Journal of Universal Computer Science, vol. 15, no. 13 (2009), 2528-2546
submitted: 31/10/08, accepted: 13/6/09, appeared: 1/7/09 © *J.UCS*

Splice Site Prediction using Support Vector Machines with Context-Sensitive Kernel Functions

Yifei Chen
(Vrije Universiteit Brussel, Belgium
yifechen@vub.ac.be)

Feng Liu
(Vrije Universiteit Brussel, Belgium
fengliu@vub.ac.be)

Bram Vanschoenwinkel
(Vrije Universiteit Brussel, Belgium
bvschoen@vub.ac.be)

Bernard Manderick
(Vrije Universiteit Brussel, Belgium
bmanderi@vub.ac.be)

Abstract: This paper focuses on the use of support vector machines on a typical context-dependent classification task, splice site prediction. For this type of problems, it has been shown that a context-based approach should be preferred over a transformation approach because the former approach can easily incorporate statistical measures or directly plug sensitivity information into distance functions. In this paper, we designed three types of context-sensitive kernel functions: polynomial-based, radial basis function-based and negative distance-based kernels. From the experimental results it becomes clear that the radial basis function-based kernel with information gain weighting gets the best accuracies and can always outperform their simple non-sensitive counterparts both in accuracy and in model complexity. And with well designed features and carefully chosen context sizes, our system can predict splice sites with fairly high accuracy, which can achieve the $FP95\%$ rate, 3.94 for donor sites and 5.98 for acceptor sites, an approximate state of the art performance for the moment.

Key Words: support vector machines, kernel functions, splice site prediction
Category: I.2.6, I.5.4, J.3

1 Introduction

An important task in bio-informatics is the analysis of genome sequences for the location and structure of their genes, often referred to as gene finding. Without going into detail, we will consider the case of eukaryotic species which are characterized by the fact that they have cells with visible nuclei surrounded by a nuclear membrane. Humans for example are eukaryotic species. In general, a gene can be defined as a region of DNA that controls a certain hereditary characteristic, although other definitions exist. More precisely, the region of the

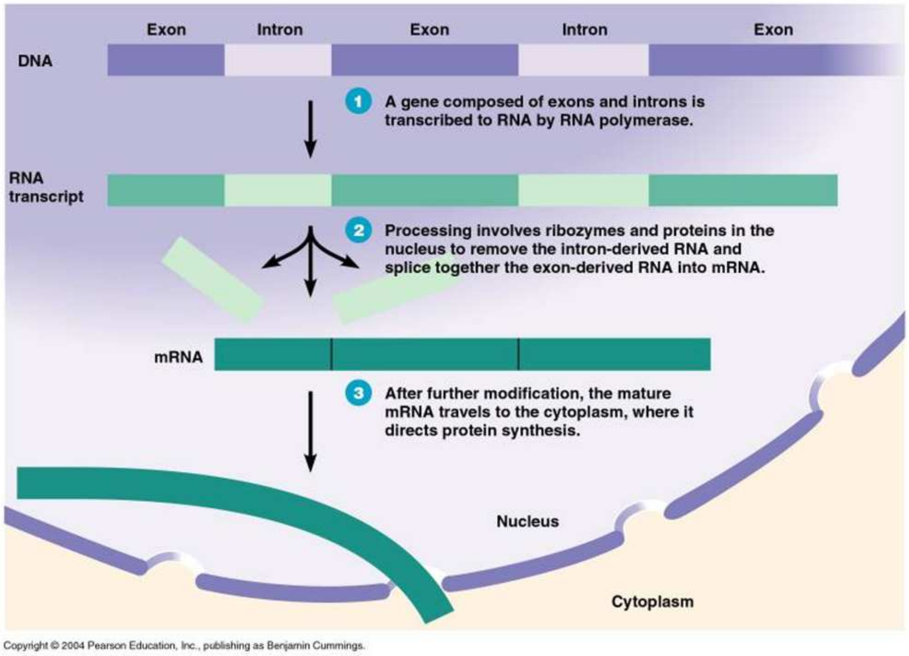

Figure 1: Transcription and translation in eukaryotes species. First a RNA sequence is transcribed from the gene on the DNA, next the RNA is spliced to form the mRNA which then travels to the cytoplasm where it is translated into a protein.

DNA sequence corresponding to the gene is used in the production of a specific protein. In an organism, each cell essentially has the same set of genes, but it can have different functions by making certain genes active and other genes inactive.

The expression of genes as proteins occurs in two steps: i) making the gene active by means of the transcription of DNA into RNA and ii) the translation of RNA into proteins. Taken together these two steps form the central dogma of biology. We will concentrate on the first step here and explain it in some more details next.

Transcription itself occurs in three steps as outlined in [Fig. 1]. In the first step the raw DNA is transcribed to RNA. Transcription starts when the enzyme called RNA *polymerase* binds to the *promoter region* of the gene and from this starting point RNA polymerase moves downstream continuously synthesizing RNA until a terminator sequence, called the *stop codon*, is reached.

In the second step, the RNA is transformed into messenger RNA or mRNA for short. It is called this way because it is the role of the mRNA to move the information contained in DNA to the translation machinery, this is actually the last step in the transcription process and is shown as step three in [Fig. 1]. But now back to step two, i.e. the transformation of RNA into mRNA. One necessary step in the process of obtaining mature mRNA is called *splicing*.

