Fast Splice Site Classification Using Support Vector Machines in Imbalanced Data-sets

Jair Cervantes[1], Asdrúbal López Chau[2], Adrian Trueba Espinoza[1] and José Sergio Ruiz Castilla[1]
[1]Department of Computer Sciences, UAEM-Texcoco, Texcoco, MX 56259, México
chazarra17@gmail.com
[2]Department of Computer Sciences, CINVESTAV-IPN, México D.F. 07360, México.
achau@computacion.cs.cinvestav.mx

Abstract— *Splice sites prediction is an important objective of genome sequencing. In last years, careful attention has been paid in order to the improve the performance of the algorithms used, but the study of most feasible methods to improve the performance in large and imbalanced data-sets is still of immense importance. This paper presents a novel SVMs classification method which works with gene data, the proposed method reduces significantly the training time and obtain a high accuracy on huge and imbalanced data-sets. Experimental results show that the accuracy obtained by the proposed algorithm is slightly better (*98.9%*) in comparison with other SVMs implementations such as SMO (*98.6%*), LibSVM (*98.6%*), and Simple SVM (*98.2%*). Furthermore the proposed approach can be used in large and imbalanced data-sets obtaining high classification accuracy.*

Keywords: SVM, Splicing, Imbalanced data-sets

1. Introduction

The advances and development in DNA sequencing technologies have resulted in a impressive increase in the size of genomic sequences. This growth of sequence data demands effective techniques to processing huge amounts of biological information. Identifying genes is an important issue in bioinformatics, and the accurate identification of splice sites in DNA sequences plays one of the central roles of gene structural prediction in eukaryotic cells. An effective detection of splice sites requires the knowledge of characteristics, dependencies, relationship of nucleotides in the splice site surrounding region and an effective encoding method.

The classification of gene sequence into regions that code for genetic material and regions that do not is a challenging task in DNA sequence analysis. It is not an easy challenge. It is due to size of DNA sequences and sometimes regions that encode in proteins (exons) can be interrupted by regions that do not encode (introns). These sequences are characterized, however they are not clearly defined by local characteristics at splicing sites. Identifying exons into DNA sequences presents a computational challenge. In some organisms the introns are small regions and the splicing sites are fully characterized. However, in some other sequences, including human genome, it is a great problem to localize the correct transition between the regions that encode and the ones that not. Furthermore, the genes in many organisms splice of different way, which complicates considerably the task. On the other hand, splice sites fall into two categories: donor sites of introns and acceptor sites of introns. These sites display some characteristic patterns, e.g. 99% of donor sites begin with base pairs GT while 99% acceptor sites end with based pairs AG. However, not all locations with base pairs GT or AG are necessarily splice sites. Some occurrences of AG or GT occur outside of a gene or inside an exon. These are called decoys, because they do not indicate the presence of a splice site. Furthermore, the majority of gene data-sets are imbalanced and the bulk of classifiers generally perform poorly on imbalanced data-sets because making the classifier too specific may make it too sensitive to noise and more prone to learn an erroneous hypothesis. Another factor is that in imbalanced data-sets an instance can be treated as noise and ignored completely by the classifier. Due to it, efficient methods and fast techniques that aims to tackle this problem are necessary.

In this paper, we use a novel approach for train and predict acceptor and donor splice sites in huge and imbalanced data-sets using Support Vector Machines (SVM). SVM has received considerable attention due to its optimal solution, discriminative power and performance. Lately some SVM classification algorithms have been used in splice site detection with acceptable accuracies [1] [2] [3] [10] [12] [14]. Cheng et al [2] use SVMs in order to predict mRNA polyadenylation sites [poly(A) sites] the method can help identify genes, define gene boundaries, and elucidate regulatory mechanisms. Damaevicius [3] and Xia [12] use SVMs in order to detect splice-junction (intron-exon or exon-intron) sites in DNA sequences. In [14] the authors use a SVM in order to discover sequence information that could be used to distinguish real exons from pseudo exons. Baten et al. [1] make use of SVM with polynomial kernel in order to obtain an effective detection of splice sites, the authors used a first order Markov model as a pre-processing step of DNA sequences. Some authors have been using SVM for the detection of splicing sites. However, when faced SVM with imbalanced data-sets the performance of SVM drops

significantly. Other important disadvantage of SVMs is due to memory requirements grows with square of input data points, so training complexity of SVMs is highly dependent on the size of a data-set.

