Localized Random Shapelets

Mael Guillemé[1,2], Simon Malinowski[2], Romain Tavenard[3], and Xavier Renard[4]

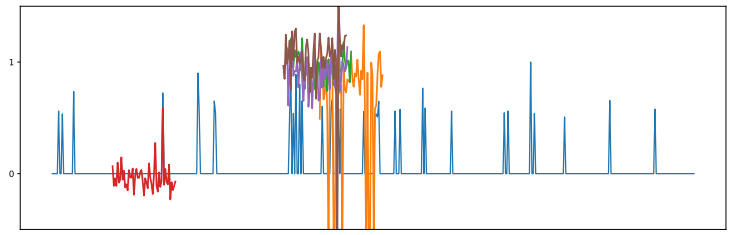
[1] Energiency
mael.guilleme@energiency.com
[2] Univ Rennes, Inria, CNRS, IRISA
[3] Univ Rennes, CNRS, LETG, IRISA
[4] AXA

Abstract. Shapelet models have attracted a lot of attention from researchers in the time series community, due in particular to its good classification performance. However, such models only inform about the presence / absence of local temporal patterns. Structural information about the localization of these patterns is ignored. In addition, end-to-end learning shapelet models tend to generate meaningless shapelets, leading to poorly interpretable models. In this paper, we aim at designing an interpretable shapelet model that takes into account the localization of the shapelets in the time series. Time series are transformed into feature vectors composed of both a distance and a localization information. Then, we design a hierarchical feature selection process using regularization. This process can be tuned to select, for each shapelet, either only its distance information or both distance and localization information. It is hence possible for every selected shapelet to analyze whether only the presence or the presence and the localization contributed to the decision process improving interpretability of the decision. Experiments show that this feature selection process has competitive performance compared to state-of-the-art shapelet-based classifiers, while providing better interpretability.

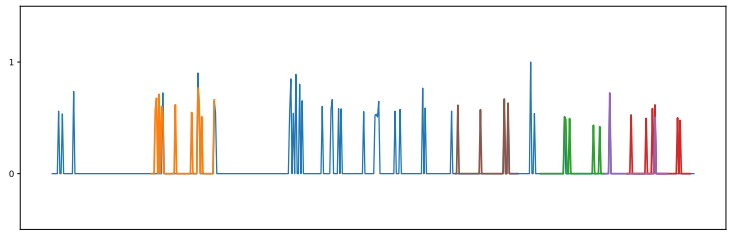
Keywords: time series · machine learning · shapelets

1 Introduction

Time series classification has recently gained an increasing attention from researchers, due in particular to its possible application in various domains such as economics, agriculture, and health for instance. There are two main families of methods for that task: some deal with raw time series and use or design dedicated (dis-)similarity measures, while others rely on feature extraction to embed time series in metric spaces in which standard machine learning tools can be considered. The work presented in this paper is part of this latter category. Feature-based methods include works relying on hand-crafted features [3,12] as well as learning-based approaches, among which the shapelet model plays an important role. Shapelets have first been introduced in [19]. They correspond to subsequences that are able to discriminate classes. The concept of shapelet



(a) Learning Time-Series Shapelets (LS)



(b) Localized Random Shapelets (LRS)

Fig. 1: Comparison between shapelets extracted by the *Learning Time-Series Shapelets* (LS) algorithm and our Localized Random Shapelets (LRS) approach. This Figure has been generated using `tslearn` implementation of LS [14].

transform has then been proposed in [7]. It consists in transforming time series into a vector whose components represent the similarity between the time series and shapelets that have been selected beforehand (or learned, as in [5]). After this transformation, time series are embedded in a Euclidean space where classifiers like Multi-Layer Perceptron or Support Vector Machines can be used. Techniques based on the shapelet transform are amongst the most accurate ones for time series classification (an interesting survey and comparison of time series classification methods can be found in [1]). However, they have three main drawbacks. First, the step of generating (or learning) shapelets that lead to accurate classification is computationally demanding (especially when dealing with large time series datasets). Some works have been proposed in order to fasten this step, by relying on time series approximation [8] or by drawing random shapelets [18,10]. Second, no information about the localization of the shapelets in the time series is available after the transformation. Classical shapelet transform only makes use of the similarity between a shapelet and a time series. Information about when the shapelet occurs in the time series might be of importance to discriminate classes. This can be related to previous works that have shown that localization of extracted features in time series improve classification accuracy [15]. To better understand why, let us consider, for example, a sign language sentence recognition task. In this use case, being able to recognize salient patterns (gesture atoms) is key, but localizing them in the sentence is

