34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Improving the prediction of sub-cellular locations of proteins with a particle swarm optimization-based boosting strategy

Sebastián García-López[1], Jorge Alberto Jaramillo-Garzón[1,2] and Germán Castellanos-Domínguez[1]

Abstract— **Learning from imbalanced data sets presents an important challenge to the machine learning community. Traditional classification methods, seeking to minimize the overall error rate of the whole training set, do not perform well on imbalanced data since they assume a relatively balanced class distribution and put too much strength on the majority class. This is a common scenario when predicting sub-cellular locations of proteins since proteins belonging to certain specific locations are naturally more abundant or have been more extensively studied. In this work, a new method to learn from imbalanced data, called SwarmBoost, is proposed in order to reduce overlapping and noise of imbalanced datasets and improve prediction performances. The method combines oversampling, subsampling based on particle swarm optimization and ensemble methods. Our results show that SwarmBoost equals and in several cases outperforms other common boosting algorithms like DataBoost-Im and AdaBoost, constituting a useful tool for improving sub-cellular location predictions.**

I. Introduction

The latest advances in science and technology have generated an exponential growth of information, leading to a great evolution in the field of engineering data analysis focused on large-scale data. The above phenomenon has been present in fields such as proteomics and molecular biology, including the identification and functional annotation of novel proteins. The sub-cellular localization of proteins can provide useful information on how proteins interact with each other and with other molecules, thus providing important clues to reveal their functionality [1]–[3]. Although this type of information can be acquired by conducting various biochemical experiments, it is usually very time consuming and practically cumbersome. For that reason, it is highly desirable to develop computational methods for identifying sub-cellular localizations of novel proteins [3]. However, proteins with certain specific locations are more abundant, creating a high degree of disparity in the number of samples belonging to each class [4] and, since machine-learning classifiers with unbalanced data usually generate larger bias [5], [6], proteins of interest get misclassified and prediction performances are low.

The fundamental issue with the imbalanced learning problem is the the property of imbalanced data to significantly compromise the performance of most standard learning algorithms. Such algorithms assume or expect balanced class distributions or equal misclassification costs. Therefore, when presented with complex imbalanced data sets, they fail to properly represent the distributive characteristics of the data and resultantly provide unfavorable accuracies [7].

To solve this bottleneck several strategies have been proposed, most of them focused on under-sampling examples of the majority class [8], over-sampling examples of the minority class [9], combined under-sampling and over-sampling [10] or weighting examples to bias the learning towards the minority class [9]. Lately, ensembles have emerged as a promising technique with the ability to improve the performance of weak classification algorithms [11], [12]. Ensembles of classifiers consist of a set of individually trained classifiers whose predictions are combined to classify new instances. In particular, boosting is an ensemble method where the performance of weak classifiers is improved by focusing on hard examples which are difficult to classify. Boosting produces a series of classifiers and the outputs of these classifiers are combined using weighted voting in the final prediction of the model [13], [14]. Recent studies have indicated that boosting algorithms are applicable to a broad spectrum of problems with great success [13], [15]. However, there are also some drawbacks for boosting methods: they can fail to perform well if there is not enough training data [16] or if the training data contains too much noise [17]. With the aim of solving this problem, combined techniques that merge Oversampling and Boosting strategies like DataBoost-IM [14] have been proposed. However, the use of Oversampling techniques in Boosting does not guarantee a better training subset to induce a better generalization capability, mainly because most oversampling techniques tend to produce over generalization and add computational complexity [18].

In this paper, a new method that combines oversampling, particle swarm optimization subsampling and Boosting is presented. The aim of this method is to improve the predictive accuracies of both the majority and minority classes, by using a selective sampling criterion that chooses the most representative samples of the training dataset and thus represents more efficiently the probabilistic distribution of the data. The approach was tested in sub-cellular location prediction of Yeast and E. Coli proteins. The results show improved prediction performances compared with standard and advanced boosting techniques like AdaBoost and DataBoost-IM for such datasets.