This paper presents a novel splice sites fast classification model using SVM for imbalanced data-sets. The proposed method reduces intelligently the input data-set, tackling the problem of imbalanced data-sets with SVM and reducing significatively the training time. The rest of the paper is organized as following: Section II reviews some preliminaries of SVM. Section III focuses on explaining the methodology of proposed SVM classification algorithm. Section IV shows experimental results. Conclusions are given in Section V.

2. Preliminaries

2.1 Support Vector Machines

Support Vector Machines aim at estimating an optimal classification function using labeled training data from X_{tr} such that f will correctly classify unseen examples (test data). In our case, input space X will contain simple representations of sequences A,C, G, T while corresponds to true splice and decoy sites, respectively. Considering binary classification, we assume that a training set X_{tr} is given as:

$$(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n) \tag{1}$$

i.e. $X_{tr} = \{x_i, y_i\}_{i=1}^{n}$ where $x_i \in R^d$ and $y_i \in (+1, -1)$ is the label of example x_i. The generated classification function can be written as

$$g(x) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b\right) \tag{2}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_l]$ is the input data, α_i and y_i are Lagrange multipliers. SVM training obtain a set of real-valued weights $\alpha_i \geq 0$ such the normal vector can be expressed as a linear combination of input vectors, $w = \sum_{i=1}^{n} y_i \alpha_i x_i$. Input vectors x_i having non-zero weight are called support vectors and they determine the SVM solution. Once the SVM is trained, a new object x can be classified using (2). The vector $\mathbf{x}_i$ is shown only in the way of inner product. The $\alpha_i s$ are Lagrange multipliers and b is the usual bias which are the result of SVM training.

The principal disadvantage of SVMs is due to complexity that grows with square of input data points. Sequential minimal optimization (SMO) breaks the large Quadratic Programming (QP) problem into a series of smallest possible QP problems [9]. These small QP problems can be solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The memory required by SMO is linear in the training set size, which allows SMO to handle very large training sets [9]. A requirement in (3) is $\sum_{i=1}^{n} \alpha_i y_i = 0$, it is enforced throughout the iterations and implies that the smallest number of multipliers can be optimized at each step is two. At each step SMO chooses two elements α_i and α_j to jointly optimize, it finds the optimal values for these two parameters while all others are fixed. The choice of the two points is determined by a heuristic algorithm, the optimization of the two multipliers is performed analytically.

2.2 Methods for imbalanced classification

The classification of imbalanced data-sets is a crucial problem in machine learning because it normally causes negative effects on the performance of a classification method. There are two methods to tackle this problem. At the data level, re-sampling training data is a popular solution to classification of imbalanced data-sets, the most important techniques used at the data level or by preprocessing data exist are Over-sampling and Under-sampling.

2.2.1 Over-sampling

This technique over samples the minority class to balance the class distribution of a training data-set. Specifically, the minority class is over sampled until the size is equal to the size of the maximum class. Over sampling is a popular technique tackle some imbalanced classification problems. However in SVM increases significatively the training time.

2.2.2 Under-sampling

This technique under samples the majority class to balance the class distribution of a training data-set. Specifically, the majority class is under sampled until the size is equal to the size of the minimum class. Some previous studies showed that under sampling was better than over sampling in classification of imbalanced data-sets. It should also noted that under sampling usually reduces the training time but discard some potentially useful training examples and may degrade the performance of the classifier.

On the other hand, at the algorithmic level, weighting training data assign a larger weight to the minority class in order to balance the input data-set.

3. Methodology

In the following, we describe the methodology for splice sites recognition. Given a sequence, the proposed algorithm starts by encoding the DNA sequences. DNA encoding is crucial to successful intron/exon prediction. The next step is done by training SVMs on the training data and tuning their hyperparameters on the validation data.

