1 Efficient Splice Site Prediction with Context-Sensitive Distance Kernels

Bernard Manderick, Feng Liu and Bram Vanschoenwinkel

This paper presents a comparison between different context-sensitive kernel functions for doing splice site prediction with a support vector machine. Four types of kernel functions will be used: linear-, polynomial-, radial basis function- and negative distance-based kernels. Domain-knowledge can be incorporated into the kernels by incorporating statistical measures or by directly plugging in distance functions defined on the splice site instances. From the experimental results it becomes clear that the radial basis function-based kernels get the best accuracies. However, because classification speed is of crucial importance to the splice site prediction system, this kernel is computationally too expensive. Nevertheless, in general incorporating domain knowledge does not only improve classification accuracy, but also reduces model complexity which in its turn again increases classification speed.

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. In general, a gene can be defined as a region in a DNA sequence that is used in the production of a specific protein. In many genes, the DNA sequence coding for proteins, called 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 introns are subtracted from the exons.

Hence we can make a distinction between two different splice sites: i) the exon-intron boundary, referred to as the donor site and ii) the intron-exon boundary, referred to as the acceptor site. Splice site prediction, an important subtask in gene finding systems, is the automatic identification of those regions in the DNA sequence that are either donor sites or acceptor sites [?].

Because splice site prediction instances can be represented by a context of a number of nucleotides before and after the candidate splice site, it is called a context-dependent classification task. In this paper we do splice prediction with support vector machines (SVMs) using kernel functions that take into account the information available at different positions in the contexts. In this sense the kernel functions are called context-sensitive. This is explained in Sections ?? and ??.
More precisely, in a support vector machine, the data is first mapped non-linearly from the original input space $X$ to a high-dimensional Hilbert space called the feature space $F$ and then separated by a maximum-margin hyperplane, i.e. linearly, in that space $F$. By making use of the kernel trick, the mapping $\emptyset : X \rightarrow F$ remains implicit, and as a result we avoid working in the high-dimensional feature space $F$. Moreover, because the mapping is non-linear, the decision boundary which is linear in $F$ corresponds to a non-linear decision boundary in the input space $X$. One of the most important design decisions in SVM learning is the choice of kernel function $K : X \times X \rightarrow \mathbb{R}$ because the maximum-margin hyperplane is defined completely by inner products of vectors in the Hilbert feature space $F$ using the kernel function $K$. Since $K$ takes elements $x$ and $y$ from the input space $X$ and calculates the inner products of $\emptyset(x)$ and $\emptyset(y)$ in the Hilbert feature space $F$ without having to represent or even to know the exact form of the elements $\emptyset(x)$ and $\emptyset(y)$. As a consequence the mapping $\emptyset$ remains implicit and we have a computational benefit [?]. In the light of the above it is not hard to see that computational efficiency of $K$ is crucial for the success of the classification process. We refer to Section ?? for more on theoretical background concerning the SVM.

As a result, the learning process can benefit a lot from the use of special purpose similarity or dissimilarity measures in the calculation of $K$ [?,?,?,?]. However, incorporating such knowledge in a kernel function is non-trivial since a kernel function $K$ has to satisfy a number of properties that result directly from the definition of an inner product. In this paper we will consider two types of kernel functions that can make direct use of distance functions defined on contexts themselves: i) negative distance kernels and ii) radial basis function kernels. This is explained in Section ??.

Furthermore, because classification speed is of crucial importance to a splice site prediction system the used kernel functions should be computationally very efficient. For that reason, in related work on splice site prediction with a SVM a linear kernel is chosen in favor of computational efficiency but at the cost of some accuracy [?]. In this light most of the kernels presented here will probably be too expensive, therefore we also show results for context-sensitive linear kernels and from these results it can be seen that the classification speed can be further increased while at the same time precision and accuracy of the predictions are a little higher. This is discussed in Section ??.

