

Efficient Temporal Kernels between Feature Sets for Time Series Classification

Romain Tavenard, Simon Malinowski, Laetitia Chapel, Adeline Bailly, Heider Sanchez, Benjamin Bustos

▶ **To cite this version:**

Romain Tavenard, Simon Malinowski, Laetitia Chapel, Adeline Bailly, Heider Sanchez, et al.. Efficient Temporal Kernels between Feature Sets for Time Series Classification. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, Sep 2017, Skopje, Macedonia. halshs-01561461

HAL Id: halshs-01561461
https://shs.hal.science/halshs-01561461

Submitted on 12 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Temporal Kernels between Feature Sets for Time Series Classification

Romain Tavenard[1], Simon Malinowski[2], Laetitia Chapel[3], Adeline Bailly[1], Heider Sanchez[4], and Benjamin Bustos[4]

[1] Univ. Rennes 2 / LETG-Rennes COSTEL, IRISA
[2] Univ. Rennes 1 / IRISA
[3] Univ. Bretagne Sud / IRISA
[4] Dept. of Computer Science, Univ. Chile

Abstract. In the time-series classification context, the majority of the most accurate core methods are based on the Bag-of-Words framework, in which sets of local features are first extracted from time series. A dictionary of words is then learned and each time series is finally represented by a histogram of word occurrences. This representation induces a loss of information due to the quantization of features into words as all the time series are represented using the same fixed dictionary. In order to overcome this issue, we introduce in this paper a kernel operating directly on sets of features. Then, we extend it to a time-compliant kernel that allows one to take into account the temporal information. We apply this kernel in the time series classification context. Proposed kernel has a quadratic complexity with the size of input feature sets, which is problematic when dealing with long time series. However, we show that kernel approximation techniques can be used to define a good trade-off between accuracy and complexity. We experimentally demonstrate that the proposed kernel can significantly improve the performance of time series classification algorithms based on Bag-of-Words.

1 Introduction

Time series classification has many real-life applications in various domains such as biology, medicine or speech recognition [17, 24, 27] and has received a large interest over the last decades within the data mining and machine learning communities. Three main families of methods can be found in the literature in this context: similarity-based methods, that make use of similarity measures between raw time series, shapelet-based methods aiming at extracting small subsequences that are discriminant of class membership, and feature-based methods, that rely on a set of feature vectors extracted from time series. The interested reader can refer to [1] for an extensive presentation of time series classification methods.

The most used dissimilarity measures are the euclidean distance (ED) and the dynamic time warping (DTW). The computational cost of ED is lower than the one of DTW, but ED is not able to deal with temporal distortions. The combination of DTW and k-Nearest-Neighbors (k-NN) is one of the seminal approaches for time series classification thanks to its good performance. Cuturi [9]

introduces the Global Alignment Kernel that takes into account all possible alignments in order to produce a reliable similarity measure to be used at the core of standard kernel methods such as Support Vector Machines (SVM).

Ye and Keogh [29] introduce shapelets which are sub-sequences of time series that have a high discriminating power between the different classes. In this framework, classification is done with respect to the presence of absence of such shapelets in tested time series. Hills et al [15] use shapelets to transform the time series into feature vectors representing distances from the time series to the extracted shapelets. Grabocka et al [13] propose a new classification objective function (applied to the shapelet transform) to learn the shapelets, that improves accuracy and reduces the need to search for too many candidates.

Feature-based methods rely on extracting, from each time series, a set of feature vectors that describe it locally. These feature vectors are quantized into words, using a learned dictionary. Every time series is finally represented by a histogram of word occurrences that then feeds a classifier. Many feature-based approaches for time series classification can be found in the literature [2, 3, 4, 19, 25, 26, 27] and they mostly differ in the features they use.

This Bag-of-Word (BoW) framework has been shown to be very efficient. However, it suffers from a major drawback: it implies a quantization step that is done via a fixed partitioning of the feature space. Indeed, for a given dataset, words are obtained by clustering the whole set of features and this fixed clustering might not reflect very accurately the distribution of features for every individual time series. This problem has been studied in the computer vision domain for image retrieval for instance. To overcome this drawback, similarity measures operating directly on feature sets have been considered [5, 6, 16]: every instance is represented by its own raw feature sets, which models more accurately every single distribution of features. These measures have been shown to improve accuracy in the image classification context. However, their associated computational cost is quadratic with the size of the feature sets, which is a strong limitation for their direct use in real-world applications. Moreover, they have never been applied to time series classification purposes to the best of our knowledge.

In this paper, we propose a novel temporal kernel that takes as input a set of feature vectors extracted from the time series and their timestamps. Unlike standard Bag-of-Word approaches, this kernel takes feature localization into account, which leads to significant improvement in accuracy. The distance between feature sets is computed using the signature quadratic form distance (SQFD for short [5]) that is a very powerful tool for feature set comparison but has a high computing cost. We hence introduce an efficient variant of our feature set kernel that relies on kernel approximation techniques.