also important in order to understand the meaning of the sentence. Third, most shapelet-based models suffer from a lack of interpretability in the sense that (i) it is difficult to understand the impact of each shapelet on the final classification decision and/or (ii) extracted shapelets might not always be related to original time series. Figure 1 compares shapelets learned by the Learning Shapelet (LS) algorithm [5] on the EarthQuakes dataset and those extracted (at random) by our algorithm. Time series are rescaled to lie in the $[0, 1]$ range before learning the shapelets. However, we can see that LS-extracted shapelets have very little in common with the time series and do not even fit in the $[0, 1]$ range. In other words, shapelets cannot be seen as realistic time series patterns. For methods based on the shapelet transform [7], shapelets are extracted from original time series. But they then feed an ensemble of classifiers, which makes it difficult to understand relationship between shapelets and class belongings.

In this paper, we propose a novel shapelet model that tackles these drawbacks. First, in order to reduce the computing cost of selecting the most discriminant shapelets, our model selects shapelets randomly from training data. Hence, extracted shapelets are, by definition, closely related to the time series. Moreover, this random shapelet framework allows us to easily take the shapelet localization information into account. Second, we propose a dedicated feature selection method called Semi-Sparse Group Lasso (SSGL) capable of either ignoring a shapelet, keeping only its distance information or using both the distance and localization information. Third, we show that obtained shapelets are meaningful and the resulting model can be easily analyzed to get insights about which are the important features (for the classification task) in the dataset and for each of these features, what kind of information (presence only or presence and localization) contributed to the decision. Overall, we show that we are able to reach competitive performance w.r.t. Learning Shapelets [5] (even outperforming this baseline when larger datasets are considered) with a more interpretable model. The rest of this paper is organized as follows. Our Localized Random Shapelet model is detailed in Section 2 and its interpretability is discussed in Section 3. Section 4 evaluates the benefit of the proposed model on time series classification.

2 Localized Random Shapelet Model

2.1 Background on shapelets and shapelet transform

A shapelet $S = s_1, \ldots, s_l$ is a temporal sequence (that can be extracted from existing time series or not). Given a time series $T = t_1, \ldots, t_L$, the distance between s and T is defined as :

$$d(T, S) = \min_{1 \leq j \leq L-l+1} \sqrt{\sum_{i=1}^{l} (s_i - t_{i+j-1})^2}. \quad (1)$$

In other words, euclidean distances between s and every subsequence of T (of length l) are computed and only the best match (minimum distance) is kept.

Given a set $\mathcal{S} = \{S_1, \ldots, S_K\}$ of K shapelets, the shapelet transform of T is defined as the vector $v_1, \ldots, v_K$ such that $v_k = d(T, S_k)$ for all k. The original way to select a set of shapelets for a classification task is to evaluate the discriminatory power of a shapelet candidate by using for instance the information gain [19,7], and to keep the ones with the higher gain. A strategy based on learning the shapelets that minimize an objective function was proposed in [5]. Other works consider the use of random shapelets and then rely on classical feature selection algorithms together with the classification step. In [10], it has been shown that with this idea, a few thousands subsequences are enough to reach state-of-the-art classification performance on a standard benchmark [2].

2.2 Localized Random Shapelet model

In this framework, no information about the localization of the shapelets in the time series is used, while it has been shown that this kind of information helps improving classification performance [15]. In this section, we explain how we can integrate such information in the shapelet transform framework and derive a feature selection algorithm that keeps localization information only when needed.

