A New Approach to Detect Splice-Sites Based on Support Vector Machines and a Genetic Algorithm

Jair Cervantes[1], De-Shuang Huang[2], Xiaoou Li[3], and Wen Yu[4]

[1] Posgrado e Investigacíon, UAEM-Texcoco, Av. Jardín Zumpango s/n
Fraccionamiento El Tejocote, Edo. Mex., C.P. 56259
jcervantesc@uaemex.mx
[2] Department of Control Science & Engineering, Tongji University
Cao'an Road 4800, Shanghai, 201804 China
dshuang@tongji.edu.cn
[3] Departmento de Computación, CINVESTAV-IPN, Mexico City, Mexico
lixo@cs.cinvestav.mx
[4] Departamento de Control Automático, CINVESTAV-IPN, Mexico City, Mexico
yuwen@ctrl.cinvestav.mx

Abstract. This paper presents a method for classification of imbalanced splice-site classification problems, the proposed method consists of the generation of artificial instances that are incorporated to the dataset. Additionally, the method uses a genetic algorithm to introduce just instances that improve the performance. Experimental results show that the proposed algorithm obtains a better accuracy to detect splice-sites than other implementations on skewed data-sets.

Keywords: SVM, Skewed datasets, Classification DNA splice sites.

1 Introduction

Recognizing boundaries of exons and introns is a challenging task in DNA sequence analysis. To identify exons into DNA sequences present a computational challenge due to the genes in many organisms splices of different way. Moreover, most of gene datasets are imbalanced and the bulk of classifiers generally performs poorly on imbalanced datasets because making the classifier too specific may make it too sensitive to noise and more prone to learn an erroneous hypothesis. Moreover, sometimes an instance can be treated as noise and ignored completely by the classifier if the dataset is imbalanced. Consequently, an effective detection of splice sites requires not just to know features, dependencies, relationship of nucleotides in the splice site surrounding region or an effective encoding method, but also a good method which tackles the disadvantage of imbalanced in datasets. In this paper, we use a novel SVM approach to detect splice sites in imbalanced datasets. The proposed method generates new synthetic instances in a similar form of SMOTE [5], the key idea of this model is to introduce artificial instances in the

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part II, LNCS 8259, pp. 342–349, 2013.

region of positive SV, decreasing the skew of the margin and improving the generalization capacity. The proposed technique not only modifies the margin also modifies the region of the minority class improving the generalization power of the classifier. However, to introduce incorrectly new synthetic instances can reduce the classifier performance because this is a sensible region to small changes. To avoid this fundamental issue, we incorporate a Genetic Algorithm which guides the search in the sensible region generating intelligently new synthetic instances. The proposed algorithm, tackles the disadvantage of imbalanced data-sets with SVM. The rest of the paper is organized as following: Section 2 shows the SVM imbalanced problem. Section 3 focuses on explaining the methodology of proposed SVM classification algorithm. Section 4 shows experimental results. Conclusions are given in Section 5.

2 Related Work

SVM has received considerable attention due to its optimal solution, discriminative power and performance. SVM has been applied in many fields, some SVM algorithms have been used in splice site detection with acceptable accuracies. There are a lot of works about Splice sites detection with several methods in the literature. However, the works most representative of splice sites detection with SVM are [1] [3] [4]. Baten [1] uses SVM with polynomial kernel to obtain an effective detection of splice sites, Cheng [2] uses SVM to predict mRNA polyadenylation sites [poly(A) sites], the method helps to identify genes, define gene boundaries, and elucidate regulatory mechanisms, [3] and [4] use SVM to detect splice-junction (intron-exon or exon-intron) sites in DNA sequences. In all this works, the accurate splice-site detection is a critical component of all analytic techniques. However, before mentioned methods do not consider datasets with high imbalance. Lately has been showed that SVM performance drops significantly with imbalanced data-sets. Some important algorithms based on Undersampling, Oversampling or SMOTE techniques[5] had been developed to tackle this problem. SMOTE over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. The SMOTE technique is better than under-sampling and over-sampling and a promising technique to tackle this problem. Some other proposals inspired in SMOTE can be seen in [6].