2. Context-Dependent Classification

In this paper we consider classification tasks where it is the purpose to classify a focus symbol in a sequence of symbols, based on a number of symbols before and after the focus symbol. The focus symbol, together with the symbols 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. Here, each symbol is one of the four nucleotides $\{A,C,G, T\}$. 
2.1 Contexts

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

**Definition 1.** A context \( s_p^q \) is a sequence of symbols \( s_i \in D \) with \( p \) symbols before and \( q \) symbols after a focus symbol \( s_p \) at position \( p \) as follows

\[
s_p^q = (s_0, \ldots, s_p, \ldots, s_{p+q})
\]

with \((p + q) + 1\) the length of the context, with \( D \) the dictionary of all symbols, 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. Essentially, DNA is a sequence of nucleotides represented by a four character alphabet or dictionary \( D = \{A,C,G,T\} \). Moreover, an acceptor site always contains the AG dinucleotide and a donor site always contains the GT dinucleotide. In this light 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, \( \ldots CCATTGGTGGCAGCCAG \ldots \) the candidate donor site given by the dinucleotide GT can be represented by a context in terms of Definition ?? as

\[
s_p^q = \left(\begin{array}{c}
\end{array}\right)
\]

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.

Furthermore, for splice site prediction there is no need to represent the AG/GT dinucleotides, because two separate classifiers are trained, one for donor sites and one for acceptor sites. In this light the only possible symbols occurring in the contexts are given by the dictionary \( D \).

Note that, for reasons of computational efficiency, in practice the symbols in the contexts will be represented by an integer, more precisely by assigning all the symbols that occur in the training set a unique index and subsequently using that index in the context instead of the symbols themselves.

2.2 The Overlap Metric

The most basic distance function defined on contexts is called the overlap metric, it simply counts the number of mismatching symbols at corresponding positions in two contexts.
Definition 2. Let $S^n$ be a set with contexts $s_p^q$ and $t_p^q$ with $n = (p + q) + 1$ the length of the contexts, with symbols $s_i$, $t_i \in D$ the dictionary of all distinct symbols with $|D| = m$ and let $w \in \mathbb{R}^n$ be a context weight vector. Then the overlap metric $d_{OM} : S^n \times S^n \to \mathbb{R}^+$ is defined as

\[
S^n \times S^n \to \mathbb{R}^+ : d_{OM} (\bar{s}, t) = \sum_{i=0}^{n-1} w_i \delta (s_i, t_i)
\]

with $\delta : S \times S = \mathbb{R}^+$ defined as

\[
\delta (s, t) = \begin{cases} w_i & \text{if } s_i = t_i \\ 0 & \text{else} \end{cases}
\]

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

Next, we make a distinction between two cases: i) if all $w_i = 1$ no weighting takes place and the metric is referred to as the simple overlap metric $d_{SOM}$ and ii) otherwise 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 [?], ii) gain ratio [?] and iii) shared variance [?]. For more details the reader is referred to the related literature.

2.3 The Modified Value Difference Metric

The Modified Value Difference Metric (MVDM) [?] is a powerful method for measuring the distance between sequences of symbols like the contexts considered here. The MVDM is based on the Stanfill-Waltz Value Difference Metric introduced in 1986 [?]. The MVDM determines the similarity of all the possible symbols at a particular context position by looking at co-occurrence of the symbols with the target class. Consider the following definition.

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

\[
S^n \times S^n \to \mathbb{R}^+ : d_{MVDM} (\bar{s}, \bar{t}) \sum_{i=0}^{n-1} (s_i, t_i)^r
\]

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:
Key Note 1: Efficient Splice Site Prediction with Context-Sensitive Distance Kernels

\[ \delta(s_i, t_i) = \sum_{j=1}^{M} |p(y_j | s_i) - p(y_j \mid t_i)|^\gamma \]  

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

3.1 Theoretical Background

Remind that in the SVM framework classification is done by considering a kernel induced feature mapping \( \Phi \) 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 \( K : X \times X \to \mathbb{R} \) is a symmetric function so that for all \( x \) and \( x' \) in \( X \), \( K(x, x') = \langle \phi(x), \phi(x') \rangle \) where \( \phi \) is a (non-linear) mapping from the input space \( X \) into the Hilbert space \( F \) provided with the inner product \( \langle \cdot, \cdot \rangle \).

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 \( \| x - x' \| \) \[\text{[?],[?],[?]}\].

One type of CPD kernel that is of particular interest to us is given in [?] from which we quote the following two theorems.

