An Unsupervised Learning Approach to Resolving the Data Imbalanced Issue in Supervised Learning Problems in Functional Genomics

Kihoon Yoon, Stephen Kwek
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX 78249
{kyoon, kwek}@cs.utsa.edu

Abstract

Learning from imbalanced data occurs very frequently in functional genomic applications. One positive example to thousands of negative instances is common in scientific applications. Unfortunately, traditional machine learning treats the extremely small instances as noise. The standard approach for this difficulty is balancing training data by resampling them. However, this results in high false positive predictions. Hence, we propose preprocessing majority instances by partitioning them into clusters. This greatly reduces the ambiguity between minority instances and instances in each cluster. For moderately high imbalance ratio and low in-class complexity, our technique gives better prediction accuracy than undersampling method. For extreme imbalance ratio like splice site prediction problem, we demonstrate that this technique serves as a good filter with almost perfect recall that reduces the amount of imbalance so that traditional classification techniques can be deployed and yield significant improvements over previous predictor. We also show that the technique works for subcellular localization and post-translational modification site prediction problems.

1 Introduction

Recent technological advances enable biologists to collect huge amount of genomic data by using automated DNA sequencers, microarrays that generate gene expressions information of an entire organism, and other advanced techniques. These data contain valuable information that may lead to treatments of deadly disease and improve our quality of life. Although, in principle, machine learning techniques can serve as valuable tools for analyzing genomic data, some surveys indicated that the results are far from idea [2]. In many genomic applications, we are faced with the challenging issue of extremely high imbalanced data where we may see one positive instance (e.g. splice site) only after having seen thousands of negative instances. Henceforth, we assume the minority class is the positive class. Similarly, in the area of computer security, most traces in computer system logs are normal non-malicious usage and hence training data for building an automatic intrusion detection system is highly imbalanced. Standard treatment of such imbalance is to undersample the majority class to obtain the more balanced training and test instances. Such undersampling method makes the problems more tractable and yields good accuracy on the test instances. Unfortunately, the classifiers constructed often make too many false-positive predictions when deployed since the data they are trained on are far from the actual real-world imbalance distribution. Here, we propose a better technique for imbalance reduction by considering the entire data sets so that there is information loss is minimized.

In this paper, we proposed a technique for dealing with the imbalance data problem when the majority class exhibits moderately low in-class complexity. Although the imbalance ratios in many functional genomic applications are extremely high, fortunately the in-class complexity of the majority class tends to be moderately low. This allows us to partition the majority instances into dense clusters. We construct a base classifier for each cluster to distinguish the majority instances in it from all the minority instances. The data set for constructing the base classifier besides being more balanced, has a lower boundary complexity (since the majority class instances form a tight cluster). More importantly, unlike traditional undersampling method, we use all the majority instances so there is no information loss.

We tested our proposed technique on a sample of three representative functional genomic problems:

splice site, protein subcellular localization and phosphorylation site prediction problems. Briefly, in the splice splice site prediction problem, a gene can be viewed as a sequence of 4 letters (nucleotides) A, G, T and C. Each gene in eukaryotes consists of alternating segments of intron and exon regions. After the transcription process, the introns are spliced out and discarded while the exons are concatenated to form the messenger RNA (mRNA). As the name suggest, splice site prediction problem problem is to determine where splicing occurs. For human genome, the imbalance ratio of splice sites to non-splice sites is extremely high, possibly one to many thousands. Splice site prediction is a very important problem as it is the first step toward a cDNA library construction which is a working set of genes. The mRNA then goes through a translation process to produce protein which consists of a sequence of (20 possible) amino acids. The protein is then transported to its designated subcellular location to perform its function or interact with other proteins. The subcellular localization problem is to determine final destination of protein within the cell. This information will provide valuable clues to the functions of the proteins and how they interact. Thus, it also provides some clue to the function of a particular gene. As there are possibly more than hundred possible locations, it is a multi-class problem. The class imbalance problem occurs because of the different number of available instances for each class. Most proteins also undergo some post-translational modifications (PTMs) (phosphorylation, glycosylation, sulfation, and ubiquitination) which are functionally relevant. Among the possible PTMs, phosphorylation is the most studied and perhaps the most important. In phosphorylation a phosphate group is transferred from Adenosine Tri-Phosphate (ATP) to the hydroxyl side chains of serine, threonine, or tyrosine amino acid residues in the protein sequence. The phosphorylation site prediction problem is to predict where phosphorylations occur in a given protein sequence. Due to page limitation, we shall restrict focus our discussion the result obtained on splice site prediction problem. A more detailed discussion on results on all the three problems will be presented in a longer version of this extended abstract.

