classifier has a different power of decision depending on its performance during training procedure.

2.3 Cost Sensitive Strategies

Cost Sensitive Learning. This strategy attempts to minimize costs associated to their decisions rather than simply reaching high precision. Given a cost specification for either correct or incorrect predictions, sample can be assigned to that class that leads to lower expected cost, where the expected value is computed using conditional probability of each class for a given sample. So, assuming c_{ij} as inputs associated to a cost matrix C holding cost to predict a class i when the true class is j. If $i = j$, prediction is assumed as correct, whereas if $i \neq j$, prediction is incorrect. Given a sample $\boldsymbol{x}$, optimal prediction for each i is a class that minimizes:

$$L(\boldsymbol{x}, i) = \sum\nolimits_j P(j|\boldsymbol{x}) c_{ij} \tag{1}$$

where $L(\boldsymbol{x}, i)$ is the sum over alternative possibilities for true class of sample $\boldsymbol{x}$. In this framework, the goal of algorithm is to produce a classifier estimating probability $P(j|\boldsymbol{x})$. So, for a given $\boldsymbol{x}$, prediction can be carried out as if i were the true class. As quoted in [9], the main idea behind decision-making based on cost sensitivity learning is that it may be optimal to act as if one class were true even when other class looks like more probable. In the biclass case, the optimal prediction is the class 1, if and only if, expected cost of prediction is less than or equal to the one predicting class 0, i.e.:

$$P(j=0|\boldsymbol{x})c_{10} + P(j=1|\boldsymbol{x})c_{11} < P(j=0|\boldsymbol{x})c_{00} + P(j=1|\boldsymbol{x})c_{01} \tag{2a}$$

$$(1-p)c_{10} + pc_{11} < (1-p)c_{00} + pc_{01}, \tag{2b}$$

where $p = P(j = 1|\boldsymbol{x})$. Therefore, threshold for optimal decision making is as follows:

$$p^* = \frac{c_{10} - c_{00}}{c_{10} - c_{00} + c_{01} - c_{11}} \tag{3}$$

MetaCost. This technique assumes an unaltered classifier, but adjusts learning according to a given cost matrix. Initially, training set is taken to constitute multiple subsets via bootstrap, where each obtained subset is used to build an classifier ensemble making the final decision [8]. Classifiers are combined through a majority vote rule to determine the probability of each data object $\boldsymbol{x}$ belonging to each class label. Next, each data object of training data is relabeled based on evaluation provided for an introduced conditional risk function, as follows:

$$R(i|\boldsymbol{x}) = \sum\nolimits_j P(j|\boldsymbol{x}) c_{ij} \tag{4}$$

Lastly, the classification algorithm makes the decision over relabeled training data.

3 Proposed Optimal Cost Matrix Based on CuckooCost Search

The *Cuckoo Search* is based on parasitic behavior exposed by some species of Cuckoo birds that has as natural strategy to leave eggs in host nest created by

Table 1. Scaled cost matrix.

	Actual negative	Actual positive
Predicted negative	$c_{00} = 0$	$c_{01} = k^+/k^-$
Predicted positive	$c_{10} = 1$	$c_{11} = 0$

other birds. This eggs presents the particularity to have a big similitude with host eggs, so, the more similar they are, the greater your chance of survival.

Based on this statement, Cuckoo Search uses three hypothesized rules:

- Each cuckoo lays one egg at a time, but dumps it in a randomly chosen nest.
- The best nests with high quality of eggs (solutions) should carry over to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

For simplicity, the last assumption can be approximated by a fraction p_a of the n nests being replaced by new nests (with new random solutions at new locations). For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. Given a solution at time$t$, noted as $\boldsymbol{x}_i^t$, thus, generation of new solutions along the time t is defined as:

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^t + \alpha \oplus \text{Lévy}(\lambda) \tag{5}$$

where α is a scale parameter while λ is the step size of the Cuckoo Search optimization. Notation $\oplus$ stands for a direct summation operator.

