A Probabilistic Kernel for Splice Site Prediction

Ya Zhang[*] Chao-Hisen Chu[*] Hongyuan Zha[†] Yixin Chen[‡] Xiang Ji[§]

Abstract

One of the most important tasks in correctly annotating genes in higher organism is to accurately locate the DNA splice sites. Although relatively high accuracy has been achieved by the existing methods, most of these prediction methods are computationally very expensive. Considering the enormousness of DNA sequences, the computing speed is an important issue. In this paper, we propose to use a probabilistic kernel-based method to predict DNA splice sites, which project the sequence data into a new probabilistic feature space. We then use Support Vector Machines to recognize the true splice sites. While the performance is comparable to the results obtained with polynomial kernels, the computation is performed much faster.

1 Introduction

The advances in sequencing technologies have resulted in a large amount of DNA sequence information and therefore a dramatic increase in the size of genetic and genomic databases. With the whole genomes for many organisms available, an important goal in bioinformatics is to accurately annotate genes from DNA sequence information. Many efforts have been made to predict gene structures [1] from DNA sequences and aid the wholesale analysis of the DNA sequences, including recognizing translation initiation site [8], discovering transcriptional factor binding sites [3], identifying DNA splice sites [2, 4, 7].

In this paper, we target on the problem of identifying DNA splice sites. Splicing is one of the primary post-processing steps of gene expression in eukaryotes. During splicing, the introns, the non-coding regions, are removed from the primary transcripts, and the exons, the coding regions, are joined to form a continuous sequence that specifies a functional polypeptide. The pairs of residues GT and AG are highly conserved at the donor and acceptor splice sites respectively. However, this canonical GT-AG rule does not always hold. Thus, it is natural to model the prediction of splice sites as a two-class classification problem, using DNA sequences with experimentally confirmed splice sites as positive training examples and those DNA sequences with GT-AG structure but confirmed not to be real splice sites as negative training examples.

Machine learning methods such as Artificial Neural Network[4], Perceptron[7], and Support Vector Machine[2] have been employed to approach the problem of recognizing true splice sites. Relatively high accuracy has been achieved with the methods currently available. However, almost all of the existing methods are computationally very demanding, and as a matter of fact splice site prediction has been a bottle neck in gene annotation.

The DNA sequences are provided as strings while most classifiers only take numerical inputs. Thus, the very first step of classification is often to encode the DNA sequences with numbers. A widely used encoding method is sparse encoding[2], where each letter in the DNA sequence is represented in four bits. This encoding treat the four nucleotides equally and failed to consider the probability of natural mutation in DNA sequences. As a result, it may not perform well in some cases. Specifically, for our problem of classifying splice sites, the data is linearly inseparable with the common sparse encoding method. Therefore, we propose to use a probabilistic kernel, which projects the data into a new probabilistic feature space and accounts for the natural mutations in the sequences. The true splice sites and the false splice sites is better distinguished by linear SVMs at this feature space. Experimental results with SVM classifier have shown that the performance of the proposed kernel is comparable to that of polynomial kernels, in terms of accuracy, precision and recall, while its speed is significantly faster. Considering the overwhelming amount of DNA sequences that needs to be processed, this is a very desirable property.

2 The probabilistic kernel

The probabilistic kernel method is built from the bayes' rule. Suppose we have a set of examples E =

[*] School of Information Sciences and Technology, The Pennsylvania State University

[†] Department of Computer Science and Engineering, The Pennsylvania State University

[‡] Department of Computer Science, University of New Orleans

[§] NEC Laboratories America, Cupertino, CA

$\{X_1, X_2, ..., X_N\}$. Let $X_i = \{a_1, ..., a_n\}(X_i \in E)$ denotes a DNA sequence, where each a_j (j=0, 1, ..., n) is an nucleotide. Each $X_i \in E$ falls into one of the two categories: c_1 or c_{-1}, where c_1 stands for true splice sites, and c_{-1} for false splice sites. According to the Bayes' rule:

$$P(c_1|X_i) = \frac{P(X_i|c_1) \cdot P(c_1)}{p(X_i)} \quad (1)$$

$$P(c_{-1}|X_i) = \frac{P(X_i|c_{-1}) \cdot P(c_{-1})}{p(X_i)}. \quad (2)$$

Assume that $a_j(j = 1, 2, ..., n)$ are independent. Thus, we get

$$P(X_i|c_1) = \prod_{j=1}^{n} P(a_j|c_1) \quad (3)$$

$$P(X_i|c_{-1}) = \prod_{j=1}^{n} P(a_j|c_{-1}). \quad (4)$$

After a few manipulations of the above equations, Equation. 1 and 2 can be reformulated as:

$$log(P(c_1|X_i)) = \sum_{j=1}^{n} log(P(a_j|c_1)) - log(P(X_i)) + a, \quad (5)$$

$$log(P(c_{-1}|X_i)) = \sum_{j=1}^{n} log(P(a_j|c_{-1})) - log(P(X_i)) + b, \quad (6)$$

where $a = log(P(c_1))$ and $b = log(P(c_{-1}))$. With the näive bayesian classifier, the classification decision is made to maximize the log likelihood. X_i is assigned to the class c_k ($k = 1$ or -1) that would maximize $log(P(c_k|X_i))$. Thus the decision function can be expressed as:

$$f(X_i) = sgn(log(P(c_1|X_i)) - log(P(c_{-1}|X_i))). \quad (7)$$

Assuming uniform prior, i.e. $P(c_1) = P(c_{-1})$ and thus $a = b$, we get:

$$f(X_i) = sgn(\sum_{j=1}^{n} log(P(a_j|c_1)) - \sum_{j=1}^{n} log(P(a_j|c_{-1}))). \quad (8)$$

Equation. 8 can be reformulated as:

$$f(X) = sgn(\vec{w} \cdot \vec{p}), \quad (9)$$

where $\vec{w} = \{w_1, w_2, ..., w_{2n}\}$ is the weight vector, and $\vec{P} = \{p_1, p_2, ..., p_{2n}\}$ is the posterior probability vector. In näive Bayesian classifier, we have $w_i = 1$ for $i \in \{1, ..., n\}$, $w_i = -1$ for $i \in \{n + 1, ...2n\}$, $p_i = P(x_i|c_1)$ for $i \in \{1, ..., n\}$, and $p_i = P(x_{i-n}|c_{-1})$ for $i \in \{n+1, ...2n\}$. The positional profile of an alignment of DNA sequences of length l is defined as a $4 \times l$ matrix ($p_{N,i}$), where $p_{N,i}$ is the frequency of nucleotide N in the i^{th} position in the alignment. The positional profiles may be obtained from DNA sequences with true splice sites and DNA sequences with false splice sites, respectively (Step 1 of Figure 1). The p_i value can be obtained by looking up the corresponding positional profiles (Step 2 of Figure 1).

The näive bayesian classifier is guaranteed to be optimal only when the attributes are independent given the class. However, this independence assumption may not always be true. Thus, the estimation of the distribution of X_i may not be accurate. In addition, the näive bayesian classifier assumes each position is equally important, which might not be true in the case of splice site prediction. Some position may be essential while some others may be trivial. The idea here is to use the probabilistic feature mapping to project the data into a new feature space where the data are more likely to be linearly separable, and then use linear SVM to learn the optimal weight vectors. We expect that this can improve the classification accuracy gained by the näive bayesian classifier while maintaining the simplicity in computation. The overflow of the encoding process is illustrated in Fig 1.

3 Experiment

To evaluate the performance of the probabilistic kernel, Support Vector Machine (SVM)[6] is employed as the classifier. The SVMs with the probabilistic kernel (thereafter BMSVMs) is used to recognize true splice sites. A series 10-fold cross validation experiment was performed. The BMSVMs method was compared with näive bayesian classifier, with SVMs with linear kernel, and with SVMs with polynomial kernels.

Two data sets, $Dsmall$ and $Dlarge$, are used for the experiments. They contain 1,000 and 10,000 nucleotide sequences of splice site data, respectively. All the sequences are 50 bases long, and for each sequence the $GT\text{-}AG$ structure occurs at the middle. Both data sets contain examples of true splice sites as well as false splice sites.

The data set is randomly split into ten sets of equal size. Each time, one set is used for testing, and the rest nine are combined and used for training. First, the positional profiles are estimated for the true splice sites from the positive examples in the training set and for false splice sites from negative examples. We assume that the sequences in the training set are representative for all DNA sequences. Therefore, entries of the positional profiles for the true and false splice sites can be approximated with the observed frequency of occurrence