**Theorem 1.** Let \( X \) be the input space, then the function \( K : X \times X \to \mathbb{R} : \)

\[ K_{\text{nd}}(x - x') = \| x - x' \|^\beta \text{ with } 0 < \beta \leq 2 \]  

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 [?]:
Theorem 2. 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

$$K_{rbf}(x - x') = \exp (\gamma K(x - x'))$$

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 2 to work, it is assumed that $X \subseteq \mathbb{R}^n$ where $\mathbb{R}^n$ is a normed vector space. But, for contexts in particular and sequences of symbols in general one can not define a norm like in the RHS of Equation 2. 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 $d_X(x - x') = ||x - x'||$ from which it then automatically follows that $-d_X$ is CPD by application of Theorem 2.

In our case however, since the input space $X \subseteq \mathbb{S}^n$ the set of all contexts of length $n$, the distances $d_{SOM}$, $d_{WOM}$ and $d_{MVDM}$ we would like to use can therefore not be expressed in terms of Theorem 2. Nevertheless, in previous work it has been shown that $-d_{SOM}$, $-d_{WOM}$ and $-d_{MVDM}$ are CPD [?,?,?,?], this will be briefly explained next. For more details the reader is referred to the 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 [?,?,?]. In the orthonormal vector encoding every symbol 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 [?,?,?] 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 the 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 [?,?,?].

3.2 A Weighted Polynomial Kernel

The first kernel we will define 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 [?,?,?].

Definition 5. Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $s_p^q$ and $t_p^q$ with $n = (p+q)+1$ the length of the contexts and $s_p$, $t_i \in D$ the symbols at position $i$ in the contexts as before, and let $w \in \mathbb{R}^n$ be a context weight vector, then we define the simple overlap kernel $K_{SO} : X \times X \rightarrow \mathbb{R}$ as
\[ K_{\text{SOK}}(s, t) = \left( \sum_{i=0}^{n-1} \delta(s_i, t_i) + c \right)^d \]  

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

### 3.3 Negative Distance Kernels

Next, we give the definitions of three negative distance kernels employing the distances \( d_{\text{SOM}}, d_{\text{WOM}} \) and \( d_{\text{MVDM}} \), for more details we refer to [?,?,?].

We start with the definition of two negative distance kernels using the overlap metric from Definition ???. In the same way as before, we make a distinction between the unweighted, non-sensitive case \( d_{\text{SOM}} \) and the weighted, context-sensitive case \( d_{\text{WOM}} \).

**Definition 6.** Let \( X \subseteq \mathbb{S}^n \) be the input space with contexts \( \cdot_s^q \) and \( \cdot_t^q \) with \( n = (p+q)+1 \) the length of the contexts and \( s_p, t_q \in D \) the symbols at position \( i \) in the contexts as before, and let \( w \in \mathbb{R}^n \) be a context weight vector, then we define the negative overlap distance kernel \( K_{\text{NODK}} : X \times X \to \mathbb{R} \) as

\[ K_{\text{NODK}}(s, t) = -d_{\text{SOM}}(s, t)^{\frac{1}{\beta}} \]  

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

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

**Definition 7.** Let \( X \subseteq \mathbb{S}^n \) be the input space with contexts \( \cdot_s^q \) and \( \cdot_t^q \) with \( n = (p+q)+1 \) the length of the contexts and \( s_p, t_q \in D \) the symbols at position \( i \) in the contexts as before, then we define the negative modified distance kernel \( K_{\text{NMDK}} : X \times X \to \mathbb{R} \) as

\[ K_{\text{NMDK}}(s, t) = -d_{\text{MVDM}}(s, t)^{\frac{1}{\beta}} \]  

with \( 0 < \beta \leq 2 \) as before.
However, it should be noted that for $r = 1$ in the definition of the MVDM $-d_{MVDM}$ is not CPD and thus for $r = 1$ the kernel $K_{NMDK}$ will also not be CPD. Nevertheless, given the good empirical results we will use $K_{NMDK}$ with $d_{MVDM}$ and $r = 1$ anyway.

### 3.4 Radial Basis Function Kernels

Next, we 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 [? , ?, ?].

