TIME-RESOLVED SPECTRUM KERNEL FOR BIOSONAR TARGET CLASSIFICATION

M. Beigi
Computer Science Dept.
University of Tübingen
Sand 1, D-72076
Tübingen, Germany
majid.beigi@uni-tuebingen.de

M. Wang and A. Zell
Computer Science Dept.
University of Tübingen
Sand 1, D-72076
Tübingen, Germany
{maosen.wang, andreas.zell}@uni-tuebingen.de

ABSTRACT
We consider the problem of biosonar landmark classification as an example of random and non-stationary signal classification in which finding robust and structure independent features for classification is not trivial. Time-frequency domain studies show that despite the seemingly randomness of those signals, there are local temporal similarities, independent of the position of occurrence in echoes of each object that reflect the intrinsic similarities between the echoes and also a self similarity in the objects. In this paper we suggest a time resolved spectrum kernel for extracting the local similarities (subsequence similarity) in time series in general, and as an example in biosonar signals. We implemented this kernel using dynamic programming and could get accurate results using a low number of echoes needed for training compared with the methods in which finding specific features in each echo were followed.

KEY WORDS
Biosonar signal processing, Pattern recognition, Kernels.

1 Introduction

Bats can distinguish objects and their prey by emitting a series of ultrasound signals (chirps) that generally sweep covering frequencies from 22 to 100 kHz. Auditory scene analysis involves the grouping and segregation of sounds to perceptually classify information about auditory objects. The perception of sound is influenced by the spectral and temporal characteristics of acoustic signals [1]. When bats navigate in a natural habitat, the landmarks available to them are trees. To understand how bats perceive the objects we need to know the encoding mechanism of received echoes and the distribution of information in the time and frequency domain. Inspired by the bat biosonar system, researchers have utilized ultrasonic sensing techniques for mobile robots (biomimetic robots) and tried to classify different textures and landmarks using received echo signals.

McKerrow used a CTFM (Continuous Transmission Frequency Modulated) system and modelled the echoes with the acoustic density profiles, used the frequency components and energy spectra and found features, which characterize the acoustic density profiles of plants in a classification task [2]. Kuc [3] suggested a transformation of echo to pseudo-action potential as temporal point process to understand how bats recognize landmarks in the field. Müller [4] presented a neuro-spike representation of echoes in which each echo is transcribed into a spike code using a parsimonious model, and classified four foliages using three features derived from interspike intervals. Gao et. al [5] presented a template matching algorithm for classification of several types of brick walls, picket fences and hedges using sonar echoes. M. Wang et al. [6, 7, 8] used different structural features in the frequency domain and also template matching for the classification task.

In summary, we can conclude from the study of the above references and our experiments that the robustness of features for the classification task depends heavily on the experimental setup. For example, the orientation of plants due to windy weather or sudden rain can result in large changes in the reflected echoes. Hence, in this case the only temporal based features can be inefficient. But on the other hand, the local temporal similarities between different echoes of one object as an indication of its texture is a significant issue that should be considered. In this paper we explain our experimental setup in which we have considered the robustness and efficiency of features aiming at practical application in biosonar based robot navigation. We show that the efficient method for our classification task should consider both local temporal similarity and the power spectrum of echoes and suggest a kernel based classification method considering those parameters.

2 Problem

We used a sonar head system consisting of three ultrasound transducers, one for emission chirp signals (Polaroid 7000) and two for reception (Polaroid 6000). The emitted pulse was a linearly frequency modulated chirp sweeping from 20kHz to 120kHz in 1 ms (Fig.1). We used three artificial trees with a similar height but different density and size of leaves as shown in Fig. 2.

