Shapelet Evaluation for Time Series Classification

Adam Charane[1], Matteo Ceccarello[2] and Johann Gamper[1]

[1]Free University of Bozen-Bolzano, Dominikanerplatz 3, Italy - 39100, Bozen-Bolzano

[2]University of Padova, Via Gradenigo 6/b 35131 - Padova Italy

Abstract

Shapelet based classification is a promising time series classification approach, which usually results in accurate predictions that are competitive with sophisticated and more complex classifiers, while it provides interpretability for the predictions. An important step in classification using shapelets is to select candidate subsequences based on some evaluation criteria. We adapt the Silhouette score, used originally in the context of clustering, in order to rank and select shapelet candidates for classification. We demonstrate empirically that our approach is faster compared to other methods in the literature, while being competitive in terms of the accuracy of classification. In particular, when the number of shapelets used for classification is small, our approach is superior to all other evaluation methods.

Keywords

Time Series Classification, Shapelets, Evaluation

1. Introduction

Time series classification (TSC) is a very active research area, where the aim is to assign a class label from a finite set to a high dimensional data point. The main difference with classic tabular data classification is that the attributes of the time series are ordered and the ordering of the points is crucial, because it defines the local behaviors of the time series. Different models have been developed based on different techniques such as Dictionaries [1, 2], Intervals [3], Distances [4] and many others [5].

In 2009, a technique called Shapelets have been introduced by Ye and Keogh [6]. This technique is based on finding local patterns where similar patterns appear in other time series from the same class. This has many benefits since it deals with local patterns of the time series rather than the global features of the time series. Another benefit is interpretability. Once a discriminating subsequence is found, it can be traced back to the original time series. This intuitive and simple approach turned out to be very effective in many time series mining tasks, and in particular for classification. However, it came with a computational challenge: a naive approach to find a shapelet needs to consider $\mathcal{O}(n^2m)$ candidates, resulting in an $\mathcal{O}(n^3m^2)$ time complexity, where n and m are the length and number of time series, respectively.

To improve the running time, most algorithms use some function to evaluate shapelets and to prune computations based on this evaluation. For instance, in [6] the authors evaluate the quality of shapelets using the information gain, which is based on the Shannon entropy. If the upper bound of the gain is less than the best shapelet found so far, the computation stops and the next candidate is considered. The authors also use early abandoning while computing the distances. Even though the time complexity remains the same, the pruning improves the running time significantly.

The main drawback of information gain is the running time, since it needs to evaluate many splitting points and compute the gain for each. Hills et al. [7] introduce different measures for evaluating shapelets, which are faster to compute compared to the information gain.

To address the above-mentioned issues, we introduce a new evaluation method for shapelets based on the Silhouette score [8], which is both fast and results in higher accuracy. We adapt the Silhouette score, which is originally used to evaluate the quality of clusters and the fitting of each object in a cluster, to the evaluation of shapelets, where we consider the clusters as the distances between the shapelets with time series from the same and from different classes. We select only those shapelets that contribute positively according to the Silhouette score. Our proposed evaluation method is faster and more accurate than the existing approaches, especially when the number of shapelets used for the classification is small.

The rest of this document is organized as follows. In Section 2 we review related works. Section 3 introduces some notations and background necessary for the rest of the paper. Section 4 introduces our approach and compares it to existing methods. In Section 5 we experimentally compare our approach with others using datasets from the UCR archive [9]. Finally, in Section 6 we summarize the findings and discuss future work.

DOLAP 2024: 26th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data, co-located with EDBT/ICDT 2024, March 25, 2024, Paestum, Italy

acharane@unibz.it (A. Charane); matteo.ceccarello@unipd.it (M. Ceccarello); johann.gamper@unibz.it (J. Gamper)
https://www.dei.unipd.it/~ceccarello/ (M. Ceccarello); http://www.inf.unibz.it/~gamper/ (J. Gamper)
0000-0002-0339-5322 (A. Charane); 0000-0003-2783-0218 (M. Ceccarello); 0000-0002-7128-507X (J. Gamper)

