



HAL
open science

Dense Bag-of-Temporal-SIFT-Words for Time Series Classification

Adeline Bailly, Simon Malinowski, Romain Tavenard, Thomas Guyet, Laetitia Chapel

► **To cite this version:**

Adeline Bailly, Simon Malinowski, Romain Tavenard, Thomas Guyet, Laetitia Chapel. Dense Bag-of-Temporal-SIFT-Words for Time Series Classification. Lecture Notes in Artificial Intelligence, 2016. hal-01252726v3

HAL Id: hal-01252726

<https://hal.science/hal-01252726v3>

Submitted on 24 May 2016 (v3), last revised 25 May 2016 (v4)

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.

Advanced Bag-of-Temporal-SIFT-Words for Time Series Classification

Adeline Bailly¹, Simon Malinowski², Romain Tavenard¹,
Thomas Guyet³, and Lætitia Chapel⁴

¹ Université de Rennes 2, IRISA, LETG-Rennes COSTEL, Rennes, France

² Université de Rennes 1, IRISA, Rennes, France

³ Agrocampus Ouest, IRISA, Rennes, France

⁴ Université de Bretagne Sud, Vannes ; IRISA, Rennes, France

Abstract. Time series classification is an application of particular interest with the increase of data to monitor. Classical techniques for time series classification rely on point-to-point distances. Recently, Bag-of-Words approaches have been used in this context. Words are quantized versions of simple features extracted from sliding windows. The SIFT framework has proved efficient for image classification. In this paper, we design a time series classification scheme that builds on the SIFT framework adapted to time series to feed a Bag-of-Words. We then refine our method by studying the impact of normalized Bag-of-Words, as well as densely extract point descriptors. Proposed adjustments achieve better performance. The evaluation shows that our method outperforms classical techniques in terms of classification.

Keywords: time series classification, Bag-of-Words, SIFT, BoTSW

1 Introduction

Classification of time series has received an important amount of interest over the past years due to many real-life applications, such as environmental modeling, speech recognition. A wide range of algorithms have been proposed to solve this problem. One simple classifier is the k -nearest-neighbor (k NN), which is usually combined with Euclidean Distance (ED) or Dynamic Time Warping (DTW) [13]. Such techniques compute similarity between time series based on point-to-point comparisons, which is often not appropriate. Classification techniques based on higher level structures are most of the time faster, while being at least as accurate as DTW-based classifiers. Hence, various works have investigated the extraction of local and global features in time series. Among these works, the Bag-of-Words (BoW) approach (also called bag-of-features) has been considered for time series classification. BoW is a very common technique in text mining, information retrieval and content-based image retrieval because of its simplicity and performance. For these reasons, it has been adapted to time series data in some recent works [2, 3, 11, 14, 17]. Different kinds of features based on simple statistics have been used to create the words.

In the context of image retrieval and classification, scale-invariant descriptors have proved their efficiency. Particularly, the Scale-Invariant Feature Transform (SIFT) framework has led to widely used descriptors [12]. These descriptors are scale and rotation invariant while being robust to noise. We build on this framework to design a BoW approach for time series classification where the words correspond to the description of local gradients around keypoints, that are first extracted from the time series. This approach can be seen as an adaptation of the SIFT framework to time series.

This paper is organized as follows. Section 2 summarizes related work, Section 3 describes the proposed Bag-of-Temporal-SIFT-Words (BoTSW) method, and Section 4 reports experimental results. Finally, Section 5 concludes and discusses future work.

2 Related work

Our approach for time series classification builds on two well-known methods in computer vision: local features are extracted from time series using a SIFT-based approach and a global representation of time series is built using Bag-of-Words. This section first introduces state-of-the-art methods in time series classification, then presents standard approaches for extracting features in the image classification context and finally lists previous works that make use of such approaches for time series classification.

Data mining community has, for long, investigated the field of time series classification. Early works focus on the use of dedicated metrics to assess similarity between time series. In [13], Ratanamahatana and Keogh compare Dynamic Time Warping to Euclidean Distance when used with a simple k NN classifier. While the former benefits from its robustness to temporal distortions to achieve high efficiency, ED is known to have much lower computational cost. Cuturi [5] shows that DTW fails at precisely quantifying dissimilarity between non-matching sequences. He introduces Global Alignment Kernel that takes into account all possible alignments to produce a reliable dissimilarity metric to be used with kernel methods such as Support Vector Machines (SVM). Douzal and Amblard [6] investigate the use of time series metrics for classification trees.

