A novel kernel-based method for local pattern extraction in random process signals

Majid M. Beigi[*], Andreas Zell

Computer Science Dept. - University of Tübingen
Sand 1, D-72076 - Tübingen, Germany

Abstract

In many applications, one is interested to detect certain patterns in random process signals. We consider a class of random process signals which contain sub similarities at random positions representing the texture of an object. Those repetitive parts may occur in speech, musical pieces and sonar signals. We suggest a warped time resolved spectrum kernel for extracting the subsequence similarity in time series in general, and as an example in biosonar signals. Having a set of those kernels for similarity extraction in different size of subsequences, we propose a new method to find an optimal linear combination of those kernels. We formulate the optimal kernel selection via maximizing the Kernel Fisher Discriminant criterion (KFD) and use Mesh Adaptive Direct Search (MADS) method to solve the optimization problem. Our method is used for biosonar landmark classification with promising results.

Key words: Time-resolved spectrum kernel, SVM, Fisher discriminant, Mesh adaptive direct search.

1 Introduction

Time series are an important type of data occurring in many scientific disciplines. A common task with time series is to compare one sequence with another. In some domains, a very simple distance method measure, such as Euclidian distance will suffice. In the case that two time series have similar parts but not at similar positions, we use a more efficient method for similarity extraction known as Dynamic Time Warping (DTW), in which the time of one (or both) sequences is "warped" before an alignment. Dynamic programming

[*] Corresponding author. Tel:+49 7071 2978983; fax:+49 7071 295091.
Email Address: majid.beigi@uni-tuebingen.de

is used to measure the similarity score in DTW.
Kernel methods have also been used as a popular method to extract the similarity. Different kernels correspond to different notions of similarity. A kernel function implicitly defines a feature space that in many cases we do not need to construct explicitly. 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. Then, the kernel function can be used directly in Support Vector Machines (SVMs) based classifiers.

There have been methods proposed to embed the time alignment operation and DTW into a kernel function [1,2]. These methods especially are useful in speech recognition, in which the information lies in the whole time series.

However, in some time series the information lies in a fixed (or not very varied) size window of time events (subsequence), independent of the actual time. So we have subsequences at random positions whose similarities should be measured. Those repetitive parts may occur in speech, musical pieces and sonar signals. Therefore, the algorithms for finding similar time series should not consider the whole time series but look for informative subsequences. Then, in kernel based methods for similarity extraction, we need kernels, which can extract similarities between all subsequences. Hence, the main task is to find a map that reflects the suitable and common features of those time series and gives a good indiction of the sub similarity, we would like to capture. On the other hand, we should be able to calculate those inner products efficiently.

A similar situation happens in text classification and also remote homology detection in protein families, where we must detect a remote relation between unknown sequence and a family of proteins. Those proteins contain domains whose positions are not similar in proteins of a family. There again we should measure the local similarities between all subsequences as an indication of similarity between two sequences.

Proposed kernels for text classification and remote homology detection in protein families include the Fisher kernel [6], spectrum kernel [3], mismatch kernel [4], and the string kernel proposed by Lodhi et al. [5]. A dynamic programming technique makes the computation of those kernel very efficient.

Inspired by the solutions for remote homology detection in protein families and the string kernel proposed by Lodhi et al. [5], in our previous work [7] we suggested a similar kernel called *time-resolved spectrum kernel* to measure the similarity of two time series. The p-length subsequence of that kernel simply measures the occurrences of fixed p-length subsequences for each of the time series in consideration. The more time series share similar p-length subsequences, the more similar they are.

In this paper, we implement a more general kernel called *warped time-resolved*

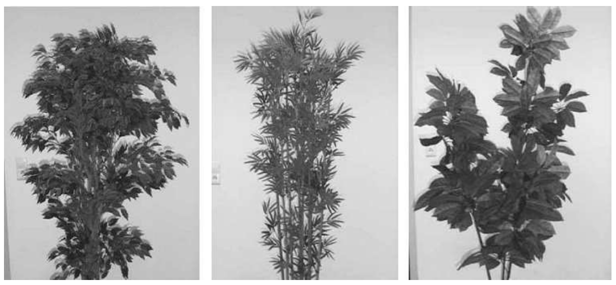


Fig. 1. Three different trees as biosonar landmarks. From left to right: Ficus, Bamboo, Schefflera.

spectrum kernel, which considers warping in the subsequences. The warped time-resolved spectrum kernel measures the whole similarities of all warped non contiguous subsequences of the two time series, independent of their positions.

Having a set of those kernels for different size of subsequences, we find the optimal kernel selection via maximizing the Kernel Fisher Discriminant criterion (KFD) [8] to build the optimal linear combination of kernels. This criterion ascertains that the obtained kernel maximizes the similarity score between signals of one class and minimizes the similarity score between signals of two different classes. Given that criterion, to solve the optimization problem we use a Mesh Adaptive Direct Search (MADS) method. MADS as defined by Audit and Dennis [9] is a class of algorithms for nonlinear optimization. It computes a series of points that get closer and closer to the optimal point. The algorithm searches a set of randomly selected points, called a mesh, around the current point-the point computed at the previous step of the algorithm. The mesh is formed by adding the current point to a scalar multiple of a set of vectors called a pattern and the point in the mesh that improves the objective function becomes the current point at the next step. The routine continues until a stopping criterion is fulfilled.

In this paper, we study the classification of biosonar signals as an example of the random process signals which contain those local similarities. Bats can distinguish objects by emitting a series of ultrasound signals (chirps) that generally sweep covering frequencies from 22 to 100 kHz [10]. 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 [11,12].

Sonar signals reflected by different trees and flowers as landmarks (Fig. 1) contain the information of their textures. Those textures represent the geometric specifications like the size of leaves or branches. They are random and nonstationary in the temporal dimension and small changes in the orientation of the plant result in changes in the position of energy features along the time series.