II. Materials and methods

II-A. DataBoost-IM

DataBoost-IM is a reinforcement methodology designed to manage unbalanced datasets [14]. It combines data ge-

[1] Grupo de Control y Procesamiento Digital de Señales. Universidad Nacional de Colombia, sede Manizales. Km 7. Vía al Magdalena. Manizales, Colombia. (e-mail: {sgarcialop; jajaramillog; cgcastellanosd }@unal.edu.co

[2]Grupo de Automática y Electrónica. Instituto Tecnológico Metropolitano, Calle 73 No 76A-354, Medellín (Antioquia), Colombia.

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

neration and boosting procedures to improve the predictive accuracies of simple classifiers over both the majority and minority classes. Starting with a weak classifier and set of m examples, each one of them associated to a weight w_i, $w = 1, 2, \ldots, m$, DataBoost-IM first initializes all $w_i = 1/m$ and then performs an iterative process comprised of the following steps:

1. Obtain the training error of the weak classifier to identify hard examples of the training dataset and sort the examples in descending order, based on their weights. Such examples will serve as seeds for generating the new data.
2. Generate synthetic examples according to the probability distribution of hard examples from both minority and majority classes.
3. Compute the weights associated to the new examples by dividing the actual weigth of their seed example by the number of instances generated from it.
4. Add the synthetic examples to the real ones and rebalance the weights of the whole data set searching that the sum of weights in the majority class equals the sum of weights in the minority class.
5. Re-train the weak classifier using the new weights.

The algorithm ends after a predefined number of iterations and provides a set of classifiers that are able to make better predictions than each weak classifier alone. To obtain more detailed information about DataBoost-IM, see [14].

II-B. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is a population based optimization tool where the system is initialized with a set of random solutions, seeking for an optimal subset of the population satisfying some performance index over generations. Given a set of m potential solutions $\mathbf{p}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$, called particles, and a fitness scalar function $f(\mathbf{p}_i) = q_i$, PSO assigns a randomized velocity $\mathbf{v}_i \in \mathbb{R}^n$ so that particles are then "flown" through the problem space. At each time step, the particles move depending on their fitness function values. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved at that time in a vector $\mathbf{p}_i \in \mathbb{R}^n$. Furthermore, the best obtained positions among all the particles in the population is included in the vector $\mathbf{p}^g \in \mathbb{R}^n$. A new velocity for the i-th particle is updated at each time step t by equation (1).

$$\begin{aligned}\mathbf{v}_i(t+1) = \alpha\mathbf{v}_i(t) &+ c_1\phi_1(\mathbf{b}_i(t) - \mathbf{p}_i(t)) \ldots \\ &+ c_2\phi_2(\mathbf{b}^g(t) - \mathbf{x}_i(t))\end{aligned} \quad (1)$$

where c_1 and c_2 are positive constants, ϕ_1 and ϕ_2 are uniformly distributed random numbers and α is the inertia weight. Changing velocity in this way enables the particle i to search around its individual ($\mathbf{b}_i$), and global ($\mathbf{b}^g$) best position. Based on the updated velocities, each particle changes its position according to equation (2).

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1) \quad (2)$$

II-B.1. Separability criterion based on scatter matrices:

Metrics based on separability estimate the overlap between the distributions from which the data are drawn, and favour those sample sets for which this overlap is minimal (i.e., maximizing the separability). A measure of the average separation separation between two data sets, ω_1 and ω_2 can be defined as:

$$J_{as}(\omega_i, \omega_j) = \frac{1}{n_1 n_2} \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} d(\mathbf{x}_i, \mathbf{y}_j) \quad (3)$$

where $\omega_1 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m_1}\}$, $\omega_2 = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{m_2}\}$, $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^n$. The average distance between classes measured in probabilistic distances is represented in the following equation:

$$J(\omega_i, \omega_j) = \frac{1}{2} \sum_{i=1}^{C} P(\omega_i) \sum_{j=1}^{C} P(\omega_j) J_{as}(\omega_i, \omega_j) \quad (4)$$

where $P(\omega_i)$ is the prior probability of class ω_i (estimated as P_i=n_i/n, being $n, C \in \mathbb{N}$ the total sample number in all dataset and C the number of classes presents in the dataset). This separability criterion is independent of the learning process used by the classifier employed and can be computed from the between$-$class (S_b) and within$-$class (S_w) scatter matrices respectively defined in equations 5 and 6.