So as to efficiently classify images, those first have to be described accurately. Both local and global descriptions have been proposed by the computer vision community. For long, the most powerful local feature for images was SIFT [12] that describes detected keypoints in the image using the gradients in the regions surrounding those points. Building on this, Sivic and Zisserman [15] suggested to compare video frames using standard text mining approaches in which documents are represented by word histograms, known as Bag-of-Words (BoW). To do so, authors map the 128-dimensional space of SIFT features to a codebook of few thousand words using vector quantization. VLAD (Vector of Locally Aggregated Descriptors) [8] are global features that build upon local ones in the same spirit as BoW. Instead of storing counts for each word in the dictionary, VLAD preserves residuals to build a fine-grain global representation.

Inspired by text mining, information retrieval and computer vision communities, recent works have investigated the use of Bag-of-Words for time series classification [2, 3, 11, 14, 17]. These works are based on two main operations : converting time series into Bag-of-Words (a histogram representing the occurrence of words), and building a classifier upon this BoW representation. Usually, classical techniques are used for the classification step: random forests, SVM, neural networks, k NN. In the following, we focus on explaining how the conversion of time series into BoW is performed in the literature. In [3], local features such as mean, variance, extremum values are computed on sliding windows. These features are then quantized into words using a codebook learned by a class probability estimate distribution. In [17], discrete wavelet coefficients are extracted on sliding windows and then quantized into words using k -means. In [11, 14], words are constructed using the SAX representation [10] of time series. SAX symbols are extracted from time series and histograms of n -grams of these symbols are computed. In [2], multivariate time series are transformed into a feature matrix, whose rows are feature vectors containing a time index, the values and the gradient of time series at this time index (on all dimensions). Random samples of this matrix are given to decision trees whose leaves are seen as words. A histogram of words is output when the different trees are learned. Rather than computing features on sliding windows, authors of [18] first extract keypoints from time series. These keypoints are selected using the Differences-of-Gaussians (DoG) framework, well-known in the image community, that can be adapted to one-dimensional signals. Keypoints are then described by scale-invariant features that describe the shapes of the extremum surrounding keypoints. In [4], extraction and description of time series keypoints in a SIFT-like framework is used to reduce the complexity of Dynamic Time Warping: features are used to match anchor points from two different time series and prune the search space when finding the optimal path in the DTW computation.

In this paper, we design a time series classification technique based on the extraction and the description of keypoints using a SIFT framework adapted to time series. The description of keypoints is quantized using a k -means algorithm to create a codebook of words and classification of time series is performed with a linear SVM fed with normalized histograms of words.

3 Bag-of-Temporal-SIFT-Words (BoTSW) method

The proposed method is adapted from the SIFT framework [12] widely used for image classification. It is based on three main steps : (i) detection of keypoints in time series, (ii) description of these keypoints by gradient magnitude at a specific scale, and (iii) representation of time series by a BoW, words corresponding to quantized version of the description of keypoints. These steps are depicted in Fig. 1 and detailed below.