2. Related Work

Geurts [10] demonstrated that many time series classification problems can be solved by using local patterns, and introduced a technique to find and combine local patterns in order to classify time series. Subsequently, Ye and Keogh [6] introduced shapelets, which is a time series primitive that solves many time series data mining tasks, including time series classification. Furthermore, this technique made it possible for domain experts to interpret the outcome of the classification. A shapelet is a subsequence of a time series that maximally predicts a target variable. For the computation of shapelets, Ye and Keogh used exhaustive search evaluating all possible subsequences. While being accurate, this approach is computationally exhausting. Therefore, the authors introduced early abandoning while the distances between subsequences are computed. They also adopted the information gain based on entropy as a criterion to stop the computation of subsequence distances, once it is clear that a subsequence cannot achieve a higher ranking compared to other subsequences already evaluated.

In the following, many techniques were proposed to reduce the computation time. For instance, Mueen et al. [11] cached in memory some statistics and reused them to speed up the computation. Rakthanmanon and Keogh [12] used symbolic aggregate approximation (SAX) to first reduce the dimensionality of the time series. Then, the data is hashed, and the collision history is used for the identification of shapelets. Ji et al. [13] reduced the computation by first selecting a small subset of representative time series. Then, instead of considering all possible subsequences, only subsequences that contain local farthest deviation points are considered, based on the time series representation error. These two optimizations improved the computation by three orders of magnitude. Another widely used approach by Renard et al. [14] is to randomly select shapelets from the set of possible subsequences.

Despite the many works on time series classification using shapelets, surprisingly few works have focused on the evaluation metrics for the quality of the shapelets. In the paper introducing shapelets, Ye and Keogh [6] used the information gain to assess the quality of a candidate, and the gain was used as a splitting criterion for a decision tree classifier. Later on, Hills et al. [7] introduced F-statistic, which originally was proposed for the analysis of variance, Kruskal-Wallis and Moods medians, in order to evaluate and rank the shapelet candidates. In the same work, they also introduced a technique to transform datasets from the time domain to a feature domain spanned by the distances between the shapelets and the time series. The authors showed that (a) the transformation did not affect the classification accuracy and (b) that when using different classifiers instead of a decision tree, the accuracy of the classification increased. Our contribution is along the same lines as Hills et al., where we use the Silhouette score [8] to assess the quality of shapelets.

3. Background

In this section we provide notations and some background necessary for the rest of the paper.

3.1. Notation

A time series T is an ordered list of n real-valued variables $T = t_1, \dots, t_n$. A subsequence of T of length l, denoted by $T_{i,l}$, is a sequence of l consecutive values starting from i, i.e., $t_i, t_{i+1}, \dots, t_{i+l-1}$. To compute the distance between two subsequences S and Q of length l, we use the z-normalized Euclidean distance d, defined as

$$d(S,Q) = \sqrt{\sum_{i=1}^{l} (\frac{S_i - \bar{S}}{\sigma_S} - \frac{Q_i - \bar{Q}}{\sigma_Q})^2},$$

where $\bar{S}$ and σ_S are the mean and the standard deviation of S, respectively.

The above definition of distance requires that the two sequences have the same size. However, in our case we need to compare a subsequence of length l with a time series of length n with $n > l$. Following the Shapelet literature, we tackle this problem by computing the distance to all $n - l + 1$ subsequences of length l of the time series. Then, the minimal distance is considered to be the distance between the shapelet and the time series. More formally, given a sequence S of length l and a time series T of length n, the distance between T and S according to the normalized Euclidean distance d is

$$D(S,T) = \min_{i \in [1,\dots,n-l+1]} d(S, T_{i,l}).$$

For the rest of the paper, we assume that we have a dataset $\mathscr{D}$ with m time series, and each time series is assigned to a class c from a set C of classes. We denote the set of time series in a class $c \in \mathscr{C}$ as $\mathscr{D}^c$, and a shapelet S representing a class $c \in \mathscr{C}$ as S^c, i.e., a subsequence extracted from a time series in $\mathscr{D}^c$.

3.2. Shapelets

In this paper, we use the terms subsequence and shapelet interchangeably, since a shapelet is a subsequence that represents a class of time series and discriminates other classes. We would like to stress that the original definition of a shapelet introduced by Ye and Keogh [7] is not just a subsequence, but a subsequence with its optimal split point according to the information gain, which we describe in subsection 3.3. However, most papers about shapelets that were published after [15, 14, 7] do not use the same original definition. Instead, a shapelet is simply a subsequence regardless of its information gain. In fact, our contribution is an alternative approach to the information gain in the context of selecting shapelets for classification.