$$S_b = \sum_{i=1}^{C} \frac{n_i}{n} (m_i - m)(m_i - m)^T \quad (5)$$

$$S_w = \sum_{i=1}^{C} \frac{n_i}{n} \widehat{\Sigma}_i \quad (6)$$

Where $\widehat{\Sigma_i}$ is the covariance matrix of the $i-th$ class, m_i the sample mean of the $i-th$ class and m the sample mean of the whole dataset. This way, J can be rewritten as:

$$J(\omega_i, \omega_j) = Tr\{S_w + S_b\} = Tr\{\Sigma\} \quad (7)$$

This criterion is simply the total variance, which does not depend on class information. It also reduces the level of dispersion within the classes.

III. Proposed Methodology: SwarmBoost

Although the resampling procedure implemented in DataBoost-IM improves the representative set to induce the model over imbalanced data, the use of oversampling strategies does not always can guarantee a good generalization capability, specially if it is used on ensemble of classifiers structured in the Boosting scheme. This may be caused by possible noise, rare samples in the resampling process and increased the ocurrence of overlapping between classes in the balance process [7], [19].

To overcome these problems, Swarmboost performs a subsampling stage before training the weak classifier in each iteration, reducing noise and redundant examples and

thus helping to mitigate the overlapping generated in over-sampling stage [20], [21]. With removed samples, it is possible establish well-defined class clusters in the training set, which can, in turn, lead to well defined classification rules for improved classification performance [7]. In order to minimize the loss of useful information, a subsampling strategy based on PSO is employed [20]. The evaluation function for the metaheuristic is based on filter metrics in order to reduce the computational cost while preserving similar or superior performances compared with Wrapper metrics [22]. The evaluation function used in SwarmBoost attempts to find the maximun interclass separability between the minority and majority classes Algorithm 1 shows the complete methodology of SwarmBoost.

Algorithm 1: SwarmBoost

Require: Labeled examples $\langle(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)\rangle$
Require: Weak learning algorithm **WeakLearn**
Require: Number of iterations T

1: Initialize distribution $D_t(i) = 1/m$
2: **for** $t = 1, 2, ..., T$ **do**
3: Identify hard examples from the original data set for different classes.
4: Generate synthetic data to balance the training knowledge of different classes.
5: Add synthetic data to the original training set to form a new training data set.
6: Update and balance the total weights of the different classes in the new training data set.
7: Resample the new training dataset based on PSO subsampling.
8: Call **WeakLearn**, providing it with the new training set.
9: Get back a hypothesis $h_t : \mathbf{X} \to Y$.
10: Calculate the training error $h_t : \varepsilon_t = \sum_{i:h_t(\mathbf{x}_i)\neq y_i} D_t(i)$.
11: Set $\beta_t = \varepsilon_t/(1-\varepsilon_t)$.
12: Update the distribution for the new training set $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(\mathbf{x}_i) = y_i \\ 1 & \text{if otherwise} \end{cases}$ where Z_t is a normalization constant.
13: **end for**
14: **return** $h_{fin}(x) = \underset{y\in Y}{argmax} \sum_{t:h_t(y)=y} log(1/\beta_t)$

Note that a detailed description of the data generation process used in DataBoost-IM (step 3 until step 6) falls beyond the scope of this paper. Interested readers are referred to [14] for a description of this process and its evolution.

IV. Experimental setup

IV-A. Databases

To evaluate the performance of the algorithms, two protein sub-cellular localization benchmark datasets were used. Such datasets correspond to Yeast and E. Coli and were extracted from UCI machine learning repository [23]. The multi-class problems were divided into multiple bi-class problems and classes with less than 20 samples were discarded because they had not enough data to induce a reliable model. Table I summarizes the characteristics of the data sets.