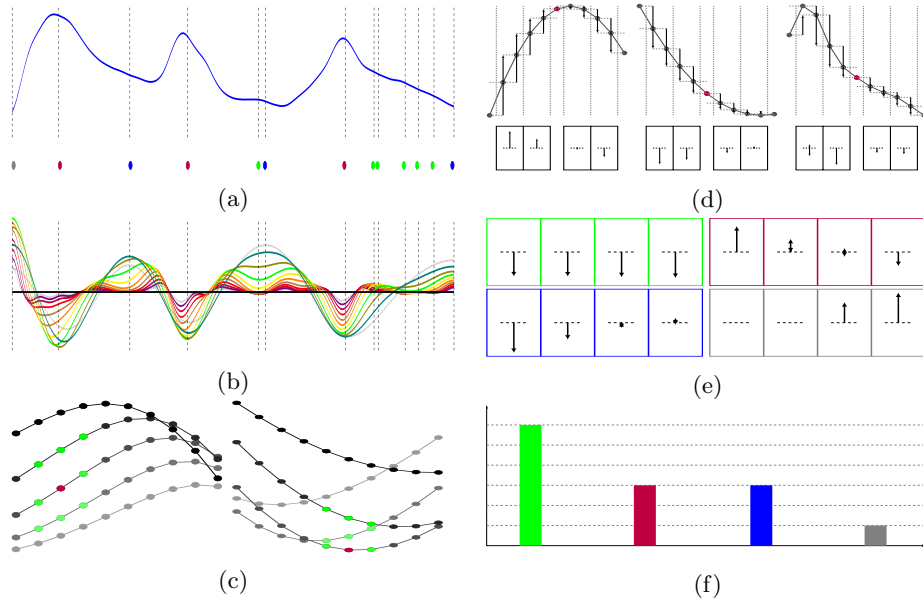


Fig. 1: Approach overview : (a) A time series and its extracted keypoints (scale-space extrema detection), (b) The Difference-of-Gaussians, computed at different scales, on which the keypoint extraction is built, (c) In red the tested point, in green the neighbors. On the right, a scale-space extrema, (d) Keypoint description is based on the time series filtered at the scale at which the keypoint is extracted. Descriptors are quantized into words (feature vectors). (e) Codewords obtained via k -means (where $k = 4$), the color is associated with the dots under each keypoints in (a). (f) These histograms are given to a classifier (linear SVM here) that learns boundaries between the different classes. Best viewed in color.

3.1 Detection of keypoints in time series

The first step of our method is to select keypoints in time series. Two approaches are proposed, one based on scale-space extrema detection and another one based on dense extraction.

Scale-space extrema detection. Following the SIFT framework, keypoints in time series correspond to local extrema both in terms of scale and location. These scale-space extrema are identified using a DoG function, which establishes a list of scale-invariant keypoints. Let $L(t, \sigma)$ be the convolution ($*$) of a Gaussian function $G(t, \sigma)$ of width σ with a time series $S(t)$:

$$L(t, \sigma) = G(t, \sigma) * S(t).$$

DoG is obtained by subtracting two filtered time series at consecutive scales:

$$D(t, \sigma) = L(t, k_{sc}\sigma) - L(t, \sigma),$$

where k_{sc} controls the scale ratio between two consecutive scales. A keypoint is detected at time index t and scale j if it corresponds to an extremum of $D(t, k_{sc}^j \sigma)$ in both time and scale (8 neighbors : 2 at the same scale, and 6 in adjacent scales) If a point is higher (or lower) than all of its neighbors, it is considered as an extremum in the scale-space domain and hence a keypoint of S .

Dense extraction. Previous researches have shown that efficient classification can be achieved by using densely extracted local descriptors features [16]. Dense sampled extracts keypoints at regular locations. Counter to scale-space extrema detection, keypoints detected with dense extraction do not represent extremum in the scale-space domain but are extracted with a regular step between each one of them.

3.2 Description of the extracted keypoints

Next step in our process is the description of keypoints. A keypoint at (t, j) is described by gradient magnitudes of $L(\cdot, k_{sc}^j \sigma)$ around t . n_b blocks of size a are selected around the keypoint. Gradients are computed at each point of each block and weighted using a Gaussian window of standard deviation $\frac{a \times n_b}{2}$ so that points that are farther in time from the detected keypoint have lower influence. Then, each block is described by storing separately the sums of magnitude of positive and negative gradients. Resulting feature vector is of dimension $2 \times n_b$.

Bag-of-Words normalization. We will consider the basic description as our baseline (as in [1]). Advanced representations will include normalizations of Bag-of-Words. We introduce 3 normalizations adapted to BoW in order to (possibly) improve classification rate. Changes resulting from BoW normalization will be study in section 4.