We start with the definition of two radial basis function kernels employing the overlap metric from Definition ???. 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 8.** Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $s_p$ and $t_p$ with $n = (p+q)+1$ the length of the contexts and $s_p$, $t_i \in D$ the symbols at position $i$ in the contexts as before, and let $w \in \mathbb{R}^n$ be a context weight vector, then we define the overlap radial basis function kernel $K_{ORBF}: X \times X \rightarrow \mathbb{R}$ as

$$K_{ORBF}(s, t) = \exp \left(-\gamma d_{SOM}(s, t)\right)$$

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

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

**Definition 9.** Let $X \subseteq \mathbb{S}^n$ be the input space with contexts $s_p$ and $t_p$ with $n = (p+q)+1$ the length of the contexts and $s_p$, $t_i \in D$ the symbols at position $i$ in the contexts as before, then we define the modified radial basis function kernel $K_{MRBF}: X \times X \rightarrow \mathbb{R}$ as

$$K_{MRBF}(s, t) = \exp \left(-\gamma d_{MVDM}(s, t)\right)$$

with $\gamma > 0$ as before.

It should however be noted that, with respect to the discussion above, i.e. that for $r = 1$ the distance $-d_{MVDM}$ and corresponding kernel $K_{NMDK}$ are not CPD and therefore here for $r = 1$ the kernel $K_{MRBF}$ will not be PSD. Nevertheless, given the good empirical results we used it anyway.
4. Experiments

We have done a number of experiments, first of all we wanted to validate the feasibility of our approach and compare our kernel functions that operate on contexts directly and see whether they are doing at least as well and hopefully better than the traditional kernels. In these experiments the left and right context length was set to 50. Second, we set up some experiments to find the optimal left and right context length for each classifier. Third, we also looked at di- and trinucleotides to find out whether this gave better performance than the single nucleotide case.

In the next sections, we describe the software and the data sets used in our experiments, we discuss how we have set the different parameters for the SVM, we present and discuss the results obtained and finally we give an overview of related work.

4.1. Software and Data

We did the experiments with LIBSVM [?], 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 [?]. Each instance is represented by a fixed context size of 50 nucleotides before and 50 nucleotides after the candidate splice site based on the initial design strategy in [?]. Since, we train one 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 ??.

4.2. 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 $\gamma$.

Table 1. Overview of the data sets that have been used for the splice site prediction experiments.

<table>
<thead>
<tr>
<th>data set</th>
<th>genes</th>
<th>GT+</th>
<th>GT−</th>
<th>AG+</th>
<th>AG−</th>
</tr>
</thead>
<tbody>
<tr>
<td>HumGS</td>
<td>1115</td>
<td>5733</td>
<td>484714</td>
<td>5733</td>
<td>655822</td>
</tr>
<tr>
<td>training</td>
<td>/</td>
<td>4586</td>
<td>4586</td>
<td>4586</td>
<td>4586</td>
</tr>
<tr>
<td>testing</td>
<td>/</td>
<td>1147</td>
<td>96943</td>
<td>1147</td>
<td>131165</td>
</tr>
</tbody>
</table>

We performed a fine grid search for values of $C$ and between $2^{-16}$ and $2^5$. For the NODK, NWDK and NMDK only the cost parameter $C$ has to be optimized because we choose 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 $2^2$ have been considered.

The LINEAR kernel in the results below refers to the the overlap kernel from Definition ?? with $d = 1$ and $c = 0$. For the SOK and the WOK we take $d = 2$ and $c = 0$ as previous work pointed out that higher values for d actually leads 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). For more details the reader is referred to [?].

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% [?] 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.

4.3. Results

Table ?? gives 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 $\alpha = 0.05$ [?]. A FP95% rate outside of these intervals is assumed to be significantly different from the related FP95% rate at a confidence level of $\alpha = 0.05$. In addition to the final FP95% results we also give as an illustration two FP% plots, for donor sites, comparing the context-sensitive kernels with those kernels that are not context-sensitive. Figure ?? does this for the negative distance kernel making use of the MVDM and Figure ?? does this for the radial basis function kernel making use of the WOK with GR, IG and SV weights.
Table 2. Splice site prediction, results for all kernels, for donor sites and for acceptor sites.