Methods based on Genetic Algorithms (GA) have also been pursued to tackle imbalanced problems. Since evolutive methods provide state-of-the-art techniques for many of todays data engineering applications, the use of evolutive methods to understand imbalanced learning has naturally attracted growing attention recently. Zou et al. [7] use a GA to balance the data-sets. In [8] the authors propose a classification system using hierarchical fuzzy rule and a genetic algorithm to select the most important rules and to eliminate conflicting rules or rules which perturb the performance. Garcia et al. [9] implement an algorithm which performs an optimized selection of previously defined generalized examples obtained by a heuristic. An excellent state of the art about imbalanced classification can be found in [10]. Despite the early proposed methods to improve the performance, these algorithms use GA's to balance the data-sets, obtain rules or to select instances intelligently, but not to generate new instances as the proposed algorithm. The proposed algorithm permits to create new instances and evaluate the discriminative power of these new instances in the data set. The region where the instances are created (region of SV) retains valuable information, but is necessary to use a GA to guide the search of best instances.

3 Methodology

The proposed algorithm is based in the sparse property of SVM, where the solution is given for a small subset from the original data-set called Support Vectors (SV). Formally, given a data set $\{(x_i, y_i)\}_{i=1}^{n}$ and separating hyperplane $f(x) = w_i^T x + b = 0$, the shortest distance from separating hyperplanes to the closest positive example and closest negative example in the non separable cases are given by

$$\gamma_+ = \min \gamma_i, \forall \gamma_i \in class + 1 \tag{1}$$

$$\gamma_- = \min \gamma_i, \forall \gamma_i \in class - 1 \tag{2}$$

where γ_i is given by

$$\frac{y_i(w_i^T K \langle x_i \cdot x_j \rangle + b_i)}{\|w\|} \tag{3}$$

Margin is the optimal separating hyperplane obtained by training a SVM and it is given by $\gamma = \gamma_+ + \gamma_-$. This algorithm takes advantage of this fact, the key idea of this model is to introduce artificial instances from positive SV, It permits not only modify the margin also modify the region of the minority class and decrease the skew of the margin, but also improve the generalization capacity. Is clear that, introduce new synthetic instances in this region can affect negatively the SVM performance by introducing noise in the data-set. However, in this paper we use a GA to guide the search of the best regions and include just the best data instances in the margin region. Figure 1 shows the framework of the proposed method.

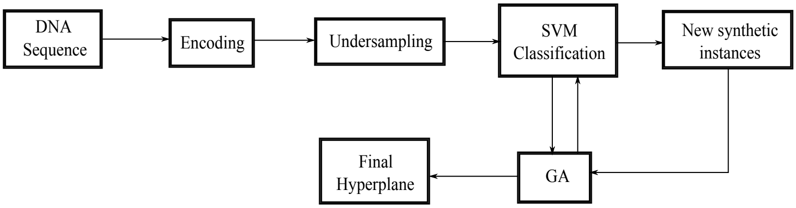

Fig. 1. Stages of the proposed algorithm

3.1 DNA Encoding

DNA sequences are given as strings of nucleotides and is necessary to encode it. Sparse encoding is a widely used encoding schema which represents each nucleotide with four bits: $A \rightarrow 1000, C \rightarrow 0100, G \rightarrow 0010$ and $T \rightarrow 0001$ [11]. We use 18 additional features with the sparse encoding schema. The first 16 components define the nucleotide pairs into a DNA sequence, which are defined by $\beta = \{(x_{AA}), (x_{AC}), (x_{AG}), (x_{AT}),\ldots,(x_{TA}),(x_{TC}),(x_{TG}),(x_{TT})\}$. When some nucleotide pair is in the sequence, it is marked with 1 and an absence of this pair is marked with 0. The last two components correspond to the informative function of each triples in the sequence ranked by their *F-value*. For each triple, we specify its location relative (pre and post) and its mean frequency between exons and decoys $\mu_k^+ - \mu_k^-$ respectively.

The *F-value* criterium is that used by Golub et al [12]. For each triple x_k,$k = 1, ..., n$, we calculated the mean $\mu_k^+(\mu_k^-)$ and the standard deviation $\sigma_k^+(\sigma_k^-)$ using positive and negative examples. The *F-value* criterium is given by

$$F(x_k) = \left| \frac{\mu_k^+ - \mu_k^-}{\sigma_k^+ + \sigma_k^-} \right| \tag{4}$$

where x_k is the $k - esime$ triple, the *F-value* serves as a simple heuristic for ranking the triples according to how well they discriminate. The last point in the vector is represented by the relative presence of each triple of nucleotides.

This encoding schema allows to obtain the nucleotides of each sequence, showing the importance of some pairs in the sequence, and obtaining the importance of each triple at the begin and at the end of each sequence.