In many genes, the DNA sequence coding for proteins, or *exons*, may be interrupted by stretches of non-coding DNA, called *introns*. A gene starts with an exon, is then interrupted by an intron, followed by another exon, intron and so on, until it ends in an exon. Splicing is the process by which the non-coding sequences (i.e. the introns) are subtracted from the coding sequences (i.e. the exons). In the light of the previous we can make a distinction between two different splice sites: i) the *exon-intron boundary* which is referred to as the *donor site* and ii) the *intron-exon boundary* which is referred to as the *acceptor site*. Splice site prediction is the automatic identification of those regions in the DNA sequence that are intron-exon or exon-intron boundaries. To see this it should first be noted that DNA is essentially a sequence of nucleotides represented by a four letter alphabet $D = \{A,C,G,T\}$. Next, an acceptor site is observed to always contain the AG dinucleotide and the donor site is observed to always contain the GT dinucleotide.

Because splice site prediction instances can be represented by a *context* of a number of nucleotides before and after the AG/GT dinucleotides, it is called a *context-dependent classification* task, which will be discussed in [Section 2] in detail. Support vector machines (SVMs) are employed to do splice site prediction in this paper. In practice a classifier is trained for each type of splice site, i.e. the problem is split up into two binary classification problems: one classifier is trained to distinguish acceptor sites from pseudo-acceptor sites and one classifier is trained to distinguish donor sites from pseudo-donor sites.

More precisely, in SVM learning the data is mapped non-linearly from the original input space X to a high-dimensional feature space F and subsequently separated by a maximum-margin hyperplane in that space F. By making use of the kernel trick, the mapping to F can stay implicit, and we can avoid working in the high-dimensional space. Moreover, because the mapping to F is non-linear, the decision boundary which is linear in F, corresponds to a non-linear decision boundary in X. One of the most important design decisions in SVM learning is the choice of kernel function K because the hyperplane is defined completely by inner products between vectors in F and calculated through the kernel function K. Moreover, K takes vectors from the input space X and directly calculates inner products in F without having to represent or even know the exact form of these vectors, hence the implicit mapping and computational benefit [Cristianini and Shawe-Taylor 2004]. In the light of the above it is not hard to see that the way in which K is calculated is crucial for the success of the classification process.

Notice that an inner product is actually one of the most basic similarity measures between vectors since it gives much information about the position of these vectors in relation to each other. The learning process can benefit a lot from the use of special purpose similarity or dissimilarity measures in the calculation of K [Schölkopf 2000, Vanschoenwinkel and Manderick 2004, Vanschoenwinkel et al. 2006]. However, incorporating such knowledge in a kernel function is not trivial as a kernel function has to satisfy a number of properties that result directly from the definition of the inner product.

Applying SVMs on contexts involves some issues that need to be addressed, i.e. SVMs are defined on real vectors and not on contexts. Generally speaking two approaches exist: i) the transformation approach, i.e. transform contexts to real vectors, for example, bag-of-words approach [Joachims 2002] and orthonormal vector approach [Vanschoenwinkel et al. 2005] and ii) the direct approach, i.e. define kernel functions that work on contexts but calculate real inner products in F. The transformation approach has been successfully applied to many classification problems. Nevertheless, here we do not make use of the transformation approach but the direct approach. In previous work [Vanschoenwinkel et al. 2005, Vanschoenwinkel et al. 2006], we have found that it is better to work directly on contexts instead of on a transformed high-dimensional sparse format, because in this way it is much easier to incorporate special purpose similarity measures into the kernel function as such measures are defined on the contexts and not on a high-dimensional representation of the contexts. Therefore these kernel functions are called *context-sensitive kernel functions*.

The rest of this paper is organized as follows: [Section 2] shows what contexts are and two popular distance functions defined on contexts, an overlap metric and a modified value difference metric, which have already been introduced in previous work and achieved good results [Vanschoenwinkel and Manderick 2004, Vanschoenwinkel et al. 2005, Vanschoenwinkel et al. 2006]. Motivated by this, in [Section 3], we introduce a number of kernel functions that make direct use of the distance functions mentioned in [Section 2]. Next, [Section 4] shows some experimental results on gene splice site prediction and finally, [Section 5] gives a conclusion.

2 Context-Dependent Classification

In this paper we consider classification tasks where it is the purpose to classify a focus string in a sequence of strings, based on a number of strings before and after the focus string. The focus string, together with the strings before and after it, is called a *context* and applications that rely on such contexts will be called *context-dependent*. Splice site prediction is a typical example of a context-dependent classification task.

2.1 Context

We start with a definition of a context followed by an illustration in the framework of splice site prediction.

Definition 1. A *context* $\bar{\mathbf{s}}_p^q$ is a sequence of strings $\mathbf{s}_i \in D$ with p strings before and q strings after a focus string $\mathbf{s}_p$ at position p as follows

$$\bar{\mathbf{s}}_p^q = (\mathbf{s}_0, \ldots, \mathbf{s}_p, \ldots \mathbf{s}_{p+q}) \tag{1}$$

with $(p+q)+1$ the length of the context, with D the dictionary of all strings, with $|D| = m$ and with p the left context size and with q the right context size.

Example 1. Remind from the introduction that in splice site prediction it is the purpose to automatically identify those regions in a DNA sequence to be donor sites or acceptor sites. Splice site prediction instances can be represented by a context of a number of nucleotides before and after the AG/GT dinucleotides. More precisely, given a fragment of a DNA sequence, ...CCATTG<u>GT</u>GGCAGCCAG... the candidate donor site given by the dinucleotide GT can be represented by a context in terms of [Definition 1] as

$$\bar{\mathbf{s}}_p^q = \left(\underbrace{\mathtt{A}, \mathtt{T}, \mathtt{T}, \mathtt{G}}_{\mathbf{s}_0, \ldots, \mathbf{s}_{p-1}}, \underbrace{\mathrm{GT}}_{\mathbf{s}_p}, \underbrace{\mathtt{G}, \mathtt{G}, \mathtt{C}}_{\mathbf{s}_{p+1}, \ldots, \mathbf{s}_{p+q}} \right)$$

with $p = 4$ the left context size and $q = 3$ the right context size and with $(p+q)+1 = 8$ the total length of the context. Notice that in this example single characters are considered to be strings of length 1.