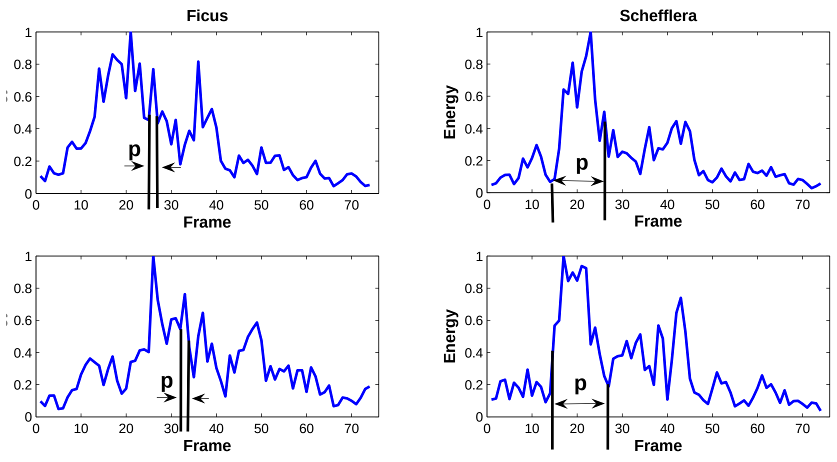


Fig. 2. The time-resolved spectrum kernel tries to find the local similarities in windows of size p in echoes of one object.

However, as we see in Fig. 2, despite the randomness of those signals, there are some local similarities (shown by p) in echoes from one tree. But their positions are random. Then, we should find the size or sizes of subsequences of the time series *independent of the positions of occurrences* that have maximum similarities in echoes of each object. We use our suggested kernel to extract the similarities between those reflected echoes.

This paper is organized as follows: In Sec. 2 we describe the warped time resolved spectrum kernel. In Sec. 3 we describe our method for optimal kernel selection. We report experimental results in Sec.4 and give conclusions in Sec. 5.

2 Warped time resolved spectrum kernel

A kernel function can often be considered as a measure of similarity. Different kernels correspond to different notions of similarity. 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. However, the main task is to find a map ϕ that reflects the suitable and common features of data and gives a good indication of the similarity we would like to capture. The warped time-resolved spectrum kernel measures the whole similarities of all warped non contiguous subsequences of the two time series, independent of their positions. The more two time series share similar subsequences, the more similar they are. We use dynamic programming for efficient calculation of that kernel.

A time sequence $s = s_1 ... s_n$ is a sequence of data points at successive times with $s_i \in \mathbb{R}^d$, where $1 \leq i \leq n$ and d is the dimension of data points. We denote $|s|$ the length of s, $s(i-p+1:i)$ the p-length subsequence of s from position $i-p+1$ to position i. We consider $\mathbf{I}_p^{|s|}$ the set of indices defining all the p-long (both *contiguous* and *non-contiguous*) subsequences of s: $\mathbf{I}_p^s = \{\mathbf{i} : \mathbf{i} \in \mathrm{N}^p, 1 \leq i_1 < ... < i_p \leq |s|\}$ and $u = s_{\mathbf{i}}$ as a subsequence of s in positions given by $\mathbf{i} = (i_1, ..., i_{|u|})$. The number of gaps in the subsequence is $g_{\mathbf{i}} = (i_{|u|} - i_1 + 1) - |\mathbf{i}|$. For example, if we consider $s = s_1s_2s_3s_4s_5$, $u = s_1s_3s_5$ is a subsequence of s in the positions $\mathbf{i} = (1, 3, 5)$ of length $|\mathbf{i}| = 3$ and $g_{\mathbf{i}} = 2$.

In p-length warped time resolved spectrum kernel, we add the similarities of all (possibly warped) p-length subsequences of times series s and t.

For $u \in \Sigma^{p\times d}$, the set of all subsequences with size p and dimension d, the implicit embedding map ϕ brings s to a vector space F ($\phi : s \rightarrow (\phi_u(s)) \in F$) and the u component of our feature vector is:

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

where $\gamma \in (0, 1)$ is a decay factor as a cost for warping (non-contiguousity) in the time series and φ is an implicit map that satisfies:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) =< \varphi_u(s_{\mathbf{i}}),\ \varphi_u(t_{\mathbf{j}}) > \qquad \mathbf{i} \in \mathbf{I}_p^s,\ \mathbf{j} \in \mathbf{I}_p^t,\ u \in \Sigma^{p\times d} \tag{1}$$

in which κ_p is a kernel function that measures the local similarity between two p-length 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 of s and t represents the *warped time resolved p-spectrum kernel*:

$$\mathcal{K}_p(s,t) = \langle \phi_u^p(s), \phi_u^p(t) \rangle = \textstyle\int_{\mathbf{R}^{d\times p}} \phi_u^p(s)\phi_u^p(t)du$$

$$= \sum_{\mathbf{i}\in\mathbf{I}_p^s} \sum_{\mathbf{j}\in\mathbf{I}_p^t} \gamma^{g_{\mathbf{i}}}\gamma^{g_{\mathbf{j}}} \textstyle\int_{R^{d\times p}} \varphi_u(s_{\mathbf{i}})\varphi_u(t_{\mathbf{j}})du$$

Regarding the above kernel definition for local similarity(Eq. 1), we conclude:

$$\mathcal{K}_p(s,t) = \sum_{\mathbf{i}\in\mathbf{I}_p^s} \sum_{\mathbf{j}\in\mathbf{I}_p^t} \kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})\gamma^{g_{\mathbf{i}}+g_{\mathbf{j}}} \tag{2}$$

As we see from the above equation, the kernel adds all similarity scores between subsequences, considering all possible degrees of warping. Needless to say, the calculation of that kernel has a very high computational cost. We use dynamic programming to calculate it in an efficient manner and justifiable time.

Considering the definitions of $\mathbf{I}_p^s$ and $\mathbf{I}_p^t$, we can rebuild the Eq. 2:

$$\mathcal{K}_p(s,t) = \sum_{i=1}^{|s|} \sum_{j=1}^{|t|} \sum_{(\mathbf{i},\mathbf{j}) \in \mathbf{I}_p^{s(1:i)} \times I_p^{t(1:j)}} \kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) \gamma^{g_{\mathbf{i}} + g_{\mathbf{j}}}$$

To express the kernel using a suffix version of that, we define the suffix kernel as:

$$\mathcal{K}_p^S(s(1:i), t(1:j)) = \sum_{(\mathbf{i},\mathbf{j}) \in \mathbf{I}_p^{s(1:i)} \times \mathbf{I}_p^{t(1:j)}} \kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) \gamma^{g_{\mathbf{i}} + g_{\mathbf{j}}} \tag{3}$$

So we have:

$$\mathcal{K}_p(s,t) = \sum_{i=1}^{|s|} \sum_{j=1}^{|t|} \mathcal{K}_p^S(s(1:i), t(1:j) \tag{4}$$