3.2 Classification Algorithm

The first step in the proposed algorithm consists in encode the DNA sequence, we use the method described early. In the next step, the algorithm obtains subsets from the entire data-set. To separate input data set, 70% of examples from data set are selected as training set labeled as tr. We select tr with 70%, tf 15% and te 15% of input data maintaining almost equal proportion in class distribution over the data. For instance, if there are two class values (say X^- and X^+) in a classification problem P with 1000 examples in total, and the number of examples of class-types: X^- and X^+ are respectively 800 and 200. Then, 560 and 140 examples of class-types X_{tr}^- and X_{tr}^+ respectively are assumed to be included into tr by random selection, and X_{tf}^-,X_{tf}^+, X_{te}^- and X_{te}^+ with 120, 30, 120 and 30 examples respectively. Figure 1 show the steps of proposed algorithm which are described in detail in the algorithms 1 and 2. X_{tr}^+ and X_{tr}^- are used to train a SVM and to find an introductory hyperplane H_1 (X_r^+, X_r^-), from H_1 we obtain the SV x_{svi}^- and x_{svi}^+ and generate new synthetic examples from it. We use the SMOTE technique to generate the first population of new synthetic instance x_{svg}. which is given by $x_{svg} = x_{svi}^+ + \delta \cdot (x_{svi-n}^+ - x_{svi}^+)$, where x_{svg} denotes one synthetic instance, x_{svi-n}^+ is the nearest neighbors of x_{svi}^+ in the positive class, and $\delta \in [0, 1]$. This procedure is repeated for all the positive instances. The initial

Algorithm 1. General SVM classification procedure

Input: Nucleotides Sequence **Output**: Improved hyperplane $H_f : (X^+_{te}, X^-_{te})$

```
1. Encode the nucleotides sequences $\{x_i \in X : y = \pm 1\}, i = 1, \ldots, n$
2. From $X^+$ and $X^-$ obtain $X^+_{tr}$, $X^-_{tr}$, $X^+_{tf}, X^-_{tf}$, $X^+_{te}$, $X^-_{te}$ with 70%, 15% and 15%
   respectively
3. Train $SVM$ with $\left(X^+_{tr}, X^-_{tr}\right) \rightarrow H_1$
4. Obtain SVs $x^-_{svi}$ and $x^+_{svi}$ from $H_1$
5. Obtain initial population according to (3.2).
6. Obtain best data points $(X^+_{GA}, X^-_{GA})$ using the GA (Algorithm 2)
7. Obtain final hyperplane $trainSVM(X^+_{GA}, X^-_{GA}) \rightarrow H_f$
```

population is conformed by $x^-_{svi} \cup x^+_{svj} \cup x_{svg}$. It is manipulated using several genetic operators to improve the population in each iteration and optimizing the solution, i.e. DNA sequences are slightly modified from the DNA sequences with best discrimination power improving the classifier performance, this process is obtained by the GA defined in algorithm 2.

Second algorithm describes the functioning of the GA. We used a gray coding to represent each individual in the population and the fitness function.

Genetic operators can find a solution in a small space by crossover operators, and explore new areas in the space by mutation operators. The fitness function ensures the evolution towards optimization by the fitness score for each DNA sequence with high discriminative power in the population. The process continues until a predefined termination criterion has been met.

In the proposed technique, we use the F-measure as fitness, it provides a way to arrive the search solutions, and also controls the selection process. F-measure is defined by

$$\frac{2 \times precision \times recall}{precision + recall} \tag{5}$$

where precision $= \frac{TP}{TP+FP}$ and recall $=\frac{TP}{TP+FN}$, TP represents true positive rate defined by the fraction of true positives out of the positives and FP false positive rate defined by the fraction of false positives out of the negatives.

Selection is based in ranking selection with elite preserving. Each individual survives in the next generation in proportion to the rank of its fitness value. The best individual in the population is made to remain to the next generation in order to prevent the best individual from being eliminated by stochastic genetic drifts.

In the proposed algorithm, we used crossover and mutation operators. Crossover operator unifies the genetic information of two individuals (parents), obtained by selection operator, and creates two new individuals (children) called as offspring. We use two points crossover. A crossover operator permits the fitness function to evolve towards optimization. The mutation operator helps to find the global optimal solution to the problem. It is called exploration operator. We use a crossover probability of $p_c = 0.9$, and a mutation probability