The rest of this paper is organized as follows. An overview of time series classification approaches based on BoW is given in Section 2, together with alternatives to BoW mainly used in the image community. The Signature Quadratic Form Distance [5] is detailed in Section 3. In Section 4, we introduce a kernel based on this distance together with a temporal variant of the latter kernel that

enables taking temporal information into account. We also propose an approximation scheme that allows efficient computation of both kernels. In the experimental section, we evaluate the proposed kernel using SIFT features adapted to time series [2, 7] and show that it significantly outperforms the original algorithm (based on quantized features) on the UCR datasets [8].

2 Related work

In this section, we first give an overview of state-of-the-art techniques for time series classification that are based on the BoW framework, as the approach proposed in this paper aims at going beyond this framework. We focus on core classifiers, as the one proposed here. Such classifiers can be integrated into ensemble classifiers in order to build more accurate overall classifiers. More information about the use of ensemble classifiers in this context can be found in [1]. Then, we give an insight about similarity measures defined on feature sets for object comparison purposes. Such measures have been widely used in the image community, but never for time series classification to the best of our knowledge.

2.1 Bag-of-Words methods for time series classification

Inspired by the text mining and computer vision communities, recent works in time series classification have considered the use of Bag-of-Words [2, 3, 4, 19, 25, 26, 27]. In a BoW approach, time series are first converted into a histogram of word occurrences and then a classifier is built upon this representation. In the following, we focus on explaining how the conversion of time series into BoW is performed in the literature.

Baydogan et al [4] propose Time Series Bag-of-Features (TSBF), a BoW approach for time series classification where local features such as mean, variance and extrema are extracted on sliding windows. A codebook learned by a class probability estimate distribution is then used to quantize the features into words. In [27], discrete wavelet coefficients are computed on sliding windows and then quantized into words by a k-means algorithm. A similar approach denoted BOSS using quantized Fourier coefficients is proposed in [25]. The SAX representation introduced in [18] can also be used to construct words. Histograms of n-grams of SAX symbols are computed in [19] to form the Bag-of-Patterns (BoP) representation. In [26], they propose the SAX-VSM method, which combines SAX with Vector Space Models. SMTS, a symbolic representation of Multivariate Time Series (MTS) is designed in [3]. This method works as follows: a feature matrix is built from MTS and the rows of this matrix are feature vectors composed of a time index, values and gradients of the time series on all dimensions at this time index. A dictionary of words is computed by giving random samples of this matrix to different decision trees. Xie and Beigi [28] extract keypoints from time series and describe them by scale-invariant features that characterize the shapes around the keypoints. Bailly et al [2] compute multi-scale descriptors based on the SIFT framework and quantize them using a k-means algorithm. Features

are either computed at regular time locations or at specific temporal locations discovered by a saliency detector.

2.2 Alternatives to BoW quantization for feature set classification

Such BoW approaches usually include a quantization step based on a fixed partitioning of the whole set of features. This step might prevent from accurately modeling the distribution of features for every single instance. To overcome this drawback, several similarity measures defined directly on feature sets have been designed. Huttenlocher et al [16] propose the Hausdorff distance that measures the maximum nearest neighbor distance among features of different instances. The Fisher Kernel [22] relies on a parametric estimation of the feature distribution using a Gaussian Mixture Model. Beecks et al [5] propose the Signature Quadratic Form distance (SQFD) that is based on the computation of cross-similarities between the features of different sets called *feature signatures*. As we will see later in this paper, SQFD is closely related to match kernels proposed in [6]. Beecks et al [5] show that SQFD is able to reach higher accuracy than other above classical measures for image retrieval purposes. This makes SQFD a good candidate for being an alternative to BoW quantization in the time series classification context.

3 Signature Quadratic Form Distance

In the following, we assume that a time series $\mathcal{S}$ is represented as a set of n feature vectors $\{x_i \in \mathbb{R}^d\}$. Any algorithm presented in Section 2.1 can be used to extract such a set of feature vectors from the time series and one should note that considered time series may have different number of features extracted. SQFD is a distance that enables the comparison of instances (time series in our case) represented by weighted sets of features called *feature signatures*. In this section, we review the SQFD distance.

The feature signature of an instance is defined as follows:

$$\mathcal{F} = \{(x_i, w_i)\}_{i=1,\dots,n}, \text{ with } \sum_{i=1}^{n} w_i = 1. \tag{1}$$

$\mathcal{F}$ can either be composed of the full set of features $\{x_i\}$, in which case weights $\{w_i\}$ are all set to $1/n$, or by the result of a clustering of the full set of feature vectors from the instance into n clusters. In the latter case, $\{x_i\}$ are the centroids obtained after clustering and $\{w_i\}$ are the weights of the corresponding clusters. We explain here how the SQFD measure is defined (adopting the time series point of view).