1. The first normalization is the ℓ_2 -normalization i.e. $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$.
2. The second normalization corresponds to the Term-Frequency and Inverse Document Frequency normalization (TF.IDF). BoW weights are adapted to increase the influence of specific words and to decrease the influence of more common words.
3. The Signed Square Root normalization (SSR) introduced by Jegou and Chum [7] will be the last normalization tested. SSR-normalization can be written as $x_i \xrightarrow{SSR} \text{sign}(x_i) \sqrt{|x_i|}$

3.3 Bag-of-Temporal-SIFT-Words for time series classification

Features are then quantized using a k -means algorithm to obtain a codebook of k words. Words represent different kinds of local behavior in the time series. For a given time series, each feature vector is assigned to the closest word of the codebook. The number of occurrences of each word in a time series is computed. The BoTSW representation of a time series is the normalized histogram (*i.e.* frequency vector) of word occurrences. These histograms are then passed to a classifier to learn how to discriminate classes from this BoTSW description.

4 Experiments and results

4.1 Experimental setup

In this section, we investigate the impact of both the dense extraction of the key-points and the normalization of the Bag-of-Words on classification error rates. Experiments are conducted on 86 datasets from the UCR repository [9]. Parameters a , n_b , k and C_{SVM} of BoTSW are learn, we set remaining parameters as follows: $\sigma = 1.6$, $k_{sc} = 2^{1/3}$. These values have shown to produce stable results. Parameters a , n_b , k and C_{SVM} vary inside the following sets : $\{4, 8\}$, $\{4, 8, 12, 16, 20\}$, $\{2^i, \forall i \in \{5..10\}\}$ and $\{1, 10, 100\}$ respectively. Codebooks are obtained *via* k -means quantization and a linear SVM is used to classify times series represented as BoTSW. Each dataset is composed of a train and a test set. For our approach, the best sets (considering the accuracy) of (a, n_b, k, C_{SVM}) parameters are selected by performing a leave-one-out cross-validation on the train set. These best sets of parameters are then used to build the classifier on the train set and evaluate it on the test set. Experimental error rates (ER) are reported in Table 1, together with baseline scores publicly available at [9].

4.2 Experiments on BoW normalization

Table 2 allows us to analyze the impact of different types of BoW normalization on the classification rate. Considering the normalization as a parameter, we learn the best sets of parameters for each dataset for each one of the couple {experimental setup detailed in 4.1 + normalization}, *i.e.* sets of best parameters for ℓ_2 -normalization and TF.IDF-normalization with the same experimental setup can be dissimilar.

Several observations can be made. First, we can observe that if we normalize the Bag-of-Words with ℓ_2 -normalization or with TF.IDF-normalization, our classification rates are lower than without normalization for more than half of the tested datasets. If we associate SSR-normalization with TF.IDF-normalization, classification rates are comparable than without normalization. Finally, using exclusively SSR-normalized BoTSW provides better classification rates than without normalization for most of the UCR datasets.

Fig. 2 enables us to compare error rates obtain with unnormalized BoTSW and with BoTSW associated with SSR-normalization. Basic BoTSW has a better