Another important aspect we would like to highlight is that in our work, similar to the original, a shapelet is a subsequence that is extracted from the set of time series. However, this is not necessarily the case. For instance, Grabocka et al. [16] introduced an algorithm that formalizes an optimization objective function to learn the shapelets instead of extracting them from the data. The result are shapelets that indeed represent a class and discriminate others, while they do not actually exist in the dataset.

3.3. Classification with Shapelets

Classifying time series using shapelets is a 3-step process:

1. extraction of shapelet candidates,
2. evaluation of candidates, and
3. transformation of data using shapelets.

Extraction of Shapelet Candidates. There exists many approaches in the literature to extract shapelets candidates. The first approach was brute force, i.e., slide a window of a fixed size over the time series with different window lengths and consider each subsequence as a candidate [6, 17]. This method results in a set of candidates in the order of $\mathcal{O}(mn^2)$. A faster approach introduced later reduced the number of candidates by randomly selecting subsequences from different positions with different lengths [14, 15]. A more recent method introduced in [13] reduces the space of candidates by first selecting some representative time series, which reduces the number of time series to be considered, and second, by ignoring non-interesting subsequences. Instead, it generates shapelet candidates by finding the important data points (IDP) in a time series, which is the point with the largest fitting error when the time series is represented with a linear representation, and keep finding IDPs recursively. The IDPs found are then used to extract shapelets.

Evaluation of Candidates. The extraction step usually results in a large set of candidates. Then, one has to evaluate these candidates in order to choose a smaller subset to be used as a primitive for classification. The first shapelet evaluation method is the information gain [6]. To evaluate a candidate S, the algorithm first computes the distance between S and all the time series in the dataset. The distances are then sorted, and $m-1$ thresholds are considered, where a threshold (also called split point) is the average between two consecutive distances. For a threshold t, the data is then separated into two sets $D_{S_<}$ and $D_{S_>}$, where $D_{S_<}$ is the set of time series which have a distance to S that is smaller than t. The information gain is computed as

$$G(D_S, t) = b - a,$$

where

- b is the gain before a split: $b = H(D_S)$,
- a is the gain after a split:

$$a = \frac{|D_{S_<}|}{|D_S|} H(D_{S_<}) + \frac{|D_{S_>}|}{|D_S|} H(D_{S_>})$$

,

- $|X|$ is the cardinality of a set X, and
- H is the Shannon binary entropy, which can be computed for a Bernoulli random variable X, where $\mathbb{P}(X = 1) = p$, as:

$$H(X) = -p \log(p) - (1-p) \log(p).$$

One drawback of the information gain is the computation time, since distances between time series and shapelets have to be sorted in order to find the optimal splitting point. For that, Hills et al. [7] introduce other evaluation methods that are much faster to compute, and the accuracy is not significantly different. Two of the methods were non-parametric statistical tests, namely Kruskal-Wallis and Moods medians, to test if samples originated from the same distribution or not based on their medians. The third method introduced was the F-statistic for variance analysis. Compared to the two non-parametric tests and the information gain, the F-statistic was the fastest in terms of running time and the time to find the best shapelet. It had also the best classification accuracy. At the end, the authors recommended that the F-statistic should be the default when evaluating the quality of shapelets. For the rest of this paper, we will compare our evaluation approach, Silhouettes, with both the F-statistic and the information gain.

Transform Data Using Shapelets. The shapelets selected from the large pool of candidates are of different lengths, and standard classifiers, such as K-Nearest Neighbor, logistic regression and neural networks cannot be directly applied to these subsequences of different size. In the original paper [6], the shapelets were included in a decision tree by using the shapelet's information computed as a splitting criterion at each node, which resulted in the coupling of shapelet evaluation and scoring.

In the shapelet transform paper [7], the authors introduced a data transformation, and show empirically that dissociating shapelet discovery from classification does not reduce the accuracy. When the data is transformed, it becomes possible to use any out of the box standard classifiers. Furthermore, the authors have shown that when the classification is done with a different model than a decision tree, the classification accuracy becomes higher.