<table>
<thead>
<tr>
<th>Kernel and Weights</th>
<th>FP95%</th>
<th>#S Vs</th>
<th>FP95%</th>
<th>#S Vs</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEAR</td>
<td>8.18 ± 0.92</td>
<td>2398</td>
<td>12.78 ± 1.45</td>
<td>3621</td>
</tr>
<tr>
<td>LINEAR/GR</td>
<td>7.86 ± 1.05</td>
<td>1902</td>
<td>11.50 ± 1.55</td>
<td>2499</td>
</tr>
<tr>
<td>LINEAR/IG</td>
<td>7.92 ± 1.08</td>
<td>1905</td>
<td>11.78 ± 1.82</td>
<td>2494</td>
</tr>
<tr>
<td>LINEAR/SV</td>
<td>7.88 ± 1.02</td>
<td>2072</td>
<td>11.60 ± 2.01</td>
<td>2500</td>
</tr>
<tr>
<td>SOK</td>
<td>7.19 ± 0.70</td>
<td>3414</td>
<td>10.00 ± 1.22</td>
<td>3635</td>
</tr>
<tr>
<td>WOK/GR</td>
<td>6.51 ± 0.72</td>
<td>2126</td>
<td>9.06 ± 1.17</td>
<td>2698</td>
</tr>
<tr>
<td>WOK/IG</td>
<td>6.38 ± 0.80</td>
<td>2151</td>
<td>9.04 ± 1.11</td>
<td>2647</td>
</tr>
<tr>
<td>WOK/SV</td>
<td>6.43 ± 0.67</td>
<td>2156</td>
<td>9.07 ± 1.23</td>
<td>2695</td>
</tr>
<tr>
<td>NODK</td>
<td>7.97 ± 1.02</td>
<td>3372</td>
<td>11.36 ± 1.44</td>
<td>3696</td>
</tr>
<tr>
<td>NWDK/GR</td>
<td>6.43 ± 0.68</td>
<td>2803</td>
<td>9.71 ± 1.52</td>
<td>3143</td>
</tr>
<tr>
<td>NWDK/IG</td>
<td>6.40 ± 0.66</td>
<td>3009</td>
<td>9.66 ± 1.57</td>
<td>3380</td>
</tr>
<tr>
<td>NWDK/SV</td>
<td>6.38 ± 0.70</td>
<td>3169</td>
<td>9.76 ± 1.55</td>
<td>3252</td>
</tr>
<tr>
<td>NMDK (r = 1)</td>
<td>6.26 ± 0.75</td>
<td>2995</td>
<td>10.70 ± 1.46</td>
<td>2902</td>
</tr>
<tr>
<td>NMDK (r = 2)</td>
<td>6.38 ± 0.59</td>
<td>2625</td>
<td>12.63 ± 1.46</td>
<td>3146</td>
</tr>
<tr>
<td>ORBF</td>
<td>7.46 ± 0.77</td>
<td>4327</td>
<td>10.50 ± 1.66</td>
<td>4927</td>
</tr>
<tr>
<td>WRBF/GR</td>
<td>6.25 ± 0.72</td>
<td>2346</td>
<td>8.60 ± 1.65</td>
<td>2881</td>
</tr>
<tr>
<td>WRBF/IG</td>
<td>6.21 ± 0.57</td>
<td>2348</td>
<td>8.49 ± 1.73</td>
<td>2906</td>
</tr>
<tr>
<td>WRBF/SV</td>
<td>6.27 ± 0.75</td>
<td>2440</td>
<td>9.06 ± 1.43</td>
<td>2703</td>
</tr>
<tr>
<td>MRBF (r = 1)</td>
<td>5.81 ± 0.68</td>
<td>2696</td>
<td>10.36 ± 1.28</td>
<td>3652</td>
</tr>
<tr>
<td>MRBF (r = 2)</td>
<td>6.40 ± 0.78</td>
<td>2364</td>
<td>12.19 ± 1.67</td>
<td>2836</td>
</tr>
</tbody>
</table>

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. There is however one exception, i.e. the MRBF with $r = 2$ for acceptor sites performs worse than its non-sensitive counterpart. 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 is obtained by the MRBF with $r = 1$, but for acceptor sites this is the WRBF with IG weights. Therefore it is clear that the success of the used metric and the used weights depends, for a great deal, on the properties of the data under consideration so that it is worthwhile trying different metrics and different context weights to see which one gives the best result.
Finally, if one would like to use the LINEAR kernel in favour of classification speed but at the cost of some accuracy, it can be seen from the results that the weighted LINEAR kernel outperforms its unweighted counterpart, although the difference is not significant at a confidence level $\alpha = 0.05$. Nevertheless, it can be seen that the number of support vectors is significantly lower than for the unweighted LINEAR kernel and this will result in faster classification, because classification of a new instance happens by comparing it with every support vector in the model through the kernel function $K$.