2.2 The Overlap Metric

The most basic distance function defined on contexts is called the *overlap metric*, which simply counts the number of mismatching strings at corresponding positions in two contexts.

Definition 2. Let $\mathbb{S}^n$ be a set with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p+q)+1$ the length of the contexts, with strings $\mathbf{s}_i, \mathbf{t}_i \in D$ the dictionary of all distinct strings with $|D| = m$ and let $\mathbf{w} \in \mathbb{R}^n$ be a context weight vector. Then the overlap metric $d_{OM} : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}^+$ is defined as

$$\mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}^+ : d_{OM}(\bar{\mathbf{s}}, \bar{\mathbf{t}}) = \sum_{i=0}^{n-1} w_i - \delta(\mathbf{s}_i, \mathbf{t}_i) \tag{2}$$

with $\delta : \mathbb{S} \times \mathbb{S} \to \{w_i, 0\}$ defined as

$$\delta(\mathbf{s}_i, \mathbf{t}_i) = \begin{cases} w_i & \text{if } \ \mathbf{s}_i = \mathbf{t}_i \\ 0 & else \end{cases} \tag{3}$$

with $w_i \geq 0$ a context weight for the string at position i.

Next, we make a distinction between two cases: i) for $\mathbf{w} = \mathbf{1}$ no weighting takes place and the metric is referred to as the *simple overlap metric* d_{SOM} and ii) for $\mathbf{w} \neq \mathbf{1}$ a position dependent weighting does take place and the metric is referred to as the *weighted overlap metric* d_{WOM}. A question that now naturally rises is: What measures can be used to weigh the different context positions?

Information theory provides many useful tools for measuring statistics in the way described above. In this work we made use of three measures known as i) *information gain* [Quinlan 1986], ii) *gain ratio* [Quinlan 1993] and iii) *shared variance* [White and Liu 1994]. For more details the reader is referred to the related literature.

2.3 The Modified Value Difference Metric

The *Modified Value Difference Metric* (MVDM) [Cost and Salzberg 1993] is a powerful method for measuring the distance between symbolic-valued vectors, like the contexts considered here. The MVDM is based on the *Stanfill-Waltz Value Difference Metric* [Stanfill and Waltz 1986] introduced in 1986. The MVDM determines the similarity of all the possible strings at a particular context position by looking at co-occurrence of the strings with the target class. Consider the following definition.

Definition 3. Let $\mathbb{S}^n$ be a set with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p + q) + 1$ the length of the contexts as before, with components $\mathbf{s}_i$ and $\mathbf{t}_i \in D$ the dictionary of all distinct strings with $|D| = m$. Then the modified value difference metric $d_{MVDM} : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}^+$ is defined as

$$\mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}^+ : d_{MVDM}(\bar{\mathbf{s}}, \bar{\mathbf{t}}) = \sum_{i=0}^{n-1} \delta(\mathbf{s}_i, \mathbf{t}_i)^r \tag{4}$$

with r a constant often equal to 1 or 2 and with $\delta : D \times D \to \mathbb{R}$ the difference of the conditional distribution of the classes as follows:

$$\delta(\mathbf{s}_i, \mathbf{t}_i)^r = \sum_{j=1}^{M} |P(y_j|\mathbf{s}_i) - P(y_j|\mathbf{t}_i)|^r \tag{5}$$

with y_j the class labels and with M the number of classes in the classification problem under consideration.

3 Context-Sensitive Kernel Functions

In this section we will introduce a number of kernel functions that make direct use of the distance functions d_{SOM}, d_{WOM} and d_{MVDM} defined in the previous section. In the case of d_{WOM} and d_{MVDM} the kernels are called context-sensitive as they take into account the amount of information that is present at different context positions as discussed in the previous section.

3.1 Theoretical Requirements

Remind that in the SVM framework classification is done by considering a kernel induced feature mapping ϕ that maps the data from the input space X to a high dimensional Hilbert space F and classification is done by means of a maximum-margin hyperplane in that space F. This is done by making use of a special function called a *kernel*.

Definition 4. A *kernel* is a symmetric function $K : X \times X \rightarrow \mathbb{R}$ so that for all $\mathbf{x}$ and $\mathbf{x}'$ in X, $K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ where ϕ is a (non-linear) mapping from the input space X into the Hilbert space F provided with the inner product $\langle ., . \rangle$.

However, not all symmetric functions over $X \times X$ are kernels that can be used in a SVM, because a kernel function needs to satisfy a number of conditions imposed by the fact that it calculates an inner product in F. More precisely, in the SVM framework we distinguish two classes of kernel functions: i) *Positive Semi-Definite* kernels (PSD) and ii) *Conditionally Positive Definite* (CPD) kernels.

Whereas a PSD kernel can be considered as one of the most simple generalizations of one of the simplest similarity measures, i.e. the inner product, CPD kernels can be considered as generalizations of the simplest dissimilarity measure, i.e. the distance $\|\mathbf{x} - \mathbf{x}'\|^2$ [Berg et al. 1984, Schölkopf 2000]. Consider the following theorem.

