

Kernels for Large Margin Time-Series Classification

Sivaramakrishnan K R, Karthik K and C. Bhattacharyya

Abstract— In this paper we propose a novel family of kernels for multivariate time-series classification problems. Each time-series is approximated by a linear combination of piecewise polynomial functions in a Reproducing Kernel Hilbert Space by a novel kernel interpolation technique. Using the associated kernel function a large margin classification formulation is proposed which can discriminate between two classes. The formulation leads to kernels, between two multivariate time-series, which can be efficiently computed. The kernels have been successfully applied to writer independent handwritten character recognition.

I. INTRODUCTION

In recent years with the advancement in computer storage and sensor technologies it has become possible to efficiently gather and store temporal data in real world applications. These improvements have made temporal learning a plausible and important area of research in machine learning. Temporal learning involves many learning problems of interest including alignment matching, classification and clustering. Temporal/time-series classification is manifested in many real world applications including online handwritten text recognition, speech recognition, financial anomaly detection and many other business and personal applications. The problem of online handwriting recognition is to infer the written character and/or character sequence given a time-series written on a touch sensitive device. A similar situation arises in speech processing where utterances of various words, vowels etc are represented by multidimensional time-series of LPC coefficients.

Recently, mobile handheld devices like PDA have become very popular mode of personal computing. These devices have very stringent size and power/processing constraints. Traditional input interfaces like keypads are not suitable and new interfaces like touchpads have become popular. These interfaces are required to be as efficient as traditional interfaces. These devices are mainly used for interpersonal communication, hence language interfaces like speech/handwriting need to be efficient and accurate. The usage of these interfaces depends on the efficiency and accuracy of the methods applied to solve these problems. Time series classification provides a general framework for language independent interfaces.

Handwritten character based interfaces are one of the simplest and popular mobile touchpad interfaces. Handwritten character recognition(HCR) is one of the widely researched problem in pattern recognition, with methods that handle

offline and online handwritten character recognition. Real-time applications of HCR requires usage of online handwritten character recognition methods. Time series classification provides a general setting for Multi-lingual online HCR. In this paper, we provide an efficient time series classification method which has been successfully applied to language independent online handwritten character recognition.

In general a time-series is represented as a set of time ordered vectors

$$\mathcal{F} = \{\mathcal{F}(t_1), \mathcal{F}(t_2), \dots, \mathcal{F}(t_n)\}$$

where \mathcal{F} is the time-series, $\mathcal{F}(t_k)$ denotes a vector sampled at the time instant $t_k \in \mathbb{R}$, $t_k < t_l$. where $k < l$ and n is the total number of samples. Note that the time differences $t_k - t_{k-1}$ are not necessarily constant.

The problem of time-series classification has been well studied in the area of speech processing/recognition where Hidden Markov Models (HMMs) have emerged as a powerful tool. The applicability of HMMs to online handwriting recognition was explored in [1] with mixed results. There are other methods based on Dynamic Programming approach such as Dynamic Time Warping which are efficient and have been successfully applied to the problem of time-series classification. A variant of DTW which uses clustering techniques was proposed in [2]. Current literature suggests that the state of the art for online handwriting recognition is online Scanning N-Tuple(SNT) algorithm[3]. The SNT algorithm converts the time series into strings using chain coding techniques, a language modelling technique is later applied to the strings to build a statistical model [4]. The statistical models so built are used for classification.

In recent times, Support Vector Machines (SVMs)[5] have emerged as a powerful method for classifying fixed length vectors. SVM is a large margin classification formulation which tries to minimize the generalization error while keeping the empirical error low. Given N vectors and their associated labels $D = \{(x_1, y_1) \dots (x_N, y_N)\}$, such that $x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$ and $i \in \{1 \dots N\}$. SVM learns a linear discriminant function

$$f(x) = \text{sign}(w^T x + b)$$

such that

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \\ \text{subj to} \quad & y_i (w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \\ & 1 \leq i \leq N. \end{aligned}$$

This is a quadratic programming problem which can be efficiently solved using the dual. The dual is given by,

Sivaramakrishnan, Karthik and Dr Bhattacharyya are with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India (email:sivaramakrishnan@gmail.com, karthikk@csa.iisc.ernet.in, chiru@csa.iisc.ernet.in)