Compared with the method of other researchers for sensing the object [4, 3], we used a different method. We used a 0.5 degree angular stepsize for our scans, each

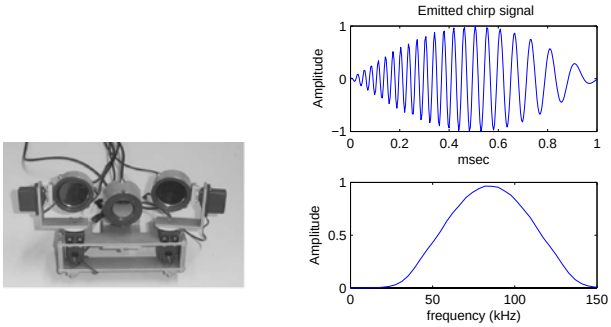

Figure 1. Biosonar head (left). Emitted chirp signal and its frequency content (right).

Figure 2. Three different trees as biosonar landmarks. From left to right: Ficus, Bamboo, Schefflera.

tree was scanned 360 degrees in a circular movement of the robot and we collected echoes from all orientations of leaves and tree. The reflected echo contains the information about the geometry of the tree and is the superposition of all reflections. Fig. 3 shows the block diagram of the data acquisition and preprocessing procedure of reflected echoes. We passed the reflected echoes through a bank of 10 gammatone filters between 20 kHz and 120 kHz. In order to extract the envelope of the filtered signals, they were delivered to half-wave rectifiers.

The next step is *frame blocking*. In this step the signal is blocked to frames of N samples and is separated

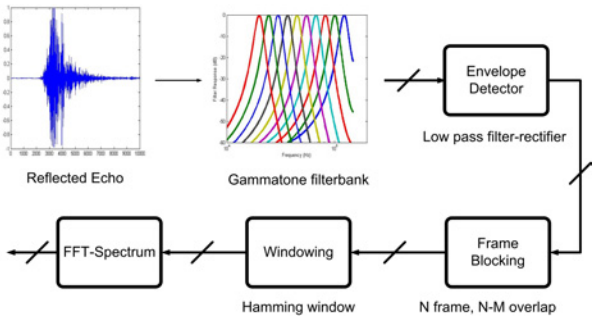

Figure 3. Block diagram of the preprocessing steps for reflected echoes.

from adjacent frames by M $(M < N)$ samples and has $N - M$ overlaps. Considering the sampling frequency of the data acquisition part (1 MHz) and the minimum width of leaves of trees and axial resolution of transducers, we selected $N = 32$ and 50% overlap for frames. The next step in the data preprocessing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. We used a *Hamming window* for this purpose. The last step is to calculate the average energy of each band of gammatone filter bank in each frame. The result is a feature matrix where each column is a vector showing the average energy of each channel in one time frame. Fig. 4 shows the examples of the preprocessed echoes of Ficus and Schefflera trees. We use those feature matrix for our classification task.

But as noted before, the problem is that the biosonar signals are random and nonstationary in the temporal dimension. For example, the location of leaves in the plant determines the acoustic energy throughout the frames and small changes in the orientation of the plant result in changes in those features along the frames of time. But, as we see in Fig. 4, despite the seemingly randomness of those signals there are some local similarities (shown by p) in echoes from one tree. Then, if we can find the sizes of windows in which we have maximum similarity between data of one object it can help us to classify that object from others. We consider the output of the block diagram shown in Fig. 3, a time series in which each point is a time frame and its value is a vector of features (the average energy of each channel of gammatone filter bank). We should find the subsequences of the time series *independent of the positions of occurrences* that have maximum similarities in echoes of each object. The intuition behind our idea is that the structure of objects and, as an example, the size of leaves or branches, should be considered in the classification task. The size of the subsequence that we are looking for, can be related to the size of the leaves or branches of the tree. In another way, the energy reflected by the leaves or branches of the tree can be related to the size of those similar subsequences of the time series.