In bi-class problems, the minority class (i.e., the category with less samples) is assumed to have higher misclassification cost k^+ (usually, this samples relate to category of interest). Due to big amount of data, likewise, category with more samples must have lower misclassification cost k^-. From the above, if a cost matrix is given, the optimal decisions remain unchanged if their cost (in this case the inputs of the cost matrix) are multiplied by a scaling factor [17]. This normalization allows to change the baseline in which costs are measured. Therefore, if each element of the cost matrix that is multiplied by $1/k^-$, can be expressed as shown in Table 1.

Since costs are normalized to the unchanged optimal decision, value k^- can always be set to 1. Therefore k^+/k^- must be bigger than 1 [9]. This relationship is termed *cost-sensitive rescale ratio* or cost ratio [17]. In order to deal with class-imbalance using rescaling, different costs are to be incurred for different classes. So, the optimal rescale ratio (*imbalance rescale ratio*) of positive class to negative class, r, is defined as $r_I = n^-/n^+$. Therefore, to handle unequal misclassification and class-imbalance at the same time, both the cost-sensitive rescale ratio r_C and the imbalance rescale ratio r_I should be taken under consideration [17]. Merging both scale factors, cost ratio of the cost matrix can obtained as $\varphi = r_C r_I$, where $\varphi \geq r_I$.

Suggested cost using Cuckoo Search, termed *CuckooCost*, accomplishes optimal parameter values to achieve the best possible classification performance. Each nest represents a solution set in the searching space, i.e., each egg on the nest represents a parameter to be used in the model optimization. In this case, the cost ratio and classifier parameters are handled to improve the performance of cost sensitive learning.

It is worth noting that in Cuckoo Search, both parameters p_a and α are to explore efficiently over searching space and allow to find together globally and locally improved solutions. Additionally, these parameters directly influence the convergence rate of used optimization algorithm. For instance, if value p_a tends to be small and α value is large, the algorithm tends to increment the iteration number to converge to an optimal value. On the other hand, if p_a is large but α is small, the algorithm convergence speed tends to be very high but it is more likely to converge to a local optimum. In this work, an improvement to Cuckoo Search proposed in [23] is used that consists in restraining range of p_a and α; within this values search may change dynamically during each iteration, through the following equations:

$$c = \frac{1}{N_T} \ln\left(\frac{\alpha_{\min}}{\alpha_{\max}}\right), \; l\alpha \in [\alpha_{\min}, \alpha_{\max}] \tag{6a}$$

$$p_a = p_{\max} - \frac{N_I}{N_T}(p_{\max} - p_{\min}), \; p_a \in [p_{\min}, p_{\max}] \tag{6b}$$

$$\alpha = \alpha_{\max} \exp(c\, N_I) \tag{6c}$$

where N_T is the total number of iterations present in the optimization, N_I is the current iteration in the algorithm,

The best nest is the one holding the optimal parameters to induce a dependable cost sensitive model.

4 Experimental Setup

4.1 Database

Used database that is a subset of the one constructed in [15] holds 1098 proteins belonging to *Embryophyta* taxonomy of the Uniprot [14], with at least one annotation in the molecular function ontology of the Gene Ontology Annotation project [3]. Sequences predicted by computational tools and with no real experimental evidence are discarded. Proteins are associated to one or more of the seven categories that are shown in Table 2. The dataset does not contain protein sequences with a sequence identity superior to 40 % in order to avoid neither bias nor overtraining in the training dataset. All the proteins are mapped into feature vectors enclosing several statistical and physical-chemical attributes (see Table 3).

Table 2. Dataset description.

Class	Biological name	Samples	Imbalance ratio
GO 0003677	DNA binding	143	1 : 7.68
GO 0003700	Sequence-specific DNA binding transcription factor activity	102	1 : 10.76
GO 0003824	Catalytic activity	401	1 : 2.74
GO 0005215	Transporter activity	133	1 : 8.26
GO 0016787	Hydrolase activity	237	1 : 4.63
GO 0030234	Enzyme regulator activity	46	1 : 23.87
GO 0030528	Transcription regulator activity	152	1 : 7.22

Table 3. Description of the feature space (taken from [15]).