TABLE I
Data sets information

Class	Minority class instances	Features	Imbalance ratio
Yeast - CYT	463	8	1:2.20
Yeast - EXC	37	8	1:39.1
Yeast - MIT	244	8	1:5.1
Yeast - ME2	51	8	1:28.1
Yeast - ME3	163	8	1:8.11
Yeast - NUC	429	8	1:2.46
Ecoli - cp	143	8	1:1.34
Ecoli - im	77	8	1:3.36
Ecoli - imU	35	8	1:8.6
Ecoli - om	20	8	1:15.8
Ecoli - pp	52	8	1:5.46

IV-B. Class imbalance and classification schemes

To obtain a predictive model for all datasets, a decision tree C4.5 classifier was chosen. The proposed methodology was compared with AdaBoost and DataBoost-IM, all of them with three iterations. As objective function for the PSO-subsampling stage used on SwarmBoost, it was used the total variance filter (J1) described in [22] with 10 iterations of the PSO algorithm.

V. Results and discussion

Table II shows the results obtained by each method with a $10-$fold cross-validation scheme. Given the high class imbalance, it is misleading to employ simple classification accuracy as performance measure. So, a more balanced measure, the square root of the product between sensitivity and specificity (geometric mean), is used instead. In all cases the experiments comprised three iterations of the boosting schemes excepting when explicitly marked.

TABLE II
Performance with three boosting schemes on all datasets

Datasets	SWARMBOOST	DATABOOST-IM	ADABOOST
Yeast-CYT	**62,83**	62,81	30,26 [1]
Yeast-EXC	**83,49**	78,54	55,95 [2]
Yeast-ME2	**83,44**	76,13	53,54 [2]
Yeast-ME3	89,32	87,12	**89,34**
Yeast-NUC	64,98	67,32	**67,76**
Yeast-MIT	**75,13**	73,09	70,78
Ecoli-cp	95,20	**95,39**	94,49
Ecoli-im	87,24	84,17	**88,35**
Ecoli-imU	**87,68**	78,44	86,86
Ecoli-om	**93,34**	88,59	70,25
Ecoli-pp	**89,18**	88,51	83,42

[1] after 50 iterations
[2] after 8 iterations

It can be seen that the datasets with most similar performances across the three methods were Ecoli-cp, Yeast-ME3, and Yeast-NUC, and it is noticeable that these datasets have some of the lowest imbalance ratios (excepting Yeast-NUC). On the other hand, for datasets with the highest imbalace

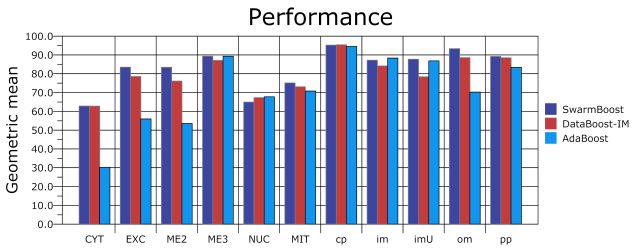

Fig. 1. Performance of the three boosting schemes on all datasets

ratios: Yeast-Exc, Yeast-ME2 and Ecoli-om, SwarmBoost clearly shown better performances than the other two methods. This phenomenon may be due to the increased noisy samples generated by oversampling (DataBoost-IM) within the training set used in Boosting techniques, which in turn increases the complexity of the training data set and can lead to a loss of generalization capabilities. In the case of AdaBoost, three iterations were insufficient to obtain geometric means superior to zero (the majority class completely dominated the minority class) for the datasets CYT, EXC and ME2. The algorithm responded after 50, 8, and 8 iterations respectively.

In general terms, AdaBoost showed the worst results, presumably due to the absence of representative samples in minority class to induce the learning model. SwarmBoost, presented the best performances in seven out of eleven datasets, being very close to the best method in the remaining four. This fact suggests a better generalization capability and therefore superiority among the tested algorithms. Figure 1 depicts the results in a graphic way.