3.1 DNA Encoding

DNA encoding has been extensively researched in recent years [5][8]. Each technique is based on the most important features to be shown. Sparse encoding is a widely used encoding schema which represents each nucleotide with 4 bits: $A \rightarrow 1000, C \rightarrow 0100, G \rightarrow 0010$ and $T \rightarrow 0001$ [7].

Suppose we have a DNA sequence of AGGCGTATGAGG. With the sparse encoding, the sequence is represented as: 1000 | 0010 | 0010 | 0100 | 0010 | 0001 | 1000 | 0001 | 0010 | 1000 | 0010 | 0010. where | is a virtual separator used to illustrate the example.

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 as $\beta = \{(x_{AA}), (x_{AC}), (x_{AG}), (x_{AT}),\ldots,(x_{TA}),(x_{TC}),(x_{TG}),(x_{TT})\}$. When some nucleotide pair is in the sequence, it is marked with 1 and an absence of this pair is marked with 0. The DNA sequence, AGGCGTATGAGG can be encoding by this schema as: 0 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0.

The last two components correspond to the informative function of each triples in the sequence ranked by their $F\text{-}value$. For each triple, we specify its location relative (pre and post) and its mean frequency among exons and decoys $\mu_k^+ - \mu_k^-$ respectively.

The $F\text{-}value$ criterium is that used by Golub et al [6]. 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 $F\text{-}value$ criterium is given by

$$F(x_k) = \left| \frac{\mu_k^+ - \mu_k^-}{\sigma_k^+ + \sigma_k^-} \right| \tag{3}$$

where x_k is the $k - esime$ triple, the $F\text{-}value$ 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. If this sequence AGGCGTATGAGG belong to data-set of example 1 can be encoding by this schema as: $\gamma = \{f_{AGG}, f_{AGG}\} = \{0.231, 0.231\}$, where γ is computed using the $F\text{-}value$ criterium. The $F\text{-}value$ is repeated because the triple AGG is in the sequence pre and post $(AGG...CGTATG...\ AGG)$.

The proposed encoding schema allows to obtain the nucleotides of each sequence, encoding the pairs show the importance of some pairs in the sequence, and obtain the importance of each triple at the begin and at the end of each sequence. The previous DNA sequence can be encoding by the complete schema as: 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0 |0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0 | 0.231, 0.231. Where | is a virtual separator which objective is just illustrate the three techniques used. With the proposed encoding schema SVM can use the features and discriminate between the categories.

1000 | 0010 | 0010 | 0100 | 0010 | 0001 | 1000 | 0001 | 0010 | 1000 | 0010 | 0010.

3.2 Classification algorithm

SVM classification aim at estimating a classification function $H : X \rightarrow \{\pm 1\}$ using labeled training data from $X \times \{\pm 1\}$ such that H will correctly classify unseen examples (testing data). In our case, input space X will contain simple representations of sequences $\{A, C, G, T\}^N$, while ± 1 corresponds to true splice and decoy sites, respectively.

Learning with imbalanced data is one of the recent challenges in machine learning. There are some techniques proposed in order to find a solution for this problem, such as the application of a preprocessing stage focused on balancing data, in preprocessing data two tendencies exist: reduce the set of examples (under-sampling) or replicate minority class examples (over-sampling). Over-sampling of minority classes can be done by re-sampling the examples from minority classes thus increasing the bias of the learned classifier towards them and increasing the accuracy on minority classes. Under-sampling with imbalanced data-sets could be considered as a prototype selection procedure which the majority class can reduce the bias of the learned classifier towards it and thus improve the accuracy on the minority classes. In this paper, we used under-sampling, the selection process under-sample the majority class in order to remove noisy and redundant training instances however the proposed algorithm recover the most important data points and the outliers keeping all the information in the training data-set. Our goal in this case is to retain and use this information, because even though under-sampling the majority class provokes an inherent loss of valuable information.