Feature	Description	Number
Chemical-physical	Length of the sequences	1
	Molecular weight	1
	Percentage of positively charged residues (%)	1
	Percentage of negatively charged residues (%)	1
	Isoelectric point	1
	GRAVY - hydropathic index	1
Primary structure	Frequency of each aminoacids	20
	Frequency of each dimers	400
Secundary structure	Frequency of structures	3
	Frequency of dimers in structures	9
	TOTAL	**438**

4.2 Fitness Function Approach

Performance of CuckooCost depends largely on a function that can properly guide searching process of the optimal hyperparameters. For this purpose, we propose the following fitness function that combines two variables that directly influence the classification process: area under ROC curve, $\mho$, and the total cost, ς, as follows:

$$\Theta(\mho, \varsigma) = \lambda(\mho) + (1 - \lambda)(\varsigma) \tag{7}$$

The aim of proposed fitness function is to maximize the overall classification performance as well as to minimize the cost associated with the wrong classified samples. Optimal value of free parameter of the fitness function is searched within a range interval [0, 1.] Heuristically, the best value is fixed at $\lambda = [0.1]$.

4.3 Class Imbalance and Classification Schemes

To mitigate the effect generated by multi-label samples in the dataset, as well as to reduce classification complexity and to obtain a better interpretation of results, *against-vs-all* learning strategy is used. Nevertheless, the usage of this strategy leads to additional problems such as highly class imbalance on data space. To overcomes this issue, the following class balance strategies are considered: AdaBoost (Ada), SMOTE, Subsampling based on particle swarm optimization (SPSO), and cost sensitive learning (CS). Also, the proposed MetaCost (MC) is considered in two versions: (i) without matrix cost optimization via CuckooCost (MC), (*ii*) Cost sensitive learning and MetaCost within CuckooCost (MCCu). During classification testing, support vector machines (SVM) with Gaussian Kernel is used, except the test with AdaBoost, for which Naive Bayes is employed as weak classifier and twenty iterations for Boosting technique. Parameter tuning needed in SVM and Gaussian Kernel (penalty constant C and dispersion γ) are carried out by using particle swarm optimization. However, PSO is not accomplished in cost sensitive learning strategies (CS and MetaCost), mainly, since the optimization based on Cuckoo Search turns to be more effective that PSO. So, CuckooCost takes γ and penalty constant C as hyperparameters in the optimization problem. To evaluate the performance of molecular function classification, cross-validation is used over ten folds. Besides, chosen a priori search parameter ranges of CuckooCost are the following:

$$\begin{aligned} 1 &\leq \varphi \leq 1.5R_d \\ 0.00030518 &\leq C \leq 4096 \\ 0.000030518 &\leq \gamma \leq 32 \end{aligned}$$

where φ is the cost ratio extracted from cost matrix, and R_d is the imbalance ratio.

4.4 Evaluation Metrics

Performance measures non-susceptible to unbalance data phenomena are used to obtain a reliably evaluation of accomplished classification. Measures such as sensitivity, specificity, geometric mean, and ROC area (AUC) are used, which are defined as:

$$\text{Sensitivity}: \quad S_e = \frac{T_P}{T_P + F_N} \tag{8a}$$

$$\text{Specificity}: \quad S_p = \frac{T_N}{T_N + F_P} \tag{8b}$$

$$\text{Geometric mean}: \quad \mu_G = \sqrt{S_e S_p} \tag{8c}$$

$$\text{ROC area}: \quad \mho = \frac{1 + T_P^{'} - F_P^{'}}{2} \tag{8d}$$

where T_P, T_N, F_N, and F_P are the true positive, true negative, false negative, and false positive values obtained from confusion matrix, respectively; $T_P^{'}$ is the true positive rate, $F_P^{'}$ is the false positive rate.