Formally, after the k shapelets are selected, the shapelet transform of the data is

$$\mathscr{T} = \begin{pmatrix} D(T_1, S_1) & D(T_1, S_2) & \cdots & D(T_1, S_k) \\ D(T_2, S_1) & D(T_2, S_2) & \cdots & D(T_2, S_k) \\ \cdots & \cdots & \cdots & \cdots \\ D(T_m, S_1) & D(T_m, S_2) & \cdots & D(T_m, S_k) \end{pmatrix}$$

where T_i represents the i'th time series in the dataset $\mathscr{D}$ and S_j is the j'th selected shapelet. The matrix $\mathscr{T}$ and the labels of the time series are then fed to a standard classifier for training. In order to classify an unlabeled time series T, the time series is first transformed: $[D(T, S_1), D(T, S_2), \cdots, D(T, S_k)]$ and then the classification is done using the trained classifier.

4. Evaluating Shapelets Using Silhouettes

This section describes the idea behind our approach and compares how Silhouettes differs from information gain and F-statistic.

4.1. Silhouettes Description

The goal of shapelet evaluation is to select a good subset from the set of shapelet candidates, which can then be used for classification and achieves a high accuracy. Each approach assigns a real number to the candidates, termed evaluation score or rank of the shapelet, where usually a higher score means higher accuracy. Then, the shapelets with the highest rank are selected. Our method exploits the idea of Silhouettes introduced by Rousseeuw [8] and originally used to evaluate the outcome of clustering algorithms by comparing the within and between cluster dissimilarities. The score assigned to each object is between -1 and 1. A high value close to 1 means that the object is well assigned to the cluster it lies in; -1 means that the object is assigned to a wrong cluster; and a value around 0 means that the object is between two clusters.

In our case, we want to select from the large set of all possible subsequences, a smaller set in which each subsequence chosen represents a class and discriminates the other classes. In other words, we want to find a set of patterns S^c representing some class c, such that the patterns will have a small distance to time series in $\mathscr{D}^c$ compared to time series in $\mathscr{D}^j$, $c \neq j$. We use the Silhouette score to assign a rank to each subsequence, and we select the top ranked ones. Formally, the Silhouette score for a candidate shapelet S^c is

$$s(S^c) = \frac{b - a}{\max(a, b)},$$

where

- $a = \frac{\sum_{T \in \mathscr{D}^c} D(S^c, T)}{|\mathscr{D}^c|}$ is the average distance between the shapelet and time series from the same class, and
- $b = \frac{\sum_{T \notin \mathscr{D}^c} D(S^c, T)}{|\mathscr{D} \setminus \mathscr{D}^c|}$ the average distance between the shapelet and time series from different classes.

Notice that the score of the Silhouette is based only on whether a time series is in the same class of the shapelet or not. In the original work of Rousseeuw [8], the Silhouette works with many clusters, and b is the distance to the closest cluster that the object is not assigned to. Our decision is following the recommendation from Bostrom and Bagnall [17] where they introduce the binary shapelet. The authors show that, when a shapelet is selected to represent one class against all other classes instead of how well it splits all the classes, the classification results in higher accuracy for multiple-class datasets; additionally, it allows speeding up the computations as it facilitates frequently early abandoning.

4.2. Difference with F-statistic and Information Gain

In this subsection we would like to highlight the major differences between our approach and others.

Information Gain. The first difference is that Silhouettes do not need to sort the distances and find the best splitting. The overhead of sorting the distances to all time series for every shapelet makes the information gain approach slower [7]. The second and most important difference is that the gain assigned does not consider if the split differentiates between a specific class and the rest, or just finds a good balance. This is not a problem by itself since the classifier will learn how to classify the data based on how well each shapelet splits the classes. However, problems start occurring when the dataset has multiple classes, and especially if a class is more distinct from the others. The outcome is a set of many redundant shapelets for a class, which can discriminate a class very well but not the others. This behavior has been noted in [17], and to address it, the authors decided to extract and evaluate shapelets from each class independently. This way, even if one class is easy to classify, which will result in many shapelets that have a high score, they guarantee that shapelets representing other classes will be selected.