Theorem 5. *Let X be the input space, then the function $K : X \times X \rightarrow \mathbb{R}$:*

$$K_{nd}\Big(\mathbf{x}, \mathbf{x}'\Big) = -\left\|\mathbf{x} - \mathbf{x}'\right\|^{\beta} \ \ \textit{with } 0 < \beta \leq 2 \tag{6}$$

is CPD. The kernel K defined in this way is referred to as the negative distance kernel.

Another result that is of particular interest to us relates a CPD K to a PSD kernel $\tilde{K}$ by plugging in K into the exponent of the standard radial basis function kernel, this is expressed in the following theorem [Berg et al. 1984]:

Theorem 6. *Let X be the input space and let $K : X \times X \rightarrow \mathbb{R}$ be a kernel, then K is CPD if and only if*

$$K_{rbf}\Big(\mathbf{x}, \mathbf{x}'\Big) = \exp\Big(\gamma K\left(\mathbf{x}, \mathbf{x}'\right)\Big) \tag{7}$$

is PSD for all $\gamma > 0$. The kernel K_{rbf} defined in this way is referred to as the radial basis function kernel.

For [Theorem 5] to work however, it was implicitly assumed that $X \subseteq \mathbb{R}^n$, because for non-vectorial data, like contexts, we can not define a norm or a normed difference like in the RHS of [Equation 6]. More precisely, given the results above, if we want to use an arbitrary distance d_X, defined on the input space X, in a kernel K, we should be able to express it as $d_X(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$ from which it then automatically follows that $-d_X$ is CPD by application of [Theorem 5].

In our case however, the input space is non-vectorial, i.e. $X \subseteq \mathbb{S}^n$ the set of all contexts of length n and the distances d_{SOM}, d_{WOM} and d_{MVDM} we would like to use can therefore not be expressed in terms of [Theorem 5]. Nevertheless, in previous work [Vanschoenwinkel et al. 2006, Liu et al. 2006] it has been shown that $-d_{SOM}, -d_{WOM}$ and $-d_{MVDM}$ are CPD, which will be briefly explained next. For more details the reader is referred to the related literature.

More precisely, for the overlap metric defined on the contexts it can be shown that it corresponds to an orthonormal vector encoding of those contexts [Vanschoenwinkel et al. 2006]. In the orthonormal vector encoding every string in the dictionary D is represented by a unique unit vector and complete contexts are formed by concatenating these unit vectors. Notice that this is actually the standard approach to context-dependent classification with SVMs [Hua and Sun 2001] and in this light the non-sensitive linear, polynomial, radial basis function and negative distance kernels employing the simple overlap metric (i.e. the unweighted case) presented next, are actually equivalent to the standard linear, polynomial, radial basis function and negative distance kernel applied to the orthonormal vector encoding of the contexts.

Finally, for MVDM with $r = 2$, it can be shown that it corresponds to the Euclidean distance in a transformed space, based on a probabilistic reformulation of the MVDM presented in [Kasif et al. 1998, Liu et al. 2006]. However, it should be noted that for MVDM with $r = 1$, $-d_{MVDM}$ is not CPD [Liu et al. 2006] thus we can't use it in our work.

3.2 A Weighted Polynomial Kernel

The first kernel defined here is based on [Equation 2] of the definition of the overlap metric from [Definition 2]. In the same way as before, we make a distinction between the unweighted non-sensitive case and the weighted context-sensitive case, for more details the reader is referred to [Vanschoenwinkel et al. 2006].

Definition 7. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p+q)+1$ the length of the contexts and $\mathbf{s}_i, \mathbf{t}_i \in D$ the strings at position i in the contexts as before, and let $\mathbf{w} \in \mathbb{R}^n$ be a context weight vector, then we define the *simple overlap kernel* $K_{SOK} : X \times X \to \mathbb{R}$ as

$$K_{SOK}\left(\bar{\mathbf{s}}, \bar{\mathbf{t}}\right) = \Big(\sum_{i=0}^{n-1} \delta(\mathbf{s}_i, \mathbf{t}_i) + c\Big)^d \tag{8}$$

with $c \geq 0$, $d > 0$ and $\mathbf{w} = \mathbf{1}$, the *weighted overlap kernel* $K_{WOK} : X \times X \to \mathbb{R}$ is defined in the same way but with a context weight vector $\mathbf{w} \neq \mathbf{1}$.

3.3 Negative Distance Kernels

Next, we give the definitions of three negative distance kernels employing the distances d_{SOM}, d_{WOM} and d_{MVDM}, for more details we refer to [Liu et al. 2006]. We start with the definition of two negative distance kernels using the overlap metric from [Definition 2]. Similarly, we make a distinction between the unweighted, non-sensitive case d_{SOM} and the weighted, context-sensitive case d_{WOM}.

Definition 8. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p + q) + 1$ the length of the contexts and $\mathbf{s}_i, \mathbf{t}_i \in D$ the strings at position i in the contexts as before, and let $\mathbf{w} \in \mathbb{R}^n$ be a context weight vector, then we define the *negative overlap distance kernel* $K_{NODK} : X \times X \to \mathbb{R}$ as

$$K_{NODK}\left(\bar{\mathbf{s}}, \bar{\mathbf{t}}\right) = -d_{SOM}(\bar{\mathbf{s}}, \bar{\mathbf{t}})^{\frac{1}{2}\beta} \tag{9}$$

with $0 < \beta \leq 2$ and $\mathbf{w} = \mathbf{1}$ as before, the *negative weighted distance kernel* $K_{NWDK} : X \times X \to \mathbb{R}$ is defined in the same way but substituting d_{WOM} for d_{SOM} in the RHS of [Equation 9], i.e. with a context weight vector $\mathbf{w} \neq \mathbf{1}$.