2 Related Work in Imbalanced Data

Most supervised learning algorithms tend to focus on obtaining high accuracy on the observed labeled training data. To further aggravate this difficulty, almost all algorithms tend to follow the Occam's razor principle (or related minimum description length MDL principle) where there is a preference toward simple hypothesis [13]. Short decision trees and neural networks with small weights are preferred. The underlying assumption here is that events (instances) that occur infrequently are considered as noise. This further discriminates against the minority class so as to achieve high overall prediction accuracy. For highly imbalance data, the classifiers constructed using these algorithms would simply predict negative all the time and achieve almost 100% accuracy! This is nonsensical for applications in functional genomic (and computer security) where the aims are to detect minority instances within a certain reasonable tolerance of false positive mistakes.

Various approaches [8, 14] have been proposed to tackle the challenge posed by the imbalance ratio problem. These approaches fall into two different categories, namely weighting or resampling based methods. Weighting methods either assign heavier weights to the minority training instances or penalties for misclassifications of minority instances [1, 7, 19, 21]. The other way is to preprocess training data to minimize discrepancy between the classes. Oversampling [5] the minority class and undersampling [11] the majority class are the data level approaches. Ling and Li [12] combining oversampling and undersampling methods but did not achieve significant improvement in the "lift index" metric that they used. Both methods effectively change the training distribution to one that no longer resemble the original (highly imbalance) distribution, resulting in overfitting. Other important related works similar to resampling approaches are to focus on solving small disjuncts problem within each class. Japkowicz [9, 10] discussed about the cause for lower performance in standard classifiers is actually small disjuncts of within-class. These works agree with what we observed from our experiments.

3 Proposed Approach

The intuition behind our approach is to build a filter to identify large number of majority instances without losing too many minority instances. This allows us to reduce the imbalance ratio which makes the learning task more tractable. Since minority instances are scarce, it is very crucial that any imbalance reduction procedure should try not to eliminate any minority instances from the original data. The idea for achieving this goal is to find as many clusters of majority instances as possible that do not contain any minority instance or at most very few minority instances. In particularly, we will like to be able to determine majority instances that are far away from the target boundary (and hence reduce the amount of imbalance) so that we can concentrate on distinguishing the more difficult boundary instances. Thus, the key is to find clusters

that consists purely (or almost purely) of majority instances. Therefore, we developed a supervised clustering algorithm with class purity maximization function.
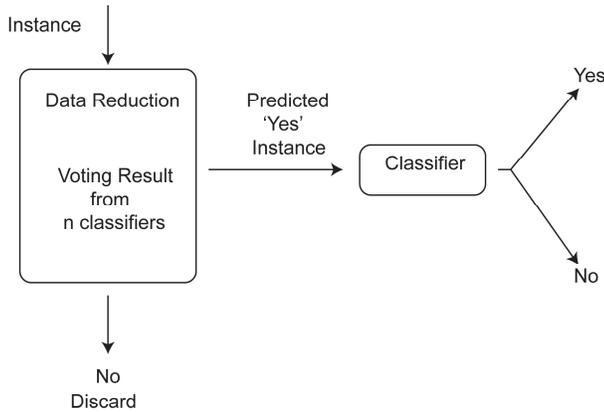
Figure 1. Overall procedure for imbalance reduction and final prediction scheme
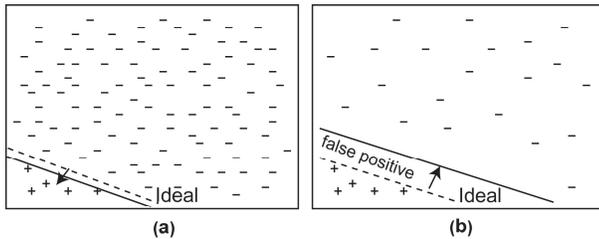
Figure 2. Illustration of imbalanced data and undersampling: (a)Imbalanced data set - decision boundary is shifted toward to minority class. (b)After undersampling - decision boundary move to majority class.