INPUT: X_{EDS}
//X_{EDS}; Entire Imbalanced data-set
OUTPUT: $H_f : \{x_i \in x_{EDS} : x_i \in SVs\}$;
Initialization;
1. $X_r^+ \leftarrow 0$ /* training data-set with positive labels begins empty */
2. $X_r^- \leftarrow 0$ /* training data-set with negative labels begins empty */
3. $X_r^+ \leftarrow \{x_i \in x_{EDS} : y_i = +1\}\ , i = 1, 2, ..., p;$
4. $X^- \leftarrow$ $get\_RandomSampling\ \{x_i \in x_{EDS} : y_i = -1\}\ , i = 1, 2, ..., p;$
5. Obtain outliers (O^+, O^-) using Algorithm 2;
6. Obtain $\left(X_f^+, X_f^-\right)$ using Algorithm 2;
7. $X_{RD}^+ \leftarrow \left(X_f^+ \cup O^+\right)$;
8. $X_{RD}^- \leftarrow \left(X_f^- \cup O^-\right)$;
9. $H_f\left(X_{RD}^+, X_{RD}^-\right) \leftarrow trainSVM\left(X_{RD}^+, X_{RD}^-\right)$;
10. **return** $H_f\left(X_{RD}^+, X_{RD}^-\right)$

Algorithm 1: SVM training

In this paper, we propose a fast SVM algorithm to work with imbalanced data-sets. The proposed algorithm is based in the sparse property of SVM When using SVM for classification, in most cases has been found that after the

training, the number of SV is very small compared with the number of elements of the training data-set, so taking advantage of this fact, the basic idea behind the reduction of the training data-set strategy is to select elements most likely to be SV. The Algorithm 1 shows the general process to detect splices sites or decoys by our technique.

The first step in the proposed algorithm consists in obtain the minority class which contains p instances, in the imbalanced data-set and label them as positive X_r^+, we also randomly select from the entire data-set X_{EDS} and label them as negative X_r^-.

X_r^+ and X_r^- are used by the algorithm 2 in order to find an introductory hyperplane $H_1\left(X_r^+, X_r^-\right)$, from H_1 we obtain SV, non-SV and $O^+ \cup O^-$ by testing the hyperplane obtained in the entire data-set, the data-set $O^+ \cup O^-$ contains all data points that are misclassified with H_1 and contains valuable information in this process. In order to obtain the most important data points in the entire data-set we train a SVM and obtain $H_2\left(X_{ch}^+, X_{ch}^-\right)$ where $X_{ch}^+ and X_{ch}^-$ represent the data points that are SV and non SV with H_1 respectively. Testing H_2 in the entire data-set we obtain the most important data points and eliminate redundant training instances.

The small size of (X_{RD}^+, X_{RD}^-) contributes to speed up the training of the proposed method. Furthermore, the reduced data-set obtained contains the most important data points in the entire data-set.

$\cap$

INPUT: X_r^+, X_r^+
//X_{Tr}; Training data-set
OUTPUT: X_f^+, X_f^-, O^+, O^-;
Initialization;
1. $H_1\left(X_r^+, X_r^-\right) \leftarrow trainSVM\left(X_r^+, X_r^-\right)$;
2. $SV \leftarrow get\_SV(H_1\left(X_r^+, X_r^-\right)$;
3. $nonSV \leftarrow get\_nonSV(H_1\left(X_r^+, X_r^-\right)$;
4. $X_r^+ \leftarrow 0$ /* positive outliers or missclassified data points with H_1 are empty */ ;
5. $X_r^- \leftarrow 0$ /* negative outliers or missclassified data points with H_1 are empty */ ;
6. $O^+ \cup O^- \leftarrow testing\_SVMH_1\left(X_r^+, X_r^-\right)$;
7. $X_{ch}^+ \leftarrow SV$;
8. $X_{ch}^- \leftarrow nonSV$;
9. $H_2\left(X_{ch}^+, X_{ch}^-\right) \leftarrow trainSVM\left(X_{ch}^+, X_{ch}^-\right)$;
10. $\left(X_f^+, X_f^-\right) \leftarrow testing\_SVMH_2\left(X_{ch}^+, X_{ch}^-\right)$;
11. **return** X_f^+, X_f^-, O^+, O^-.