We consider $s' = s(1:|s'|)$, $t' = t(1:|t'|)$, $1 \leq |s'| \leq |s|$ and $1 \leq |t'| \leq |t|$ as prefixes of s and t. If we add a new data point x to the time series s', using the above equation we can calculate $\mathcal{K}_p(s'x, t')$:

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

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

Then,

$$\mathcal{K}_p(s'x, t') = \mathcal{K}_p(s', t') + \sum_{j=1}^{|t'|} \mathcal{K}_p^S(s'x, t'(1:j)) \tag{5}$$

We accept a constraint on choosing the kernel function $\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})$ (Equation 1), we suppose:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) = \prod_{i=1}^{p} \kappa^*(s_{\mathbf{i}_i}, t_{\mathbf{j}_i}) \tag{6}$$

in which κ^* is an arbitrary function that measures the similarity between two data points of the time series. In this study, as a suitable and arbitrary 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:

$$\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}}) = \prod_{i=1}^{p} \kappa^*(s_{\mathbf{i}_i}, t_{\mathbf{j}_i}) = \exp\left(-\frac{||s_{\mathbf{i}} - t_{\mathbf{j}}||^2}{2\sigma^2}\right) \tag{7}$$

That, $\kappa_p(s_{\mathbf{i}}, t_{\mathbf{j}})$ is a gaussian kernel of width σ and suitable for measuring the local similarity of subsequences in time series. This also ensures the positive definiteness of our suggested kernel (Eq. 2).

If we add another new data point y to the time series t', considering the assumption for κ_p and the above definition of $\mathcal{K}_p^S$ (Eq. 4), we have:

$$\mathcal{K}_p^S(s'x, t'y) = \kappa^*(x, y) \sum_{i=1}^{|s'|} \sum_{j=1}^{|t'|} \gamma^{|s'|-i+|t'|-j} \mathcal{K}_{p-1}^S(s'(1:i), t'(1:j)) \tag{8}$$

It means when new points are added, to measure the new p-suffix kernel, we must calculate similarities of $p-1$ length subsequences in the suffixes considering all possible degrees of warping. To evaluate $\mathcal{K}_p^S$ recursively, we define:

$$\mathcal{K}_p^{Sw}(k, l) = \sum_{i=1}^{k} \sum_{j=1}^{l} \gamma^{k-i+l-j} \mathcal{K}_{p-1}^S(s'(1:i), t'(1:j)) \tag{9}$$

Then equation 8 becomes:

$$\mathcal{K}_p^S(s'x, t'y) = \kappa^*(x, y) \mathcal{K}_p^{Sw}(|s'|, |t'|) \tag{10}$$

to express the above kernel recursively, we use the relation:

$$\sum_{i=1}^{a} \sum_{j=1}^{b} f(i,j) = f(a,b) + \sum_{i=1}^{a-1} \sum_{j=1}^{b} f(i,j) + \sum_{i=1}^{a} \sum_{j=1}^{b-1} f(i,j)$$
$$- \sum_{i=1}^{a-1} \sum_{j=1}^{b-1} f(i,j)$$

Let $f(i,j) = \gamma^{k-i+l-j} \mathcal{K}_{p-1}^S(s'(1:i), t'(1:j))$, $a = k$ and $b = l$, we have the following algorithm:

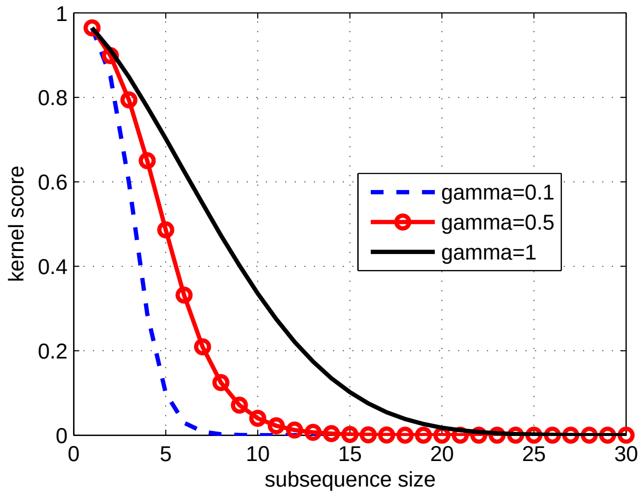


Fig. 3. Kernel score between two echoes of Ficus tree with different warping costs

Algorithm: *Recursive computation of the warped time resolved spectrum kernel.*

$$\mathcal{K}_p^{Sw}(k,l) = \mathcal{K}_{p-1}^S(s'(1:k),t'(1:l)) + \gamma\mathcal{K}_p^{Sw}(k,l-1) + \gamma\mathcal{K}_p^{Sw}(k-1,l) - \gamma^2\mathcal{K}_p^{Sw}(k-1,l-1)(11$$

$\mathcal{K}_p^S(s'x,t'y) = \kappa^*(x,y)(x,y)\mathcal{K}_p^{Sw}(|s'|,|t'|)$

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

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

The computation of the kernel follows a dynamic programming technique with the order of $O(p|s||t|)$. 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 $|t'| = |t|$. 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 of our algorithm to calculate a linear combination of all p-spectrum kernels is $O(p|s||t|)$.

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]. Fig.3 plots the kernel

Algorithm Warped Time resolved spectrum kernel

Input : Time series s and t of length n and m, max subsequence length l and warping cost γ;

Output: Array of spectrum kernel $\mathcal{K}[]$ with different sizes of subsequence-length from 1 to l);