The Main Idea. Our CPM algorithm selects a pair of minority and majority instances as centers. The other instances are then partition into two subsets according to their nearest centers, with at least one subset having high class purity. This process is repeated recursively for each of the two subsets until we can no longer form two clusters, with at least one yielding higher class purity than its parent cluster. A collection of samples is then constructed by adding all minority instances to each non-pure cluster, and a decision tree is built for each sample. Figure 1 illustrates the overall imbalance reduction and classification step. Given an unlabeled test instance, we first run through the imbalance reduction process (i.e. CPM) to estimate the best possible cluster that it might be belong. If the instance belongs to a pure majority instance cluster, it is simply discarded (as a majority instance). Only those instances belonging to a non-pure cluster is passed onto the decision tree committee. If the majority voted it as a

minority instance, then it is filtered out to the final classifier, which is constructed using a neural network.

Effect of the dense majority instance clusters. A classifier that trains on the entire data set will encounter a lot of negative (majority) instances closed to the ideal boundary, simply because they are the majority class. This pushes the decision boundary toward the minority positive instances. When the ratio between majority and minority becomes larger, a classifier might treat minority instances as noisy (figure 2(a)). Figure 2(b) shows the decision boundary shifting after undersampling. Area between ideal and shifted decision boundary is responsible for false positive predictions. Unlike various undersampling techniques, clustering will split majority instances based on their distribution into meaningful clusters (Figure 3). The instances in a good cluster, by definition, tend to lie in a tight region. In this case, a classifier can find a decision boundary that favors more on minority class even though the number of majority instances is much higher. Another good characteristic is that the decision boundary of each classifier is dramatically different from each other. A negative instance that is wrongly classified as positive by a classifier may be corrected by the other classifiers (with different decision boundary).

Class Purity Maximization (CPM) Clustering. The CPM algorithm is shown in Figure 4. It calls itself recursively. It attempts to find a pair of centers, one being a minority instance while the other is the majority instance (Lined 3). Using these centers, it partitions all the instances into two clusters C_1 and C_2. If either of the clusters has class impurity less than its parent's impurity (Imp) then we have found our clusters. Here, the impurity of a set of instances is simply the proportion of minority instances. It then recursively partitions each of these clusters into subclusters (in Line 8 and 9). Thus, it forms a hierarchical clustering. If the impurity cannot be improved then we stop the recursion (Line 3). A slight detail that is missing in Figure 4 is that we require that the clusters cannot be too small. This is to avoid the extreme case of having singleton clusters which always have a purity of 1. The distance measure used is simply the Euclidean distance. CPM is quite different from Expectation Maximization (EM) Clustering, in the sense that CPM uses the class labels to decide how to partition the instances. Unlike EM, CPM does not estimate the parameters of the mixture of Gaussian distributions. One of the advantages of CPM is it runs much faster than EM algorithm.

Performance Measure. The general performance measure, (estimated) test error, is not a good metric for imbalanced data. For many important bioinformatics or
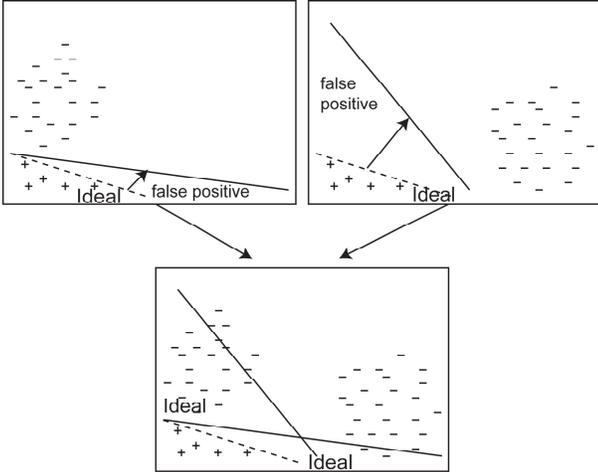
Figure 3. Effect of small and dense subsets - give more space to minority class. Any instances placed between relaxed decision boundary and minority instances will be predicted as a minority class.