Next, we look at the experiments to find the optimal left and right context length for each classifier. Then, we look at di- and trinucleotides to find out whether this gave better performance than the single nucleotide case. For these experiments we used the WRBF/GR kernel, this choice was based on the fact that WRBF performs second best for donor sites and best for acceptor sites. Moreover, since IG (information gain), GR (gain ratio) and SV (shared variance) were not significantly different in our experiments we used GR (gain ratio) as the weighting scheme. This follows from the experiments described above. The results are shown in Table 3.

### Table 3. Splice site prediction, for the WRBF/GR kernel, for donor sites and for acceptor sites.

<table>
<thead>
<tr>
<th></th>
<th>donor sites</th>
<th>acceptor sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr. nucleotides</td>
<td>FP95% left context right context</td>
<td>FP95% left context right context</td>
</tr>
<tr>
<td>Single nucleotide</td>
<td>6.54 ± 0.92 60 40</td>
<td>8.79 ± 1.45 40 100</td>
</tr>
<tr>
<td>Dinucleotide</td>
<td>5.52 ± 0.70 60 40</td>
<td>7.46 ± 1.22 80 100</td>
</tr>
<tr>
<td>Trinucleotide</td>
<td>5.87 ± 1.02 60 60</td>
<td>8.46 ± 1.44 80 100</td>
</tr>
</tbody>
</table>

4.4. Related Work

The number of papers on splice site prediction and the related problem of gene finding is enormous and hence it is impossible to give an exhaustive overview. We will give some popular references (according to citeseer) and discuss some recent work.

The problem of recognizing signals in genomic sequences by computer analysis was pioneered by Staden [?] and the recognition of splice sites using neural networks was first addressed by Brunak et al. [?]. They trained a backpropagation feedforward neural network with one layer of hidden units to recognize donor and acceptor sites, respectively. The input consist of a sliding window centered on the nucleotide for which a prediction is to be made. The window is encoded as a numerical vector. The best results were obtained by combining a neural network to recognize the consensus signal at the splice site with another one that predicted coding regions based on the statistical properties of the codon usage and preference. This tool is available online at http://www.cbs.dtu.dk/services/NetGene2/.
Kulp et al. [?] and Reese et al. [?] build upon the work of Brunak by explicitly taking into account the correlations between neighboring nucleotides around a splice site by using dinucleotides as input features instead of single nucleotides. This tool is available online at http://www.fruitfly.org.

Genesplicer [?] uses a combination of a hidden Markov model and a decision tree. They obtained good performance compared to other leading splice site detectors at that time.

Rätsch and Sonnenburg[?] use a SVM with a special kernel to classify nucleotides as either donor or acceptor sites. There is one SVM for donor sites and one for acceptor sites. The system predicts the correct splice form for more than 92% of these genes. This approach is quite similar to ours but the kernel is different.

Finally, a list of online tools for splice site prediction and gene finding is available at http://www.cbs.dtu.dk/biolinks/pserve2.php.

5. Conclusions

In this article it was 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 became clear that the sensitivity information has a positive effect on the results.

So far, this was shown on only one data set because the SVM is computationally very expensive but we have shown that kernel functions that operate on contexts directly gives additional benefits. At the moment, we are running experiments on a number of other data sets to show that the increased performance is not due to bias to the data sets. Apart from that, we are running experiments with more complex features based on the improved design strategy in [?], where a FP95% rate of 2.2% for donor and 2.9% for acceptor sites is obtained. In this light it remains to be seen whether the positive effect of the sensitivity information will still be significant in a system that already performs at very high precision without such information. Finally, we plan to compare our results with the ones obtained by other classifiers on the same data sets.