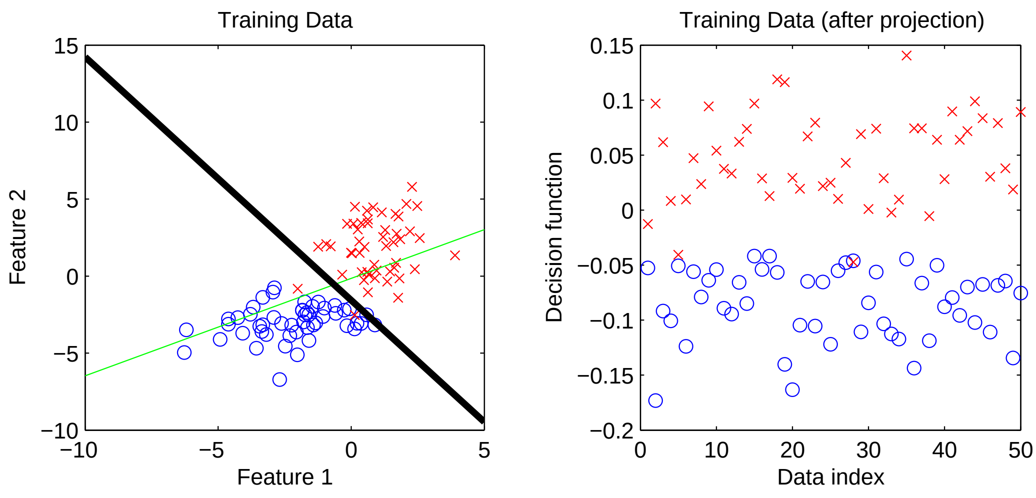
```
$KPSw(0:n, 0:m) = 0$;
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $m$ do
        $KPS(i,j) = \kappa^*(s_i, t_j)$;
        $\mathcal{K}[1] = \mathcal{K}[1] + KPS(i,j)$;
    end
end
for $p \leftarrow 2$ to $l$ do
    for $i \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $m$ do
            $KPSw(i,j) = KPS(i-1,j-1) + \gamma KPSw(i,j-1) + \gamma KPSw(i-1,j) - \gamma^2 KPSw(i-1,j-1)$;
            $KPS(i,j) = \kappa^*(s_i, t_j) KPSw(i-1,j-1)$;
            $\mathcal{K}[p] = \mathcal{K}[p] + KPS(i,j)$;
        end
    end
end
return $\mathcal{K}[]$
```

score of two samples of echoes reflected by a Ficus tree with different values of warping cost. We see that as the gamma parameter gets closer to 1 we let subsequences of two time series warp more and the similarity score (kernel score) increases. When gamma is equal to zero, the kernel is equal to the time-resolved spectrum kernel [7].

In practice and especially 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 $\theta_i \geq 0$. The weighted kernel is:

$$K(s,t) = \sum_{i=1}^{p} \theta_i \mathcal{K}_i^{norm}(s,t) \tag{12}$$

The parameter θ_i shows the weight of each i-length kernel and the optimum selection of those parameters extracts maximum similarities in the signals in consideration. It is a case of more general problem known as optimal kernel selection. For this task we selected the optimal kernels via maximizing the Kernel Fisher Discriminant (KFD) criterion ([8]).



(a) Projection (b) Linear discriminant function

Fig. 4. Linear Fisher Projection. a)Projection axis (green) and decision boundary between means of both classes (black bold) b) Linear discriminant function

3 Optimal kernel selection

3.1 Fisher discriminant based optimal kernel selection

Having two classes of labelled data, Fisher's idea was to look for a direction w that separates the classes means well (when projected onto the found direction) while achieving a small variance around these means. The hope is that, using this projection a classifier can classify the unlabelled data with a small error. The quantity measuring the difference between the means is called between class variance and the quantity measuring the variance around these class means is called within class variance, respectively. Then, in Linear Fisher Discriminant Analysis the goal is to find a direction that maximizes the between class variance while minimizing the within class variance at the same time (Fig. 4).

The Kernel Fisher Discriminant (KFD) is a non-linear extension of the linear Fisher discriminant analysis. Optimizing the selection of kernels using this criterion in kernel spaces ensures us that the obtained kernel scores represent the maximum similarity within signals of a class and minimum similarity between two different classes.

Given a set of n_+ positive training data $\chi_+ \subset \mathbb{R}^d$ and a set of n_- negative data $\chi_- \subset \mathbb{R}^d$, ($n = n_+ + n_-$, all data), and a map $\phi: \; u \rightarrow \phi(u) \in \mathcal{F}$, the aim is to find a direction $w = \sum_{i=1}^{n} \alpha_i \phi(u_i)$ in the feature space $\mathcal{F}$ given by weights $\alpha = [\alpha_1, ..., \alpha_n]$ which maximizes the separation of the mean scaled in the feature space and minimizes the variance in that direction (KFD criterion).

Considering the kernel matrix K:

$$K_{i,j} = k(x_i, x_j) =< \phi(x_i), \phi(x_j) > \tag{13}$$

For the direction w, the KFD criterion will be in the form of ([8]):

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T (N + \lambda I)\alpha} \tag{14}$$

The parameter λ is a regulation factor and M and N are gained in terms of the kernel matrix K:

$$M = (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T \tag{15}$$

where $\mu_+ = \frac{1}{n_+} \sum\limits_{x \in \chi_+} K_x$ and $\mu_- = \frac{1}{n_+} \sum\limits_{x \in \chi_-} K_x$ are scaled means in the feature space and:

$$N = KDK^T \tag{16}$$

where $D = \begin{bmatrix} I_{n_+} - \frac{1}{n_+} 1_{n_+} 1_{n_+}^T & 0 \\ 0 & I_{n_-} - \frac{1}{n_-} 1_{n_-} 1_{n_-}^T \end{bmatrix}_{n \times n}$

in which 1_n and I_n denote the vector of all ones and the identity operator in $\mathbb{R}^d$, respectively.

The parameter α that maximizes the KFD criterion is obtained via:

$$\alpha_{\max} = (N + \lambda I)^{-1}(\mu_+ - \mu_-) = (KDK^T + \lambda I)^{-1} K y$$

where:

$$y = \begin{bmatrix} (1/n_+) 1_{n_+} \\ (-1/n_-) 1_{n_-} \end{bmatrix}_{n \times 1}$$

which results in:

$$J_{\max}(K) = \alpha_{\max}^T K y = y^T K (K D^T K + \lambda I)^{-1} K y \tag{17}$$

If we consider the variable K as a linear combination of a set of kernel matrices, in the next step we try to find the matrix K, which maximizes the above equation. Considering equations 12 and 17, the problem of finding the optimal kernel in terms of maximizing the Fisher discriminant ratio can be written as:

$$\begin{array}{lc} \min & f(\sum_{i=1}^{l} \theta_i \mathcal{K}_i^{norm}) = -J_{\max}(\sum_{i=1}^{l} \theta_i \mathcal{K}_i^{norm}) \\ \text{subject to} & \theta \succeq 0, \quad 1^T\theta = 1 \end{array}$$

It is easy to prove the convexity of the above objective function. We suppose $f(x,y) = x'y^{-1}x$, $h(K) = Ky$ and $g(K) = KD'K + \lambda I$, considering the convexity of f, h and g, we conclude the convexity of $f(h,g)$ and so the above objective function. Then, any local optimum answer for the objective function is a global one of that, too. One suggested method (Kim et. al [13]) was to use the convex optimization and bring the objective function in the from of Semi-Definite Programming (SDP) via the Schur complement technique [15]. In their method, the SDP solver of SeDuMi [14] was used to solve the SDP.