In our Localized Random Shapelet (LRS) model, each shapelet S is drawn uniformly at random from the set of all training time series snippets. Each shapelet leads to two features for each time series T. The first feature is the same as in the classical shapelet transform, i.e. the shapelet distance[1] $d(T, S)$ between T and s as defined in Equation (1). The second feature corresponds to the first time instant at which this distance is reached. It is computed as

$$l(T, S) = \underset{1 \leq j \leq L-l+1}{\operatorname{argmin}} \sqrt{\sum_{i=1}^{l} (s_i - t_{i+j-1})^2}. \tag{2}$$

Let $\mathcal{T} = T_1, \ldots, T_N$ be a dataset of N time series. This set is transformed by the localized random shapelet model into a feature matrix $\mathbf{X}$, such that:

$$\mathbf{X} = \begin{pmatrix} d(T_1, S_1) & l(T_1, S_1) & \cdots & d(T_1, S_K) & l(T_1, S_K) \\ \vdots & \vdots & & \vdots & \vdots \\ d(T_N, S_1) & l(T_N, S_1) & \cdots & d(T_N, S_K) & l(T_N, S_K) \end{pmatrix}. \tag{3}$$

Once this feature matrix computed, we feed it to a standard multi-layer perceptron classifier as shown in Figure 2. This model takes as input a $2 \times K$-dimensional vector and outputs probabilities for the different classes at stake. The number of hidden layers can be adapted to application needs. We denote the whole set of parameters of this model by $\boldsymbol{\theta}$, and inside this set of parameters, we isolate the parameters of the first layer in the model and denote them $\boldsymbol{\beta}$.

[1] Note that the term distance is an abuse of notation since $d(T, S)$ is not a distance, mathematically speaking.

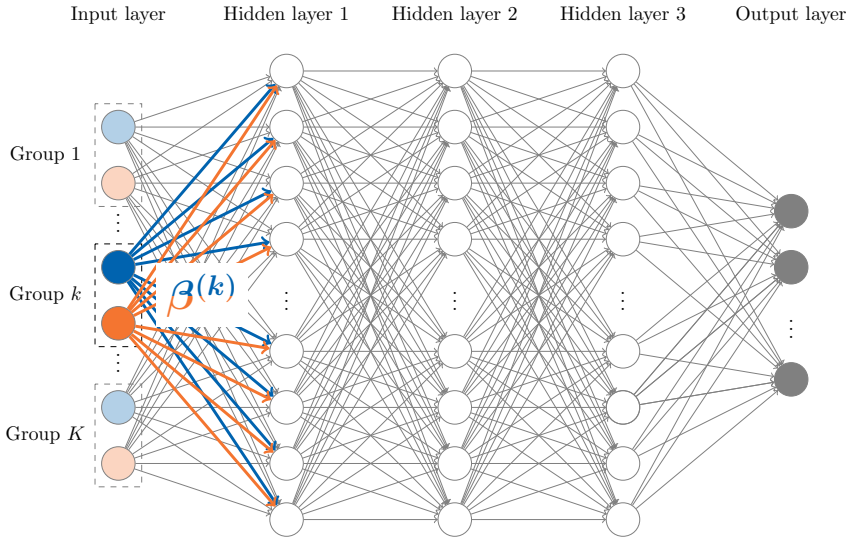


Fig. 2: Overview of our localized random shapelet model. Blue circles indicate distance features while orange ones correspond to location features. For each shapelet, a group is formed whose weights are denoted $\boldsymbol{\beta^{(k)}}$ (where k is the shapelet index). Note that the number of hidden layers may vary from one application to the other.

2.3 Structured feature selection

As explained above, the proposed shapelet model relies on random shapelets (taken from the original time series) in order to fasten the shapelet generation step. When dealing with such shapelets, a feature selection strategy should be applied before (or jointly with) the classification step, in order to simplify the resulting representation, which tends to improve overall accuracy [9]. It can be seen in Equation (3) that the extracted features are structured. Distance features tell how well a shapelet matches a time series, while localization features inform about the location of the match. Classical feature selection strategies are hence not adapted to this kind of feature matrix. Indeed, it does not seem reasonable to exclude a distance feature related to a shapelet while keeping the associated localization feature. However, removing a localization feature and keeping the associated distance feature might be meaningful in cases where the localization of a shapelet does not impact the class belonging of the times series. In the following, we design a feature selection strategy adapted to the kind of features extracted from the localized random shapelet model. This strategy, based on regularization, is described below.