Let $\mathcal{F}^1 = \{(x_i^1, w_i^1)\}_{i=1,\dots,n}$ and $\mathcal{F}^2 = \{(x_i^2, w_i^2)\}_{i=1,\dots,m}$ be two feature signatures associated with two time series $\mathcal{S}^1$ and $\mathcal{S}^2$. Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a similarity function defined between feature vectors. The SQFD between $\mathcal{F}^1$ and $\mathcal{F}^2$ is defined as:

$$\text{SQFD}(\mathcal{F}^1, \mathcal{F}^2) = \sqrt{w_{1-2}\, A\, w_{1-2}^T}, \tag{2}$$

where $w_{1-2} = (w_1^1, \ldots, w_n^1, -w_1^2, \ldots, -w_m^2)$ is the concatenation of the weights of $\mathcal{F}^1$ and the opposite of the weight of $\mathcal{F}^2$, and A is a square matrix of dimension $(n+m)$:

$$A = \left(\begin{array}{c|c} A^1 & A^{1,2} \\ \hline A^{2,1} & A^2 \end{array} \right), \tag{3}$$

where A^1 (resp. A^2) is the similarity matrix (computed using k) between features from $\mathcal{F}^1$ (resp. $\mathcal{F}^2$), $A^{1,2}$ is the cross-similarity matrix between features from $\mathcal{F}^1$ and those from $\mathcal{F}^2$, and $A^{2,1}$ is the transpose of $A^{1,2}$. It is shown in [5] that the RBF kernel

$$k_{\mathrm{RBF}}(x_i, x_j) = e^{-\gamma_f \left\| x_i - x_j \right\|^2}, \tag{4}$$

where γ_f is called the kernel bandwidth, is a good choice for computing local similarity between two features.

SQFD is hence the square root of a weighted sum of local similarities between features from sets $\mathcal{F}^1$ and $\mathcal{F}^2$. When no clustering is used, pairwise local similarities are all taken into account, resulting in a very fine grain estimation of the similarity between series that we will refer to as *exact SQFD* in the following. However, its calculation has a high cost, as it requires the computation of a number of local similarities that is quadratic in the size of the sets. A reasonable alternative presented in [5] consists in first quantizing *each* set using a different k-means and then computing SQFD between feature signatures representing the quantized version of the sets. In the following, we will refer to this latter alternative as the k-means approximation of SQFD (SQFD-k-means for short).

4 Efficient temporal kernel between feature sets

In this section, we first derive a kernel from the SQFD distance, considering an RBF kernel as the local similarity, and extend it to a time-sensitive kernel. We then propose a way to alleviate its computational burden.

4.1 Feature set kernel

Let us consider the equal weight case in the SQFD formulation, which is the one considered when no pre-clustering is performed on the feature sets. By expanding Eq. (2) in this specific case, we get:

$$\begin{aligned} \mathrm{SQFD}(\mathcal{F}^1, \mathcal{F}^2)^2 = & \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k_{\mathrm{RBF}}(x_i^1, x_j^1) + \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} k_{\mathrm{RBF}}(x_i^2, x_j^2) \\ & - \frac{2}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathrm{RBF}}(x_i^1, x_j^2). \end{aligned} \tag{5}$$

Note that SQFD then corresponds to a biased estimator of the squared difference between the mean of the samples $\mathcal{F}^1$ and $\mathcal{F}^2$ which is classically used

to test the difference between two distributions [14]. One can also recognize in Eq. (5) the match kernel [6] (also known as set kernel [11]). By denoting K the match kernel associated with k_{RBF} (in what follows, we will always denote with capital letters kernels that operate on sets whereas kernels operating in the feature space will be named with lowercase k), we have:

$$\text{SQFD}(\mathcal{F}^1, \mathcal{F}^2)^2 = K(\mathcal{F}^1, \mathcal{F}^1) + K(\mathcal{F}^2, \mathcal{F}^2) - 2K(\mathcal{F}^1, \mathcal{F}^2). \tag{6}$$

In other words, SQFD is the distance between feature sets embedded in the Reproducing Kernel Hilbert Space (RKHS) associated with K. Finally, we build a feature set kernel, denoted K_{FS} by embedding SQFD into an RBF kernel:

$$K_{\text{FS}}(\mathcal{F}^1, \mathcal{F}^2) = e^{-\gamma_K \, \text{SQFD}(\mathcal{F}^1, \mathcal{F}^2)^2}, \tag{7}$$

where γ_K is the bandwith of K_{FS}. This kernel can then be used at the core of standard kernel methods such as Support Vector Machines (SVM).

4.2 Time-sensitive Feature Set Kernel

Kernel K_{FS} as defined in Eq. (7) ignores the temporal location of the features in the time series, only taking into account cross-similarities between the features. In order to integrate temporal information into k_{RBF}, let us augment the features with the time index at which they are extracted: for all $1 \leq i \leq n$, we denote $x_i^t = (x_i, t_i)$. The time-sensitive kernel between features associated with their time of occurrence, denoted k_{tRBF}, is defined as:

$$k_{\text{tRBF}}(x_i^t, x_j^t) = e^{-\gamma_t \, (t_j - t_i)^2} \cdot k_{\text{RBF}}(x_i, x_j). \tag{8}$$

In practice, t_i and t_j are relative timestamps ranging from 0 (beginning of the considered time series) to 1 (end of the time series) so that features extracted from time series of different lengths can easily be compared. As the product of two positive semi-definite kernels, k_{tRBF} is itself a positive semi-definite kernel. It can be seen as a temporal adaptation of the convolutional kernel for images introduced in [20]. Fig. 1 illustrates the impact of the parameter γ_t of k_{tRBF} on the resulting similarity matrices A for SQFD: k_{RBF} ($\gamma_t = 0$) takes all matches into account without considering their temporal locations, whereas our time-sensitive kernel favors diagonal matches, γ_t controlling the rigidity of this process.