Instead of that method we use a newly suggested method for local optimization known as *Mesh Adaptive Direct Search method* that needs less run time (approximately one third in our experiments) on a similar machine.

3.2 Mesh Adaptive Direct Search method

MADS (defined by Audit and Dennis [9]) is a class of algorithms for nonlinear optimization. It is a modification of the generalized pattern search (GPS [16]) algorithm for local optimization. In summary, this algorithm computes a series of points that get closer and closer to the optimal point. It searches a set of random selected points, called a *mesh*, around the *current point*-the point computed at the previous step of the algorithm. The mesh is formed by adding the current point to a scalar multiple of a set of vectors called the *mesh size* (pattern). (The GPS algorithm uses fixed direction vectors, whereas the MADS algorithm uses a random selection of vectors to define the mesh). The point in the mesh that improves the objective function becomes the current point at the next step. The value of the objective function either decreases or remains the same from each current point to the next. The routine continues until a stopping criterion is fulfilled. The formal definitions and algorithm from [9] follow:

Suppose that $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a given function under general constraint $x \in \Omega \subseteq \mathbb{R}^n, \ \Omega \neq 0$.

If $\Omega \neq \mathbb{R}^n$ (constraint optimization), the algorithm attempts to locate a minimizer of function f over Ω by means of a *barrier* function:

$$f_\Omega = \begin{cases} +\infty & \text{if } p \notin \Omega \\ f & \text{otherwise.} \end{cases}$$

The algorithm does not require the use of the approximations or derivatives of f (free-derivative method). This is useful (especially in our case) when ∇f is not available or can not be accurately estimated.

MADS is an iterative algorithm where at each iteration k a finite number of trial points are generated and their objective function values are compared with the current incumbent value $f_\Omega(x)$ as the best objective function value found so far. Each of these trial points lies on the *current mesh*, constructed from a finite fixed set of n_D directions $D \subset \mathbb{R}^n$ scaled by a *mesh size* parameter $\Delta_k^m \in \mathbb{R}_+$. The mesh size parameter controls the coarseness or fineness of search at iteration k. Δ_{k+1}^m is adjusted from Δ_k^m depending on the success of that iteration.

$D_{n\times n_D}$ must be a positive spanning set, *i.e.*, nonnegative linear combinations of its elements must span $\mathbb{R}^n$, and each direction $d_j \in D$ ($j \in [1, n_D]$) must be the product of some fixed nonsingular generating matrix $G \in \mathbb{R}$ by an integer vector $z_j \in \mathbb{Z}^n$. We consider Z a matrix whose columns are z_j, for $j = 1, 2, ..., n_D$, and use matrix multiplication, $D = GZ$. If S_k is the set of points where the objective function f had been evaluated at the start of iteration k, at iteration k, the current mesh is defined as:

$$M_k = \bigcup_{x \in S_k} \{x + \Delta_k^m Dz : \; z \in N^{n_D}\}$$

The above definition ensures that all previously visited points lie on the mesh, and that new trial points can be selected around any of them.

Each iteration consists of SEARCH and POLL steps. In the SEARCH step the value of f_Ω at any finite number of mesh points is evaluated. When a improved mesh point, at which f_Ω is less that $\min_{x\in S_k} f_\Omega$, is generated, the iteration may stop, or it may continue to find a better improved mesh point. Otherwise the POLL steps begins and the algorithm generates and evaluates f_Ω around the current incumbent x_k, where $f_\Omega(x_k) = \min_{x\in S_k} f_\Omega(x)$. The poll size parameter Δ_k^p limits the distance between x_k and the new trail points. The set of new trail points is called a *frame* and x_k is the *frame center.* This frame is generated using x_k, Δ_k^p, Δ_k^m, and D to obtain a set D_k of positive spanning directions.

At iteration k, the *MADS frame* is defined to be the set:

$$P_k = \{x_k + \Delta_k^m d : \ d \in \ D_k\} \subset M_k$$

in which D_k is a positive spanning set ($0 \notin D_k$) and for each $d \in D_k$:

- $d \neq 0$ is nonnegative integer combination of the directions in D,
- $\Delta_k^m d$, the distance from the frame center, is bounded by a constant times the poll size parameter: $\Delta_k^m \|d\| \leq \Delta_k^p \ \max\left\{\left\|d'\right\| : d' \in D\right\}$,
- limits of the normalized set D_k are positive spanning sets [17].

To ensure the convergence, the radii of successive frames must converge to zero at a slower rate than the mesh size parameter. It means $\Delta_{k+1}^m \leq \Delta_{k+1}^p$ and it must satisfy

$$\lim_{k\to\infty} \inf \Delta_k^p = \lim_{k\to\infty} \inf \Delta_k^m = 0$$

The algorithm evaluates f_Ω at points in the frame P_k until it finds an improved point t with $f_\Omega(t) < f_\Omega(x_k)$ or until it has evaluated f_Ω at all of the points in P_k. When the POLL step fails to generate an improved mesh point then the frame is called a *minimal frame*, and the fame center p_k is said to be a minimal frame center and the poll size parameter should be updated.

At iteration k, the mesh size parameter is updated according to:

$$\Delta_{k+1}^m = \begin{cases} \Delta_k^m/4, & \text{if } p_k \text{ is a minimal frame center} \\ 4\Delta_k^m, & \text{if an improved mesh point is found, and} \Delta_k^m \leq \frac{1}{4} \\ \Delta_k^m, & \text{Otherwise.} \end{cases}$$

and the poll size parameter as:

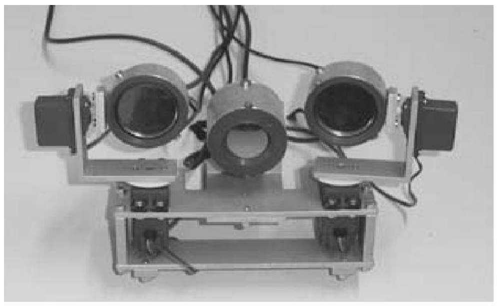
$$\Delta_k^p = \sqrt{\Delta_k^m}$$

These rules guarantee that Δ_k^m is always a power of $1/4$ less than or equal to one, and $\Delta_k^m \leq \Delta_k^p$ for all k. We can select a default minimum value of mesh size as stopping criterion to be fulfilled.