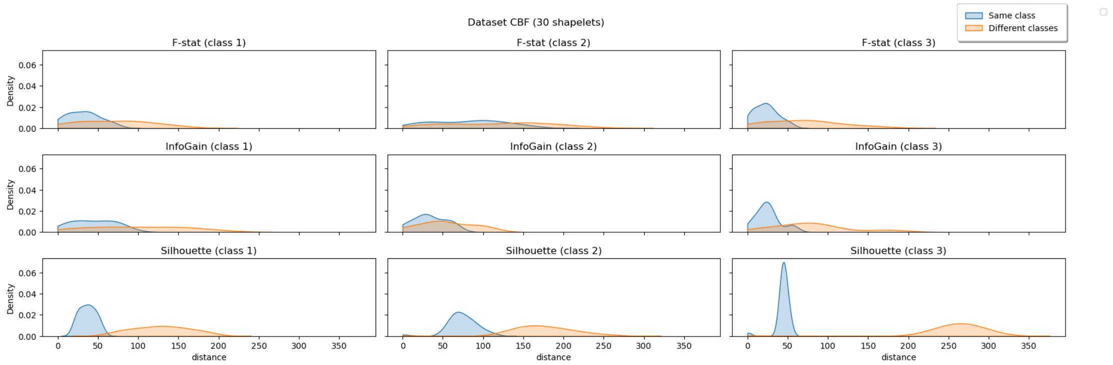

Figure 1: Distribution of distances of shapelets selected by each evaluation method.

F-statistic. F-statistic used originally for the analysis of variance is very similar to our approach. It is computed as the ration of *between group variability* and *within group variability*. For a shapelet S^c, the F-statistic is defined as

$$F(S^c) = \frac{\sum_{i=1}^{C}(\mu_i - \mu)^2 \times \frac{|\mathscr{D}^i|}{C-1}}{\sum_{i=1}^{C}\sum_{j=1}^{|\mathscr{D}^i|}(d_{ij} - \mu_i)^2 \times \frac{1}{m-C}},$$

where

- μ_i is the average distance of time series from $\mathscr{D}^i$,
- μ is the average distance with all time series, and
- d_{ij} is the distance to the j'th time series from $\mathscr{D}^i$.

The major difference between F-statistic and Silhouette is the use of variance, whereas in Silhouettes we use the mean. Despite the similarity between the two methods, the resulting shapelets are very different. Figure 1 shows the distribution of distances from shapelets selected by each of the three approaches for the CBF dataset from the UCR archive [9]. The CBF dataset has 3 classes. We used 30 time series in training set: 10 for class 1, 12 for class 2, and 8 for class 3. The test set had 900 time series, namely 300, 298, and 302 for the classes 1, 2, and 3, respectively. We extract 30 shapelets from each class using the three evaluation methods, and we color the distribution of distances with time series from the same class with blue, and the distribution of distances to other classes with orange. The X-axis represents the distance, and the Y-axis represent the density of distances. Notice that all the three approaches managed to select shapelets that are close to 0 from the same class compared to other classes. It is clear from the figure that Silhouettes select more shapelets (higher density) that have a small distance to time series from the same class, and at the same time, the two distributions (blue and orange) are separate from each other. Our assumption is that if we select shapelets that separate well the two distributions, then standard classifiers should be able to easily learn how to separate between different classes, and thus result in a high classification accuracy.

5. Experiments

This experimental evaluation aims to answer the following questions:

- How is the classification accuracy affected by the shapelets selected using the three approaches?
- How does the evaluation score of each approach correlate with the overall classification performance of the dataset?
- How does the evaluation scores for each class correlate with the true positives and false negatives of instances of that class?
- How does the running time of the Silhouette score compare to existing approaches?

For all our experiments, we use 94 datasets from the UCR archive [9]. From the 94 datasets, 37 are binary classification datasets, 16 have 3 classes, 9 have 4 classes, and 32 have 5 classes or more. There are 6 datasets where the most prevalent class has over 10 times the number of instances compared to the least represented, while 10 datasets exhibit an imbalance exceeding a factor of 5.