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_i \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^N y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \\ & 1 \leq i \leq N. \end{aligned}$$

where α_i 's are the Lagrange multipliers. The w and b can be calculated from Karush-Kuhn-Tucker conditions. In the dual problem and the discriminant function $f(\cdot)$ the input vectors are only seen in the form of dot products. Using kernel functions [6] one can extend the classification framework to arbitrary feature spaces. The kernel function $K(\cdot, \cdot)$ is essentially positive semi-definite symmetric function of the input data and it can be interpreted as a dot product $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ between samples $\phi(\cdot)$ in a suitably defined feature space. SVMs have been proved to have good generalization ability and have been applied to large number of problems.

The main research issue in extending the SVM formulation to time-series data is the design of these kernels. The Fisher kernel [7] was an important breakthrough which was successfully applied to classifying protein sequences. It combines a generative model like a HMM with discriminative methods like SVM. The kernel function is derived from a HMM. Each kernel computation requires two passes of a forward backward algorithm on a pre-trained HMM and hence is computationally heavy. Alternatively it is possible to derive kernels based on dynamic programming based alignment for classification of sequences [8]. Gaussian Dynamic Time Warping (GDTW) is an instance of such an approach, which was explored by [9] for online handwriting recognition with encouraging results over HMM based classifiers. GDTW combines the RBF kernel with the score obtained by the DTW comparison between two time series. The GDTW kernel is calculated as,

$$K(\mathcal{F}_i, \mathcal{F}_j) = e^{-\gamma D(\mathcal{F}_i, \mathcal{F}_j)}$$

where \mathcal{F}_i and \mathcal{F}_j are input sequences and D is the DTW distance defined as.

$$D(\mathcal{F}_i, \mathcal{F}_j) = \min_{\psi_i, \psi_j} \frac{1}{n} \sum_{k=1}^n d(\mathcal{F}_i(\psi_i(k)), \mathcal{F}_j(\psi_j(k)))$$

where ψ is the alignment path and d is the local distance measure, Euclidean distance can be used as d in most applications. (See [10] for more details). For SVMs to be applicable to online HCR applications an efficient method of kernel computation between time-series is needed.

In [11] a piecewise linear function approximation was used to re-sample the time series vector, which was used to train a SVM classifier. In this paper we study the problem of classifying multivariate time-series and develop efficient kernels suitable for online handwriting recognition on a handheld device like PDA.

The main contribution of the paper is to propose a set of kernels for time-series which can be used not only for

classification, but other applications like clustering, novelty detection etc. The other important contribution is to generalize the piecewise linear interpolation scheme used in [11] to piecewise polynomials in a Reproducing Kernel Hilbert Space (RKHS) setting. Lastly the paper also gives an efficient $O(n)$ algorithm for computing the interpolation step rather than an $O(n^3)$ algorithm proposed in [11]. This allows for efficient computation of the decision, in online HCR applications, on handheld devices.

This paper is organized as follows. Section II describes a scheme for interpolation of a time-series by sum of polynomials in a RKHS. The scheme is used to derive a continuous function describing the time-series. In Section III a large margin formulation is later derived to discriminate these interpolated functions. Section IV describes the empirical evaluation of the proposed formulation with the state of the art on several representative benchmark datasets. Section V describes the contributions and concludes the paper.

II. RKHS BASED PIECEWISE POLYNOMIAL INTERPOLATION

In this section we start by briefly describing an interpolation scheme due to Moore [12], which uses piecewise polynomial functions as basis functions. The interpolation scheme is used to represent each time-series as a function and using a large margin approach we show how to discriminate between such functions. Kernels are extracted from the formulation and an algorithm is provided for time series classification.

Let $0 \leq s_1 < s_2 < \dots < s_{n-1} < s_n \leq 1$ be a sequence of points and let $\mathcal{F}_k = \mathcal{F}(s_k)$ be the function $\mathcal{F} : [0, 1] \rightarrow \mathbb{R}$ evaluated at s_k . The interpolation problem can be viewed as approximating \mathcal{F} given $\mathcal{D} = \{(s_k, \mathcal{F}_k) | 1 \leq k \leq n\}$.