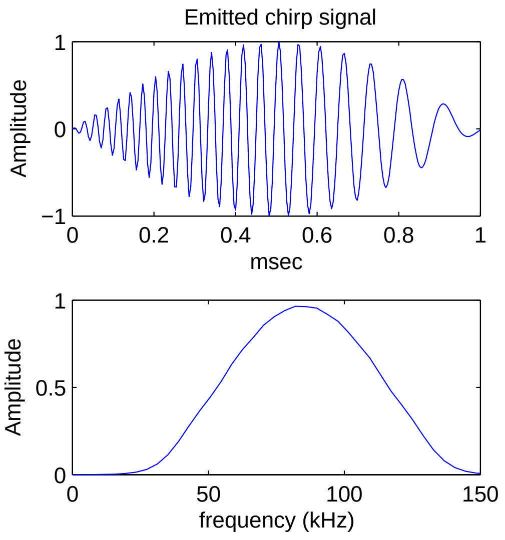
In summary, the MADS algorithm is described as follows:

Algorithm Mesh Adaptive Direct Search Algorithm

```
step 0  [Initialization] Given $x_0$, $\Delta_k^m \leq \Delta_k^p$ and $D_{n\times n_D}$ a positive spanning
    set. Set $\Delta_0^m = 1$, and the Iteration counter k:=0
step 1  [SEARCH step] Evaluate $f_\Omega$ on a finite subset of trial points on the
    mesh $M_k$ as defined above. If an improved trail point $t$ that $f_\Omega(t) < f_\Omega(x_k)$
    is found, declare $k$ successful and go to step 3.
step 2  [POLL step] Evaluate $f_\Omega$ at points from the frame $P_k$ until $x_{k+1}$ that
    $f_\Omega(x_{k+1}) < f_\Omega(x_k)$ is found. If no such point exists declare $k$ unsuccessful.
step 3  [Parameter update] If iteration $k$ was declared unsuccessful, then set
    $p_{k+1} = p_k$ (minimal frame center). Otherwise $p_{k+1}$ is an improved mesh
    point. Update $\Delta_{k+1}^m$ and $\Delta_{k+1}^p$. If an appropriate stopping criteria has not
    been met, set k:=k+1 and go to step 1.
```



(a)



(b)

Fig. 5. (a) Biosonar head and (b) Emitted chirp signal and its frequency content.

4 Experiment and Results

We used a sonar head system consisting of three ultrasound transducers, one for emission chirp signals (Polaroid 7000), two for reception (Polaroid 6000)(Fig. 5.a) and tried to classify three trees as landmarks (Fig. 1). The emitted pulse was a linearly frequency modulated chirp sweeping from 20kHz to 120kHz in 1 ms (Fig. 5.b). The reflected echo contains the information about the geometry of the tree.

Fig. 6 shows the block diagram of the data 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*

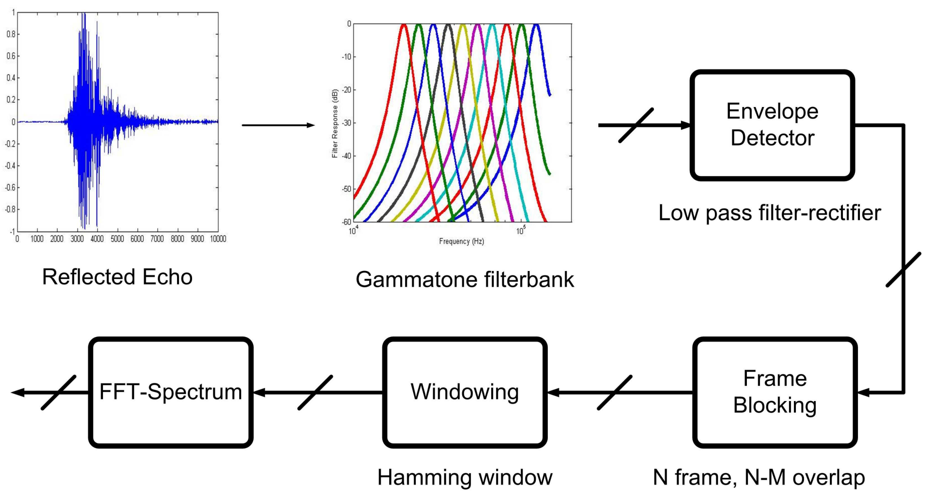


Fig. 6. Block diagram of the preprocessing steps for reflected echoes.

blocking. In this step the signal blocked to frames of N samples, is separated 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. 2 shows the examples of the preprocessed echoes of Ficus and Schefflera trees.

We consider the output of the block diagram shown in Fig. 6, 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). The task is to classify the echoes of each object using those features.

We gathered the sonar data, 720 echoes for each tree. After the preprocessing steps for each echo (Fig. 6), 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 $\sigma \in \{1, 10, 100, 1000\}$ (Eq. 7), where $s[m]$ and $s[n]$ are the m-th and n-th of pre-processed echoes and l is the length of the windowed time series (in our experiment 90). Using the optimal kernel selection noted above, we found the optimal value for θ_i in Eq. 12 and calculated the matrix K:

$$\mathbf{K}(i, j) = K(s[i], s[j]) = \sum_{k=1}^{l} \theta_i^{opt} \mathcal{K}_l^{norm}(s[i], s[j]) \tag{18}$$

in which i, $j \in [1, 300]$ and $s[i]$ is i-th echo. For Ficus echoes $i \in$[1,100], for Bamboo $i \in$ [101,200] and for Schefflera $i \in$[201,300]. In this study, we found that suitable values for σ (Eq. 7) are in the range [10,100].

A SVM learns a classification function $f(x)$ of the form:

$$f(x) = \sum_{i;x_i \in \chi_+} \lambda_i K(x, x_i) - \sum_{i;x_i \in \chi_-} \lambda_i K(x, x_i) \tag{19}$$

where non-negative λ_i weights are computed during training by maximizing a quadratic objective function and $K(.,.)$ is the kernel matrix. 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. After training the classifier, we used the remaining data (1860 echoes) for testing.

Figure 7 shows the accuracy of the classifier for those trees with different warping costs (γ) and $\sigma = 100$, based on the number of echoes as observation. It shows a high accuracy even for a low number of echoes. We see that the parameter γ can affect the accuracy of the classifier and the accuracy of the time-resolved spectrum kernel [7] (γ=0, without warping) increases in each tree by changing the parameter γ.