Similarly, for the MVDM from [Definition 3] we can define a negative distance type kernel as follows.

Definition 9. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p + q) + 1$ the length of the contexts and $\mathbf{s}_i, \mathbf{t}_i \in D$ the strings at position i in the contexts as before, then we define the *negative modified distance kernel* $K_{NMDK} : X \times X \to \mathbb{R}$ as

$$K_{NMDK}\left(\bar{\mathbf{s}}, \bar{\mathbf{t}}\right) = -d_{MVDM}(\bar{\mathbf{s}}, \bar{\mathbf{t}})^{\frac{1}{2}\beta} \tag{10}$$

with $0 < \beta \leq 2$ as before.

3.4 Radial Basis Function Kernels

Next, we will give the definitions of three radial basis function kernels employing the distances d_{SOM}, d_{WOM} and d_{MVDM}, for more details we refer to [Vanschoenwinkel et al. 2006, Liu et al. 2006].

We start with the definition of two radial basis function kernels employing the overlap metric from [Definition 2]. In the same way as before, we make a distinction between the unweighted non-sensitive case d_{SOM} and the weighted context-sensitive case d_{WOM}.

Definition 10. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p+q)+1$ the length of the contexts and $\mathbf{s}_i, \mathbf{t}_i \in D$ the strings at position i in the contexts as before, and let $\mathbf{w} \in \mathbb{R}^n$ be a context weight vector, then we define the overlap radial basis function kernel $K_{ORBF} : X \times X \to \mathbb{R}$ as

$$K_{ORBF}\Big(\bar{\mathbf{s}}, \bar{\mathbf{t}}\Big) = \exp\Big(-\gamma d_{SOM}(\bar{\mathbf{s}}, \bar{\mathbf{t}})\Big) \tag{11}$$

with $\gamma > 0$ as before, with $\mathbf{w} = \mathbf{1}$ and the weighted radial basis function kernel $K_{WRBF} : X \times X \to \mathbb{R}$ is defined in the same way but substituting d_{WOM} for d_{SOM} in the RHS of [Equation 11], i.e. with a context weight vector $\mathbf{w} \neq \mathbf{1}$.

Similarly, for the MVDM from [Definition 3] we can define a radial basis function type kernel as follows.

Definition 11. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $\bar{\mathbf{s}}_p^q$ and $\bar{\mathbf{t}}_p^q$ with $n = (p+q)+1$ the length of the contexts and $\mathbf{s}_i, \mathbf{t}_i \in D$ the strings at position i in the contexts as before, then we define the *modified radial basis function kernel* $K_{MRBF} : X \times X \to \mathbb{R}$ as

$$K_{MRBF}\Big(\bar{\mathbf{s}}, \bar{\mathbf{t}}\Big) = \exp\Big(-\gamma d_{MVDM}(\bar{\mathbf{s}}, \bar{\mathbf{t}})\Big) \tag{12}$$

with $\gamma > 0$ as before.

4 Experiments

4.1 Overview

In this section, we will perform two series of experiments. The first series of experiments are to see which context-sensitive kernel function is the most suitable kernel for splice site prediction. Then in the second series of experiments, we will explore more useful features to improve the performance further.

4.2 Software and Data

The experiments are conducted with LIBSVM [Chang and Lin 2001], which is a Java/C++ library for SVM learning. The dataset we use in the experiments is a set of human genes, which is referred to as HumGS [Degroeve 2004]. They obtained the data set from [Pertea et al. 2001] who on their turn obtained it from the GenBank, but checked it for errors and redundancy. In total, the data

data set	genes	$GT+$	$GT-$	$AG+$	$AG-$
HumGS	1115	5733	484714	5733	655822
training	/	4586	4586	4586	4586
testing	/	1147	96943	1147	131165

Table 1: Overview of the data sets used for the splice site prediction experiments.

set contains 1115 genes which at the level of the splice sites comes down to 5733 donor sites (denoted by $GT+$) and 484714 pseudo-donor sites (denoted by $GT-$) and 5733 acceptor sites (denoted by $AG+$) and 655822 pseudo-acceptor sites (denoted by $AG-$). Because we train a classifier to predict donor sites and another classifier to predict acceptor sites, separate training and test sets are constructed for donor and acceptor sites. For the purpose of training the classifiers, we constructed balanced training sets. For testing however we want a reflection of the real situation and keep the same ratio as given in the original set HumGS. This is shown in [Table 1].

Notice that we are limited by the number of actual splice sites as there are only 5733, both for donor sites and acceptor sites, at our disposition. For this reason, we do not construct a separate development set and parameter selection is done through 5-fold cross validation on the training set.

4.3 Parameter Selection and Accuracy

Parameter selection is done by 5-fold cross validation on the training set. For the ORBF, WRBF and MRBF, there are two free parameters that need to be optimized: The SVM cost parameter C (which is a trade-off for the model complexity and the model accuracy) and the radial basis function parameter γ. We performed a fine grid search for values of C and γ between 2^{-16} and 2^5. For the NODK, NWDK and NMDK only the cost parameter C has to be optimized because we choose β fixed to 1 as this gives very good results, more precisely for $\beta = 2$ results are not good at all, other values have not been tried. Again, values for C between 2^{-16} and 2^5 have been considered.

For the SOK and the WOK we take $d = 2$ and $c = 0$ as previous work pointed out that higher values for d actually led to bad results, while taking values for $c > 0$ does not have a significant impact on the results.

As a weighting scheme for the weighted kernels, we used three different weights: Information Gain (IG), Gain Ratio (GR) and Shared Variance (SV).