computer security applications, the minority instances may be less than 1% of the entire data. By simply predicting according to the majority class, we can achieve more than 99% accuracy. Clearly such predictor is not useful at all. For applications with high imbalance ratio, we frequently want to recall as many minority instances as possible. Further, we want to be precise so that when we predict an unlabeled instance to be minority class instance, there is a good chance that we are right. These two goals are often contradictory goals and we need to strike a compromise. We use F-measure to measure the overall performance (as c compromise between recall and precision) of the algorithms studied. The exact definitions of the recall (R) and, precision (P) were first introduced in the information retrieval community. Recall (a.k.a. over-prediction) is defined as $R = \frac{CP}{TP} \times 100$ where CP is the number of instances that are correctly predicted as positive and TP is the number of actual positive instances. Precision (a.k.a. under-prediction) is defined as $P = \frac{CP}{PP} \times 100$ where PP is total number instances predicted as positive. As achieving high recall and high precision are often conflicting goal, we use F-measure as a measure of how good a "compromise" is being achieved. F-measure is defined as $F = 2 \times \left(\frac{R \times P}{R+P} \right)$ which is a harmonic mean between recall and precision. F-measure becomes zero if either R or P is zero. It becomes 1 when both R and P are 1. Ten-fold cross validation was used to estimate R and P for this paper.

```
Input: Imp: cluster impurity of parent cluster
       parent: parent cluster ID
Output: subclusters C_i rooted at parent
CPM(Imp, parent)
1.  impurity ← ∞
2.  while Imp <= impurity
3.      if all the instance pairs in parent were tested then return
4.      Pick a pair of majority and minority instances as centers
5.      Partition all instances into 2 clusters C_1 and C_2
            according to nearest center
6.      impurity ← min(impurity(C_1), impurity(C_2))
7.  end while
\\ Create subclusters
8.  CPM(impurity(C_1), C_1)
9.  CPM(impurity(C_2), C_2)
```

Figure 4. The CPM Algorithm

4 Results for Splice Site Prediction

For the splice site predictor to be useful, it is important to be able to 'recall' as many positive examples as possible but keep the 'precision' high (i.e. the true positive high). Similarly, in order for a splice site predictor to be useful for constructing a gene finding system, the recall has to be high so that it does not miss out too many undiscovered gene. In splice site prediction, the precision is slightly less important as we may be able to eliminate some of the false positive predictions by some other information (e.g., snRNAs and snRNPs interactions, promoter binding sites, transcription factor binding sites, etc.). Nevertheless, as far as possible, we still want the precision to be high so that it does not generate too many false positive gene predictions, which may render the eventual gene finding system useless. Alas, the two objectives, achieving high recall and precision, are often contradictory and we need to strike a compromise.

We compared our approach with three leading splice site predictors; performance comparisons were done by running their web-based programs on test sequences for fair comparisons.

- NNSplice[1] from Berkeley Drosophila Genome Project (BDGP) - NNSplice is a sub-process of the gene finding system, Genie.[17] Two separate neural networks were used to predict donor and acceptor sites based on dinucleotide frequencies.

- GeneSplicer[2] from the Institute for Genomic Research (TIGR) - GeneSplicer is a decision tree method using Maximal Dependence Decomposition and enhanced by Markov Models [15, 4].

---

[1]The program is accessible from http://www.fruitfly.org/.
[2]The   program   is   accessible   from http://www.tigr.org/tdb/GeneSplicer/gene_spl.html.

- SpliceView[3] from the Institute of Advanced Biomedical Technologies (ITBA) - SpliceView considers the signals from the consensus sequences of the boundary regions [18].