Additionally, a metric measuring the classification bias degree, termed relative sensitivity, is used that is defined as $r_S = S_e/S_p$, as given in [22].

Data Complexity Measures. Degree of data imbalance is not the only factor leading to a biased learning. Elements associated with data complexity may also influence learning models. Particularly, data complexity can be related to difficulties inherent in data, shortcomings in classification algorithms, and the low representation present in the data space [4,12]. The following measures are used to quantify data complexity:

(i) **Overlap Measures:** They explore both range and distribution of values in each category, and verify the overlap between them. The measures following overlap measures are used:

- *Volume of Overlap Region* (VOR): For a given feature set, $\{f_i\}$, VOR measures the amount of overlap in boundary region between two categories, $c_i : i = (1, 2)$, and is defined as [4,13]:

$$VOR = \prod_i \frac{\min(\max(f_i, c_1), \max(f_i, c_2)) - \max(\min(f_i, c_1), \min(f_i, c_2))}{\max(\max(f_i, c_1), \max(f_i, c_2)) - \min(\min(f_i, c_1), \min(f_i, c_2))} \tag{9}$$

- *Fisher's Discriminant Ratio:* For a multidimensional problem, all features not necessarily have to contribute to class discrimination. As long as there exists one discriminating feature, the problem is suitable. Therefore, we use the maximum f over all the feature dimensions to describe a problem [4,13]. This measure also serves as indicator of quality in the dataset representation, i.e., if its value tends to be low, there is little contribution in the overall discrimination of the dataset, which may indicate a weak representation of the data. Fisher's discriminant ratio is defined as:

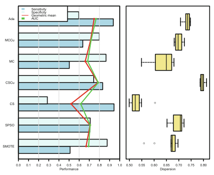
$$\kappa = \max \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{10}$$

  where μ_1 and μ_2 are the feature mean of the classes 1 and 2, while σ_1 and σ_2 are the feature variance of same classes, respectively.

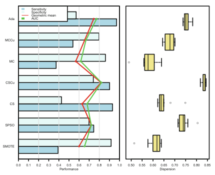
- **Scatter Matrix of Difference between inter/intra Classes:** It measures the distance between the class distribution and indicates on improved separability as its value is greater [10]. Being complementary to **VOR** and κ, this metric is described as:

$$J_4 = \text{tr}\,\{S_b - S_w\} \tag{11}$$

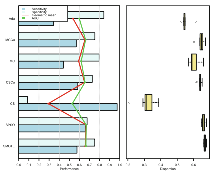
  where, $S_w = \sum_{i=1}^{C} \frac{n_i}{n} \widehat{\Sigma}_i$ and $S_b = \sum_{i=1}^{C} \frac{n_i}{n} (m_i - m)(m_i - m)^\top$, being $\widehat{\Sigma_i}$ covariance matrix of i-th class, m_i is the sample mean of i-th class and



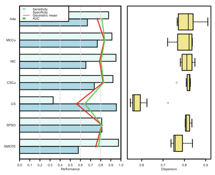
(a) DNA binding



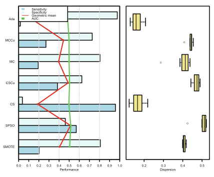
(b) Sequence-specific DNA binding - transcription factor activity



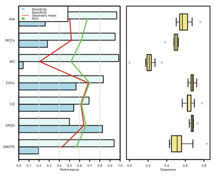
(c) Catalytic activity



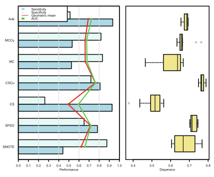
(d) Transporter activity



(e) Hydrolase activity



(f) Enzyme regulator activity



(g) Transcription regulator activity

Fig. 2. Molecular function prediction results.

m the sample mean of the whole dataset. Notation tr stands for matrix trace, C is the number of classes, and n is the number of samples in whole dataset.