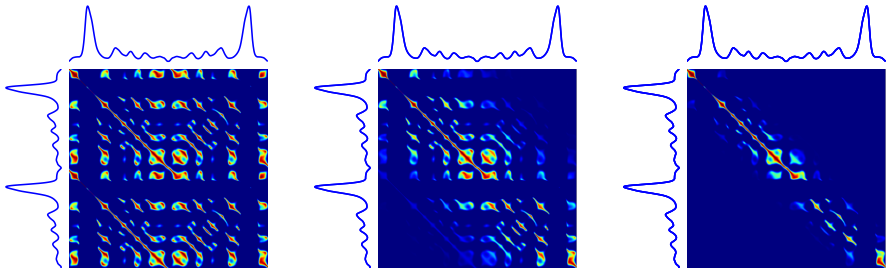
Denoting $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, Eq. (8) can be re-written as:

$$k_{\text{tRBF}}(x_i^t, x_j^t) = e^{-\gamma_f \left(\sum_{l=1}^{d} (x_{jl} - x_{il})^2 + \frac{\gamma_t}{\gamma_f} (t_j - t_i)^2 \right)}, \tag{9}$$

where γ_f is the scale parameter. By defining $g(x_i, t_i) = \left(x_{i1}, \ldots, x_{id}, \sqrt{\frac{\gamma_t}{\gamma_f}}\, t_i \right)$, we get:

$$k_{\text{tRBF}}(x_i^t, x_j^t) = e^{-\gamma_f \, \| g(x_i, t_i) - g(x_j, t_j) \|^2}. \tag{10}$$

In other words, if we build time-sensitive features $g(x, t)$ by concatenating rescaled temporal information and the raw features, the time-sensitive kernel in



(a) $\gamma_t = 0$ (b) Medium γ_t value (c) Large γ_t value

Fig. 1: Impact of the k_{tRBF} kernel on similarity matrices $A^{1,2}$. From left to right, growing γ_t values are used from $\gamma_t = 0$ (*i.e.* the k_{RBF} kernel case) to a large γ_t value that ignores almost all non-diagonal matches. Blue colors indicate low similarity whereas red colors represent high similarity.[1] Best viewed in color.

Eq. (10) can be seen as a standard RBF kernel of scale parameter γ_f operating on these augmented features. Finally, replacing k_{RBF} with k_{tRBF} in Eq. (7) defines a time-sensitive variant for our feature set kernel.

4.3 Temporal kernel normalization

As can be seen in Fig. 1, when γ_t grows, the number of zero entries in A increases, hence the norm of A decreases. It is valuable to normalize the resulting match kernel K so that, if all local kernel responses are equal to 1, the resulting match kernel evaluation is also equal to 1. The corresponding normalization factor is:

$$s^2 = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} \exp^{-\gamma_t \left(\frac{i}{n} - \frac{j}{m} \right)^2} \right)^{-1} . \tag{11}$$

When $\gamma_t = 0$, this is equivalent to the $\frac{1}{n \cdot m}$ normalization term in the match kernels. When $\gamma_t > 0$, this can be computed either through a double for-loop or approximated by its limit when n and m tend towards infinity:

$$\hat{s}^2 = \left(\int_0^1 \int_0^1 e^{-\gamma_t (t_1 - t_2)^2} dt_1 dt_2 \right)^{-1} \tag{12}$$

$$= \left(\sqrt{\frac{\pi}{\gamma_t}} \left[2F\left(\sqrt{2\gamma_t} \right) - 1 \right] + \frac{e^{-\gamma_t} - 1}{\gamma_t} \right)^{-1} \tag{13}$$

where F is the cumulative distribution function of a centered-reduced Gaussian. In practice, we observe that even for very small feature sets ($n = m = 5$),

[1] Matrices $A^{1,2}$ are shown but similar observations hold for matrices A^1, A^2 and $A^{2,1}$.

the relative approximation error done when using $\hat{s}$ normalization instead of s normalization is less than 2%, which is sufficient to efficiently rescale kernels.

4.4 Efficient computation of feature set kernels

As mentioned earlier, exact SQFD computation is demanding as it requires filling the A matrix (Eq. (3)), leading to the evaluation of $(n+m)^2$ local kernels $k_{\mathrm{RBF}}(x_i, x_j)$. To lower the computational cost of kernel K_{FS} (Eq. (7)), two standard approaches can be considered. The first one, presented in Section 3, relies on a k-means quantization of *each* feature set, leading to a time complexity of $O(k^2)$ where k is the chosen number of centroids extracted per set. Another approach is to build an explicit finite-dimensional approximation of the RKHS associated with the match kernel K and approximate SQFD as the Euclidean Distance in this space, as explained below.