Regularization strategy In the following, we assume that we have a prediction problem with a loss function $\mathcal{L}$ to be minimized. This loss function is computed over a dataset $\mathbf{X}$ of N observations associated with a target vector $\mathbf{y}$. A standard approach to bias the learning process towards sparse solutions is to derive a regularized loss function, as done for Lasso regression [16]:

$$\mathcal{L}^{\text{Lasso}}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \mathcal{L}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) + \lambda \|\boldsymbol{\beta}\|_1 \tag{4}$$

Simon *et al.* [13] introduced Sparse-Group Lasso (SGL), a more structured regularization scheme that could take feature group information into account. The resulting regularized loss function for a K-group problem is:

$$\mathcal{L}^{\text{SGL}}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \mathcal{L}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) + \alpha\lambda \|\boldsymbol{\beta}\|_1 + (1-\alpha)\lambda \sum_{k=1}^{K} \sqrt{p_k} \|\boldsymbol{\beta}^{(\boldsymbol{k})}\|_2 \tag{5}$$

where p_k is the number of features in group k and $\boldsymbol{\beta}^{(\boldsymbol{k})}$ is the sub-vector of $\boldsymbol{\beta}$ made of features from group k. Here, the $\|\boldsymbol{\beta}\|_1$ term enforces per-feature sparsity while $\|\boldsymbol{\beta}^{(\boldsymbol{k})}\|_2$ pushes towards sparsity at the group level. The α parameter hence acts as a trade-off between these two regularization terms.

We consider a slightly different setting in which, inside a group, only part of the features can be dropped. In our case, each group corresponds to a shapelet. For each shapelet, two features are available: one for the distance and one for localization of the match. We aim at designing a strategy that can, for each shapelet (or group), either:

- drop all information related to that shapelet (if the shapelet is useless for prediction),
- keep only the distance information (if the location of the shapelet does not help for prediction),
- keep both the distance and localization information.

In order to meet these constraints, we introduce the Semi-Sparse-Group-Lasso (SSGL) framework, which consists in minimizing the following loss function:

$$\mathcal{L}^{\text{SSGL}}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \mathcal{L}(\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) + \alpha\lambda \|\mathbf{M}_{\text{ind}}\boldsymbol{\theta}\|_1 + (1-\alpha)\lambda \sum_{k=1}^{K} \sqrt{p_k} \|\boldsymbol{\beta}^{(\boldsymbol{k})}\|_2 \tag{6}$$

where $\mathbf{M}_{\text{ind}}$ is an indicator diagonal matrix made of ones and zeros, the latter corresponding to variables that should be kept as soon as the group is not zeroed-out (*i.e.* variables that will not be considered for ℓ_1 regularization). In the case of our localized shapelet model, $\mathbf{M}_{\text{ind}}$ has a diagonal that alternates between zeros for dimensions corresponding to distances and ones for those related to the localization information.

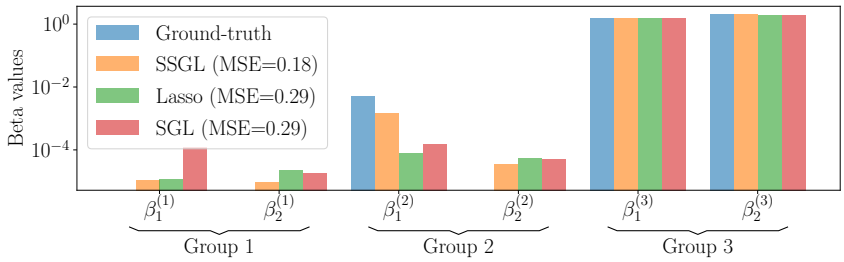


Fig. 3: Coefficients learned using different regularization schemes for a linear regression problem. Ground-truth coefficients are reported in blue.