Dataset	EDNN	DTWNN	BoTSW	Dataset	EDNN	DTWNN	BoTSW
50words	0.369	0.310	-1	MiddlePhalanx	0.247	0.352	0.413
Adiac	0.389	0.396	0.076	OutlineCorrect			
ArrowHead	0.200	0.297	0.188	MiddlePhalanxTW	0.439	0.416	0.416
Beef	0.333	0.367	0.3	MoteStrain	0.121	0.165	0.146
BeetleFly	0.250	0.300	0.1	NonInvasiveFetal			
BirdChicken	0.450	0.250	0.15	ECG_Thorax1	0.171	0.209	-1
Car	0.267	0.267	0.15	NonInvasiveFetal			
CBF	0.148	0.003	0.000	ECG_Thorax2	0.120	0.135	-1
Chlorine				OliveOil	0.133	0.167	0.133
Concentration	0.35	0.352	-1	OSULeaf	0.479	0.409	0.198
CinC_ECG				PhalangesOutlines			
_torso	0.103	0.349	0.320	Correct	0.239	0.272	-1
Coffee	0.000	0.000	0.035	Phoneme	0.891	0.772	0.785
Computers	0.424	0.300	0.404	Plane	0.038	0.000	0.009
Cricket_X	0.423	0.246	-1	ProxiamlPhalanx			
Cricket_Y	0.433	0.256	-1	OutlineAgeGroup	0.215	0.195	-1
Cricket_Z	0.413	0.246	-1	ProxiamlPhalanx			
DiatomSize				OutlineCorrect	0.192	0.216	-1
Reduction	0.065	0.033	0.091	ProximalPhalanxTW	0.292	0.263	0.24
DistalPhalanx				RefrigerationDevices	0.605	0.536	0.554
OutlineAgeGroup	0.218	0.208	0.35	ScreenType	0.640	0.603	-1
DistalPhalanx				ShapeletSim	0.461	0.350	0.072
OutlineCorrect	0.248	0.232	0.231	ShapesAll	0.248	0.232	-1
DistalPhalanxTW	0.273	0.290	0.272	SmallKitchen			
Earthquakes	0.326	0.258	0.211	Appliances	0.659	0.357	-1
ECG200	0.120	0.230	0.1	SonyAIBORobot			
ECG5000	0.075	0.076	-1	Surface	0.141	0.169	0.083
ECGFiveDays	0.203	0.232	0.041	SonyAIBORobot			
ElectricDevices	0.450	0.399	-1	SurfaceII	0.305	0.275	0.151
FaceAll	0.286	0.192	0.239	StarLightCurves	0.151	0.093	-1
FaceFour	0.216	0.170	0.034	Strawberry	0.062	0.060	0.123
FacesUCR	0.231	0.095	0.212	SwedishLeaf	0.211	0.208	0.128
FISH	0.217	0.177	0.051	Symbols	0.100	0.050	0.016
FordA	0.341	0.438	-1	synthetic_control	0.120	0.007	0.013
FordB	0.442	0.406	-1	ToeSegmentation1	0.320	0.228	0.043
Gun_Point	0.087	0.093	0.026	ToeSegmentation2	0.192	0.162	0.115
Ham	0.400	0.533	0.438	Trace	0.240	0.000	0.000
HandOutlines	0.199	0.202	0.257	Two_Patterns	0.090	0.000	0.095
Haptics	0.630	0.623	0.636	TwoLeadECG	0.253	0.096	0.023
Herring	0.484	0.469	0.406	uWaveGesture			
InlineSkate	0.658	0.616	0.658	Library_X	0.261	0.273	-1
Insect				uWaveGesture			
WingbeatSound	0.438	0.645	0.519	Library_Y	0.338	0.366	-1
ItalyPowerDemand	0.045	0.05	0.068	uWaveGesture			
LargeKitchen				Library_Z	0.35	0.342	-1
Appliances	0.507	0.205	0.288	uWaveGesture			
Lightning2	0.246	0.131	0.278	LibraryAll	0.052	0.108	-1
Lightning7	0.425	0.274	0.342	LibraryAll			
MALLAT	0.086	0.066	0.115	wafer	0.005	0.020	0.033
Meat	0.067	0.067	0.133	Wine	0.389	0.426	0.518
MedicalImages	0.316	0.263	-1	WordsSynonyms	0.382	0.351	0.250
MiddlePhalanx				WordSynonyms	0.382	0.351	0.373
OutlineAgeGroup	0.260	0.250	0.275	Worms	0.635	0.536	0.464
				WormsTwoClass	0.414	0.337	0.281
				yoga	0.170	0.164	0.304

Table 1: Classification error rates (best performance is written as bold text).

•	Basic			ℓ_2			TF.IDF			SSR			SSR+TF.IDF		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
Basic	-			24	8	15	24	6	17	11	13	23	20	7	20
ℓ_2	15	8	24	-			22	9	17	14	8	26	23	10	14
TF.IDF	17	6	24	17	9	22	-			12	5	31	13	14	20
SSR	23	13	11	26	8	14	31	5	12	-			29	7	11
SSR+TF.IDF	20	7	20	14	10	23	20	14	13	11	7	29	-		

Table 2: Win-Tie-Lose (WTL) scores comparing BoTSW with unnormalized BoW and normalized BoW. Unnormalized (Basic) BoTSW achieves a better classification rate than ℓ_2 -normalized BoTSW on 24 datasets, a similar performance on 8 datasets and does worse on 15 datasets.