Denote by \mathcal{H}^q the space of all functions $f : [0, 1] \rightarrow \mathbb{R}$, whose q^{th} derivatives are in $\mathcal{L}_2(0, 1)$. It can be shown that \mathcal{H}^q is a RKHS [12] with the inner product defined as,

$$\langle f, g \rangle = \sum_{j=0}^{q-1} \frac{f^{(j)}(0)g^{(j)}(0)}{j!} + \int_0^1 f^{(q)}(t)g^{(q)}(t)dt, \quad (1)$$

where $f, g \in \mathcal{H}^q$ and $f^{(r)}(t)$ denotes the r^{th} derivative. The reproducing kernel for the RKHS is,

$$R_s^q(t) = \sum_{j=0}^{q-1} \frac{s^j t^j}{(j!)^2} + \int_0^{\min(s,t)} \frac{(s-u)^{q-1}(t-u)^{q-1} du}{((q-1)!)^2}$$

As special cases of the above function,

$$R_s^q(t) = \begin{cases} 1 + \min(s, t) & q = 1 \\ \left. \begin{aligned} 1 + st + \frac{st^2}{2} - \frac{t^3}{6} & \text{if } t < s \\ 1 + st + \frac{s^2 t}{2} - \frac{s^3}{6} & \text{if } t > s \end{aligned} \right\} & q = 2. \end{cases}$$

Consider the function $\tilde{\mathcal{F}}$,

$$\tilde{\mathcal{F}}(s) = \sum_{k=1}^n c_k R_{s_k}^q(s), \quad (2)$$

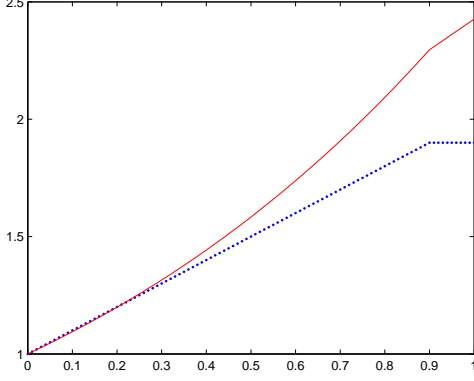


Fig. 1. Plot of basis functions with $q = 1$ and $q = 2$

where the basis function $R_{s_k}^q(s)$ are piecewise polynomial.

For any $0 \leq s \leq 1$ and $f \in H^q$, the basis functions satisfies, $\langle f, R_{s_k}^q \rangle = f(s)$, which is also referred as the RKHS property. To ensure $\bar{\mathcal{F}}$ best approximates \mathcal{F} , one can choose c by requiring that, $\langle (\bar{\mathcal{F}} - \mathcal{F}), R_{s_k}^q \rangle = 0 \forall k = 1, 2, \dots, n$. Using equation (2) and the RKHS property, the above set of equations reduces to,

$$\sum_{k=1}^n \langle R_{s_k}^q, R_{s_l}^q \rangle c_k = \mathcal{F}(s_l).$$

This is a set of linear equations in c and can be efficiently solved. It can be shown that such a choice of c is optimal and the resulting $\bar{\mathcal{F}}$ best approximates \mathcal{F} given \mathcal{D} .

III. LARGE-MARGIN CLASSIFICATION OF UNIVARIATE TIME-SERIES

A univariate time-series $\mathcal{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_n\}$ $\mathcal{F}_k = \mathcal{F}(t_k)$, evaluated at time instants $\mathcal{F}(t_k) \in \mathbb{R}$, $0 \leq t_1 < t_2 < \dots < t_n$, can be approximated by a continuous function (2). the basis coefficients c can be chosen to satisfy,

$$Gc = [\mathcal{F}_1 \dots \mathcal{F}_n]^T, \quad (3)$$

where $G_{kl} = \langle R_{s_k}^q, R_{s_l}^q \rangle$, is the Gram Matrix for the basis elements $R_{s_k}^q$.