Kernel functions compute the inner product between two feature vectors embedded on a feature space thanks to a feature map Φ:

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \,. \tag{14}$$

In the case of an RBF kernel, the associated feature map Φ_{RBF} is infinite dimensional. Let us now assume that one can embed feature vectors in a space of dimension D such that the dot product in this space is a good approximation of the RBF kernel on features. In other words, let us assume that there exists a finite mapping ϕ_{RBF} such that:

$$k_{\mathrm{RBF}}(x_i, x_j) \approx \langle \phi_{\mathrm{RBF}}(x_i), \phi_{\mathrm{RBF}}(x_j) \rangle \,. \tag{15}$$

Then, the match kernel K becomes:

$$K(\mathcal{F}^1, \mathcal{F}^2) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathrm{RBF}}(x_i^1, x_j^2) \tag{16}$$

$$\approx \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} \left\langle \phi_{\mathrm{RBF}}(x_i^1), \phi_{\mathrm{RBF}}(x_j^2) \right\rangle \tag{17}$$

$$\approx \left\langle \underbrace{\frac{1}{n} \sum_{i=1}^{n} \phi_{\mathrm{RBF}}(x_i^1)}_{\phi(\mathcal{F}^1)}, \underbrace{\frac{1}{m} \sum_{j=1}^{m} \phi_{\mathrm{RBF}}(x_j^2)}_{\phi(\mathcal{F}^2)} \right\rangle \tag{18}$$

Hence, approximating k_{RBF} using ϕ_{RBF} is sufficient to approximate the match kernel itself and the explicit feature map ϕ for K is the barycenter of explicit feature maps ϕ_{RBF} for features in the set.

Using Eq. (6), we can derive:

$$\mathrm{SQFD}(\mathcal{F}^1, \mathcal{F}^2)^2 \approx \left\langle \phi(\mathcal{F}^1), \phi(\mathcal{F}^1) \right\rangle + \left\langle \phi(\mathcal{F}^2), \phi(\mathcal{F}^2) \right\rangle - 2 \left\langle \phi(\mathcal{F}^1), \phi(\mathcal{F}^2) \right\rangle \tag{19}$$

$$\approx \|\phi(\mathcal{F}^1) - \phi(\mathcal{F}^2)\|^2 \tag{20}$$

In other words, once feature sets are projected in this finite-dimensional space, approximate SQFD computation is performed through a Euclidean distance computation in $O(D)$ time, where D is the dimension of the feature map.

In this piece of work, we use the Random Fourier Features [23] to build a finite mapping that approximates the local kernel k_{RBF}. Other kernel approximation techniques [10] could be used, but our experience showed that Random Fourier Features reached very good performance for our problem, as showed in the next Section. Random Fourier Features represent each datapoint as its projection on a Fourier basis. The inverse Fourier transform of the RBF kernel is a Gaussian distribution $p(u) = \mathcal{N}(0, \sigma^{-2}I)$ and the Random Fourier Features are obtained by projecting each original feature into a set of sampling Fourier components $p(u)$, before passing through a sinusoid:

$$\phi_{\text{RBF}}(x_i) = \sqrt{\frac{2}{D}} \left[\cos(u_1^T x_i + b_1), \ldots, \cos(u_D^T x_i + b_D)\right]^T \tag{21}$$

where coefficients b_i are drawn from a uniform distribution in $[-\pi, \pi]$.

Overall, building on a kernelized version of SQFD, we have presented a way to incorporate time in the representation as well as a scheme for efficient computation of the kernel. In the following Section, we will show experimentally that these improvements help reaching very competitive performance for a wide range of time-series classification problems.

5 Experimental results with temporal SIFT features

Our feature set kernel (with and without temporal information) can be implemented in any algorithm in lieu of a BoW approach. To evaluate the performances of this kernel, we use it in conjunction with time series-based Scale-Invariant Feature Transform (SIFT) features as it has been demonstrated in [2] that they significantly outperform most state-of-the-art local-feature-based time series classification algorithms applied on UCR datasets. In this section, we first recall the temporal SIFT features. We then evaluate the impact of kernel approximation in terms of both efficiency and accuracy. We also analyze the impact of integrating temporal information in the feature set kernel. Finally, we compare our proposed approach with state-of-the-art time series classifiers.

5.1 Dense extraction of temporal SIFT features

In [2], time series-derived SIFT features are used in a BoW approach for time series classification. We review here how these features are computed (the interested reader can refer to [2] for more details). A time series $\mathcal{S}$ is described by keypoints extracted every τ_{step} time instants. Each keypoint is composed of a set of features that gives a description of the time series at different scales. More formally, let $L(\mathcal{S}, \sigma) = \mathcal{S} * G(t, \sigma)$ be the convolution of a time series with a Gaussian filter $G(t, \sigma) = \frac{1}{\sqrt{2\pi}\ \sigma}\ e^{-t^2/2\sigma^2}$ of width σ. A keypoint at time instant

t is described at scale σ as follows. n_b blocks of size a are selected around the keypoint. At each point of these blocks, the gradient magnitudes of $L(\mathcal{S}, \sigma)$ are computed and weighted so that points that are farther in time from the keypoint have less influence. Then, each block is described by two values: the sum of the positive gradients and the sum of negative gradients in the block. The feature vectors of all the keypoints computed at all scales compose the feature set describing the time series $\mathcal{S}$.