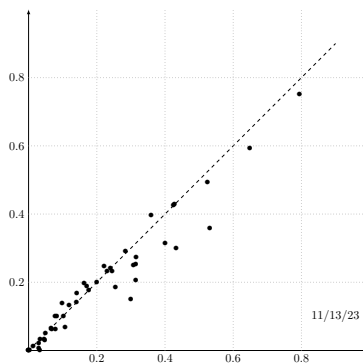


Fig. 2: Error rate of unnormalized BoTSW compared to BoTSW associated with SSR-normalization. Unnormalized BoTSW has a better classification than BoTSW + SSR-normalization on 11 datasets, an equivalent classification rate on 13 datasets and does worse on 23 datasets.

classification rate than BoTSW with SSR-normalization when a point is situated above the diagonal. Furthermore, the farthest a point is from the diagonal, the most significant the difference is. From Table 2, we already know that the Win-Tie-Lose score between unnormalized BoTSW and SSR-normalized BoTSW is 11/13/23. As it can be seen in Fig. 2, BoTSW associated with SSR-normalization leads to comparable (close to the diagonal) or better classification rates than unnormalized BoTSW. SSR-normalized BoTSW produces for example an 0.359 error rate whereas unnormalized BoTSW obtains a 0.531 error rate for Herring dataset.

The Win-Tie-Lose scores coupled with Fig. 2 show that amongst all normalizations, SSR-normalization of BoTSW should be the one to be privileged.

•	Basic			Dense			$\ell_2 +$ Dense			SSR+ Dense			SSR + TF.IDF + Dense		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
Basic	-			11	8	26	8	5	32	15	4	21	10	4	26
Dense	26	8	11	-			18	9	19	19	8	13	11	12	17
$\ell_2 +$ Dense	32	5	8	19	9	18	-			20	6	14	12	11	17
SSR + Dense	21	4	15	13	8	19	14	6	20	-			12	10	18
SSR+TF.IDF + Dense	26	4	10	17	12	11	17	11	12	18	10	12	-		

Table 3: Win-Tie-Lose (WTL) scores comparing BoTSW {unnormalized BoW associated with scale-space extrema detection} and {(un)normalized BoW associated with dense extraction}. The scale-space extrema detection achieves a better classification rate than dense extraction on 11 datasets, a similar performance on 8 datasets and does worse on 26 datasets.

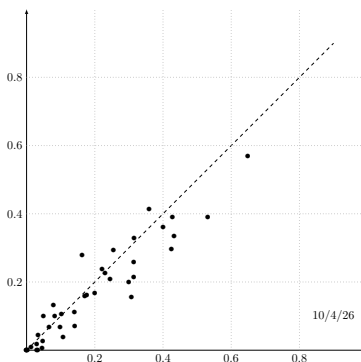


Fig. 3: Error rate of basic BoTSW compared to Dense BoTSW associated with SSR / TF.IDF normalizations. Basic BoTSW has a better classification than Dense BoTSW + SSR / TF.IDF-normalization on 11 datasets, an equivalent classification rate on 8 datasets and does worse on 26 datasets.

4.3 Experiments on Dense extraction

As it can be seen in Table 3, BoTSW associated with dense extraction achieves generally better classification rates than BoTSW with scale-space extrema detection. If we associate Bag-of-Words normalization with dense extraction, it also improved the performance for most datasets.

The best classification rates are obtained when we associate dense extraction with both SSR and TF.IDF normalizations. As it can easily be seen in Fig. 3,

all the plotted points (which correspond to error rates) are near or below the diagonal. Consequently, dense BoTSW associated with SSR and TF.IDF normalizations leads to similar or better classification rates than unnormalized BoTSW.

Amongst all dense extractions, SSR and TF.IDF normalized dense BoTSW achieves the best classification.

4.4 Conclusion on BoTSW

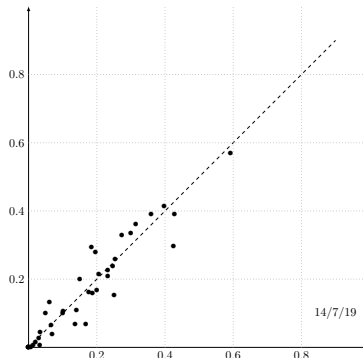


Fig. 4: Error rate of SSR-normalized BoTSW compared to Dense BoTSW associated with SSR / TF.IDF normalizations.