The learning problem can be posed as that of computing a classifier on the dataset

$$\mathcal{T} = \{(\bar{\mathcal{F}}^i, y_i) | \bar{\mathcal{F}}^i \in \mathcal{H}^q, y_i \in \pm 1, \forall i = 1, 2, \dots, N\}$$

More precisely the problem can be stated as finding a $w \in \mathcal{H}^q$ and $b \in \mathbb{R}$ so that the decision function given by,

$$y = \text{sign}(\langle \bar{\mathcal{F}}, w \rangle + b),$$

can correctly predict the class label of a given time-series. A w chosen such that it has the minimum norm $\langle w, w \rangle$ provides the best generalization [5]. The formulation for finding w and b is,

$$\begin{aligned} \min_{w \in \mathcal{H}^q, b} \quad & \frac{1}{2} \langle w, w \rangle \\ \text{subject to} \quad & y_i (\langle w, \bar{\mathcal{F}}^i \rangle + b) \geq 1 \\ & 1 \leq i \leq N \end{aligned} \quad (4)$$

As $w \in \mathcal{H}^q$, one can express w as linear combination of the basis functions in \mathcal{H}^q , specifically,

$$w(t) = \sum_{l=1}^M d_l R_{m_l}^q(t),$$

where m_l are normalized time instants such that $m_l \in [0, 1]$. The vector $m = [m_1, \dots, m_M]$ needs to be chosen appropriately depending on the data. The inner product between two functions $\bar{\mathcal{F}}_1 = \sum_{k=1}^{n_1} c_k R_{s_k}^q$ and $\bar{\mathcal{F}}_2 = \sum_{l=1}^{n_2} d_l R_{s_l}^q$ is given by,

$$\langle \bar{\mathcal{F}}_1, \bar{\mathcal{F}}_2 \rangle = \sum_{k=1}^{n_1} \sum_{l=1}^{n_2} c_k d_l \langle R_{s_k}^q, R_{s_l}^q \rangle. \quad (5)$$

Using the above definition the inner product of w and data $\bar{\mathcal{F}}$ is,

$$\langle w, \bar{\mathcal{F}} \rangle = \sum_{l=1}^M \sum_{k=1}^n d_l c_k \langle R_{m_l}^q, R_{s_k}^q \rangle = d^T A c,$$

where $c = [c_1, \dots, c_n]$ is a vector determined by (3). The A matrix is given by

$$A_{lk} = \langle R_{m_l}^q, R_{s_k}^q \rangle \quad 1 \leq l \leq M, 1 \leq k \leq n, \quad (6)$$

where M is the number of basis functions describing w . Again from the definition of the inner-product (5), the objective can be stated as,

$$\|w\|^2 = \langle w, w \rangle = \sum_{k=1}^M \sum_{l=1}^M d_k d_l \langle R_{m_k}^q, R_{m_l}^q \rangle. \quad (7)$$

Using (7) the optimization problem (4) can be restated as,

$$\begin{aligned} \min_{d, b} \quad & \frac{1}{2} d^T B d \\ \text{subject to} \quad & y_i (d^T A^{(i)} c^{(i)} + b) \geq 1 \\ & 1 \leq i \leq N, \end{aligned} \quad (8)$$

where B is the Gram matrix for the basis functions given by,

$$B_{kl} = \langle R_{m_k}^q, R_{m_l}^q \rangle$$

from equation (7), $A^{(i)}$ and $c^{(i)}$ is the A matrix and c vector respectively for the i^{th} data point.

The matrix B is positive semi-definite and hence the optimization problem (8) is an instance of convex quadratic programming and can be solved by standard tools. The decision function can now be evaluated as,

$$y = \text{sign}(d^T A c + b).$$

The above formulation (8) provides a unified scheme for time-series classification. The formulation is extended to handle multivariate data. A simple $O(n)$ scheme is also suggested for linear interpolation.

A. The time-series kernel

The Gram matrix B can be factorized as $B = U\Sigma U^T$. Where U is the matrix formed by normalized eigen vectors and Σ is a diagonal matrix, the eigen values being the diagonal elements. It can be shown that $UU^T = U^T U = I$. Based on this we can extract a kernel from (8) that can be used to calculate similarity between sequences of time-series. Let $d = U^T \Sigma^{-\frac{1}{2}} u$, where $\Sigma^{-\frac{1}{2}}$ is the inverse of the matrix square root of the diagonal matrix Σ . Substituting in (8) one obtains the following formulation.