Table 1 shows the accuracy of the classifier when it decides based on only one observation. Best accuracyies for Ficus, Bamboo and Schefflera trees are gained for $\gamma = 0.1$, $\gamma = 0.3$ and $\gamma = 0.2$, respectively. This parameter lets the kernel to consider a warping (with a cost) for the subsequences of the time series and extract their similarity. Considering that parameter in our classification task is justifiable, because the echoes reflected by the adjacent leaves of each tree can have somehow similar patterns but not exactly the same, so we need to have a parameter (γ) that can let the kernel capture those similarities, too. The optimal value of that parameter for each tree can be related to the physical specification of each tree. We see if γ gets closer to 1 (no cost for warping) the accuracy decreases.

Comparing with the previous works of our group (Wang et al. [18]), The new classifier shows a notable improvement in accuracy. The best result for classification gained before was through template matching in 2D biosonar acoustic images (using a 2D Discrete Cosine Transform). The classification was made via extracting the maximum normalized cross correlation between the acoustic templates (Fig. 9). As shown in Fig. 8, we could get higher accuracy in both single and repeated observations (even with fewer echoes) compared with Fig. 9 (note the different horizontal and vertical axes).

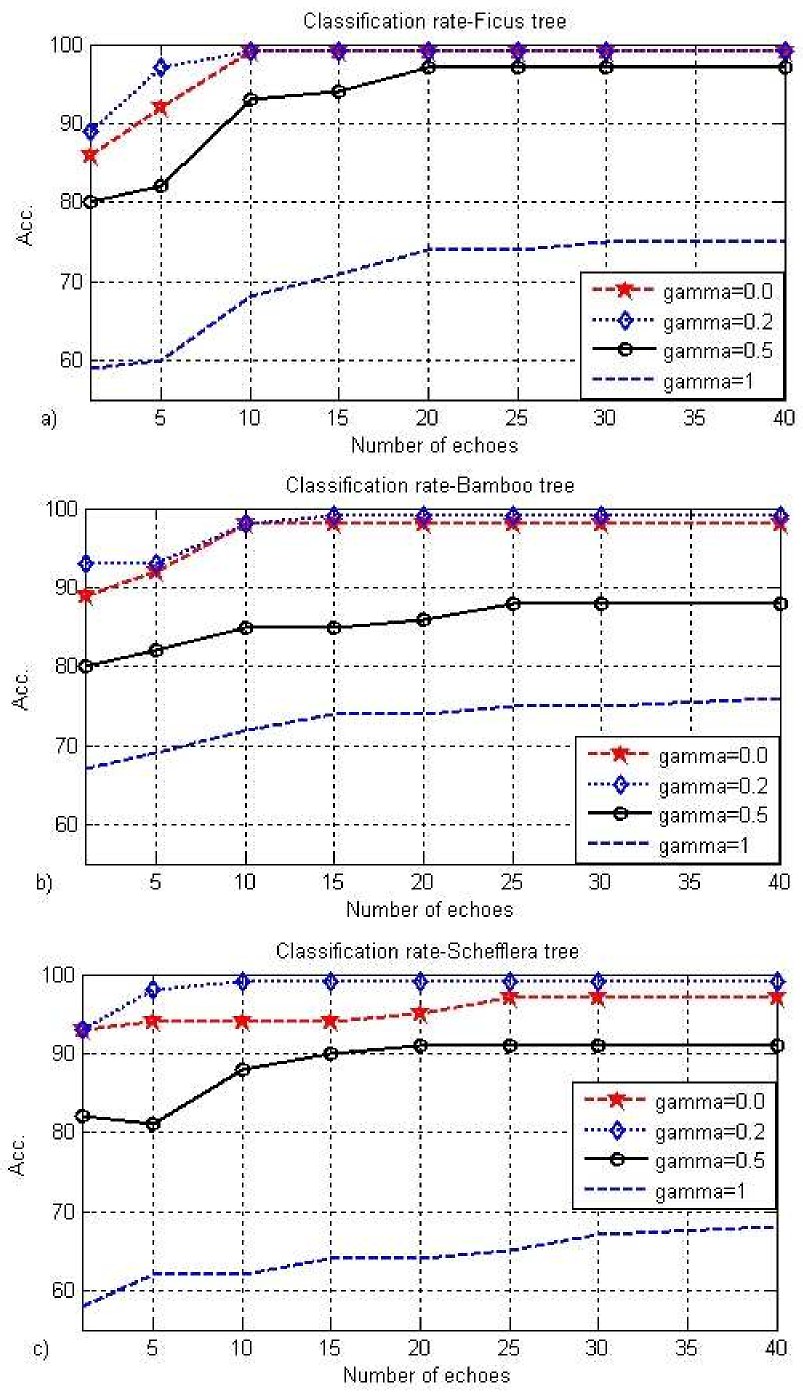


Fig. 7. The accuracy of the classifier using our suggested kernel with different warping costs ($\sigma = 100$) for a) Ficus, b) Bamboo and c)Schefflera. For $\gamma = 0$, the kernel is similar to the time-resolved spectrum kernel [7].

5 Conclusion

In the previous study [7] we presented a spectrum kernel for local similarity extraction in time series. That kernel measured the local similarities without

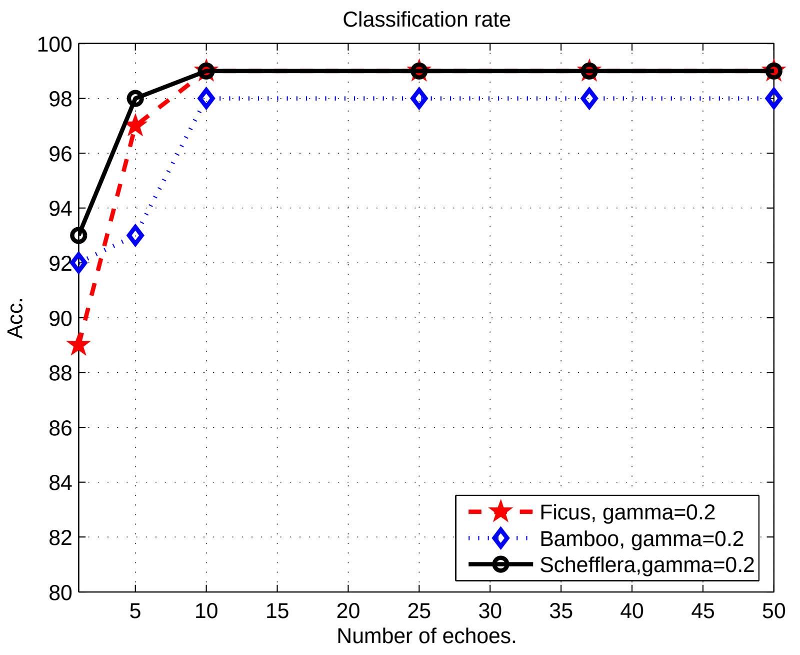


Fig. 8. The accuracy of classifiers using different numbers of echoes for testing with the Warped time-resolved spectrum kernel ($\gamma = 0.2$).