Algorithm 2. GA to generate artificial data of the minority class

Input: Initial population $X_{svg} = (x_{svg1}, x_{svg2}, \ldots, x_{svgm})$, Max generation. **Output:** Best data instances (X^{+}_{GA}, X^{-}_{GA})

```
1. m(k)=m(0)=m
2. for i=1 to m(k)
3.     $H_a \leftarrow trainSVM\left(x^{-}_{svi} \cup x^{-}_{svi} \cup x_{svg}(i)\right)$
4.     Obtain fitness from $H_a$ with $\left(X^{+}_{tf}, X^{-}_{tf}\right)$ by (5).
5. end for
6. Generate new population $X_{Nsvg}$ by selection, crossover and mutation.
7. Add the best individual in the current population $X_{svg}$ to the newly generated $X_{Nsvg}$ to form the next population.
8. $m(k)$ = size of new generation $X_{Nsvg}$
9. Return to 2 if the pre-specified stopping condition is not satisfied.
```

of $p_m = 1/n$, where n is the string length for Gray coded. Final hyperplane is obtained until a stop criterion has been met.

Classical methods cannot decide which new instances will improve the SVM performance in imbalanced data-sets, because the search space is often huge, complex or poorly understood. GA has the ability to explore large and new areas. Finding new instances with discriminative power can be considered a GA search problem. The crossover and mutation operators realize the search exploratory and exploitative respectively. Thus, to use GA improves the SVM performance by generating artificial instances. The new instances obtained by the GA (X^{+}_{GA}, X^{-}_{GA}) contain information with high discriminative power helping to increase the classifier performance.

4 Experimental Results

We conducted experiments on some imbalanced and balanced intron-exon data-sets taken from $http : //www.raetschlab.org/suppl/MITBookSplice/files/$, $www.archive.ics.uci.edu$ and $http : //big.crg.cat/bioinformatics_{a}nd_{g}enomics/$ Table 1 shows details of these data-sets. We compared our method against: Under-Sampling, Over-Sampling and SMOTE techniques. The proposed method and the methods before mentioned are implemented in Matlab. To evaluate classifiers on skewed data-sets, require to use an adequate metric. We report the results with True Positive Rate (TPR), False Positive Rate (FPR), Area Under the Curve (AUC) and F-measure metrics. In all the experiments, we used 10-fold cross validation.

Table 1, shows data-sets used in experimental results, length of DNA sequence (ls), imbalance ratio (r) and size of exons, acceptors, donor (positive instances) and introns, decoys (negative instances).

Table 2, shows the results obtained in our experiments. The first column shows the data-set used and next columns show the experimental results obtained over

Table 1. Imbalanced ratio and size of the data-sets used in experimetal results

Dataset	*ls*	*size*	*r*
Nobgrors (No_23)	23	111827	1:1
Nobgrors (No_09)	9	110824	1:1
Nobgrors (Starts)	20	9299	1:1
Nobgrors (Stops)	15	11077	1:1
Acc_23 (Ac_23)	23	124728/374184	2:1
Acc_39 (Ac_39)	39	120000/360000	2:1
Donor_9 (Do_09)	9	120000/360000	2:1
Genbank64 (EI_60)	60	767/2422	3.15:1
Genbank64 (IE_60)	60	768/2421	3.15:1
C. elegans (Ac_60)	60	2785/91546	31:1
C. elegans (Do_60)	60	2785/89163	31:1

the 11 datasets with four different metrics measure methods, as well as the results obtained using the proposed method, Under-sampling, Over-sampling and SMOTE approaches. The best result for each classifier is highlighted in bold. We report the average on 30 runs for the proposed method. In all the cases, the proposed method got the highest F-measure (we used F-measure as fitness function in the GA), it suggests that the GA works well as a search engine that helps to find perfectly what new instances improve the classifier performance. In experimental results obtained not only F-measure performance is better, but also sometimes AUC-ROC and TP measures are improved. Moreover, the improvement can be carried in balanced data-set (Table 2).

The experimental results show that the proposed algorithm helps to improve the classification accuracy. The proposed algorithm helps to reduce the false positive rate (see Table 2 -EI_60, Ac_60, Do_60, Ac_23-) or helps to increase the true positives rate (IE_60, Do_60, Ac_23, No_09) with by adding artificial data. The improvement in the performance depends directly of the fitness function used. To use F-measure as fitness function helps to improve effectively FP rate too, but sometimes to improve can affect the AUC-ROC and TP rate due to imbal-

Table 2. Comparison of the proposed method (PM) with some measure metrics against other techniques for imbalanced data-sets