Optimization in practice In practice, minimization of such a regularized loss function can be tackled by several means. First, building on [13], one can derive a block-wise procedure that considers feature groups one at a time and starts by deciding on whether it should be zeroed-out or not. In the case of an ordinary least square regression setting, it gives the following sufficient condition for zeroing-out group k (*i.e.* dropping all the information from the corresponding shapelet):

$$\left\| \mathbf{M}_{\text{ind}} S\left(\frac{\mathbf{X}^{(\mathbf{k})\top} \mathbf{r}_{-\mathbf{k}}}{N}, \alpha\lambda \right) \right\|_2 \leq \sqrt{p_k}(1-\alpha)\lambda, \tag{7}$$

where $\mathbf{X}^{(\mathbf{k})}$ is the submatrix of $\mathbf{X}$ in which only variables from group k are kept, $\mathbf{r}_{-\mathbf{k}}$ is the partial residual of $\mathbf{y}$, substituting all group fits other than group k and $S(\cdot,\cdot)$ is the coordinate-wise soft thresholding operator:

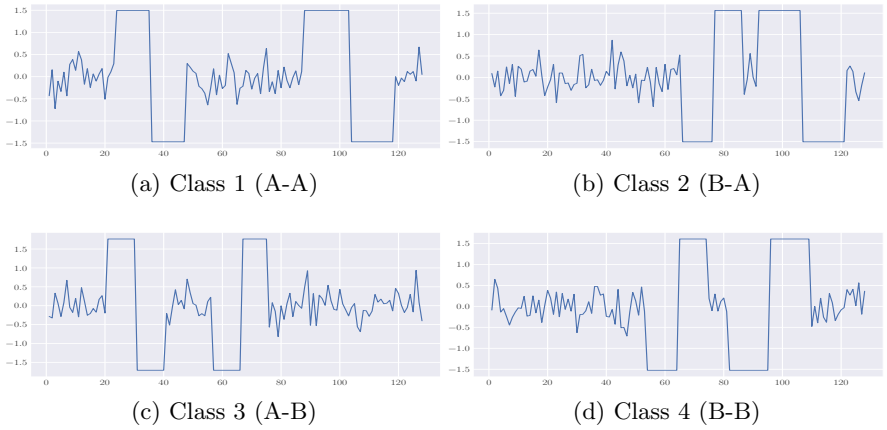
$$\left(S(\mathbf{z}, \alpha\lambda)\right)_j = \text{sign}(z_j)(|z_j| - \alpha\lambda)_+. \tag{8}$$

If this condition is satisfied, all $\boldsymbol{\beta}^{(\boldsymbol{k})}$ coefficients are set to zero and the process goes on to the next group. Otherwise, optimization of the coefficients inside group k should be performed, either using subgradient equations in a coordinate-wise algorithm or by performing gradient descent steps inside group k.

In all the experiments presented in this paper, we rather use full gradient descent on the regularized loss, in order not to face known limitations of block-wise gradient descent algorithms such as slow convergence and low parallelism capabilities. This strategy was already successfully applied in [11].

Toy example: structured linear regression Figure 3 presents a comparison of the three regularization schemes described in this section. For this comparison, we used the following model to draw samples:

$$\mathbf{y} = \mathbf{X} \cdot \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{9}$$



(a) Class 1 (A-A) (b) Class 2 (B-A)

(c) Class 3 (A-B) (d) Class 4 (B-B)

Fig. 4: An example of each class of TwoPatterns dataset.

where $\mathbf{X}$ and $\boldsymbol{\varepsilon}$ are drawn from centered normal distributions of standard deviation 1 and 0.01 respectively and $\boldsymbol{\beta} = \left[\beta_1^{(1)}, \beta_2^{(1)}, \beta_1^{(2)}, \beta_2^{(2)}, \beta_1^{(3)}, \beta_2^{(3)}\right]$ has only three non-zero components: $\beta_1^{(2)}$, $\beta_1^{(3)}$ and $\beta_2^{(3)}$ (*cf.* Figure 3).

We assume to have a problem similar to the shapelet setting, *i.e.* we have groups of two variables among which only $\beta_2^{(k)}$ is concerned by ℓ_1-norm regularization (for group k). This structural information is used for SGL and SSGL variants in the experiments, while Lasso is blind to such structure.

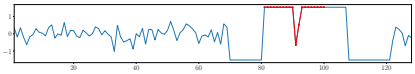
For this example, we use a simple model without any hidden layer. First, SSGL outperforms both SGL and Lasso in terms of mean squared error (MSE), showing the benefit of taking variable structure into account in the model. Second, from a more qualitative perspective, the structure of the ground truth coefficients is better preserved with SSGL (all three null coefficients have low value estimators and $\beta_2^{(1)}$ is better estimated thanks to the intra-group specific regularization scheme).