Kernel methods have widely been used for string classification. Examples of those kernels for text classification and remote homology detection in protein families include the spectrum kernel [9], mismatch kernel [10], and the string kernel proposed by Lodhi et al. [11]. Similar to the remote homology detection in proteins, where a classifier must detect a remote relation between an unknown sequence and a family of proteins, in our classification task, the algorithms for finding similar time series should not consider the whole time series but look for informative subsequences, and we need kernels, which can extract similarities between subsequences. Inspired by the work of Lodhi et al. [11], we use a time-resolved spectrum kernel for our classification task. We implement the spectrum kernel algorithm for time series and use a fast algorithm to calculate that kernel. In the next section we will discuss the method and the constraints which should be considered.

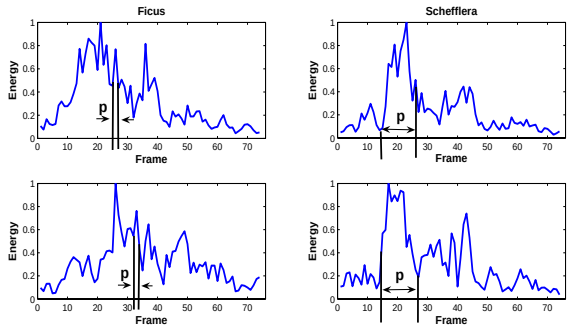

Figure 4. The energy spectrum in each time frame for Ficus and Schefflera trees (output of gammatone filter centered around 50 kHz). The time-resolved spectrum kernel tries to find the local similarities in window of size p in echoes of one object.

3 Time-resolved spectrum kernel

A kernel function can often be considered as a measure of similarity. Different kernels correspond to different notations of similarity. The structure of the data and our knowledge of the particular time series suggest a way of comparison that we can consider in our kernel function. The use of a kernel makes it possible to perform the mapping into that feature space and to calculate the inner product between those maps. But the main task here is to find a map ϕ that reflects the suitable and common features of those time series and gives a good indication of the similarity we would like to capture. The time-resolved spectrum kernel simply measures the whole similarities of all subsequences of the time series in consideration, independent of their positions. The more two time series share similar subsequences, the more similar they are.

3.1 Definitions

Definition 1. A time sequence $s = s_1...s_n$ is a sequence of data points at successive times $(s_1, s_2, ..., s_n)$ with $s_i \in \Re^d$, $1 \leq i \leq n$. Concerning the notation, we denote:
$|s|$ the length of s and $s(i-p+1:i)$ the p-length subsequence of s from position $i-p+1$ to position i.
Definition 2. We denote $\mathbf{I}_p^{|s|}$ the set of indices, defining all the p-long *contiguous* subsequence of s:

$$\mathbf{I}_p^s = \{\mathbf{i} : \mathbf{i} \in \mathrm{N}^p, 1 \leq i_1 < ... < i_p \leq |s|\}$$

and $s_{\mathbf{i}}$ is a subsequence of s in positions given by $\mathbf{i} = (i_1, i_2, ..., i_p)$. For $u \in \Sigma^{p\times d}$, the infinite set of all subsequences with size p and dimension d, the implicit embedding map ϕ brings s to vector space F, $\phi : s \rightarrow (\phi_u(s)) \in F$. The u component of our feature vector is defined as:

$$\phi_u^p(s) = \sum_{\mathbf{i}\in\mathbf{I}_p^s,\ u\in\Sigma^{p\times d}} \varphi_u(s_{\mathbf{i}})$$

where φ is an implicit map that satisfies:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) =< \varphi_u(s_{\mathbf{i}}),\ \varphi_u(t_{\mathbf{j}}) >\ ,\ \text{for}\quad \mathbf{i} \in \mathbf{I}_p^s,\ \mathbf{j} \in \mathbf{I}_p^t$$

in which κ_p is a kernel function that measures the local similarity between two p-length contiguous subsequences $s_{\mathbf{i}}$ and $t_{\mathbf{j}}$ of the time series in consideration. In words, $\phi_u^p(s)$ is a sum over all similarities between p-long subsequences of s and u. The dot product of those feature vectors represents the time resolved p-spectrum kernel (spectrum kernel with subsequence size of p):