Splice site prediction systems are often evaluated by means of the percentage of FP classifications at a particular recall rate. This measure is referred to as

$FP\%$ [Degroeve 2004] and is calculated as follows:

$$FP\% = \frac{\#\text{ false positives}}{\#\text{ false positives} + \#\text{ true negatives}} \times 100$$

We used this evaluation measure for a recall rate of 95%, in this case the measure is referred to as $FP95\%$, i.e. the $FP95\%$ measure gives the percentage of the predictions falsely classified as actual splice site at a level where the system has found 95% of all actual splice sites in the test set. Note that it is the purpose to have $FP95\%$ as low as possible.

4.4 Experiments 1: Find out the most suitable kernel function.

In the first step, our purpose is to try out all the kernel functions mentioned above to see which one is the most suitable kernel. In order to keep the model simple, here we only make use of one feature *single nucleotide* to represent the instance and choose a fixed context size of 50 nucleotides before and 50 nucleotides after the candidate splice site.

[Table 2] and [Table 3] give an overview of the final $FP95\%$ results and model complexity in terms of the number of support vectors of the different kernels on the splice site prediction task. Note that the confidence intervals have been obtained by bootstrap resampling, at a confidence level $\alpha = 0.05$ [Noreen 1989]. A $FP95\%$ rate outside of these intervals is assumed to be significantly different from the related $FP95\%$ rate at a confidence level of $\alpha = 0.05$.

From the results it can be easily seen that in all cases the context-sensitive kernels making use of the WOM with IG, GR and SV weights and the MVDM always outperform their simple non-sensitive counterparts both in accuracy and in model complexity. Moreover in almost all cases this happens with a significant difference. Another overall observation is that the difference in the results between different context weights is not significant at all. Finally, it can be seen that the best result for donor sites and acceptor sites is obtained by the WRBF with IG weights. It should be noticed that WRBF with IG doesn't outperform other context-sensitive kernels ,e.g., MRBF, NWDK, WOK, considering a confidence level of $\alpha = 0.05$. However, based on our knowledge on kernel theory [Cristianini and Shawe-Taylor 2004] and previous study [Liu et al. 2006], it shows that RBF kernel can always achieve the best performance. Hence we choose WRBF with IG weights for the next experiments.

4.5 Experiments 2: Continue to improve performance further.

From the previous experiments, we know that the WRBF with IG can achieve the best results. However, it is clearly not good enough. So for the next step, we consider improving the performance further:

	Donor Site		
Kernel and Weights	Fixed Parameters	$FP95\%$	$\#SVs$
SOK	$c = 0, d = 2$	7.19 ± 0.70	3414
WOK/GR	$c = 0, d = 2$	6.51 ± 0.72	2126
WOK/IG	$c = 0, d = 2$	6.38 ± 0.80	2151
WOK/SV	$c = 0, d = 2$	6.43 ± 0.67	2156
NODK	$\beta = 1$	7.97 ± 1.02	3372
NWDK/GR	$\beta = 1$	6.43 ± 0.68	2803
NWDK/IG	$\beta = 1$	6.40 ± 0.66	3009
NWDK/SV	$\beta = 1$	6.38 ± 0.70	3169
NMDK	$\beta = 1, r = 2$	6.38 ± 0.59	2625
ORBF	/	7.46 ± 0.77	4327
WRBF/GR	/	6.25 ± 0.72	2346
WRBF/IG	/	$\mathbf{6.21 \pm 0.57}$	2348
WRBF/SV	/	6.27 ± 0.75	2440
MRBF	$r = 2$	6.40 ± 0.78	2364

Table 2: Splice site prediction, results for all kernels for donor sites.

	Acceptor Site		
Kernel and Weights	Fixed Parameters	$FP95\%$	$\#SVs$
SOK	$c = 0, d = 2$	10.00 ± 1.22	3635
WOK/GR	$c = 0, d = 2$	9.06 ± 1.17	2698
WOK/IG	$c = 0, d = 2$	9.04 ± 1.11	2647
WOK/SV	$c = 0, d = 2$	9.07 ± 1.23	2695
NODK	$\beta = 1$	11.36 ± 1.44	3696
NWDK/GR	$\beta = 1$	9.71 ± 1.52	3143
NWDK/IG	$\beta = 1$	9.66 ± 1.57	3380
NWDK/SV	$\beta = 1$	9.76 ± 1.55	3252
NMDK	$\beta = 1, r = 2$	12.63 ± 1.46	3146
ORBF	/	10.50 ± 1.66	4927
WRBF/GR	/	8.60 ± 1.65	2881
WRBF/IG	/	$\mathbf{8.49 \pm 1.73}$	2906
WRBF/SV	/	9.06 ± 1.43	2703
MRBF	$r = 2$	12.19 ± 1.67	2836

Table 3: Splice site prediction, results for all kernels for acceptor sites.

Given a fragment of DNA sequence:

c	t	c	t	c	GT/AG	g	g	a	c	t
-5	-4	-3	-2	-1	0	1	2	3	4	5

⇩

Extracted feature sets:

SN feature sets	[c , t , c , t , c , g , g , a , c , t]
DN feature sets	[ct , tc , ct , tc , gg , ga , ac , ct]
TN feature sets	[ctc , tct , ctc , gga , gac , act]

Figure 2: Feature sets extracted for splice site prediction.

	Donor Site	Acceptor Site
SN (left/right)	60/40	40/100
DN (left/right)	60/40	80/100
TN (left/right)	60/60	80/100

Table 4: Optimal left/right context sizes for donor sites and acceptor sites. Notice that if two left/right context combinations can get the same result, we choose one with smaller context sizes in order to reduce the computational complexity.