VI. Conclusions and Future work

In this paper, a new tool for improving protein sub-cellular prediction, SwarmBoost, was proposed. The main purpose of the method was to create a boosting scheme to manage unbalanced data by combining the advantages of oversampling to have more information (DataBoost) with a better representation achieved by subsampling, thus inducing a model with better generalization abilities. Results show that SwarmBoost performs better than other Boosting schemes as the class imbalance ratio increases. It should be taken into account that the performance of SwarmBoost will depend on the fitness function used as subsampling criterion. As future work, it would be desirable to test SwarmBoost with some different criteria such as other filter or wrapper metrics that could provide more information about the distribution of classes. Also it will be useful to apply this methodology in a multiclass scenario.

VII. Acknowledgement

This work was 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 "Jóvenes Investigadores e Innovadores - 2010 Virginia Gutiérrez de Pineda" fellowship.

References

[1] J. Ehrlich, M. Hansen, and W. Nelson, "Spatio-temporal regulation of rac1 localization and lamellipodia dynamics during epithelial cell-cell adhesion," *Developmental cell*, vol. 3, no. 2, pp. 259–270, 2002.

[2] E. Glory and R. Murphy, "Automated subcellular location determination and high-throughput microscopy," *Developmental cell*, vol. 12, no. 1, pp. 7–16, 2007.

[3] K. Chou and H. Shen, "Plant-mploc: a top-down strategy to augment the power for predicting plant protein subcellular localization," *PLoS One*, vol. 5, no. 6, p. e11335, 2010.

[4] A. Al-Shahib, R. Breitling, and D. Gilbert, "Feature selection and the class imbalance problem in predicting protein function from sequence," *Applied Bioinformatics*, vol. 4, no. 3, pp. 195–203, 2005.

[5] I. Meyer, "A practical guide to the art of rna gene prediction," *Briefings in bioinformatics*, vol. 8, no. 6, pp. 396–414, 2007.

[6] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch, "Accurate splice site prediction using support vector machines," *BMC bioinformatics*, vol. 8, no. Suppl 10, p. S7, 2007.

[7] H. He and E. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.

[8] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Machine Learning - International Workshop then Conference-*. Morgan Kauffman Publishers, Inc., 1997, pp. 179–186.

[9] M. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," 2003.

[10] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[11] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Machine Learning - International Workshop then Conference-*. Morgan Kauffman Publishers, Inc., 1996, pp. 148–156.

[12] ——, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[13] H. Schwenk and Y. Bengio, "Adaboosting neural networks: Application to on-line character recognition," *Artificial Neural Networks–ICANN'97*, pp. 967–972, 1997.

[14] H. Guo and H. Viktor, "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 30–39, 2004.

[15] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[16] R. Schapire, "A brief introduction to boosting," in *International Joint Conference on Artificial Intelligence*, vol. 16. Lawrence Erlbaum Associates, 1999, pp. 1401–1406.

[17] T. Dietterich, "Ensemble methods in machine learning," *Multiple classifier systems*, pp. 1–15, 2000.

[18] D. ZEJIN, "Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics," 2011.

[19] B. Wang and N. Japkowicz, "Imbalanced data set learning with synthetic samples," in *Proc. IRIS Machine Learning Workshop*, 2004.

[20] Y. Pengyi, X. Liang, Z. Bing, Z. Zili, and Z. Albert, "A particle swarm based hybrid system for imbalanced medical data sampling," *BMC Genomics*, vol. 10, 2009.

[21] J. Luengo, A. Fernández, S. García, and F. Herrera, "Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 10, pp. 1909–1936, 2011.

[22] S. García-López, J. A. Jaramillo-Garzón, J. Higuita-Vásquez, and C. Castellanos-Domínguez., "Wrapper and filter metrics for pso-based class balance applied to protein subcellular localization," in *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms*, 2012, pp. 214–219.

[23] A. Asuncion and D. Newman. (2012) Uci machine learning repository. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html