$$\begin{aligned}\mathcal{K}_p(s,t) = \langle\phi_u^p(s), \phi_u^p(t)\rangle &= \textstyle\int_{\mathbf{R}^{d\times p}} \phi_u^p(s)\phi_u^p(t)du \\ &= \sum_{\mathbf{i}\in\mathbf{I}_p^s}\sum_{\mathbf{j}\in\mathbf{I}_p^t} \textstyle\int_{R^{d\times p}} \varphi_u(s_{\mathbf{i}})\varphi_u(t_{\mathbf{j}})du \\ &= \sum_{\mathbf{i}\in\mathbf{I}_p^s}\sum_{\mathbf{j}\in\mathbf{I}_p^t} \kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})\end{aligned}$$

Considering the definitions of $\mathbf{I}_p^s$ and $\mathbf{I}_p^t$, we can say:

$$\mathcal{K}_p(s,t) = \sum_{i=p}^{|s|}\sum_{j=p}^{|t|} \kappa(s(i-p+1:i), t(j-p+1:j)) \quad (1)$$

The evaluation of κ_p requires $O(p)$ computations, and the cost for computation of $\mathcal{K}_p(s,t)$ is of order $O(p|s||t|)$. In string p-spectrum kernels, a very fast method for computation of $\mathcal{K}_p(s,t)$ is to use an efficient data structure known as 'trie' (retrieval tree) in which we build a suffix tree for the collection of p-length subsequences of s and t, obtained by moving a p-length sliding window across each of s and t, and then calculate the kernel by traversing the tree. But because of an infinite subsequence set, that method is not applicable for the time series spectrum kernel unless the time series is quantized, symbolized and converted to a string. In this case we are faced with the quantization errors and the method for quantization and symbolization can affect the efficiency of the kernel method. Instead of that we use dynamic programming to calculate the time-resolved spectrum kernel while accepting some constraints. We accept a constraint on choosing the kernel function $\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})$, we suppose:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) = \prod_{i=1}^{p} \kappa^*(s_{\mathbf{i}_i}, t_{\mathbf{j}_i}) \quad (2)$$

in which κ^* is an arbitrary function (discussed later) that measures the similarity between two data points. Considering Equations 1 and 2, we define an auxiliary kernel, p-suffix kernel $\mathcal{K}_p^S(s',t')$ as:

$$\mathcal{K}_p^S(s',t') = \begin{cases}\kappa_p(s'(|s'|-p+1:|s'|), t'(|t'|-p+1:|t'|)) & \text{if } \min(|s'|,|t'|)\geq p \\ 0 \quad \text{otherwise.}\end{cases}$$

$$= \begin{cases}\prod_{i=0}^{p-1} \kappa^*(s'_{|s'|-i}, t'_{|t'|-i}) & \text{if } \min(|s'|,|t'|)\geq p \\ 0 \quad \text{otherwise.}\end{cases}.$$

where $s' = s(1:|s'|)$, $t' = t(1:|t'|)$, $1 \leq |s'| \leq |s|$ and $1 \leq |t'| \leq |t|$. Then we express the p-spectrum kernel in

term of its suffix version as:

$$\mathcal{K}_p(s',t') = \sum_{i=1}^{|s'|}\sum_{j=1}^{|t'|}\mathcal{K}_p^S(s'(1:i),t'(1:j))$$

