Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

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

Adeline Bailly[1], Simon Malinowski[2], Romain Tavenard[1],
Thomas Guyet[3], and Lætitia Chapel[4]

[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. Experimental results show competitive performance with respect to classical techniques.

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 (kNN), which is usually combined with Euclidean Distance (ED) or Dynamic Time Warping (DTW) [11]. 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 [1, 2, 9, 12, 14]. 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 [10]. 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 [11], Ratanamahatana and Keogh compare Dynamic Time Warping to Euclidean Distance when used with a simple kNN 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 [4] 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 [5] 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 [10] that describes detected keypoints in the image using the gradients in the regions surrounding those points. Building on this, Sivic and Zisserman [13] 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) [6] 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 [1, 2, 9, 12, 14]. 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, kNN. In the following, we focus on explaining how the conversion of time series into BoW is performed in the literature. In [2], 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 [14], discrete wavelet coefficients are extracted on sliding windows and then quantized into words using k-means. In [9, 12], words are constructed using the SAX representation [8] of time series. SAX symbols are extracted from time series and histograms of n-grams of these symbols are computed. In [1], 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 [15] 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 [3], 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

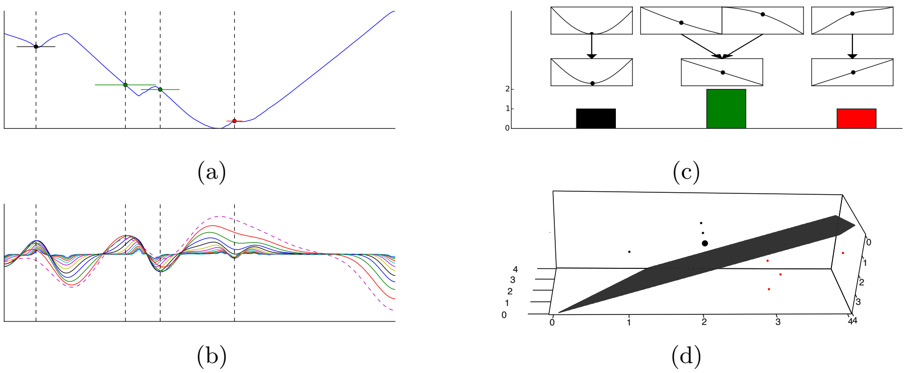
The proposed method is adapted from the SIFT framework [10] widely used for image classification. It is based on three main steps : (i) detection of keypoints (scale-space extrema) in time series, (ii) description of these keypoints by gradient magnitude at a specific scale, and (iii) representation 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.

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 time series filtered at consecutive scales:

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



(a) (c) (b) (d)

Fig. 1: Approach overview : (a) A time series and its extracted keypoints (the length of the horizontal lines for each point is proportional to the keypoint scale), (b) The Difference-of-Gaussians, computed at different scales, on which the keypoint extraction is built, (c) Keypoint description is based on the time series filtered at the scale at which the keypoint is extracted. Descriptors are quantized into words, and time series are represented by a histogram of words occurrence. For the sake of readability, neighborhoods are shown here instead of features. (d) These histograms are given to a classifier (linear SVM here) that learns boundaries between the different classes. The bigger dot here represents the description of the time series in (a), whose coordinates are $(1, 2, 1)$. Best viewed in color.

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.

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 $2 \times n_b$.

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.*

Dataset	BoTSW + linear SVM			BoTSW + 1NN			ED + 1NN	DTW + 1NN
	k	n_b	ER	k	n_b	ER	ER	ER
50words	512	16	0.363	1024	16	0.400	0.369	**0.310**
Adiac	512	16	0.614	128	16	0.642	**0.389**	0.396
Beef	128	10	0.400	128	16	**0.300**	0.467	0.500
CBF	64	6	0.058	64	14	0.049	0.148	**0.003**
Coffee	256	4	**0.000**	64	12	**0.000**	0.250	0.179
ECG200	256	16	**0.110**	64	12	0.160	0.120	0.230
Face (all)	1024	8	0.218	512	16	0.239	0.286	**0.192**
Face (four)	128	12	**0.000**	128	6	0.046	0.216	0.170
Fish	512	16	**0.069**	512	14	0.149	0.217	0.167
Gun-Point	256	4	0.080	256	10	**0.067**	0.087	0.093
Lightning-2	16	16	0.361	512	16	0.410	0.246	**0.131**
Lightning-7	512	14	0.384	512	14	0.480	0.425	**0.274**
Olive Oil	256	4	**0.100**	512	2	**0.100**	0.133	0.133
OSU Leaf	1024	10	**0.182**	1024	16	0.248	0.483	0.409
Swedish Leaf	1024	16	**0.152**	512	10	0.229	0.213	0.210
Synthetic Control	512	14	0.043	64	8	0.093	0.120	**0.007**
Trace	128	10	0.010	64	12	**0.000**	0.240	**0.000**
Two Patterns	1024	16	0.002	1024	16	0.009	0.090	**0.000**
Wafer	512	12	**0.001**	512	12	**0.001**	0.005	0.020
Yoga	1024	16	**0.150**	512	6	0.230	0.170	0.164