1. consider not only single nucleotide (SN), but also di-nucleotide (DN) and tri-nucleotide (TN).

2. try the different left/right context size to get the optimal context size.

So far we only utilize single nucleotides. Inspired from our previous work, sometimes the combinations of nucleotides also can contribute some useful information. Hence we extract not only SN, but also DN and TN as features to represent the instance. The brief procedure is illustrated in [Fig. 2].

Then, for each feature set (SN, DN and TN), in order to find the optimal combination of left/right context size, we do a grid search for left/right context size between 20 and 100. [Table 4] lists the optimal left/right context size for donor site and acceptor site. And [Table 5] and [Table 6] represents all the results by doing a grid search to find the optimal left/right context size of SN, DN and TN for both donor site and acceptor site, respectively.

Finally, we combine all the feature sets (SN, DN and TN) with their corresponding optimal context sizes to construct the final feature set. We choose the WRBF kernel with IG weighting to perform 5-fold cross validation on the training data to find the optimal parameters, C and γ (chosen from 2^{-16} to 2^{5}),

		Donor Site ($FP95\%$)				
Left Context	Right Context	20	40	60	80	100
SN	20	7.82	7.54	7.67	7.80	7.84
	40	7.28	6.89	7.23	7.04	7.37
	60	6.97	**6.54**	6.56	6.82	6.82
	80	7.04	6.67	6.97	6.62	6.99
	100	6.99	6.86	6.75	6.93	6.97
DN	20	6.89	6.34	6.56	6.47	6.60
	40	6.23	5.82	5.95	6.14	6.32
	60	5.84	**5.51**	5.51	5.60	5.66
	80	6.10	5.77	5.77	5.84	5.84
	100	6.19	5.88	5.93	5.80	6.06
TN	20	6.99	6.67	6.78	6.91	7.06
	40	6.49	6.19	6.17	6.36	6.54
	60	6.47	5.93	**5.86**	6.04	5.99
	80	6.27	6.04	5.88	5.99	6.14
	100	6.14	6.04	5.97	5.95	6.04

Table 5: Results for a grid search to find the optimal left/right context sizes for donor sites. All the results are obtained using the kernel WRBF with IG and for the parameters, C is chosen from 2^{-16} to 2^5, γ is also chosen from 2^{-16} to 2^5.

build the model on the entire training data with the optimal parameters and apply the obtained model on the test data to do the splice site predictions.

The final results are shown in [Fig. 3] compared with the result of Experiments 1. It can seen clearly that the $FP95\%$ values of Experiments 2 are much lower than those of Experiments 1, i.e., 2.27 lower for donor sites and 2.51 lower for acceptor sites. The model built here can always outperform that built in [Subsection 4.4]. This means that the DN and TN feature sets indeed have a positive effect on the performance and hence it is worth incorporating them into our prediction model.

4.6 Analyzing Experimental Results.

In this section, we will compare our final experimental results with other leading systems, GeneSplicer[Pertea et al. 2001] and Maxentscan[Yeo and Burge 2003], which have been already published in the literature. GeneSplicer uses a decision tree method called maximal dependence decomposition (MDD) and enhances it with Markov models that capture additional dependencies among neighboring

		Acceptor Site ($FP95\%$)				
Left Context	Right Context	20	40	60	80	100
SN	20	10.75	10.36	9.40	9.49	9.42
	40	10.31	10.05	9.33	8.92	**8.78**
	60	10.68	9.70	9.35	8.87	8.92
	80	10.53	9.79	9.15	9.07	8.98
	100	10.55	9.92	9.02	9.03	8.98
DN	20	9.48	9.00	8.50	7.95	7.93
	40	9.57	8.70	8.17	7.82	7.54
	60	9.26	8.54	8.11	7.80	7.58
	80	9.18	8.67	8.39	7.76	**7.45**
	100	9.41	8.67	8.41	7.98	7.69
TN	20	11.55	10.81	9.92	9.24	9.11
	40	11.27	10.24	9.44	8.59	8.61
	60	11.14	10.00	9.07	8.78	8.52
	80	10.90	10.16	9.31	8.94	**8.46**
	100	10.70	10.35	9.46	8.94	8.50

Table 6: Results for a grid search to find the optimal left/right context sizes for acceptor sites. All the results are obtained using the kernel WRBF with IG and for the parameters, C is chosen from 2^{-16} to 2^5, γ is also chosen from 2^{-16} to 2^5.

	Donor Site ($FP95\%$)	Acceptor Site ($FP95\%$)
Our system	3.94 ± 0.49	5.98 ± 0.72
GeneSplicer	6.50	5.95
Maxentscan	7.80	11.00

Table 7: Evaluation results of our system compared with other systems.

bases in a region around the splice sites. And Maxentscan makes use of Maximum Entropy principle. The detailed results are listed in [Table 7].

From [Table 7], it is clear that for donor site, our system can outperform other two systems with confidence level of $\alpha = 0.05$ while for acceptor site, it can achieve the approximately same result as GeneSplicer and two times better performance than Maxentscan does. Therefore the experimental results conform to our expectation. SVMs can make use of kernel trick to avoid computing the inner product in high-dimensional feature space so that SVMs won't suffer from local minimum problem. Moreover, by incorporating special purpose similarity or

Figure 3: Comparison the $FP95\%$ results of between Experiments 1 and 2. It can be shown that the difference is significant with 95% confidence interval.

dissimilarity measures in kernel functions, we can capture more essential domain knowledge to increase the accuracy of our system.