$$\begin{aligned} \min_{u,b} \quad & \frac{1}{2} u^T u \\ \text{subject to} \quad & y_i (u^T X_i + b) \geq 1, \\ & 1 \leq i \leq N, \end{aligned} \quad (9)$$

where

$$X_i = \Sigma^{-\frac{1}{2}} U A^{(i)} c^{(i)}.$$

The matrix $\Sigma^{-\frac{1}{2}} U$ can be pre-computed for faster operation. The computation cost of each X_i involves the computation of $c^{(i)}$'s (from (3)) which is $O(n^3)$ and $A^{(i)}$'s (from (6)) which is $O(n^2)$. The formulation (9) is equivalent to an SVM [5] with the Kernel,

$$K(X_i, X_j) = X_i^T X_j.$$

In the case of handwritten data the PDA gives two time-series, one for each coordinate or in case of speech data there can be multiple time varying features. The theory developed so far can handle only univariate time-series. In the following we generalize the approach to handle multivariate time-series.

In general consider a vector of functions with L dimensions, $\tilde{F} = [\tilde{F}_1, \dots, \tilde{F}_L]^T$. The inner product of two such functions F and G , each with L dimension

$$\langle F, G \rangle = \sum_{p=1}^L \langle F_p, G_p \rangle.$$

Using this definition one can compute the kernel as,

$$K(i, j) = \sum_{p=1}^L X_{i_p}^T X_{j_p} = X_i^T X_j, \quad (10)$$

where $X_i = [X_{i_1}^T, \dots, X_{i_L}^T]^T$, which is equivalent to concatenating the univariate vectors X_{i_p} .

As in standard SVM procedure one can derive the dual of the formulation in equation (9) and relax the formulation to handle non-separable data by using any positive definite kernel and restricting the α_i 's to be less than a user defined constant $C > 0$ [5].

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(X_i, X_j) - \sum_i \alpha_i \\ \text{subject to} \quad & \sum_i y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, 1 \leq i \leq N. \end{aligned} \quad (11)$$

We have experimented with the Radial Basis kernel, $K(X_i, X_j) = e^{-\gamma \|X_i - X_j\|^2}$.

B. Fast inversion scheme for piecewise linear basis functions

The calculation of vectors X_i , depends on time-series interpolation, which is quadratic in size of the time-series ($O(n^2)$). However, for piecewise linear basis functions, $q = 1$ (see equation (1)), it is possible to do the interpolation in time linear in size of the time-series ($O(n)$). Note that the basis functions can be written as,

$$R_t(s) = \begin{cases} 1+s & 0 \leq s \leq t \\ 1+t & t \leq s \leq 1. \end{cases}$$

The structure of the basis functions can be exploited to obtain a recursive formula for c_k which leads to a fast algorithm for interpolation.

Consider a function f whose values are available at various time instants, s_k , $1 \leq k \leq n$. We are interested in finding c_k such that $f(s) = \sum_k c_k R_{s_k}(s)$. See that $s_k < s_l$ whenever $k < l$ which gives

$$f(s_l) = \sum_{k < l} c_k (1 + s_k) + (1 + s_l) \left(\sum_{k \geq l} c_k \right).$$

Also note that,

$$f(s_{l-1}) = \sum_{k < l-1} c_k (1 + s_k) + (1 + s_{l-1}) \left(\sum_{k \geq l-1} c_k \right).$$

Subtracting one from the other we have,

$$f(s_l) - f(s_{l-1}) = (s_l - s_{l-1}) \sum_{k \geq l} c_k.$$

The coefficients c can be calculated as follows.

$$c_l = \frac{f(s_l) - f(s_{l-1})}{s_l - s_{l-1}} - \sum_{k > l} c_k. \quad (12)$$

The boundary conditions being,

$$c_n = \frac{f(s_n) - f(s_{n-1})}{s_n - s_{n-1}}$$

and

$$c_1 = \frac{f(s_1)}{1 + s_1} - \sum_{k > 1} c_k$$

This sets up a recursive formula for calculating c_k , see that it proceeds backwards starting from n . The worst case time complexity of the above algorithm is $O(n)$ which is again considerably cheaper than the matrix inversion step.

Suitable changes can be made to the algorithm to make it work for higher order interpolation functions ($q > 1$).

C. A comparison with resampled approach

In [11] a two step scheme for interpolation and resampling was used with SVM classification. The classification kernel in such a case is a simple L_2 product of the resampled vector.

$$K_{ij} = (A^{(i)} c^{(i)})^T (A^{(j)} c^{(j)}),$$

where c is the coefficients of basis functions (3), and A is as defined in (6). In the proposed formulation, the kernel obtained after optimization is given by,

$$K_{ij} = X_i^T X_j = (A^{(i)} c^{(i)})^T B^{-1} (A^{(j)} c^{(j)}).$$

Thus if the matrix B is identity, then the two formulations are identical.