In sections 4.2 and 4.3, we notice that Bag-of-Words normalization and dense extraction of keypoints improve the performance. Prior to compare our method to state-of-the-art methods, we have to determine which association has the best performance. We previously reduce the possibilities to SSR-normalized BoTSW (*A-BoTSW*) and SSR / TF.IDF normalized dense BoTSW (*B-BoTSW*).

Fig. 4 shows that *A-BoTSW* has a better classification than *B-BoTSW* on 14 datasets, an equal classification rate on 7 datasets and does worse on 19 datasets. Error rates of *A-BoTSW* and *B-BoTSW* are alike, both associations are therefore equivalent. Error rates obtained with dense BoTSW associated with SSR/TF.IDF normalizations will be use in the next section.

4.5 Results

In this section, we refer to dense BoTSW associated with SSR/TF.IDF normalizations as BoTSW. We computes the Win-Tie-Lose scores between BoTSW and several state-of-the-art methods. The UCR repository provides error rates for the 86 datasets with Euclidean distance (ED) and Dynamic Time Warping (DTW). We use available error rates for TSBF (45), SAX-VSM (51), SMTS (45) and BoP (20).

Win-Tie-Lose scores from Table 4 shows that BoTSW is better than ED (+1NN), DTW (+1NN), TSBF [3], SAX-VSM [14], SMTS [2] and BoP [11] separately. Experiments show that BoTSW outperforms performance of state-of-the-art methods.

•	ED+			DTW+			TSBF			SAX-			SMTS			BoP		
	1NN			1NN						VSM								
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
BoTSW+	41	3	8	39	4	9	22	0	5	21	3	5	17	1	9	9	2	2
SVM																		

Table 4: Win-Tie-Lose (WTL) scores comparing BoTSW to state-of-the-art methods. For instance, BoTSW+linear SVM reaches better performance than ED+1NN on **41** datasets, and worse performance on **8** datasets.

5 Conclusion

BoTSW transforms time series into histograms of quantized local features. Using dense extraction and normalizing Bag-of-Words enables to efficiently and accurately classify time series. We believe classification performance could be further improved by taking time information into account and/or reducing the impact of quantization losses in our representation.

Acknowledgments

This work has been partly funded by ANR project ASTERIX (ANR-13-JS02-0005-01), Région Bretagne and CNES-TOSCA project VEGIDAR.

References

1. A. Bailly, S. Malinowski, R. Tavenard, T. Guyet, and L. Chapel. Bag-of-Temporal-SIFT-Words for Time Series Classification. *ECML-PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2015.
2. M. G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *DMKD*, 29(2):400–422, 2015.
3. M. G. Baydogan, G. Runger, and E. Tuv. A Bag-of-Features Framework to Classify Time Series. *IEEE PAMI*, 35(11):2796–2802, 2013.
4. K. S. Candan, R. Rossini, and M. L. Sapino. sDTW: Computing DTW Distances using Locally Relevant Constraints based on Salient Feature Alignments. *Proc. VLDB*, 5(11):1519–1530, 2012.
5. M. Cuturi. Fast global alignment kernels. In *Proc. ICML*, pages 929–936, 2011.
6. A. Douzal-Chouakria and C. Amblard. Classification trees for time series. *Elsevier Pattern Recognition*, 45(3):1076–1091, 2012.