5.2 Experimental setting

For the sake of reproducibility, all presented experiments are conducted on public datasets from the UCR archive [8] and the Python source code used to produce the results (which heavily relies on `sklearn` [21] package) is made available for download[2]. All experiments are run on dense temporal SIFT features extracted using the publicly available software presented in [2]. For SIFT feature extraction, we choose to use fixed parameters for all datasets. Features are extracted at every time instant ($\tau_{\text{step}} = 1$), at all scales, with the block size $a = 4$ and the number of blocks per feature $n_b = 12$, resulting in 24-dimensional feature vectors. By using such a parameter set that achieves robust performance across datasets, we restrict the numbers of parameters to be tuned during cross-validation, without severely degrading performance. Finally, all experiments presented here are repeated 5 times and medians over all runs are reported.

5.3 Impact of kernel approximation

We analyze here the impact of the kernel approximation (in terms of trade-off between complexity and accuracy) on the proposed kernels. Timings are reported for execution on a laptop with 2.9 GHz dual core CPU and 8 GB RAM.

Effectiveness Fig. 2 presents the trade-off between accuracy and execution time obtained for the *ECG200* dataset of the UCR archive. Two methods are considered: SQFD-k-means is the approximation scheme that was proposed in [5] and SQFD-Fourier is the one used in this paper. First, this figure confirms that the assumption made in Eq. (15) is safe: one can obtain good approximation of an RBF kernel using finite dimension mapping. Then, for our approach, we observe that the use of larger dimensions leads to better kernel matrix estimation at the cost of larger execution time. The same applies for SQFD-k-means when varying the k parameter. In order to compare approximation methods, Fig. 2 can be read as follows: for a given MSE level (on the y-axis), the lower the timing, the better. This comparison leads to the conclusion that our proposed approximation scheme reaches better trade-offs for a wide range of MSE values. Note that this behaviour is observed on most of the datasets we have experimented on.

[2] `https://github.com/rtavenar/SQFD-TimeSeries`

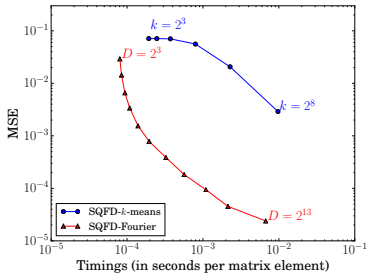
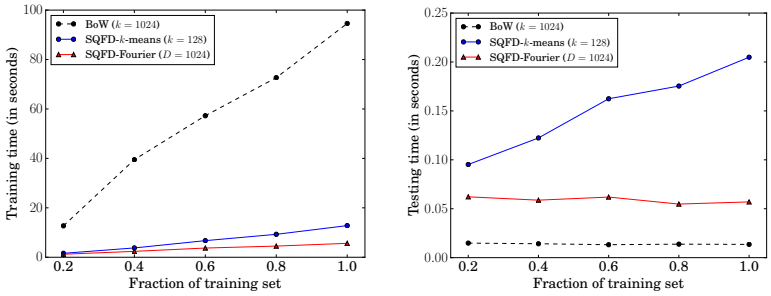


Fig. 2: Mean Squared Error (MSE) *vs* timings of the approximated kernel matrix (*ECG200*). Timings are reported in seconds per matrix element. As a reference, exact computation of feature set kernel takes 0.082 seconds per matrix element.



(a) Training time (b) Testing time

Fig. 3: Training and testing times as a function of the amount of training data (*ECG200*). Training timings correspond to full training of the method for a given parameter set whereas test timings are reported per test time series.

Sensitivity to the amount of training data In this section, we study the evolution of both training and testing times as a function of the amount of training time series. To do so, we compare both efficient approximations of K_{FS} listed above with a standard RBF kernel operating on BoW representations of the feature sets. In Fig. 3, all considered methods exhibit linear dependency between the training set size and the computation time for training. For BoW, this dependency comes from the k-means computation that has $O(nkd)$ time complexity, where n is the number of features used for quantization, k is the number of clusters and d is the feature dimension. The same argument holds for SQFD-k-means, and this explains the observed difference in slope, as lower values of k are typically used in this context. SQFD-Fourier present an even lower slope, which correspond to the computation of projected features (one per time series). Concerning testing times, BoW as well as SQFD-Fourier have almost constant computation needs, whereas SQFD-k-means computation time is linearly dependent in the number of training time series. Note that this comparison is done on *ECG200* dataset for which the number of training time series is small. In this

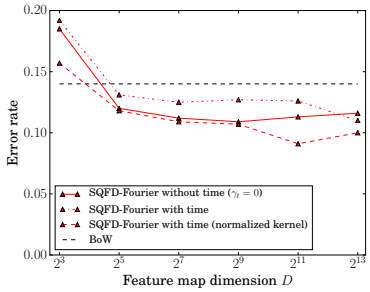


Fig. 4: Error rates as a function of the feature map dimension (*ECG200*).

context, computation of the feature set representation (k-means quantization or feature map) dominates the processing time for both training and testing. In other settings where the number of training time series is large, processing time will be dominated by the computation of pairwise similarities which is, as stated above, linear in k for BoW, quadratic in k for SQFD-k-means, and linear in D for SQFD-Fourier. Once again, our proposed approximation scheme tends to better approximate the exact kernel matrix with lower timings (both in the training and the testing phase) than its competitor.