(ii) **Measures of Geometry, Topology, and Density of Manifolds:** These metrics give indirect information about separation between categories. It is assumed that a category is composed by a collection of one or more manifolds, forming the support of the probability distribution of a given class. The shape, position and interconnectivity of manifolds give a hint of its overlap [4,13]. To evaluate the complexity of manifolds, the leave-one-out error for a one-nearest-neighbour classifier, Δ, is used.

5 Results and Discussion

Figure 2 summarizes obtained classification results that are displayed by bars and lines at different color scales. Each subfigure holds information about behavior of the geometric mean (drawn in red color), area under ROC curve (AUC) (green), sensitivity (light blue), and specificity (light cyan). Each row depicts each one considered class-balance strategies, which are ranked in ascending order according to used balance strategy, that is, oversampling (**SMOTE**), subsampling (**SPSO**), cost-sensitive learning without any optimized parameters and the same strategy using CuckooCost as well (**CS**, **CSCu**, **MC**, **MCCU**), and Boosting (**AdaBoost**). On the right side of the graph, mean values of boxplots are also shown regarding classifier dispersions obtained by each balance technique.

Table 4 that includes information concerning data complexity involved in categories describes measurements determining overlap and separability between

Table 4. Table of data complexity measurements in the datasets.

Categories	κ	VOR	$J4$	Δ (%)	Imbalance
GO 0003677	1,162564308	1,518414e-45	366,65	42	1:7,68
GO 0003700	1,258898151	8,325292e-43	153,09	54,7	1:10,76
GO 0003824	0,095424389	1,503915e-39	114,67	41,3	1:2,74
GO 0005215	1,275657636	1,974715e-67	3045,37	19,4	1:8,26
GO 0016787	0,004254501	6,654359e-07	-0,472	53,9	1:4,63
GO 0030234	0,265168845	1,247835e-26	14,712	79,3	1:23,87
GO 0030528	0,954652410	1,125151e-37	255,43	37,6	1:7,22

classes (VOR, $J4$, κ). Also included measurements of nonlinearity in the classifiers (Δ) are compared to information of imbalance level for each used dataset. This comparison is intended to provide information about difficulty to induce reliable learning models in each biclass problem.

Measures such as $J4$ and κ tend to be favorable as they increase in value, indicating a greater separability, otherwise VOR tends to be better as its value approaches zero, indicating a smaller area of overlap. According to the values given in Table 4, the most complex space is the set belonging to Hydrolase activity (GO 0016787), showing a low value at $J4$ and VOR highest compared to values achieved by other classes. This fact is proved by the results obtained for this class, as seen in Fig. 2(a), where all techniques show poor performance balance. This suggests that a very poor data representation is present in this class.

Also, as seen from values listed in Table 4, level of imbalance is not as significant as compared with the values of overlap between the data. Then, one can infer that data complexity may deteriorate more severely the learning process in protein prediction compared to the class imbalance. But this happens only when level of overlap and separability is to big compared with imbalance ratio itself. Therefore, it is convenient to use complexity measures as a complement to the imbalance degree to be certain about problem complexity.

Despite observed complexity, the best behavior for Hydrolase activity is obtained by SPSO that provides a value of geometric mean (GM) and ROC area (AUC) just over 50 %, but with very low dispersion in the prediction. Yet, obtained difference is not representative if compared to the performance of CSCu method. In datasets with higher imbalance between categories (such as Enzyme regulator activity and Sequence-specific DNA binding transcription factor activity (GO 0030234 and GO 0003700)), it is worth noting that CSCu performs considerable superiority over other compared techniques. In fact, its performance overcomes in five of the seven categories (GO 0003677, GO 0003700, GO 0003824, GO 0030234 and GO 0030528), while for the remaining 2 sets (GO 0005215 and GO 0016787), it is one of the highest performing prediction techniques, as seen in Table 5.

Table 5. Prediction performances with several balancing strategies.