3 Model interpretability

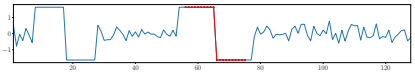
In this section, we illustrate the interpretability of our method through a simple use-case. We consider the TwoPatterns dataset from the UCR & UEA time series classification repository. TwoPatterns is a synthetic dataset in which each time series is made of two pattern occurrences surrounded by noise. There are two different patterns (named A and B in the following) in the dataset and the classification problem is hence made of four classes, corresponding to all possible permutations of patterns A and B, as shown in Fig. 4. For this illustration, we rely on a simple variant of our model in which we have no hidden layer and draw $K = 2,000$ shapelets to build our feature space. We show that, once trained, our model can be easily analyzed by scrutinizing its weights.

(a) 1st shapelet

(b) 2nd shapelet



(c) 3rd shapelet



(d) 4th shapelet

Fig. 5: Four most important shapelets (in red) extracted by our method from the TwoPatterns dataset.

More precisely, in the following, we aim at answering the following questions:

- Which shapelets are the most important for classification?
- For each of these shapelets, is the localization information important for the decision ?

Note that, in this simple setting where no hidden layer is used, this information comes in a straight-forward manner by analyzing weights, but the same analysis could be performed in the multi-layer case through gradient descent in the shapelet transform space.

First, we can extract most important shapelets (in terms of classification power) by ranking them with respect to the ℓ_2-norm of their associated weight matrix $\boldsymbol{\beta^{(k)}}$. Top-4 shapelets for dataset TwoPatterns are presented in Fig. 5. They fully match expectations since they focus on discriminative parts of the time series and cover both patterns A and B, as well as successions of these.

Then, for a shapelet that is considered discriminant, we can wonder whether its localization and distance are both used by the model or not. To assess the importance of the distance feature for a shapelet, we compute the ℓ_2-norm of the coefficients of $\boldsymbol{\beta^{(k)}}$ that are associated with its distance feature. We call this value distance coefficient. Similarly, to assess the importance of the localization feature for a shapelet, we compute the ℓ_2-norm of the coefficients of $\boldsymbol{\beta^{(k)}}$ that are associated with its localization feature. We call this value localization coefficient.

The first shapelet of Figure 5 has the 4^{th} highest localization coefficient amongst the 2,000 shapelets, together with the 14^{th} highest distance coefficient. This means that for this shapelet both the distance and the localization are important. It is coherent as this shapelet corresponds to a B pattern. The distance feature enables to discriminate class 1 from the others, while the localization features enables to discriminate class 2 and class 3. This is confirmed by the histograms of Figure 6. The second shapelet of Figure 5 has the 2^{nd} highest distance coefficient amongst the 2,000 shapelets. However, its localization coefficient is not as important as its distance coefficient (217^{th} out of 2,000). This means that for this shapelet the distance feature is important but not the localization. This is coherent as this shapelet corresponds to the end of a pattern

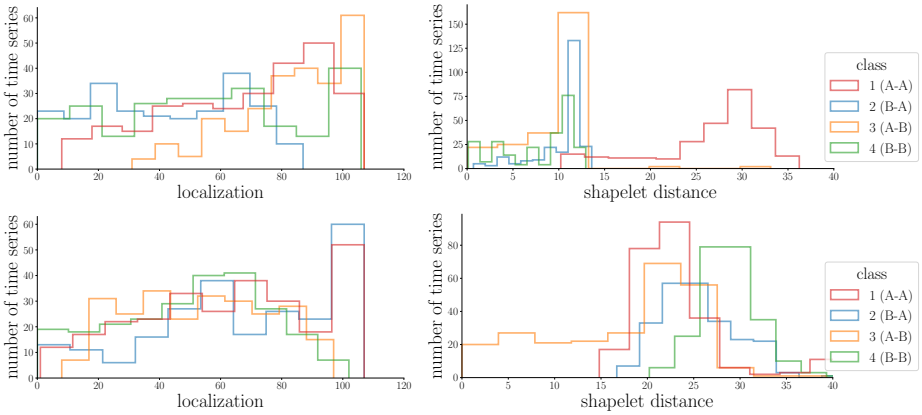


Fig. 6: The distribution of the localization (left) and distance (right) for the most important (first row) and second most important (second row) shapelet in TwoPatterns dataset.