If we add a new data point x to the time series s', using the above equation we can calculate $\mathcal{K}_p(s'x,t')$:

$$\mathcal{K}_p(s'x,t') = \sum_{i=1}^{|s'x|}\sum_{j=1}^{|t'|}\mathcal{K}_p^S(s'x(1:i),t'(1:j))$$

$$= \sum_{i=1}^{|s'|}\sum_{j=1}^{|t'|}\mathcal{K}_p^S(s'(1:i),t'(1:j)) + \sum_{j=1}^{|t'|}\mathcal{K}_p^S(s'x,t'(1:j))$$

$$= \mathcal{K}_p(s',t') + \sum_{j=1}^{|t'|}\mathcal{K}_p^S(s'x,t'(1:j))$$

On the other hand, if we add another new data point y to the time series t', considering equation 2 and the above definition of $\mathcal{K}_p^S$, we can say:

$$\mathcal{K}_p^S(s'x,t'y) = \kappa^*(x,y)\mathcal{K}_{p-1}^S(s',t')$$

It is clear that: $\mathcal{K}_p(s,t) = \mathcal{K}_p(s',t')$ if $s = s', t = t'$. Now, we define a recursive computation for $\mathcal{K}_p$:

Definition 3: *Recursive computation of the time resolved spectrum kernel.*

$$\mathcal{K}_p(s'x,t') = \mathcal{K}_p(s',t') + \sum_{k=1}^{|t'|}\mathcal{K}_p^S(s'x,t'(1:k)) \quad (3)$$

$$\mathcal{K}_p^S(s'x,t'(1:k)) = \kappa^*(x,t'_k)\mathcal{K}_{p-1}^S(s',t'(1:k-1)) \quad (4)$$

$$\begin{aligned}\mathcal{K}_0^S(s',t') &= 1 \quad for \quad all \quad s',t',\\ \mathcal{K}_i^S(s',t') &= 0, \quad if \ \min(|s'|,|t'|) < i,\\ \mathcal{K}_i(s',t') &= 0, \quad if \ \min(|s'|,|t'|) < i,\end{aligned}$$

The computation of the kernel follows a dynamic programming technique. We have recursions over the prefixes of the time series and the lengths of the subsequences and we do the routine above until $x = s_{|s|}$ and $|t'| = |t|$.

To prevent that with larger sizes of subsequences the kernel achieves a higher similarity score we normalize the kernel: $\mathcal{K}_i^{norm}(s,t) = \frac{\mathcal{K}_i(s,t)}{\sqrt{\mathcal{K}_i(s,s)\mathcal{K}_i(t,t)}}$

This operation scales the similarities in the range [0,1]. In practice and specially in our classification task, it makes sense to consider the similarity of subsequences having different sizes and calculate a linear combination of different i-spectrum kernels with different weighting $\alpha_i \geq 0$. The weighted kernel is:

$$K(s,t) = \sum_{i=1}^{l}\alpha_i\mathcal{K}_i^{norm}(s,t) \quad (5)$$

As we see from the above pseudo-code, the evaluation of the $\mathcal{K}_i^{norm}$ is of order $O(|s||t|)$ and the overall complexity

Algorithm Time resolved spectrum kernel

Input : Time series s and t of length n and m, max subsequence length l;
Output: Array of spectrum kernel $\mathcal{K}[]$ with different sizes of subsequence-length from 1 to l);

```
KPS(0:n, 0:m, 0) = 1;        (* KPS(i,j,p) stores K_p^S(s(1:i), t(1:j)) *)
KP(0:n, 0:m) = 0;            (* KP(i,j) stores K_p(s(1:i), t(1:j)) *)
for p ← 1 to l do
    KPS(0:n, 0, p) = 0;
    KPS(0, 0:m, p) = 0;
    for i ← 1 to n do
        P(0)=0;  (* P(k) stores the second term on the right side in Eq. 3 *)
        for k ← 1 to m do
            KPS(i,k,p) = κ*(s_i, t_k) KPS(i-1, k-1, p-1);
            P(k) = P(k-1) + KPS(i,k,p);
            KP(i,k) = KP(i-1,k) + P(k);
        end
    end
    K[p] = KP(n,m);          (* K_p(s,t) *)
end
return K[]
```

of our algorithm to calculate a linear combination of all p-spectrum kernels is $O(p|s||t|)$ while if Equation 1 is used the complexity is of order $O(p^2|s||t|)$.