Measure	Undersampling				Oversampling				Smote				PM			
	TP	*FP*	*ROC*	*Fm*	*TP*	*FP*	*ROC*	*Fm*	*TP*	*FP*	*ROC*	*Fm*	*TP*	*FP*	*ROC*	*Fm*
EI_60	0.979	0.097	0.941	0.918	0.992	0.037	0.983	0.968	0.99	0.038	0.99	0.965	0.995	**0.009**	0.994	**0.99**
IE_60	0.905	0.015	0.972	0.919	0.939	**0.005**	0.983	0.929	0.941	0.012	0.978	0.946	0.973	0.029	0.979	**0.958**
Ac_60	0.958	0.359	0.856	0.96	0.974	0.318	0.917	0.963	0.965	0.142	0.97	0.973	0.969	**0.128**	0.983	**0.973**
Do_60	0.952	0.335	0.807	0.953	0.972	0.357	0.861	0.93	0.972	0.323	0.971	0.963	0.976	**0.243**	0.974	**0.977**
Ac_23	0.593	0.203	0.802	0.580	0.577	0.212	0.754	0.564	0.589	0.205	0.801	0.573	0.604	**0.198**	0.788	**0.593**
Ac_39	0.411	0.245	0.676	0.446	0.42	0.223	0.726	0.451	0.433	**0.222**	0.732	0.46	0.468	0.25	0.712	**0.493**
Do_09	0.628	0.187	0.777	0.596	0.614	0.193	0.777	0.610	0.631	0.175	0.812	0.598	0.618	**0.171**	0.814	**0.605**
No_23	0.926	0.104	0.926	0.926	0.894	0.107	0.954	0.894	0.914	0.091	0.969	0.91	0.914	**0.087**	0.969	**0.934**
No_09	0.925	0.075	0.974	0.925	0.938	**0.051**	0.970	0.938	0.929	0.071	0.976	0.929	0.943	0.057	0.973	**0.942**
Starts	0.753	0.247	0.833	0.753	0.752	0.248	0.836	0.752	0.77	**0.228**	0.850	0.770	0.827	0.230	0.857	**0.832**
Stops	0.629	0.371	0.629	0.629	0.611	0.389	0.658	0.611	0.63	**0.370**	0.684	0.630	0.635	0.379	0.688	**0.640**

ance ratio. Therefore, to obtain a fitness function that improves the measures whithout loss in a metric on imbalanced data-sets can be a future research work.

5 Conclusions

In this paper, we present a novel SVM classification approach for detection of splice sites. The proposed approach obtains new synthetic instances from the SVs obtained in a first stage and includes just the instances that improve the SVM performance in the data-set. The algorithm uses a GA to evaluate and obtain better instances in each iteration. Experiments done with DNA sequences, show that the information adjoined by the synthetic instances, help to improve the SVM performance. However, the cost of evaluating each solution in the population is very high and despite the good accuracy obtained its complexity is prohibitive in large data sets.

References

1. Baten, A., Chang, B., Halgamuge, S., Li, J.: Splice site identification using probabilistic parameters and SVM classification. BMC Bioinformatics 7, S15 (2006)
2. Yiming, C., Robert, M.M., Bin, T.: Prediction of mRNA polyadenylation sites by support vector machine. Bioinformatics 22(19), 2320–2325 (2006)
3. Damaevicius, R.: Splice Site Recognition in DNA Sequences Using K-mer Frequency Based Mapping for SVM with Power Series Kernel. In: CISIS 2008, pp. 687–692 (2008)
4. Jing, X., Doina, C., Susan, B.: Exploring Alternative Splicing Features Using SVM. In: Proc. 2008 IEEE Int. Conf. on Bioinf. and Biomed, pp. 231–238 (2008)
5. Chawla, N., Bowyer, K., Hall, L.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 321–357 (2002)
6. Nguyen, H., Cooper, E., Kamei, K.: Borderline over-sampling for imbalanced data classification. Int. J. Knowl. Eng. Soft Data Paradigm 3(1), 4–21 (2011)
7. Zou, S., Huang, Y., Wang, Y., Wang, J., Zhou, C.: SVM learning from imbalanced data by GA sampling for protein domain prediction. In: ICYCS 2008, pp. 982–987 (2008)
8. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. International Journal of Approximate Reasoning 50(3), 561–577 (2009)
9. García, S., Derrac, J., Triguero, I., Carmona, C.J., Herrera, F.: Evolutionary-based selection of generalized instances for imbalanced classification. Knowledge-Based Systems 25(1), 3–12 (2012)
10. Haibo, H., Garcia, E.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21(9), 1263–1284 (2009)
11. Zhang, X.H.-F., Heller, K.A., Hefter, I., Leslie, C.S.: Sequence Information for the Splicing of Human Pre-mRNA Identified by SVM Classification. Genome Research 13, 2637–2650 (2003)
12. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)