A followed by the beginning of a pattern B, which only occurs in class 3, hence distance feature is enough to discriminate class 3 from the others, as confirmed by Figure 6. Similar conclusions can be drawn for shapelets 3 and 4.

We have seen in this section that our model allows easy extraction of the shapelets that have more contributed to discriminate classes. One can also easily analyze whether the localization of these shapelets was used by the model. This kind of information is very important as it brings a richer interpretability in the decision process. Conversely, state-of-the-art shapelet-based algorithms tend to suffer from lack of interpretability. Shapelet Transform is a weighted ensemble of several standard classifiers, which makes it difficult to assess the importance of shapelets in the decison process. In the Learning Shapelet (LS) framework, obtained shapelets are not constrained to be similar to training time series (cf. Figure 1), which limits the interpretability of such framework.

4 Experiments

In this section, we present experimental results to assess the performance of the proposed Localized Random Shapelet (LRS) model and compare it to state-of-the-art shapelet-based methods.

4.1 Experimental setup

The proposed LRS is compared to the following state-of-the-art baselines: Shapelet Transform (ST) [7], the Learning Shapelet (LS) model [5] and the Fast Shapelet (FS) one [8]. Experiments are based on the datasets from the UCR & UEA time series classification repository [2], a classical benchmark in the time series classification field. It is composed of 85 datasets with diverse time series classification

problems. A description of the datasets can be found in [1]. Classification performance for baseline methods is retrieved from the website associated to this dataset.

For each dataset, we draw $K = 2,000$ shapelets uniformly at random from the time series training set. For the length of the shapelets, we followed the procedure described in [5]. All time series are represented by a feature vector which size is twice the number of shapelets (cf. Eq. (3)). This feature vector embeds information about shapelet distance and localization.

We use a 3-hidden-layer model with 256, 128 and 64 units in the hidden layers and each internal layer is followed by a batch normalization layer as suggested in [6]. We add some dropout in the second and third layer. We use ReLU activations [4] for all internal layers and softmax activation for the final layer. The regularization in LRS requires two more hyper-parameters: λ that controls the regularization strength, and α that controls the trade-off between lasso and group-lasso terms. We cross-validated the dropout from the set $\{0.0, 0.3, 0.5, 0.7\}$, λ from the set $\{10^{-1}, 10^{-2}, \dots 10^{-7}\}$ and α from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Models are learned using RMSProp optimizer [17] with a learning rate of 0.001.

4.2 Runtime and memory cost

Experiments are run on a Laptop with a Fedora 24 system, 16 GB DDR4 RAM and dual core CPU (i7-6600U 2.6 GHz). Python code used for these experiments is publicly available[2]. As an indication of the low computing and memory cost of our method, it requires 66 seconds and 1.5 GB of memory to learn a model (for a given set of hyper-parameters) on the Electric Devices dataset that is the largest in the UCR & UEA archive.

4.3 Evaluation of the impact of the SSGL regularization

Figure 7 shows the error rates on 85 datasets for two different regularization strategies: Lasso [16] and SSGL. It can be seen that the SSGL regularization strategy (that allows selection for each shapelet of either distance feature, or distance and localization features, or none of these) slightly improves classification performance over Lasso (that makes the selection for each features independently).

4.4 Performance comparison against baselines

Critical diagrams show the average ranks of the classifiers in order, and cliques, represented by a solid bar. A clique represents a group of classifiers within which there is no significant pairwise difference. Figure 8 presents critical diagrams of the performance against the baselines when considering all datasets, and when considering only datasets with more than 300 training instances.