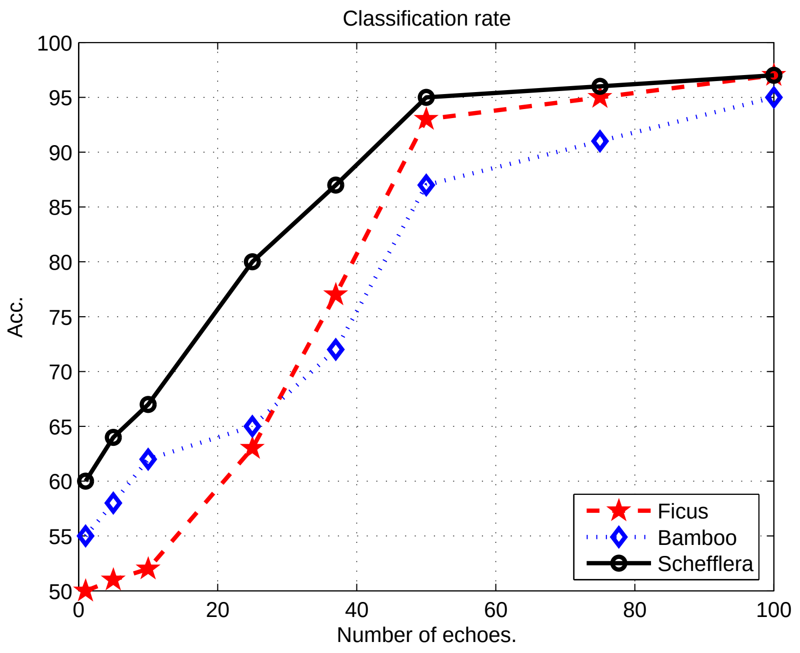


Fig. 9. The accuracy of classifiers via template matching using acoustic images of echoes (Wang et al. [19]).

γ	Ficus	Bamboo	Schefflera
$\gamma = 0$	86.2	89.5	90.2
$\gamma = 0.1$	**89.1**	90.1	91.3
$\gamma = 0.2$	88.6	91.4	**93.8**
$\gamma = 0.3$	87.6	**93,1**	91,1
$\gamma = 0.5$	80.1	81.3	82.1
$\gamma = 0.1$	59.2	67.4	58.1

Table 1
Classification rate based on different values of γ

considering the warping. In this paper we used a more general kernel that considers the warping in p-length subsequences and measures the similarity between two time series by adding all local similarities of subsequences independent of their positions. This kernel is useful for similarity extraction in time series, which contain randomly position local similarities, representing the texture of an object. With different values of p we obtain different kernel scores. We presented a method to extract an optimal linear set of those kernels based on the Fisher criterion in kernel space. With this criterion, the obtained kernel ensures the maximum similarity score of signals of one object and the minimum similarity score between signals of two different objects. We used a Mesh Adaptive Direct Search method (MADS) to solve our optimization problem. Compared with other matching methods for biosonar signals, we obtained better results with our suggested kernel based method.

References

[1] H. Shimodaira, K. -I. Noma, M. Nakai and S. Sagayama, Dynamic Time-Alignment Kernel in Support Vector, Advances in Neural Information Processing Systems,vol. 2, 2001, pp. 921-928

[2] M. Cuturi, J. -P. Vert, O. Birkenes and T. Matsui, A kernel for time series based on global alignments, Proc. ICASSP, Hawaii, 2007.

[3] C. Leslie, E. Eskin and W. S. Noble, The spectrum kernel: A string kernel for SVM protein classification, Pac. Symp. Biocomput., 2002, pp. 564575.

[4] 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, 2003, pp. 1441-1448.

[5] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins, Text Classification using String Kernels, Jounal of Machine Learning Research, 2002, pp. 419-444.

[6] T. Jaakkola, M. Diekhans and D. Haussler, Using the Fisher Kernel Method to Detect Remote Protein Homologies, 1999, pp. 149-158

[7] M. Beigi, M. Wang and A. Zell, Time-resolved spectrum kernel for biosanar target classification, In Signal Processing, Pattern Recognition, and Applications, February 2007, pp. 126-131.

[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf and K.-R. Müller, Fisher discriminant analysis with kernels. Neural Networks for Signal Processing IX, Proceedings of the 1999 IEEE Signal Processing Society Workshop, 1999, pp. 41-48.

[9] C. Audet and J.E Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM J. optim. vol. 17, No. 1, 2006, pp.188–217.

[10] C. F. Moss and A. Surlykke, Auditory scene analysis by echolocation in bats, J. Acoust. Soc. Am. 110(4),2001, 2207-2226.

[11] P. McKerrow and N. Harper, Plant acoustic density profile model of CTFM ultrasonic sensing, J. IEEE Sensors, 1(4), 2001, 245-255.

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

[13] S.-J. Kim, A. Magnani and S. Boyd, Optimal Kernel Selection in Kernel Fisher Discriminant Analysis, Proceeding of the 23th Int. Conf. on Machine Learning (ICML), 2006, pp. 465-472.

[14] J. Strum, Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones, Available from `sedumi.macmaster.ca/`.

[15] S. Boyd and L. Vandenberghe, Convex optimization, 2004, (Cambridge University Press, 2004).

[16] C. Audet and J.E Dennis Jr., Analysis of generalized pattern searches, SIAM J. optim. vol.13, 2003, pp. 889-903.

[17] I. D. Coope and C. J. Price, Frame-based methods for unconstrained optimization, J. optim. and applications vol.107 2000, pp. 261-274.

[18] M. Wang and A. Zell, Classification of natural landmarks with Biosonar, Journal of the Acoustic Society of America (JASA), vol. 116, 2004.

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