Algorithm 2: Proposed under-sampling algorithm

The main advantages of proposed model include a) it can make use of the discriminative features (features which show relevant differences between true splices sites and decoys), reducing the influence of some irrelevant and redundant features; b) it can work on imbalanced data-sets, the algorithm implements an undersampling technique in order to balance the data points and recover the most important data points in the data-set, retain valuable information with the proposed process; c) The training time obtained with the proposed method is very fast in comparison with other fast SVM implementations.

4. Experimental Results

In this section, we describe the methodology used and show the results obtained with the proposed algorithm,

4.1 Metrics for Imbalanced Classification

In order to evaluate classifiers on highly imbalanced data-sets, is necessary to use an adequate metric. With highly skewed data distribution, the overall accuracy metric is not sufficient any more. This is because with an imbalance of 99 to 1, a classifier that classifies everything negative will be 99% accurate, but it will be completely useless as a classifier to detect rare positive samples.

The medical community, and increasingly the machine learning community, use two metrics, the sensitivity and the specificity, when evaluating the performance of various tests. The sensitivity is the performance of proposed SVM to calculate the proportion of noncoding nucleotides that have been correctly predicted as noncoding and it is evaluated as

$$S_n^{false} = \frac{T_N}{T_N + F_P} \tag{4}$$

S_n is the proportion of candidate sites in the testing data-set that have been correctly predicted and it is expressed as

$$S_n = \frac{N_c}{N_t} \tag{5}$$

S_n^{true} is the proportion of coding nucleotides that have been correctly predicted as coding, i.e.,

$$S_n^{true} = \frac{T_P}{T_P + F_N} \tag{6}$$

where T_P is the number os sequences with real splice sites which are predicted to be true (true positives), T_N is the number of sequences without real splice sites which are predicted to be false (true negatives), F_P is the number of sequences without real splice sites which are predicted to be true (false positives) F_N is the number of sequences with real splice sites which are predicted to be false (false negatives), N_c is the number of exons that have been correctly predicted in the testing data-set, and N_t is the total number of exons sites in the testing data-set.

The receiver operator characteristic curve (ROC) analysis describes the sensitivity and specificity of a classification model using graphics. It is considered as an effective method to assess the performance of a classification method. We also used this metric to evaluate our classifier. We also list the

sensitivity and specificity separately to give the reader an even better idea of the performance of our classifier.

4.2 Model selection

SVM training involves to fixing several parameters. The parameters chosen have a crucial effect of the performance of the trained classifier. To be able to apply the SVM, we select the radial basis function (RBF) kernel function to train the SVM. The RBF kernel function is defined as

$$K(x_i - x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2), \gamma > 0 \qquad (7)$$

we have to find the complexity parameter C and γ, controlling the tradeoff between training error and complexity, and the kernel parameters. In order to identify an optimal hyperparameter set, we applied a "grid search" on C and γ using cross-validation.

4.3 Examples

In order to show the experimental results of the proposed method, we use two examples. First example is a small data-set with balance data-set, but the second example is an imbalanced and large data-set example.

4.3.1 Example 1

We use Primate splice-junction gene sequences(DNA) taken from Genbank64.1 (ftp site: genbank.bio.net).The DNA data-set contains 3190 DNA sequences with 62 descriptors for each sequence, 767exon/intron boundaries(referred to as EI sites), 768 intron/exon boundaries(referred to as IE sites) and 1655 neither.

In this example, we use 80% of the input data to train the SVM and 20% to test. The SVM was trained and evaluated 20 times, the experimental results are shown in the Table I. It shows the experimental results obtained with the proposed approach with the average accuracy (Acc) and the standard deviation(SD). The results obtained with S_n^{false}, S_n and S_n^{true} provide a good measure of the classifier. However, in this case the data-set is very small, the training time is almost the same with some SVM implementations like SimpleSVM, Libsvm, Sequential Minimal Optimization(SMO), but when the training data-set is large the training time grows exponentially.

Table I

Genbank 64.1 data-set			
	Av_EI	Av_IE	Av_Neither
Acc	99.37	99.18	97.8
SD	0.16	0.27	0.24
S_n^{true}	0.99	0.98	0.97
S_n^{false}	0.99	0.98	0.97
S_n	0.99	0.99	0.97

Acc.-average accuracy, SD.- standard deviation.