[2] `https://github.com/rtavenar/localized_random_shapelets`

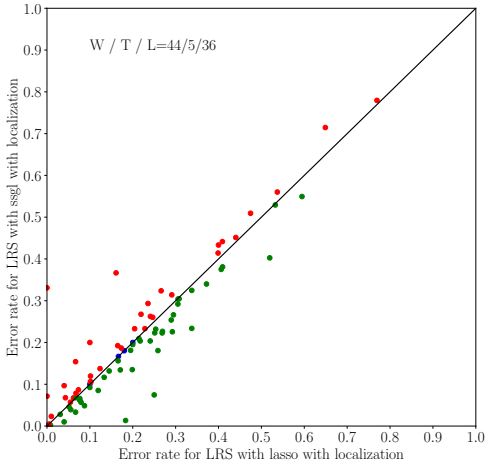


Fig. 7: Error rates comparison on 85 UCR datasets between LRS with ssgl regularization against LRS with lasso regularization.

(a) Performance on all the datasets

(b) Performance on large datasets

Fig. 8: Critical diagrams of the performance against the baselines.

Overall, in terms of classification performance, only Shapelet Transform outperforms our approach. As already stated, ST is a weighted ensemble of several standard classifiers, hence suffering from lack of interpretability about the decision process. The performance of LRS is slightly better than LS (but not significantly). When we only consider large datasets (the 42 datasets with more than 300 training instances, which are better suited to the characteristics of our method), performance of LRS is better than LS and FS and gets closer to that of ST. This seems to indicate that taking the localization information into account enables to improve classification performance provided that sufficient training data is available to correctly fit the model.

5 Conclusion

In this paper, we have proposed a novel shapelet model that has low computing cost, competitive performance and better interpretability compared to tradi-

tional shapelet-based methods. This model uses shapelet localization in addition to the traditional shapelet transform representation in a random shapelet framework. As such a framework needs many shapelets as input, we have designed a dedicated hierarchical regularization framework (SSGL) to fit our application needs. Experiments show that our model produces a ranking of realistic shapelets, and is able to explain their importance both in terms of distance and localization.This enables an easy and rich interpretability of the classification process. Moreover our classification performance is competitive with state-of-the-art shapelet-based classifiers.

References

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. Bagnall, A., Lines, J., Vickers, W., Keogh, E.: The uea & ucr time series classification repository, `www.timeseriesclassification.com`
3. Bailly, A., Malinowski, S., Tavenard, R., Chapel, L., Guyet, T.: Dense Bag-of-Temporal-SIFT-Words for Time Series Classification. In: Advanced Analysis and Learning on Temporal Data. Springer (2016). https://doi.org/10.1007/978-3-319-44412-3_2, `https://hal.archives-ouvertes.fr/hal-01252726`
4. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323 (2011)
5. 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)
6. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
7. Lines, J., Davis, L.M., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining. pp. 289–297 (2012)
8. Rakthanmanon, T., Keogh, E.: Fast shapelets: A scalable algorithm for discovering time series shapelets. pp. 668–676 (05 2013)
9. Renard, X., Rifqi, M., Erray, W., Detyniecki, M.: Random-shapelet : an algorithm for fast shapelet discovery. IEEE International Conference on Data Science and Advanced Analytics pp. 1–10 (2015)
10. Renard, X., Rifqi, M., Fricout, G., Detyniecki, M.: EAST Representation: Fast Discriminant Temporal Patterns Discovery From Time Series. ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data (2016)
11. Scardapane, S., Comminiello, D., Hussain, A., Uncini, A.: Group sparse regularization for deep neural networks. Neurocomputing **241**, 81–89 (2017)
12. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery **29**(6), 1505–1530 (2015)
13. Simon, N., Friedman, J., Hastie, T., Tibshirani, R.: A sparse-group lasso. Journal of Computational and Graphical Statistics **22**(2), 231–245 (2013)
14. Tavenard, R.: tslearn: A machine learning toolkit dedicated to time-series data (2017), `https://github.com/rtavenar/tslearn`

15. Tavenard, R., Malinowski, S., Chapel, L., Bailly, A., Sanchez, H., Bustos, B.: Efficient Temporal Kernels between Feature Sets for Time Series Classification. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery. pp. 528–543 (Sep 2017), https://halshs.archives-ouvertes.fr/halshs-01561461
16. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
17. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
18. Wistuba, M., Grabocka, J., Schmidt-Thieme, L.: Ultra-fast shapelets for time series classification. CoRR **abs/1503.05018** (2015), http://arxiv.org/abs/1503.05018
19. 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)