Categories	SMOTE		SPSO		CS		CSCu		MC		MCCu		Ada	
	AUC	GM	AUC	GM	AUC	GM	AUC	GM	AUC	GM	AUC	GM	AUC	GM
`GO 0003677`	0,693	0,668	0,708	0,707	0,615	0,519	**0.797**	**0.796**	0,684	0,659	0,718	0,713	0,766	0,747
`GO 0003700`	0,654	0,599	0,721	0,721	0,679	0,629	**0.815**	**0.810**	0,617	0,566	0,668	0,655	0,773	0,744
`GO 0003824`	0,664	0,658	0,667	0,667	0,53	0,292	**0.671**	**0.671**	0,618	0,592	0,661	0,654	0,599	0,536
`GO 0005215`	0,778	0,752	0,811	0,81	0,643	0,562	0.815	0.810	0,803	0,788	**0,839**	**0,835**	0,812	0,766
`GO 0016787`	0,505	0,405	**0,516**	**0,513**	0,497	0,188	0.491	0.473	0,499	0,395	0,499	0,443	0,485	0,128
`GO 0030234`	0,568	0,429	0,663	0,642	0,618	0,613	**0.696**	**0.683**	0,515	0,205	0,617	0,518	0,675	0,502
`GO 0030528`	0,659	0,621	0,717	0,714	0,595	0,493	**0.784**	**0.783**	0,68	0,662	0,676	0,66	0,723	0,691
Total	0,646	0,59	0,686	0,682	0,596	0,47	**0.724**	**0.718**	0,63	0,552	0,668	0,64	0,69	0,588

On the other hand, both AdaBoost and SMOTE techniques obtain the worst prediction results, especially, in Hidrolase activity, Enzyme regulator activity, and Transcription regulator activity (`GO 0016787`, `GO 0030234`, and `GO 0030528`). Since there is a high probability of inducing extra noise during training set when synthetic samples are added, therefore, we may infer that in the presence of sets with high overlap, oversampling technique is not an option making unreliable the model for prediction of molecular functions. In case of AdaBoost, a high overlap may decrease considerably the generalization capability of used classifier, which mostly is forced to have complex decision boundaries.

As shown in Fig. 2 and Table 5, the use of CuckooCost improves the performance of considered methods based on cost sensitive learning (CS, MC, CSCu, MCCu). Moreover, it clearly shows a substantial improvement in MetaCost and cost sensitive learning in overall performance (increased GM and AUC), as well as the reliability of the results by decreasing the classification dispersion in every category. Although MetaCost tends to improve when using CuckooCost strategy in transporter activity (`GO 0005215`), still there is a slight increase in terms of the variability of the results. MetaCost accomplishes the resampling procedure using Bootstrap strategy, when taking a portion of the training set to create a subset in each iteration. Further, each subset is taken by a number of base classifiers equal to the number of iterations for the algorithm selected and the final classification decision is made in committee by a vote of each classifier. So, if the number of iterations in MetaCost is not adequate and additionally the dataset has a substantial degree of imbalance, as it is in this case, the number of samples of interest, i.e., the samples belonging to this category used for each base classifier might not be enough. As a result, the variability of performance increases.

For all considered categories, generally, there exist cases where some balance techniques show very similar values of μ_G compared with their $\mho$ values, mainly, in SPSO and CSCu. It happens, particularly, when the numeric difference between sensitivity and specificity becomes too close.

Assuming $\xi = S_e - S_p$, then, it holds that:

$$T_P^{'} = \frac{T_P}{T_P + F_N} = S_e$$
$$F_P^{'} = \frac{F_P}{T_N + F_P}$$

However, $F_P/(T_N + F_P) = 1 - T_N/(T_N + F_P) = 1 - S_p$. Moreover, taking into account that the ROC curves shows a comparison between $T_P^{'}$ vs $F_P^{'}$, then, $\mho(T_P^{'}, F_P^{'}) = (1 + T_P^{'} - F_P^{'})/2$. So, $\mho$ (AUC) can be expressed in terms of S_e and S_p, as follows:

$$\begin{aligned}\mho(S_e, S_p) &= \frac{1 + S_e - (1 - S_p)}{2} \\ &= \frac{S_e + S_p}{2}\end{aligned}$$