5 Conclusions and Future Work

In this paper it has been shown how different statistical measures and distance functions can be included into kernel functions for SVM learning in context-dependent classification tasks. The purpose of this approach is to make the kernels sensitive to the amount of information that is present in the contexts. More precisely, the case of splice site prediction has been discussed and from the experimental results it becomes clear that the sensitivity information has a positive effect on the results. Moreover, with further well designed features and carefully chosen context sizes, we can improve the overall performance more. Through comparing our final experimental results with other leading splice site prediction systems, it can be seen easily that our system can outperform them significantly with 95% confidence interval.

In the future, we will consider applying our context-sensitive kernels on more complex features like combinational features, reading frame features, etc. which have been investigated in [Degroeve 2004] to see whether the sensitivity information will still be significant in a system. And we will explore some more complex distance functions into kernel functions, e.g., Levenshtein distance [Levenshtein 1966], Jaro-Winkler distance [Winkler 1999] and etc.

Acknowledgements

The first author Yifei Chen is funded by a doctoral grant of the Fonds voor Wetenschappelijk Onderzoek (FWO). The second author Feng Liu is funded by a doctoral grant of the Vrije Universiteit Brussel (VUB) and contributes equally with the first author.

References

[Degroeve 2004] Degroeve, S. : "Design and Evaluation of a Linear Classification Strategy for Gene Structural Element Recognition"; PhD thesis, Universiteit Gent, Belgium (2004).

[Cristianini and Shawe-Taylor 2004] Cristianini, N., Shawe-Taylor, J.: "Kernel Methods For Pattern Analysis"; Cambridge University Press (2004).

[Schölkopf 2000] Schölkopf, B.: "The Kernel Trick for Distances"; Microsoft Research (2000).

[Vanschoenwinkel and Manderick 2004] Vanschoenwinkel, B., Manderick, B.: "Substitution Matrix based Kernel Functions for Protein Secondary Structure Prediction"; Proceedings of International Conference on Machine Learning and Applications (2004).

[Vanschoenwinkel et al. 2006] Vanschoenwinkel, B., Liu, F., Manderick, B.: "Context-sensitive Kernel Functions : A Distance Function Viewpoint"; Lecture Notes in Artificial Intelligence 3930, Springer, Berlin (2006), 861-870.

[Vanschoenwinkel et al. 2005] Vanschoenwinkel, B., Liu, F., Manderick, B.: "Weighted Kernel Functions for SVM Learning in String Domains: A Distance Function Viewpoint"; Proceedings of International Conference on Machine Learning and Cybernetics 7, (2005) 4226-4232.

[Quinlan 1986] Quinlan, J.R.: "Induction of Decision Trees"; Machine Learning 1, (1986) 81-206.

[Quinlan 1993] Quinlan, J.R.: "C4.5: Programs for Machine Learning"; Morgan Kaufmann, CA (1993).

[White and Liu 1994] White, A.P., Liu, W.: "Bias in Information-based measures in decision tree induction"; Machine Learning 15(3), (1994) 321-329.

[Cost and Salzberg 1993] Cost, S., Salzberg, S.: "A Weighted Nearest Neighbour Algoritm for Learning with Symbolic Features" Machine Learning 10, (1993) 57-78.

[Stanfill and Waltz 1986] Stanfill, C., Waltz, D. : "Toward memory-based reasoning"; Communications of the ACM 29(12), (1986) 1213-1228.

[Berg et al. 1984] Berg, C., Christensen, J.P.R., Ressel, P.: "Harmonic analysis on semigroups. Theory of positive definite and related functions"; Graduate Texts in Mathematics, Springer-Verlag (1984).

[Liu et al. 2006] Liu, F., Vanschoenwinkel, B., Chen, Y., Manderick, B.: "A Modified Value Difference Metric Kernel for Context-Dependent Classification Tasks"; Proceedings of International Conference on Machine Learning and Cybernetics (2006).

[Hua and Sun 2001] Hua, S., Sun, Z.: "A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach"; Journal Of Molecular Biology 308(2), (2001) 397-407.

[Kasif et al. 1998] Kasif, S., Salzberg, S., Waltz, D.L., Rachlin, J., Aha, D.W.: "A Probabilistic Framework for Memory-Based Reasoning"; Artificial Intelligence 104(1-2), (1998) 287-311.

[Chang and Lin 2001] Chang, C., Lin, C.: "LIBSVM : A Library for Support Vector Machines"; (2001) `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Noreen 1989] Noreen, E.W.: "Computer-Intensive Methods for Testing Hypotheses"; John Wiley & Sons (1989).

[Pertea et al. 2001] Pertea, M., Lin, X., Salzberg, S.: "GeneSplicer: a new computational method for splice site prediction"; Nucleic Acids Research 29(5), (2001) 1185-1190.

[Yeo and Burge 2003] Yeo, G., Burge, C.: "Maximum entropy modeling of short sequence motifs with applications to rna splicing signals"; Proceedings of the 7th Intl. Conf. on Res. in Comp. Mol. Bio. (RECOMB), (2003) 322-331.

[Joachims 2002] Joachims, T.: "Learning to Classify Text Using Support Vector Machines"; Kluwer Academic Publishers (2002).

[Levenshtein 1966] Levenshtein, V. I.: "Binary codes capable of correcting deletions, insertions, and reversals"; Soviet Physics Doklady 10, (1966) 707-710.

[Winkler 1999] Winkler, W.E.: "The State of Record Linkage and Current Research Problems"; Statistics of Income Division, Internal Revenue Service Publication RR99/04 (1999).