We considered $\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})$ (Equation 2) as a multiple of similarities between data points. Different choices of that function allow different methods of comparing sub similarities. As a suitable selection we consider:

$$\kappa^*(s_{\mathbf{i}_i}, t_{\mathbf{j}_i}) = \exp\frac{-(s_{\mathbf{i}_i} - t_{\mathbf{j}_i})^2}{2\sigma^2}$$

to measure the similarity between two data points, then:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) = \prod_{i=1}^{p}\kappa^*(s_{\mathbf{i}_i}, t_{\mathbf{j}_i}) = \exp\left(-\frac{||s_{\mathbf{i}} - t_{\mathbf{j}}||^2}{2\sigma^2}\right) \quad (6)$$

$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})$ is the gaussian kernel of width σ and suitable for measuring the local similarity of subsequences in the time series.

4 Classification method and results

We gathered the sonar data,720 echoes for each tree, as explained in section 2. After the preprocessing steps for each echo (Fig. 3), we have a time series in which each point is a time frame and its value is an array of features (the average energy of each channel of gammatone filter). Using the kernel method told in the previous section, we want to extract the similarities between the echoes for our

$KPS(:,:,1)=\mathcal{K}_1^S$	ϵ	t_1	t_2	t_3
ϵ	0	0	0	0
s_1	0	$k^*(s_1,t_1)$	$k^*(s_1,t_2)$	$k^*(s_1,t_3)$
s_2	0	$k^*(s_2,t_1)$	$k^*(s_2,t_2)$	$k^*(s_2,t_3)$
s_3	0	$k^*(s_3,t_1)$	$k^*(s_3,t_2)$	$k^*(s_3,t_3)$

$\mathcal{K}_2^S$	ϵ	t_1	t_2	t_3
ϵ	0	0	0	0
s_1	0	0	0	0
s_2	0	0	$k^*(s_2,t_2)k^*(s_1,t_1)$	$k^*(s_2,t_3)k^*(s_1,t_2)$
s_3	0	0	$k^*(s_3,t_2)k^*(s_2,t_1)$	$k^*(s_3,t_3)k^*(s_2,t_2)$

$\mathcal{K}_3^S$	ϵ	t_1	t_2	t_3
ϵ	0	0	0	0
s_1	0	0	0	0
s_2	0	0	0	0
s_3	0	0	0	$k^*(s_3,t_3)k^*(s_2,t_2)k^*(s_1,t_1)$

Table 1. Calculation of $\mathcal{K}_p^S$ using $\mathcal{K}_{p-1}^S$ for $s = s_1s_2s_3$ and $t = t_1t_2t_3$. $\mathcal{K}_p(s,t) = \sum_{i=1}^{3}\sum_{j=1}^{3}\mathcal{K}_p^S(s(1:i)t(1:i))$ and $\mathcal{K}_0^S = 1$.

classification task. According to Equation 5 and 6, we need to find parameters α_i, and σ. For simplicity, we considered equal values of α in the range $[p_1, p_2]$ as follows:

$$\alpha_i = \begin{cases} 1 & p_1 \le i \le p_2 \\ 0 & \text{otherwise} \end{cases}$$

p_1 and p_2 are the minimum and maximum sizes of subsequences used to extract the similarities in each tree. To find suitable values for those parameters, we used a simple *grid search* on p_1 and p_2. We selected randomly 100 echoes of each tree and then calculated $\mathcal{K}_i^{norm}(s[m], s[n])$ for $i \in [1, l]$, $m, n \in [1, 100]$ and $\sigma \in \{1, 10, 100, 1000\}$ where $s[m]$ and $s[n]$ are the m-th and n-th of pre-processed echoes and l is the length of the time series (in our experiment 90). Then we found the optimum values p_1 and p_2 in the range [1,l] for each σ, by maximizing the average value of the kernels:

$$\max_{p_1,p_2} K = \sum_{m=1}^{100}\sum_{n=1}^{100}\left(\frac{\sum_{i=p_1}^{p_2}\mathcal{K}_i^{norm}(s[m],s[n])}{p_2-p_1}\right)$$

we found that a suitable value for σ is in the range [10,100] for all trees. Table 2 shows the optimum values for p_1 and p_2 with $\sigma = 10$. We see that for ficus and Bamboo, which have smaller leaves the p_1 has lower value. Again, for simplicity, we considered equal values of p_1 and p_2 for all trees in our classification method.