Although both our method and NNSplice share a commonality of using artificial neural networks (ANNs), we are able to reduce the amount of imbalance dramatically by using Expectation-Maximization (EM) clustering [6, ?, 16]. Further, we use a different feature construction method than all three programs. The resulting features are more indicative. As a result, we have a more accurate predictor. Two main goals were considered for designing the 'filter'. Firstly, in the construction of the filter, (unlike undersampling) we should consider the entire majority instances while making sure that the resulting filter has a high recall rate. This gives us better precision. Secondly, the resulting filter has to eliminate as many majority instances as possible without loosing any of the minority instances. The construction of the filter is as follows. The majority of the instances are clustered using the EM algorithm. After we find the clusters, we add the minority instances to each cluster. One ANN is then constructed per cluster using the sampled training data. For the final prediction of an unknown instance, the decision of the boundary site was made by voting from all ANNs. The detailed of our method will be described in methodology section. All our experiments were done with human gene sequences from TIGR web site[4], which consists of 155 human gene seqeunces of which we randomly select 55 test sequences to be our test cases. The reason for our data choice is that it would be unfair to train our classifier on the latest gene sequence data and compare it with the benchmark programs that are probably trained on smaller older data sets. As all the three programs are trained on different data sets, the best we could do in terms of fairness is to select data for GeneSplicer since it is readily available. Our result would have been better if we were to use the latest gene sequence data. Furthermore, all three programs allow us to specify that human DNA sequences were used for testing. Thus, their web-based programs are probably adapted for human gene predictions too. Our method serves as a good filter to eliminate significant number of majority instances. 97.5% of the majority instances were identified. Because preprocessed data for final classification is less imbalanced, the accuracy of the final predictions was improved dramatically. F-measure was improved by 39.8% on donor site predictions and

24.2% on acceptor site predictions relative to the best case from the three benchmark methods (See Table 1). We believe that our new technique on splice site prediction will lead to the construction of a better automated gene finding system.

From the results of the three existing methods, a clear trade-off trend between recall and precision could be noticed from the results (Table 1). As recall increases, precision tends to decrease. The trade-off seems to be unavoidable among the conventional methods. Now we would like to address which metric - recall or precision - is more important in evaluating performance of splice site predictions. SpliceView showed the highest recall among the three existing methods tested. Can we say that SpliceView is a better classifier? The answer is "not really". It is not reasonable to say that SpliceView is a more accurate method. For example, if a classifier simply predicts every instance as positives, the result will be '1' of recall, '0' of precision, '0' of F-measure. In the other extreme from the example above, the performance measure will be '0' of recall, '1' of precision, and '0' of F-measure. We probably do not trust the classifiers' predictions although we achieved perfect recall or precision. Precision must be in an acceptable range for a high recall to be meaningful. Therefore, we consider F-measure as the overall accuracy of a classifier.

Donor site predictions: The three existing methods showed clearly the problem of undersampling. Relatively high recall and very low precision are typical results from imbalanced data. However, the performance of our filtering approach has 39.8% higher on F-measure than the result of NNSplice on donor site predictions (Table 1). Both recall and precision were improved significantly with our filtering approach.

Acceptor site predictions: The recall of GeneSplicer was not degraded as much as the other methods. Filtering approach showed slightly lower recall rate than the recall rate of donor site predictions, but it showed the best result among all the tested methods here. With filtering approach, F-measures from acceptor predictions showed 24.2% improvement (Table 1). However, all the methods have lower precision as compared to the precision obtained for donor site prediction. This might indicate that acceptor sites do not have strong sequence information.

5 Other Bioinformatics Applications

Protein Subcellular Localization. We extracted from Swiss-Prot database 1450 human proteins: 644 cytoplasmic, 322 extracellular, 50 mitochondrial, and 1034 nucleus proteins (i.e 4 classes). The reason for looking at human only is to eliminate the possibility of hav-

---

[3]The program is accessible from http://l25.itba.mi.cnr.it/~webgene/wwwspliceview.html.

[4]The sequences can be obtained from http://www.tigr.org/software/traindata.shtml.

Site	Method	Homo sapiens		
		Recall	Precision	F-measure
Donor	NNSplice	74.3	21.8	33.7
	GeneSplicer	75.3	17.8	28.7
	SpliceView	94.4	6.9	12.8
	Filtering	97.3	59.0	73.5
Acceptor	NNSplice	64.3	14.8	24.1
	GeneSplicer	74.3	10.3	18.1
	SpliceView	93.8	3.9	7.5
	Filtering	92.3	32.7	48.3

Table 1. Performance comparisons among three existing methods and our approach.