5.1. Experiments Design

The extraction of shapelet candidates is an important phase in time series classification. However, in this work, our focus is on the evaluation of shapelets and its effect on the classification. For a fair comparison between the three methods, we first randomly sample a large set of shapelets for each dataset from different starting positions and with different lengths. More precisely, for each dataset we randomly extract L shapelets for each class.

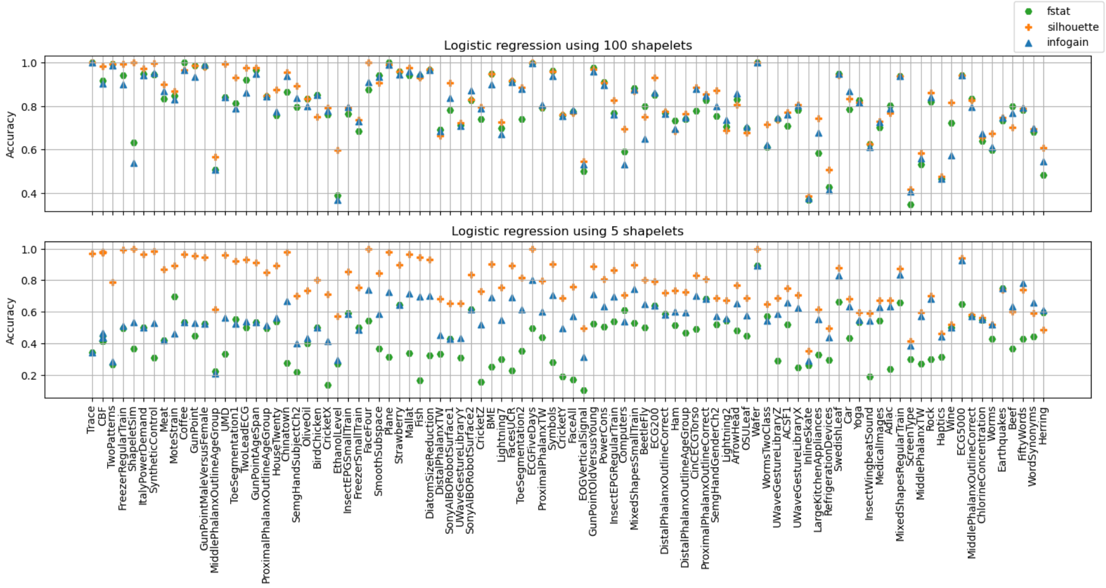

Figure 2: Accuracy on 94 datasets from the UCR archive using 100 and 5 shapelets per class, selected using different shapelet evaluation methods.

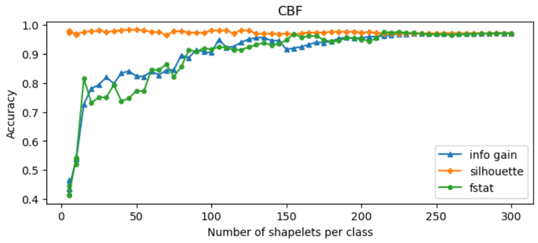

Figure 3: Comparing the effect of number of top candidates using the different evaluation methods on *CBF* dataset.

The value of L is set to the maximum of 300 and 20% of the length of the time series, i.e., $L = \max(300, 0.2 \times n)$. Next, we pre-compute and store the distances between the shapelets and the time series, since they will be needed in order to evaluate the shapelets, and also to transform the data for classification. The large set of randomly selected shapelets and their corresponding distances to all the time series will be used as a starting point for all our experiments, and they are the same for all comparisons between the approaches.

5.2. Comparison of Classification Accuracy

In this experiment we compare the effect of the evaluation using Silhouette, information gain and F-statistic on the classification. Instead of fixing an arbitrary number of shapelets to select and transform the data with, we start from a small number, namely $k = 5$ for each class. The *CBF* dataset example has 3 classes, so we select 15 shapelets in total, and we gradually increase k. For every value, we select the top k shapelets per class using the three methods, transform the data (using the pre-computed distances) and run the classification.

For the classification, we fix 6 standard classifiers. The first classifier is the decision tree as it was the standard approach in the literature when classifying with shapelets. We also include the 1-Nearest Neighbor (1-NN) and support vector classifier (SVC) since they were used in the evaluation in Hills work [7]. We also add the Logistic Regression classifier as an extra linear classifier, and K-Nearest Neighbor as an extra non-linear classifier, which is also a generalization of 1-NN. Finally, we also include an ensemble method, namely ADABoost, which iteratively builds multiple decision trees, and each new tree is trained with a penalty on instances that were wrongly classified by the previous trees.