Table 2. Optimum Values for p_1 and p_2.

tree	p_1	p_2
Ficuss	5	25
Bamboo	8	23
Schefflera	11	32

For the classification task, we used a discriminative method. Given a set of positive training data χ_+ and a set of negative data χ_-, an SVM learns a classification function $f(x)$ of the form:

$$f(x) = \sum_{i;x_i\in\chi_+}\lambda_i K(x,x_i) - \sum_{i;x_i\in\chi_-}\lambda_i K(x,x_i) \quad (7)$$

where non-negative λ_i weights are computed during training by maximizing a quadratic objective function and $K(.,.)$ is the kernel . Given this function, a new data x is predicted to belong to the positive dataset, if the value of $f(x)$ is positive, otherwise it belongs to the negative dataset.

To measure the robustness of our algorithm, we randomly selected 100 echoes of each tree (total 300 echoes) to train the classifier. Considering the same σ, p_1 and p_2 ($\sigma = 10$, p_1=5, p_2=30) for all trees, we calculated the kernel matrix $\mathbf{K}$:

$$\mathbf{K}(i,j) = K(s[i],s[j]) = \textstyle\sum_{l=p_1}^{p_2}\mathcal{K}_l^{norm}(s[i],s[j])$$

in which i, $j \in [1, 300]$ and $s[i]$ is i-th echo, where for Ficus echoes, $i \in$[1,100], for Bamboo $i \in$ [101,200] and for Schefflera $i \in$[201,300].

After calculation of the kernel matrix $\mathbf{K}$, we used the LIBSVM package for Support Vector Machines (SVMs) regression and classification [12]. It lets us use our own kernel matrix to train the classifier. We used the remaining data (1860 echoes) for test. Considering x as a test example, $K(x, x_i)$ is the kernel score of x to the training data x_i and $f(x)$ is the result of the classifier (Equation 9).

Table 3. Performance of our method in biosonar landmarks classification with 100 randomly selected echoes for training.

tree	Specificity (%)	Sensitivity (%)	Accuracy (%)
Ficus	85.2	87.5	86.3
Bamboo	87.1	90.1	89.3
Schefflera	92.6	93.4	93.8

Table 3 shows the average performance of the classifier. It should be noted that the classifier decides based on only one observation. If we use more observations and decide based on the average of the probability that an observation belongs to a class ($f(x)$ value in Equation 9) the accuracy increases. With this approach and using only 10 random observations, we could increase the accuracy of the classifier for those trees to nearly 98%. (Fig. 5.a).

Comparing with our previous works, it shows a notable improvement in accuracy. In the previous work of Wang et al. [6], the best result for classification was gained through template matching in 2D acoustic images from biosonar echo (using a 2D Discrete Cosine Transform). The classification was made via extracting the maximum normalized cross correlation between the acoustic templates (Fig. 5.b). As shown in Fig. 5, we could get higher accuracy in both single observation and repeated observations (see for example 10 observations).