Method	Recall	Precision	F-measure
Decision Tree	0.020	0.100	0.033
Undersampling	0.36	0.072	0.120
CPM	1.000	0.256	0.408
Filtered	0.635	0.392	0.458

(A) Protein Mitochondrial localization

Method	Recall	Precision	F-measure
Decision Tree	0.020	0.100	0.033
Undersampling	0.36	0.072	0.120
Final	0.635	0.392	0.458

(B) Phosphorylation Site Prediction

Table 2. Other Bioinformatic Applications

ing homologous proteins from other species which will often reveal the answer. For the classifier constructed to be useful, it is more relevant to know how well it does on protein where no close homologs exist. We constructed twelve numerical features out from these protein sequences (details omitted). We give special attention to the N-terminal and C-terminal because of the possible presence of signal peptides in these two ends that direct the protein to its destination. The problem of determining subcellular localization is not just a multi-class problem but it is also a multi-label problem. That is, some protein may have multiple localizations. While the problem of determining whether a protein is a cytoplasmic protein does not suffer severe imbalanced data problem, the ratio of mitochondrial proteins to non-mitochondrial proteins is 1:29. Simply performing a decision tree induction on the data gives a poor performance for the mitochondrial localization problem with recall and precision merely being 2% and 10% respectively. Undersampling to reduce the imbalanced ratio to 1:10 improve the performance to 36% The CPM algorithm is able to reduce the imbalance ratio by a factor of 4 (i.e precision is about 25%) without throwing away any of the positive examples. The final classifier using our approach has a higher recall and precision rates of 63.5% and 39.2% respectively.

Phosphorylation Site Prediction. The data were obtained from Human Protein Reference Database (HPRD). First we selected out the phosphorylated protein sequences which exist in Swiss-Prot database. Also negative examples were created from subsequences containing serine, threonine or tyrosine amino acid residue. The results are shown in Table 2(B) (more details will be provided in the full version of this paper.)

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In Proceedings of the 15th European Conference on Machine Learning (ECML), pages 39–50, 2004.

[2] J. Ashurst and J. Collins. Gene annotation: Prediction and testing. Annual Review of Genomics and Human Genetics, 4:69–88, 2003.

[3] C. Blake and C. Mertz. Uci repository of machine learning databases.

[4] Burge C, Karlin S, Prediction of complete gene structures in human genome DNA. J. Mol. Biol. 268, 78-94, 1997.

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

[6] Dempster AP, Laird NM, Rubin DB, Maximum-likelihood from incomplete data via the EM algorithm. J. Royal Statist. Soc Ser: B., 39, 1977.

[7] P. Domingos. How to get a free lunch: A simple cost model for machine learning applications. In Proceedings of the 2000 International Conference on Artificial Intelligence, 2000.

[8] N. Japkowicz. The class imbalance problem: Significance and strategies. In Proceedings of the 2000 International Conference on Artificial Intelligence, 2000.

[9] N. Japkowicz. Class imbalances: Are we focusing on the right issue? In Notes from the ICML Workshop on Learning from Imbalanced Data Sets II, 2003.

[10] Jo, T., Japkowicz, N.: Class Imbalances versus Small Disjuncts, ACM SIGKDD Exploration, Vol 6, No1. (2004) 40-49

[11] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In Proceedings of the 14th International Conference on Machine Learning, 1997.

[12] C. Ling and C. Li. Data mining for direct marketing problems and solutions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998.

[13] T. Mitchell. Machine Learning. The McGraw-Hill Companies, Inc., 1997.

[14] Nickerson, A., Japkowicz, N., Millos, E.: Using Unsupervised Learning to Guide Resampling in Imbalanced Data Sets. In: Proceedings of the 8th International Workshop on AI and Statistics (2001) 261-265

[15] Pertea M, Lin X, Salzberg SL, GeneSplicer : a new computational method for splice site prediction . Nucleic Acids Res. 29(5):1185-1190, 2001.

[16] Redner R, Walker H, Mixture densities, maximum likelihood and the EM algorithm. SIAM Review, 26(2), 1984.

[17] Reese MG, Eeckman FH, Kulp D, Haussler D, Improved splice site detection in Genie. Proceedings of the First Annual International Conference on Computational Molecular Biology (RE-COMB), Santa Fe, NM, ACM Press, New York, 1997.

[18] Rogozin IB, Milanesi L, Analysis of donor splice signals in different organisms. J. Mol. Evol., V.45, 50-59, 1997.

[19] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In Proceedings of the International Joint Conference on AI, 1999.

[20] I. Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann Publisher, 2000.

[21] Wu, G., Chang, E.: Class-Boundary Alignment for Imbalanced Dataset Learning. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC. (2003)