5.4 Impact of the temporal information

Let us now turn our focus to the impact of temporal information on the classification performance. To do so, we use K_{FS} in an SVM classifier. In order to have a fair comparison of the performances, all parameters (except the dimension D of the feature map) are set through cross-validation on the training set. The same applies for experiments presented in the following subsections and the range of tested parameter values are provided in the Supplementary Material. As a reference, BoW performance with RBF kernel (using the same temporal SIFT features) is also reported. To compute this baseline, the number k of codewords is also cross-validated. Fig. 4 shows the error rates for dataset *ECG200* as a function of the dimension D of the feature map, considering (i) our feature set kernel without temporal information, (ii) its equivalent with temporal information and finally (iii) the normalized temporal kernel. One should first notice that in all cases, a higher dimension D tends to lead to better performance. This figure also illustrates the importance of the temporal kernel normalization: the normalized temporal kernel reaches better performance than the feature set kernel with no time information, whereas the performance of the non-normalized one is worse. Indeed, when γ_t increases, the non-normalized version suffers from a bad scaling of kernel responses that impairs the learning process of the SVM.[3]

In the following, we use the same parameter ranges as above, and we cross-validate the parameter related to the time ($\gamma_t \in \{0\} \cup \{10^0 - 10^6\}$). By doing so,

[3] See Supplementary material for experiments on more datasets.

we offer the possibility for our method to learn (during training) whether time information is of interest or not for a given dataset. We present experiments run on the 85 datasets from the UCR Time Series Classification archive and observe that in more than 3/4 of our experiments, the temporal variant of our kernel is selected by cross-validation (*i.e.* $\gamma_t > 0$), which confirms the superiority of temporal kernels for such applications. Corresponding datasets are marked with a star in the full result table provided as Supplementary material.

5.5 Pairwise comparisons on UCR datasets

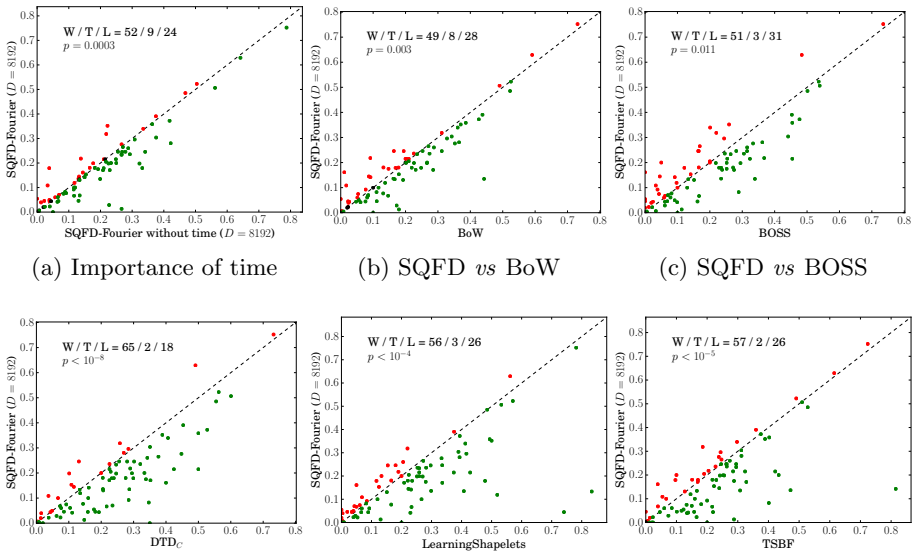
Pairwise comparisons of methods are presented in Fig. 5, in which green dots (data points lying below the diagonal) correspond to cases where the x-axis method has higher classification error rates, black ones represent cases where both methods share the same performance and red dots stand for cases for which the y-axis method has higher error rates. In these plots, Win / Tie / Lose scores are also reported where "Win" indicates the number of times the y-axis method outperforms the x-axis one. Finally, p-values corresponding to one-sided Wilcoxon signed rank tests are provided to assess statistical significance of observed differences and our significance level is set to 5%.

First, Fig. 5a confirms our observation made for *ECG200* dataset: incorporating temporal information allows, for many datasets, to improve the classification performances. We then compare the efficient version of our temporal kernel on feature sets with standard BoW approach running on the same feature sets (Fig. 5b). This figure shows improvement for a wide range of datasets: when testing the statistical differences between both methods, one can observe that our feature set kernel significantly outperforms the BoW approach. Finally, when considering other state-of-the-art competitors[4], our feature set kernel for time series show state-of-the-art performance, significantly outperforming BOSS [25], DTD_C [12], LearningShapelets (LS) [13] and TSBF [4]. This high accuracy is achieved with reasonable classification time (*e.g.* 300ms per test time series on NonInvasiveFetalECG1 dataset, one of the largest UCR datasets).