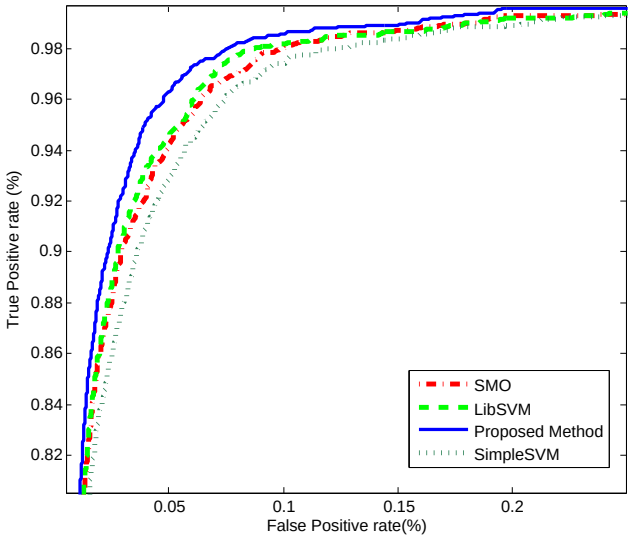

Fig. 1: ROC curves of the four classifiers. The proposed method, LibSVM, SMO and SimpleSVM.

4.3.2 Example 2

The second example is acceptor/donor data-set which was obtained from http://www2.fml.tuebingen.mpg.de/raetsch/projects/.
The data-set contains 91546 training data points and 75905(2132 true sites) testing data points for acceptors and 89163 training data points and 73784(2132 true sites) testing data points for donors. In this example we show the difference of training time between the proposed approach and other fast SVM implementations.

The Figure 1 shows the ROC curves obtained with the proposed algorithm, The AUC for the proposed method, LibSVM, SMO and SimpleSVM are 0.9894, 0.9860, 9865 and 9823 respectively. The Figure 2 shows the discriminative power of the proposed method, in the Figure 2 are shown the AUC of LibSVM with only the sparse encoding and the AUC of proposed method. It is clear that, a set of highly discriminative features could significantly improve the classification accuracy. Some features were added with the purpose of enhancing the classifier performance. Moreover, not only in the performance measure is more robust, but also we get a small training time as can we see in the Table II.

Table II

Acceptor data-set			Donnor data-set	
Algorithm	t	AUC	t	Acc
Proposed App	469	98.9	673	98.7
LIBSVM	6371	98.6	4924	98.5
SMO	123493	98.6	104525	98.4
SimpleSVM	432919	98.2	381049	98.1

$\#$ traininig data, t training time in seconds, Acc accuracy.

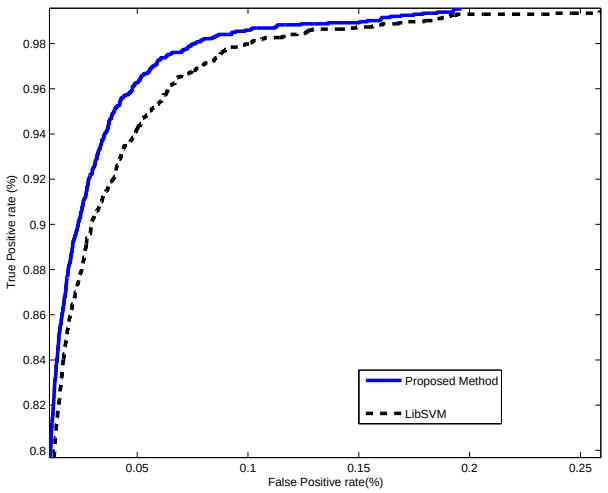

Fig. 2: ROC curves of the four classifiers. The proposed method, LibSVM, SMO and SimpleSVM.