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

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

In this section, we investigate the impact of both the number of blocks n_b and the number of words k in the codebook (defined in Section 3) on classification error rates. Experiments are conducted on 20 datasets from the UCR repository [7]. We set all parameters of BoTSW but n_b and k as follows : $\sigma = 1.6$, $k_{sc} = 2^{1/3}$, $a = 8$. These values have shown to produce stable results. Parameters n_b and k vary inside the following sets : $\{2, 4, 6, 8, 10, 12, 14, 16\}$ and $\left\{2^i, \forall\ i \in \{2..10\}\right\}$ respectively. Codebooks are obtained *via* k-means quantization. Two classifiers are used to classify times series represented as BoTSW : a linear SVM or a 1NN classifier. Each dataset is composed of a train and a test set. For our approach, the best set of (k, n_b) parameters is selected by performing a leave-one-out cross-validation on the train set. This best set of parameters is 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 [7].

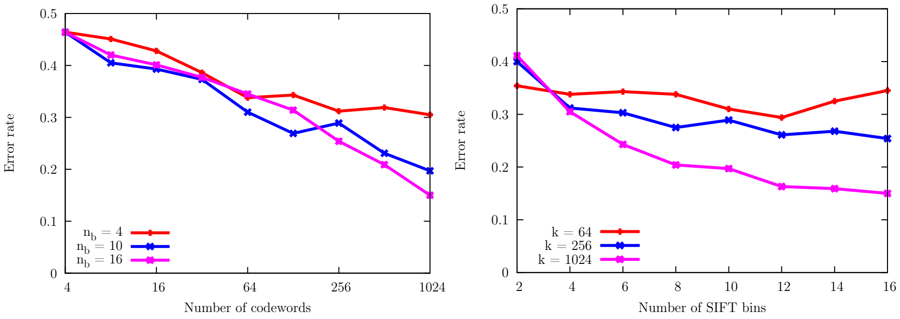


Fig. 2: Classification accuracy on dataset Yoga as a function of k and n_b.

	ED+ 1NN			DTW+ 1NN			TSBF[2]			SAX-VSM[12]			SMTS[1]			BoP[9]		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
BoTSW+lin. SVM	18	0	2	11	0	9	8	0	12	9	2	9	7	0	13	14	0	6
BoTSW + 1NN	13	0	7	9	1	10	5	0	15	4	3	13	4	1	15	7	1	12

Table 2: 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 18 datasets, and worse performance on 2 datasets.

BoTSW coupled with a linear SVM is better than both ED and DTW on 11 datasets. It is also better than BoTSW coupled with a 1NN classifier on 13 datasets. We also compared our approach with classical techniques for time series classification. We varied number of codewords k between 4 and 1024. Not surprisingly, cross-validation tends to select large codebooks that lead to more precise representation of time series by BoTSW. Fig. 2 shows undoubtedly that, for Yoga dataset, (left) the larger the codebook, the better the results and (right) the choice of the number n_b of blocks is less crucial as a wide range of values yield competitive classification performance.

Win-Tie-Lose scores (see Table 2) show that coupling BoTSW with a linear SVM reaches competitive performance with respect to the literature.

As it can be seen in Table 1, BoTSW is (by far) less efficient than both ED and DTW for dataset Adiac. As BoW representation maps keypoint descriptions into words, details are lost during this quantization step. Knowing that only very few keypoints are detected for these Adiac time series, we believe a more precise representation would help.

5 Conclusion

BoTSW transforms time series into histograms of quantized local features. Distinctiveness of the SIFT keypoints used with Bag-of-Words enables to efficiently and accurately classify time series, despite the fact that BoW representation ignores temporal order. 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. M. G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *DMKD*, 29(2):400–422, 2015.
2. M. G. Baydogan, G. Runger, and E. Tuv. A Bag-of-Features Framework to Classify Time Series. *IEEE PAMI*, 35(11):2796–2802, 2013.
3. 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.
4. M. Cuturi. Fast global alignment kernels. In *Proc. ICML*, pages 929–936, 2011.
5. A. Douzal-Chouakria and C. Amblard. Classification trees for time series. *Elsevier Pattern Recognition*, 45(3):1076–1091, 2012.
6. H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, pages 3304–3311, 2010.
7. 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/`.
8. 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.
9. J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *IJIS*, 39:287–315, 2012.
10. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
11. 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.
12. P. Senin and S. Malinchik. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. *Proc. ICDM*, pages 1175–1180, 2013.
13. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pages 1470–1477, 2003.
14. 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.
15. J. Xie and M. Beigi. A Scale-Invariant Local Descriptor for Event Recognition in 1D Sensor Signals. In *Proc. ICME*, pages 1226–1229, 2009.