6 Conclusion

Many local features have been designed for time-series representations and used in a BoW framework for classification purposes. To improve these approaches, we introduce in this paper a new temporal kernel between feature sets that gets rid of quantized representations. More precisely, we propose to kernelize SQFD for time-series classification purposes. We also derive a temporal feature set kernel, allowing one to take into account the time instant at which the features are extracted. In order to alleviate the high computational burden of this kernel and make it tractable for large datasets, we propose an approximation technique that allows fast computation of the kernel. Extensive experiments show

[4] For the sake of brevity, we focus on standalone classifiers that are shown in [1] to outperform competitors in their categories.



(a) Importance of time (b) SQFD *vs* BoW (c) SQFD *vs* BOSS

(d) SQFD *vs* DTD_C (e) SQFD *vs* LS (f) SQFD *vs* TSBF

Fig. 5: Pairwise performance comparisons. Reported values are error rates.

that the temporal information helps improving classification accuracy. The temporal information is taken into account thanks to a simple RBF kernel, and we believe that the performance could be further improved by exploring other ways to incorporate time information into the local kernels. This is the main direction for our future work. Finally, our temporal feature set kernel significantly outperforms the initial BoW-based method and leads to competitive results *w.r.t* state-of-the-art time series classification algorithms. This kernel is likely to improve performance of any time series classification approach based on BoW.

Acknowledgments

Supported by the Millennium Nucleus Center for Semantic Web Research under Grant NC120004, the ANR through the ASTERIX project (ANR-13-JS02-0005-01), École des docteurs de l'UBL as well as by the Brittany Region.

Bibliography

[1] Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery pp. 1–55 (2016)

[2] Bailly, A., Malinowski, S., Tavenard, R., Chapel, L., Guyet, T.: Dense Bag-of-Temporal-SIFT-Words for Time Series Classification. Lecture Notes in Artificial Intelligence 9785, 17–30 (2016)

[3] Baydogan, M.G., Runger, G.: Learning a symbolic representation for multivariate time series classification. Data Mining and Knowledge Discovery 29(2), 400–422 (2015)

[4] Baydogan, M.G., Runger, G., Tuv, E.: A Bag-of-Features Framework to Classify Time Series. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11), 2796–2802 (2013)

[5] Beecks, C., Uysal, M.S., Seidl, T.: Signature quadratic form distance. In: Proceedings of the ACM International Conference on Image and Video Retrieval. pp. 438–445 (2010)

[6] Bo, L., Sminchisescu, C.: Efficient match kernel between sets of features for visual recognition. In: Advances in Neural Information Processing Systems 22, pp. 135–143 (2009)

[7] Candan, K.S., Rossini, R., Sapino, M.L.: sDTW: Computing DTW Distances using Locally Relevant Constraints based on Salient Feature Alignments. In: Proceedings of the International Conference on Very Large DataBases. vol. 5, pp. 1519–1530 (2012)

[8] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (2015), `www.cs.ucr.edu/~eamonn/time_series_data/`

[9] Cuturi, M.: Fast global alignment kernels. In: Proceedings of the International Conference on Machine Learning. pp. 929–936 (2011)

[10] Drineas, P., Mahoney, M.W.: On the nyström method for approximating a gram matrix for improved kernel-based learning. The Journal of Machine Learning Research 6, 2153–2175 (2005)

[11] Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Proceedings of the International Conference on Machine Learning (2002)

[12] Górecki, T., Łuczak, M.: Non-isometric transforms in time series classification using dtw. Knowledge-Based Systems 61, 98–108 (2014)

[13] Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning time-series shapelets. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 392–401 (2014)

[14] Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in neural information processing systems. pp. 513–520 (2006)

[15] Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery 28(4), 851–881 (2014)
[16] Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(9), 850–863 (Sep 1993)
[17] Le Cun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. In: The Handbook of Brain Theory and Neural Networks. vol. 3361, pp. 255–258 (1995)
[18] Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. pp. 2–11 (2003)
[19] Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. International Journal of Information Systems 39, 287–315 (2012)
[20] Mairal, J., Koniusz, P., Harchaoui, Z., Schmid, C.: Convolutional kernel networks. In: Advances in Neural Information Processing Systems. pp. 2627–2635 (2014)
[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
[22] Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007)
[23] Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in neural information processing systems. pp. 1177–1184 (2007)
[24] Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26(1), 43–49 (1978)
[25] Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery 29(6), 1505–1530 (2014)
[26] Senin, P., Malinchik, S.: SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. Proceedings of the IEEE International Conference on Data Mining pp. 1175–1180 (2013)
[27] Wang, J., Liu, P., F.H. She, M., Nahavandi, S., Kouzani, A.: Bag-of-Words Representation for Biomedical Time Series Classification. Biomedical Signal Processing and Control 8(6), 634–644 (2013)
[28] Xie, J., Beigi, M.: A Scale-Invariant Local Descriptor for Event Recognition in 1D Sensor Signals. In: Proceedings of the IEEE International Conference on Multimedia and Expo. pp. 1226–1229 (2009)
[29] Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 947–956 (2009)