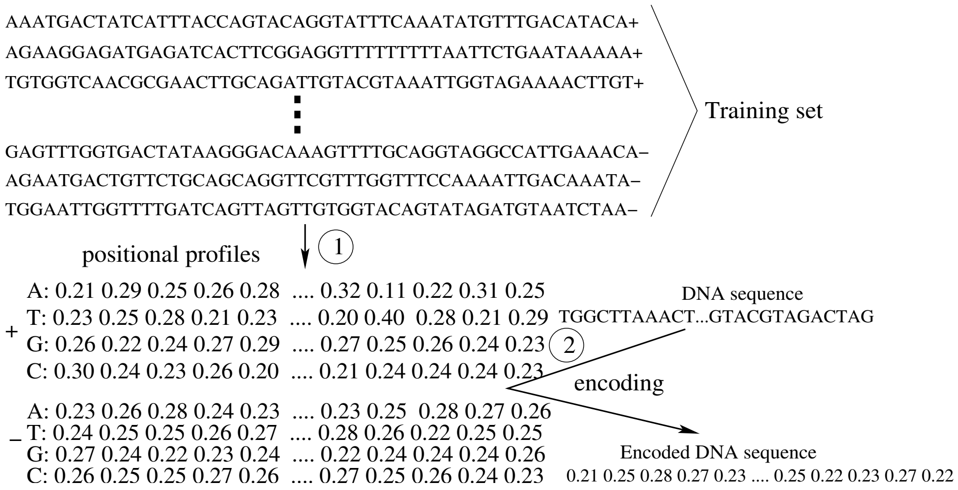
AAATGACTATCATTTACCAGTACAGGTATTTCAAATATGTTTGACATACA+
AGAAGGAGATGAGATCACTTCGGAGGTTTTTTTTAATTCTGAATAAAAA+
TGTGGTCAACGCGAACTTGCAGATTGTACGTAAATTGGTAGAAAACTTGT+
GAGTTTGGTGACTATAAGGGACAAAGTTTTGCAGGTAGGCCATTGAAACA–
AGAATGACTGTTCTGCAGCAGGTTCGTTTGGTTTCCAAAATTGACAAATA–
TGGAATTGGTTTTGATCAGTTAGTTGTGGTACAGTATAGATGTAATCTAA–
Training set
positional profiles
1
A: 0.21 0.29 0.25 0.26 0.28 .... 0.32 0.11 0.22 0.31 0.25
T: 0.23 0.25 0.28 0.21 0.23 .... 0.20 0.40 0.28 0.21 0.29
+
G: 0.26 0.22 0.24 0.27 0.29 .... 0.27 0.25 0.26 0.24 0.23
C: 0.30 0.24 0.23 0.26 0.20 .... 0.21 0.24 0.24 0.24 0.23
A: 0.23 0.26 0.28 0.24 0.23 .... 0.23 0.25 0.28 0.27 0.26
–
T: 0.24 0.25 0.25 0.26 0.27 .... 0.28 0.26 0.22 0.25 0.25
G: 0.27 0.24 0.22 0.23 0.24 .... 0.22 0.24 0.24 0.24 0.26
C: 0.26 0.25 0.25 0.27 0.26 .... 0.27 0.25 0.26 0.24 0.23
DNA sequence
TGGCTTAAACT...GTACGTAGACTAG
2
encoding
Encoded DNA sequence
0.21 0.25 0.28 0.27 0.23 .... 0.25 0.22 0.23 0.27 0.22

Figure 1: Overflow of the algorithm.

of given nucleotide at given position in the positive and negative examples of the training set, respectively. This is to estimate the posterior probability $P(x_{ik}|c_j).(i \in \{1,...,n\}.k \in \{A,T,C,G\},.j \in \{1,-1\})$ from the training set, where c_1 means the true splice sites, c_{-1} means the false splice sites, and x_i represents the i^{th} nucleotide. Then, the logarithms of each posterior probability $log(P(x_{ik}|c_j))$ was input to a linear Support Vector Machine. The Support Vector Machines software *svm-light* is downloaded from *http://www.support-vector.net*.

As a comparison, we also conducted experiments with SVMs methods with linear kernel and polynomial kernels. In these cases, DNA sequences are first encoded with sparse encoding. Similar to the experiment with the BMSVMs method, a 10-fold validation experiments were conduct for each method.

4 Results

Our BMSVMs method was compared with the näive bayesian classifier as well as with SVM classifiers with linear and polynomial kernels. The parameter C of the SVM classifiers is empirically set to be 150 based on our experiments (result not shown). We report the results in terms of $accuracy$, $precision$, $recall$ and $F\text{-}measure$. These measures are defined as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \tag{10}$$

$$Precision = \frac{tp}{tp + fp}, \tag{11}$$

$$Recall = \frac{tp}{tp + fn}, \tag{12}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{13}$$

where fp is the number of sequences with real splice sites which are predicted to be true, tn is the number of sequences without real splice sites which are predicted to be false, fp is the number of sequences without real splice sites which are predicted to be true, fn is the number of sequences with real splice sites which are predicted to be false. They are illustrated in Fig. 2.

predict real	true	false
true	tp	fn
false	fp	tn

Figure 2: Illustration of fp, tn, fp, and fn.

Table 1: Results of BMSVMs method (C=150).