In this section, we have proposed a formulation for classifying time-series and discussed several ways in which the formulation can be practically applied to real world problems. One of the strengths of our formulation is that we can employ polynomials of any positive order. It must be noted that as the order increases, the piecewise polynomials attain increasingly complicated shapes. Trade off must be decided between the complexity of the piecewise polynomial and the extent of fitting that is done.

IV. EXPERIMENTS

In this section, the efficacy of the large margin classifier proposed in the previous section is tested on three benchmark datasets from Unipen [13] and the results are compared with other state of the art algorithms such as CS-DTW [2] and Online SNT algorithm [3].

Experiments were conducted on the standard train_r01_v07 package from Unipen, more specifically experiments were conducted on three datasets namely, 1a (Digits), 1b (uppercase alphabets) and 1c (lowercase alphabets). Results from other authors [3] are available on these datasets for comparison. *Pendigits*[14], *Japanese vowels*[15] are other real world datasets on which our experiments were carried out and compared with previously known work. The data is freely available online and hence provides an easy way to benchmark the algorithms.

The data needs to be pre-processed before applying the proposed formulation (11). The first step involves converting the Unipen form data into a time-series containing a series of $[x, y, t]$ for each character. This is achieved by extracting strokes from the data which is available in the data, and obtaining the time-series for each of the strokes. Time series for a character is obtained by concatenating the time-series obtained by individual strokes in the character. Each of the series is further normalized in x, y , such that the character lies in a fixed size box of 50×50 points. The scaling is done such that the aspect ratio of the character was maintained. The character is also translated on both x and y axes such that the minimum value for these axes is zero. The code for preprocessing, training and testing are available for download from our site [16].

The pre-processed data was randomly divided into three sets, with 50% of the data used for training, 16% for validation and the remaining 33% of the data used as the unseen test data set. Classifiers were tuned on the validation set to obtain the best values for the classifier parameters – γ of the Gaussian kernel, m the number of basis vectors in the discriminating function and c the cost parameter of SVM. The classifiers so tuned was used for classification of

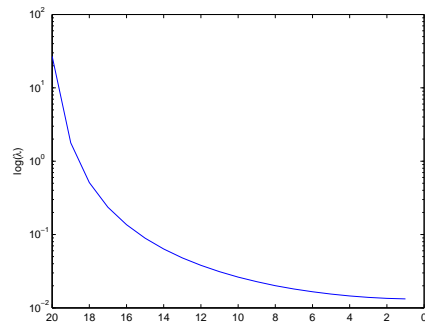


Fig. 2. Comparison of eigen values of G matrix, plotted on a log scale. The first eigen value is an order of magnitude greater than the other eigen values.

the unseen test data. The results of the experiments along with results from previous work are provided in Table I. In general, the number of basis vectors in the discriminating function m was fixed at 20 with m_k 's dividing $[0,1]$ into 19 equal intervals.

In the previous section it was shown that (8) can be solved by standard SVM solvers. The experiments were carried out by adding these custom kernels to the libsvm package[18]. The kernel in equation (10) takes different forms with different order of interpolation polynomials. We have limited our experiments upto second order interpolants ($q = 1, q = 2$ in equation (1)). The experiments can be easily extended to higher order interpolants.

The performance of our kernels is competitive to state of the art methods in all the four benchmark datasets. The interpolation kernels have a time complexity of $O(m)$ and in the above experiments the optimum value of m was 15. Thus interpolation kernels provide us with an efficient algorithm for classifying time-series on handheld devices. The performance of the linear interpolation is slightly better than that of the quadratic interpolation scheme. This may be specific to the online handwriting domain.

A. Low-rank approximation

During the experiments it was observed that the largest eigen value of the G matrix (3) was an order of magnitude larger than the other eigen values. This provides us with a possibility of using a low rank approximation for G while solving equation (3). Using such an approach with a fast Monte Carlo evaluation of the largest eigen value and the corresponding eigen vector will further speed up the algorithm.