The parameters of the models are found during the training by cross validation using five splits. The hyper-parameters are found by grid search. Finally, we train a new model using the best combination of parameters found, and report the results of the test set.

Figure 2 shows the performance of 94 datasets from the UCR archive. On the top of the figure, the classification is done with the best 100 shapelets per class. On the X-axis we list the datasets, and the Y-axis represents the accuracy. Each color refers to the result of a method.

With such a high number of shapelets, almost all methods have the same performance. However, when only a few shapelets are used (e.g., 5 shapelets as shown at the bottom of the figure), the Silhouette becomes the most accurate method, most of the time even with a big margin.

A pattern that we observed is that, when the number of shapelets is small, the Silhouette score results in a high accuracy, and it does not change much if the number of shapelets is increasing; thus the Silhouette score turns out to be a stable method. On the other hand, the accuracy achieved by the information gain and the F-statistic increases when the number of shapelets is increasing. Figure 3 shows this behavior for the CBF dataset. By only using 5 shapelets per class, the Silhouette approach already achieves 97.88% accuracy, whereas F-statistic and information gain are around 42%. In this example, the CBF data has a small training set and a much larger test set, containing 30 and 900 time series, respectively. The number of shapelets required for the F-statistic and the information gain to start approaching the same accuracy as the Silhouette is more than 200 shapelets per class. This means that the data transformed using the Silhouette score can be 40 times smaller, yet achieving a higher accuracy.

5.3. Evaluation Methods and Dataset Classification Performance

We investigate how the Silhouette score relates to the accuracy of classifying a dataset using shapelets. If the Silhouette scores assigned to the top candidate shapelets are all high (close to 1), we expect the dataset to be classified with high accuracy since the shapelets separate the different classes well. For example, from Figure 2 we can see that the dataset GunPointMaleVersusFemale has a higher accuracy (96%) compared to the Wine dataset, and it turns out that the average Silhouette scores are 0.56 and 0.25, respectively. To confirm this hypothesis, we use the results reported in Figure 2 and do the following:

1. Compute the average score of the selected shapelets using an evaluation method for each dataset.
2. Compute the correlation between the average score and the accuracy.

Table 1 reports the Pearson coefficient for each of the three approaches as well as p-values (in the parentheses). When using only a few shapelets, the Silhouette score shows the most significant correlation. When the number of shapelets is increased, the correlation between the Silhouette scores and the accuracy remains stable, but both the information gain and the F-statistic correlations increase. The correlation of the information gain becomes even more significant than the correlation of the Silhouette score. This confirms the behavior seen in Figure 3.

Table 1
Pearson coefficient between the accuracy of the dataset and the evaluation methods scores.

Method	Pearson correlation	
	5 shapelets	100 shapelets
F-statistic	0.09 (3×10^{-1})	0.21 (3×10^{-2})
Silhouette	0.29 (2×10^{-3})	0.30 (2×10^{-3})
Information gain	0.12 (2×10^{-1})	0.40 (5×10^{-5})

5.4. Evaluation Methods and Classification Performance for each Class

In the previous section, we compared the average score of each approach with the accuracy achieved on datasets. In this experiment, we want to compare the behavior of the three evaluation methods with respect to class performance, i.e., when a class in the dataset is easy or hard to distinguish compared to other classes. For that, we compute the correlation between the evaluation scores and the accuracy of each class c, which we define as

$$\mathscr{A}(c) = \frac{\mathrm{TP}^c}{\max(|\{y^c, y^c \in \hat{y}\}|, |\mathscr{D}^c|)},$$

where

- TP^c is the number of correct predictions of class c, and,
- $\{y^c, y^c \in \hat{y}\}$ is the set predictions where the classifier predicted class c.

We normalize by the maximum of the number of predictions and the actual number of instances labeled c in the test set. This is equivalent to the minimum of recall and precision, meaning that for each class the score reports whether the model trained on the transformed data is both precise (few false positives) but also making many successful predictions (few false negatives).