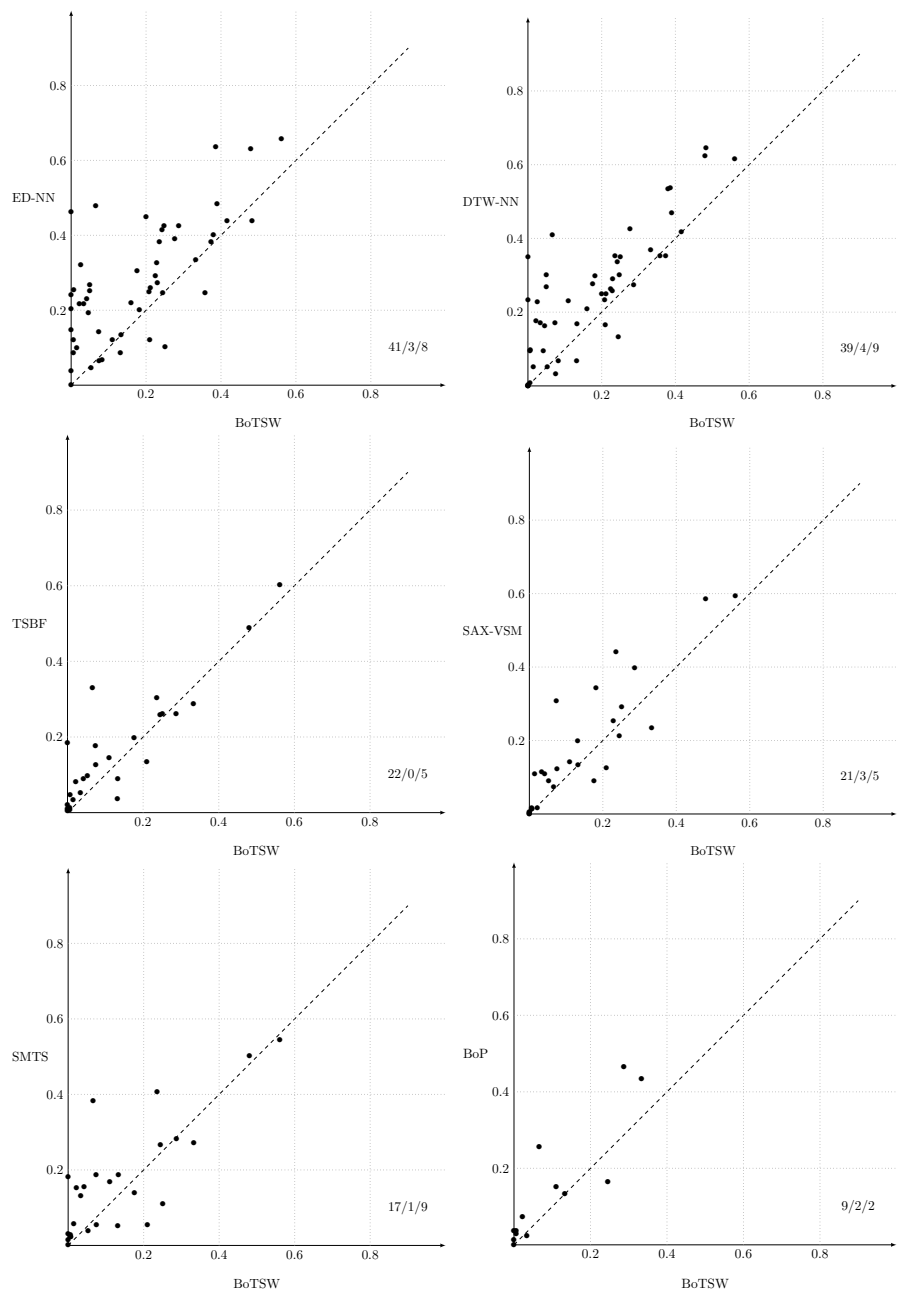


Fig. 5: Error rate of dense BoTSW with SSR-normalization versus some baselines (ED-NN, DTW-NN, TSBF, SAX-VSM, SMTS, BoP). Win-Tie-Lose score written at the bottom right corner.

7. H. Jegou and O. Chum. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. 2012.
8. H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, pages 3304–3311, 2010.
9. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR Time Series Classification/Clustering Homepage, 2011. www.cs.ucr.edu/~eamonn/time_series_data/.
10. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. ACM SIGMOD Workshop on Research Issues in DMKD*, pages 2–11, 2003.
11. J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *IJIS*, 39:287–315, 2012.
12. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
13. C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *Proc. ACM SIGKDD Workshop on Mining Temporal and Sequential Data*, pages 22–25, 2004.
14. P. Senin and S. Malinchik. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. *Proc. ICDM*, pages 1175–1180, 2013.
15. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pages 1470–1477, 2003.
16. H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. 2009.
17. J. Wang, P. Liu, M. F.H. She, S. Nahavandi, and A. Kouzani. Bag-of-words Representation for Biomedical Time Series Classification. *BSPC*, 8(6):634–644, 2013.
18. J. Xie and M. Beigi. A Scale-Invariant Local Descriptor for Event Recognition in 1D Sensor Signals. In *Proc. ICME*, pages 1226–1229, 2009.