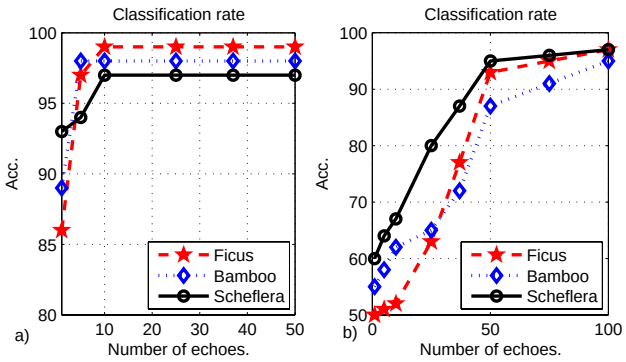

Figure 5. The accuracy of classifiers using different number of echoes for test. **a**). Time-resolved spectrum kernel method. **b**). Template matching using acoustic images of echoes (Wang et al. [6]).

To compare with other methods, we implemented several other methods for the feature matching and classification of biosonar data, like Dynamic Time Warping and Hidden Markov models. The best accuracy of those methods was less than 70%. The key point in our method is that it looks for patterns and similarities in the subsequences without dependency on the order of those subsequences while other methods look for them across the whole time series.

5 Conclusions and Further works

We considered the problem of biosonar landmark classification as an example of random and non stationary signal classification in which finding robust features for classification is not trivial. We regarded both the local temporal similarity and the power spectrum of echoes and suggested a kernel based classification method that extracts those local similarities, independent of the position of occurrences in echoes of each object. We proposed the time-resolved spectrum kernel and made a relation between those kernels and geometric specification of the objects. Our results provide evidence that this kind of kernel can be used for pattern extraction and classification in random signals. We think this kind of kernel is also suitable for pattern recognition in signals with inherent self similarity and for estimating periodicity in arbitrary time series like speech and biomedical signals. In our method, to keep the problem simple, we made a not very accurate assumption for the α_i parameters of the kernel with an equal value for all α_i. But that parameter, the weight of the similarity (kernel score) of the p-size subsequences, can represent the self similarity of one part of the object, for example the size of leaves, and also can show the geometric characteristics of that object. In future, we will try to find the optimum value of those parameters while maximizing the accuracy of the classifier using optimization algorithms. On the other hand, one may suggest a faster algorithm to calculate the time-resolved kernel or even propose better kernels for our classification task. For further work we also hope to implement the kernels that consider warping in the subsequences of time series and use them for our classification task.

References

[1] C. F. Moss and A. Surlykke. Auditory scene analysis by echolocation in bats. *J Acoust. Soc. Am.*, 110(4):2207–26, 2001.

[2] P. McKerrow and N. Harper. Plant acoustic density profile model of ctfm ultrasonic sensing. *IEEE Sensors Journal.*, 1(4):245–255, 2001.

[3] R. Kuc. Transforming echoes into pseudo-action potentials for classifying plants. *J. Acoust. Soc. AM.*, 110(4):2198–2206, 2001.

[4] R. Muller. A computational theory for the classification of natural biosonar targets based on a spike code. *Network: Computat. Neural Syst.*, 14:595–612, 2003.

[5] W. Gao and Mark Hinders. Mobile robot sonar backscatter algorithm for automatically distinguishing walls, fences, and hedges. *The International Journal of Robotics Research.*, 25(2):135–145, 2006.

[6] M. Wang. *Natural Landmark Classification with a Biosonar based Mobile Robot*. PhD thesis, University of Tuebingen, 2006.

[7] M. Wang and A. Zell. Classification of natural landmark with biosonar, journal of the acoustic society of america (jasa). *Journal of the Acoustic Society of America (JASA)*, 116:2640, 2004.

[8] M. Wang and A. Zell. Sequential sensing with biosonar for natural landmark classification. *IEEE International Workshop on safty, security and rescue robotics (SSRR 2005)*, pages 137–142, 2005.

[9] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, page 564575, 2002.

[10] C. Leslie, E. Eskin, A. Cohen, J. Weston, and W.S. Noble. Mismatch string kernel for svm protein classification. *Advances in Neural Information Processing System*, pages 1441–1448, 2003.

[11] H. Lodhi, C. Saunders J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Jounal of Machine Learning Research*, pages 419–444, 2002.

[12] C.C Chang and C.J Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.