Therefore expressing the sensitivity in terms of specificity, i.e., $S_e = S_p + \xi$, one can infer that if the numeric distance between sensitivity and specificity is shortened, that is, $\xi \rightarrow 0$, then:

$$\begin{aligned}\lim_{\xi \rightarrow 0} \mho(S_e, S_p) &= \lim_{\xi \rightarrow 0} \frac{S_e + S_p}{2} \\ &= \lim_{\xi \rightarrow 0} \frac{S_p + S_p + \xi}{2} = S_p\end{aligned}$$

Now, if considering the geometric mean, the following holds:

$$\begin{aligned}\lim_{\xi \rightarrow 0} \mu_G(S_e, S_p) &= \lim_{\xi \rightarrow 0} \sqrt{S_e\ S_p} \\ &= \lim_{\xi \rightarrow 0} \sqrt{(S_p + \xi) S_p} = S_p\end{aligned}$$

So, the above expressions indicate that if ξ tends to be much smaller, both $\mho$ and μ_G tend increasingly to the same value.

The case when $\mho = \mu_G$ also takes place when the balancing techniques have very little classification bias. This fact can be corroborated using the relative sensitivity (R_S) [22]: If $R_S \rightarrow 1$, then, $S_e/S_p \rightarrow 1$, in turn, $S_e = S_p$.

The R_S values for each technique are shown in Table 6. As seen, SPSO and CSCu techniques show less bias in their classification performance having values more close to one. That is, both classifiers tend to get a trade-off between sensibility and specificity values, as shown with the points in Fig. 2, for which $\mho = \mu_G$. On contrast, SMOTE tends to get more specificity, although sampling techniques try to become more sensitive to increase distribution of samples in those categories having lower representation. Lastly, it must be quoted that both CS and MC carry out quite a substantial improvement when they use CuckooCost to optimize their parameters. Initially, CS is to sensitive but it has a small specificity, contrary case to MC, which has a big specificity. When CuckooCost is used, both strategies accomplish similar performance, specially, in terms of CS.

Table 6. Table of relative sensitivity.

Categories	SMOTE	SPSO	CS	CSCu	MC	MCCu	Ada
GO 0003677	0,582	0,996	3,301	1.129	0,582	0,796	1,572
GO 0003700	0,427	1,067	2,185	1.239	0,433	0,676	1,701
GO 0003824	0,766	0,973	11,057	1.045	0,555	0,755	0,406
GO 0005215	0,593	1,002	2,885	0.794	0,688	0,842	0,762
GO 0016787	0,251	1,234	25,892	0.576	0,241	0,371	0,017
GO 0030234	0,208	1,652	0,783	0.684	0,044	0,297	0,269
GO 0030528	0,503	1,203	3,541	1.120	0,631	0,651	1,824

6 Conclusions and Future Work

A method to optimize free parameters associated to cost sensitive learning, which is applied to prediction of molecular functions in embryophita plants, is proposed. The method is devoted to rule directly sensitivity and specificity on classifier performance (related to the costs involved misclassifying samples belonging to each category). The optimization is carried out over cost matrix elements, which are tuned by adapting those elements outside the main diagonal, in order to build the cost ratio. The variation of the cost ratio, along with the classification parameters are used as hyperparameters in the optimization problem, since the metric intrinsically modifies the fitness function. To this purpose, a metaheuristic optimization technique called Cuckoo Search is suggested. The methodology takes as fitness function both the maximization of ROC area (AUC) and minimization of total cost; being both variables important in the classification model. This work shows that the use of models based on cost sensitivity learning are competitive, reliable, and even superior to other balance techniques in the state of the art, specially, in applications related to bioinformatics. As future work, the approach of new fitness functions that lead to better classification results is to be considered.