5. Conclusions

In this paper we present a novel SVM classification approach for large data-sets using imbalanced data-sets. In order to reduce SVM training time for large data-sets, we use a modified algorithm which overcomes the drawback that only part of the original data near the support vectors are trained. Experiments done with real world data-sets, show that the proposed method has advantage in large data-sets. Furthermore, not only in the training time is more robust, but also we get much area under the ROC curve, providing an adequate measure for the quality of the classifier. Some features have been proposed for the classification Donnor/acceptor. introducing a new encoding method. However, not all features are equally effective for the classification task. Therefore, the careful choice of features is crucial for building accurate splice detectors and if an appropiate system for imbalanced data-sets is implemented, the SVM classifier easily outperform previously proposed methods. Choosing a set of highly discriminative features could significantly improve the classification accuracy. In this work, we study the some features with the purpose of enhancing the classifier performance, and improve significatively the training time used with other fast SVM implementations.

References

[1] AKMA Baten, BCH Chang, SK Halgamuge and Jason Li. "Splice site identification using probabilistic parameters and SVM classification", *BMC Bioinformatics*, Vol. 7, S15, 2006.

[2] Yiming Cheng, Robert M. Miura and Bin Tian *Prediction of mRNA polyadenylation sites by support vector machine*, Bioinformatics, Vol. 22 no. 19, pp 2320-2325, 2006.

[3] Robertas Damaevicius *Splice Site Recognition in DNA Sequences Using K-mer Frequency Based Mapping for Support Vector Machine with Power Series Kernel*, International Conference on Complex, Intelligent and Software Intensive Systems, pp 687-692, 2008.

[4] Gideon Dror, Rotem Sorek and Ron Shamir *Accurate identification of alternatively spliced exons using support vector machine*, Bioinformatics, Vol. 21 no. 7, pp 897-901, 2005.

[5] Fickett, J. W. *Finding genes by computer: the state of the art*, Trends Genet, Vol. 12 no.8 pp 316-320, 1996.

[6] T. R. Golub and D. K. Slonim and P. Tamayo and C. Huard and M. Gaasenbeek and J. P. Mesirov and H. Coller and M. L. Loh and J. R. Downing and M. A. Caligiuri and C. D. Bloomfield, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, vol. 286, pp. 531-537, 1999.

[7] Jones, D. and Watkins, C. *Comparing kernels using synthetic dna and genomic data*,. Technical report, University of London, UK, 2000.

[8] Liew, A. W.-C., Wu, Y., and Yan, H. *Selection of statistical features based on mutual information for classification of human coding and non-coding dna sequences*. Bioinformatics technologies, In ICPR04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR04) Volume 3, pages 766-769,Washington, DC, USA, 2004.

[9] Platt J., "Fast Training of support vector machine using sequential minimal optimization. *In A.S.B. Scholkopf, C. Burges, editor, Advances in Kernel Methods: support vector machine* . MIT Press, Cambridge, MA 1998.

[10] Pritish Varadwaj, Neetesh Purohit and Bhumika Arora., "Detection of Splice Sites Using Support Vector Machine . *In Contemporary Computing, Second International Conference,Proceedings*. Vol. 40, pp. 493-502, 2009.

[11] Jing Xia, Doina Caragea and Susan Brown *Exploring Alternative Splicing Features Using Support Vector Machines*, Proceedings of the 2008 IEEE International Conference on Bioinformatics and Biomedicine, pp 231-238, 2008.

[12] Jing Xia, Doina Caragea and Susan Brown., "Exploring Alternative Splicing Features Using Support Vector Machines. *In Proceedings of the 2008 IEEE International Conference on Bioinformatics and Biomedicine*. pp. 231-238, 2008.

[13] Xiang H-F. Zhang, Katherine A. Heller, Ilana Hefter, Christina S. Leslie and Lawrence A. Chasin *Sequence Information for the Splicing of Human Pre-mRNA Identified by Support Vector Machine Classification*, Genome Research, Vol.13, pp. 2637-2650, 2003.

[14] Xiang H-F. Zhang, K.A. Heller, I. Hefter, Ch. S. Leslie and Lawrence A. Chasin, Sequence Information for the Splicing of Human Pre-mRNA Identified by Support Vector Machine Classification, *Genome Research*, Vol. 13. pp 2637-2650, 2003.