Data set		accuracy	precision	recall	F-measure
$Dsmall$	Avg	89.2	90.9	87.8	89.3
	Std	3.4	3.9	5.3	-
$Dlarge$	Avg	91.4	92.0	90.6	91.3
	Std	0.9	1.3	1.4	-

Table 1 summarizes the result of BMSVMs method. First, the accuracy, precision and recall are averaged among each ten-fold cross validation experiments. Their standard deviations are also computed. Based on the average precision and recall, we compute the F-measure. The F-measure for the $Dsmall$ data set and the $Dlarge$ data set are 89.3 and 90.3, respectively. Similarly, we present the results of näive Bayesian classifier in Table 2, and the results of SVM classifier in Table 3 and Table 4.

As can be seen from the above tables, in terms of accuracy and $F\text{-}measure$, our BMSVMs method outper-

Table 2: Results of näive bayesian as classifier.

Data set		accuracy	precision	recall	F-measure
$Dsmall$	Avg	89.0	90.5	87.9	89.2
	Std	2.7	3.3	3.7	-
$Dlarge$	Avg	91.1	90.8	91.5	91.1
	Std	1.0	1.6	1.4	-

Table 3: Results of using SVM as classifier for the $Dsmall$ data set (C=150).

Kernel		accuracy	precision	recall	F-measure
Linear	Avg	86.6	88.2	85.1	86.6
	Std	2.6	2.5	6.6	-
polynomial d=2	Avg	88.9	89.3	88.8	89.0
	std	2.7	3.6	3.9	-
polynomial d=3	Avg	89.8	90.9	88.8	89.8
	std	3.2	3.6	4.6	-

forms näive Bayesian classifier and SVM classifier with linear kernel and polynomial kernel of $d = 2$ when the $Dsmall$ data set is used. The results of SVM classifier with polynomial kernel of $d = 3$ are slightly better. When $Dlarge$ data set is used, the BMSVMs method outperforms all the other methods: näive Bayesian classifier and SVM classifier with linear kernel and polynomial kernel of $d = 2$ and $d = 3$.

5 Discussion

In this paper, we present a novel idea of constructing a probabilistic kernel mapping method from Bayesian classifier. This mapping method is then integrated with SVM classifier and applied to the problem of splice site prediction from DNA sequences. Experiments on two data sets have demonstrated that our method outperforms the benchmark methods: Näive Bayesian classifier, SVM classifier with linear kernel and polynomial kernel ($d = 2$ and $d = 3$) in terms of accuracy and F-measure.

The results show that the BMSVMs method enhances the performance of Näive Bayesian classifier. Furthermore, when the speed of computation are taken into consideration, the method is as quick as the Näive Bayesian classifier and much faster than SVM with non-linear kernel methods.

Table 4: Results of using SVM as classifier for the $Dlarge$ data set (C=150).

Kernel		accuracy	precision	recall	F-measure
Linear	Avg	91.0	91.5	90.3	90.9
	Std	1.3	1.4	2.0	-
polynomial d=2	Avg	89.2	89.0	89.4	89.2
	std	0.7	0.8	1.3	-
polynomial d=3	Avg	90.7	91.0	90.5	90.7
	std	0.9	1.0	1.3	-

Bayesian classifier is a simple generative learning method and SVM classifier represents a type of discriminative learning methods [5]. This proposed method represents an effort in integrating the generative learning methods into discriminative learning. With the success of the proposed method, more complex generative learning methods, such as Hidden Markov Model (HMM) may be integrated into SVM classifier in a similar fashion as a probabilistic kernel. Therefore, future research can introduce some other model building techniques such as HMM or improved Bayesian method to better capture of the feature distribution.

References

[1] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic dna. *J. Mol. Biol.*, 268(1):78–94, Apr 1997.

[2] D. Jones and C. Watkins. Comparing kernels using synthetic dna and genomic data. Technical report, 2000.

[3] M. E. Lim, J. S. Sim, M. G. Chung, and S. H. Park. Prediction of transcription factor binding sites with suffix arrays. *Genome Informatics*, 14:400C401, 2003.

[4] N. Mache and P. Levi. Parallel neural network training and cross validation on a cray t3e system and application to splice site prediction in human dna. Technical report, Institute of Parallel and Distributed High Performance Systems, Stuffgart, Germany, 1995.

[5] A. Ng and M. Jordan. On discrminiative vs. generative classifiers: A comparsion of logistic regresssion and naive bayes. *NIPS*, 2002.

[6] V. N. Vapnik. *Adaptive and learning systems for signal processing, communications, and control*, chapter Statistical learning theory. NY: Wiley, New York, 1998.

[7] R. Weber. Dna splice site prediction with kernels and voting. In *Proceedings of International Conference on Mathematical and Engineering Techniques in Medicine and Biological Sciences*, 2001.

[8] A. Zien, G. Ratsch, S. Mika, B. Scholkopf, T. Lengauer, and K. Muller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.