Acknowledgements. This work is partially funded by the Research office (DIMA) at the Universidad Nacional de Colombia at Manizales and the Colombian National Research Centre (COLCIENCIAS) through grant No.111952128388 and the "*jovenes investigadores e innovadores - 2010 Virginia Gutierrez de Pineda*" fellowship.

References

1. Aebersold, R., Mann, M., et al.: Mass spectrometry-based proteomics. Nat. **422**(6928), 198–207 (2003)
2. Allison, D.B., Cui, X., Page, G.P., Sabripour, M.: Microarray data analysis: from disarray to consolidation and consensus. Nat. Rev. Genet. **7**(1), 55–65 (2006)

3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nat. Genet. **25**(1), 25 (2000)
4. Basu, M.: Data Complexity in Pattern Recognition. Springer, New York (2006)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
6. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach. Learn. **40**(2), 139–157 (2000)
7. Ding, Z.: Diversified ensemble classifiers for highly imbalanced data learning and its application in bioinformatics. Ph.D thesis, Georgia State University (2011)
8. Domingos, P.: Metacost: a general method for making classifiers cost-sensitive. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 155–164. ACM (1999)
9. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
10. García-López, S., Jaramillo-Garzón, J.A., Higuita-Vásquez, J.C., Castellanos-Domínguez, C.G.: Wrapper and filter metrics for PSO-based class balance applied to protein subcellular localization. In: 2012 Biostec-Bioinformatics (2012)
11. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. J. Intell. Manuf. **16**(6), 565–573 (2005)
12. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
13. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 289–300 (2002)
14. Jain, E., Bairoch, A., Duvaud, S., Phan, I., Redaschi, N., Suzek, B., Martin, M., McGarvey, P., Gasteiger, E.: Infrastructure for the life sciences: design and implementation of the uniprot website. BMC Bioinform. **10**(1), 136 (2009)
15. Jaramillo-Garzón, J.A., Gallardo-Chacón, J.J., Castellanos-Domínguez, C.G., Perera-Lluna, A.: Predictability of gene ontology slim-terms from primary structure information in embryophyta plant proteins. BMC Bioinform. **14**(1), 68 (2013)
16. Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armañanzas, R., Santafé, G., Pérez, A., et al.: Machine learning in bioinformatics. Briefings Bioinform. **7**(1), 86–112 (2006)
17. Liu, X.Y., Zhou, Z.H.: The influence of class imbalance on cost-sensitive learning: an empirical study. In: 2006 Sixth International Conference on Data Mining, ICDM'06, pp. 970–974. IEEE (2006)
18. Liu, X.-Y., Zhou, Z.-H.: Towards cost-sensitive learning for real-world applications. In: Cao, L., Huang, J.Z., Bailey, J., Koh, Y.S., Luo, J. (eds.) PAKDD Workshops 2011. LNCS, vol. 7104, pp. 494–505. Springer, Heidelberg (2012)
19. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits Syst. Mag. **6**(3), 21–45 (2006)
20. Schapire, R.E.: A brief introduction to boosting. In: International Joint Conference on Artificial Intelligence, vol. 16, pp. 1401–1406. Lawrence Erlbaum Associates Ltd (1999)
21. Sonnenburg, S., Schweikert, G., Philips, P., Behr, J., Rätsch, G.: Accurate splice site prediction using support vector machines. BMC Bioinform. **8**(Suppl 10), S7 (2007)
22. Su, C.T., Hsiao, Y.H.: An evaluation of the robustness of MTS for imbalanced data. IEEE Trans. Knowl. Data Eng. **19**(10), 1321–1332 (2007)

23. Mohanna, E., Valian, E., Tavakoli, S.: Improved cuckoo search algorithm for global optimization. Int. J. Commun. Inf. Technol. **1**(1), 31–44 (2011)
24. Yang, P., Xu, L., Zhou, B.B., Zhang, Z., Zomaya, A.Y.: A particle swarm based hybrid system for imbalanced medical data sampling. BMC Genomics **10** (Suppl 3), S34 (2009)