V. CONCLUSION

In this paper a large margin based classification scheme is described for classifying multivariate time-series. The scheme proceeds by representing each time-series as a sum of piecewise polynomial basis functions through a kernel interpolation technique. Using the kernel a large margin formulation is developed which is capable of classifying such interpolated functions. The kernels are shown to be

TABLE I

GAUSSIAN INTERPOLATION KERNELS WITH FIRST AND SECOND ORDER INTERPOLATION COMPARED WITH OTHER STATE OF THE ART METHODS.

Data	Approach	Error (%)	$q = 1$	$q = 2$
Unipen dataset (R01/V07) 1a	Interpolation Kernel	2.87	2.87	3.9
Digits (0-9),	CS-DTW [2]	2.9		
	DAG-SVM-GDTW [9]	3.8		
	HMM [17]	3.2		
	OnSNT [3]	1.1		
Unipen dataset (R01/V07) 1b	Interpolation Kernel	8.52	8.52	11.46
Upper case alphabets (A-Z),	CS-DTW [2]	7.2		
	DAG-SVM-GDTW [9]	7.6		
	HMM [17]	6.4		
	OnSNT [3]	4.5		
Unipen dataset (R01/V07) 1c	Interpolation Kernel	10.78	10.78	13.76
Lower case alphabets (a-z),	CS-DTW [2]	9.3		
	DAG-SVM-GDTW [9]	12.1		
	HMM [17]	14.1		
	OnSNT [3]	7.9		
UCI dataset Digits(0-9)	Interpolation Kernel	2.7	2.7	2.8
	GDTW	2.14		
UCI dataset Japanese Vowels	Interpolation Kernel	2.71	2.71	2.98
	5-state HMM	3.8		

competitive in writer independent handwriting recognition on four real world datasets. We also give a fast algorithm for computing the interpolation when piecewise linear basis functions are used. As a by-product the kernel interpolation described can be useful in many other machine learning tasks.

VI. ACKNOWLEDGEMENTS

This work was supported by Department of Science & Technology, Government of India projects DSTO/ECA/CB/0655 and DSTO/ECA/CB/0660

REFERENCES

- [1] H Binsztok and Thierry Artières. Learning HMM structure for on-line handwriting modelization. In *International Workshop on Frontiers in Handwriting Recognition*, pages 407–412, Tokyo, October 2004.
- [2] Claus Bahlmann and Hans Burkhardt. The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):299–310, 2003.
- [3] E. H. Ratzlaff. Methods, report and survey for the comparison of diverse isolated character recognition results on the UNIPEN database. In *International Conference on Document Analysis and Recognition*, pages 623–628, 2003.
- [4] S. Lucas and A. Amiri. Statistical syntactic methods for high-performance OCR. *IEE Proceedings-Vision Image and Signal Processing*, 143(1):23–30, February 1996.
- [5] N. Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York., 2000.
- [6] Guyon I.M. Boser B.E. and Vapnik V. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992.
- [7] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1998.
- [8] Chris Watkins. Dynamic alignment kernels. Technical Report CSD-TR-98-11, Department of Computer Science, University of London, Egham, England, January 1999.
- [9] C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines: A kernel approach. In *International Workshop on Frontiers in Handwriting Recognition*, pages 49–54, 2002.
- [10] L. Rabiner and B-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.
- [11] K. R. Sivaramakrishnan and Chiranjib Bhattacharyya. Time series classification for online tamil handwritten character recognition - A kernel based approach. In *ICONIP*, pages 800–805, 2004.
- [12] R E Moore. *Computational Functional Analysis*. Academic Press, NY, 1985.
- [13] I Guyon, L Schomaker, R Plamondon, M Liberman, and S Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proceedings of International Conference on Pattern Recognition*, pages 29–33, 1994.
- [14] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
- [15] Mineichi Kudo, Jun Toyama, and Masaru Shimbo. Multidimensional curve classification using passing-through regions. *Pattern Recogn. Lett.*, 20(11-13):1103–1111, 1999.
- [16] Software for experiments. Code available at <http://mllab.csa.iisc.ernet.in/users/karthik/handwriting/>.
- [17] Jianying Hu, Sok Gek Lim, and Michael K. Brown. Writer independent on-line handwriting recognition using an HMM approach. *Pattern Recognition*, 33(1):133–147, 2000.
- [18] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.