Table 2 shows the results achieved by the three approaches for both 5 and 100 shapelets per class. The results reported are using 65 datasets from the UCR archive, because we only kept datasets that have 3 classes or more. The reason to keep only datasets with at least 3 classes is that in a binary classification setup, if one class can be identified well using some shapelets, the other class is automatically discriminated as well.

When using many shapelets, all three approaches are highly correlated, with the information gain having the most significant correlation. In contrast, if only a few

Table 2
Pearson coefficient between the accuracy of classes and the scores of each evaluation method.

Method	Pearson correlation	
	5 shapelets	100 shapelets
F-statistic	$0.31\ (9 \times 10^{-15})$	$0.29\ (2 \times 10^{-10})$
Silhouette	$0.18\ (1 \times 10^{-5})$	$0.18\ (5 \times 10^{-5})$
Information gain	$-0.02\ (5 \times 10^{-1})$	$0.35\ (1 \times 10^{-14})$

shapelets are used, the information gain is uncorrelated. This is due to the fact that the information gain is determining the optimal split point that results in the highest gain regardless of the class of the shapelet itself. For the F-statistic, the correlation is very high when a few shapelets is used. This can be justified by the denominator of the statistic, where the variance of distances is minimized, which favors shapelets that are similar to each other. Finally, notice again that for the Silhouette score the correlation does not change when the number of shapelets changes.

5.5. Running Time Comparison

In this experiment we compare the running time of the three methods. Table 3 shows the mean, standard deviation and the percentiles of the running time for datasets from the UCR archive. As mentioned above, the number of extracted shapelets, L, depends on the datasets. The Silhouette is the fastest method, followed by the F-statistic and finally the information gain. This corresponds also to the respective algorithmic complexities: $\mathcal{O}(Lm)$, $\mathcal{O}(Lm)$ and $\mathcal{O}(Lm\log(m))$, respectively. Even though the F-statistic and the Silhouette have the same algorithmic complexity, the F-statistic has to cache and reuse some computations in order to compute the variances in linear time (for each shapelet). For the information gain, besides the overhead of sorting, the algorithm has also to evaluate the gain at each splitting point, which results in a higher running time.

Table 3
Summary of the running time in milliseconds for 85 datasets from the UCR archive.

	F-statistic	information gain	Silhouette
mean	4.81	1592.38	1.80
std	18.75	3858.66	4.02
min	0.63	37.20	0.09
25%	0.89	103.93	0.20
50%	1.51	401.39	0.53
75%	2.96	1424.26	1.78
max	176.57	34419.93	34.43

6. Conclusion and Future Work

We introduced a new shapelet evaluation method to score the utility of the candidate shapelets for time series classification. The idea is to select shapelets based on the Silhouette score used originally for evaluating clusters. This tends to select shapelets that are very similar to the time series from the same class, but also very different from the time series of different classes. We have shown through experiments using 94 time series from the UCR archive [9] that the classification using Silhouettes is not only competitive with existing approaches in the literature in terms of the accuracy of predictions, but also much better when the number of shapelets is very small. This property results in a much smaller training data size after the transformation.

The scores of our approach correlate with the accuracy classifications. In a first experiment we computed the Pearson correlation between the average score of all the selected shapelets and the achieved accuracy. This means that a high Silhouette score for a dataset will likely result in a high accuracy. We have also seen in Table 1 that the correlation does not change with the number of shapelets. Thus, it is sufficient to use a small number shapelets for the classification since the accuracy will likely not change much, unlike for the F-statistic and the information gain, which both require many shapelets to achieve a high accuracy.

Finally, we have compared the running time between the three approaches. Silhouette and F-statistic both have acceptable running time as both have an average time in the order of milliseconds. However, the bottleneck is the actual computation of distances.

In this work, our main focus was on the evaluation of the shapelets, and its effect on the accuracy of time series classification. Given that the Silhouette score results in high accuracy with a very small number of shapelets, we plan to speed up the whole classification process using shapelets by integrating the Silhouette in the candidate selection step and also exploiting them for early computation abandoning.

References

[1] P. Schäfer, The BOSS is concerned with time series classification in the presence of noise, Data Mining and Knowledge Discovery 29 (2014) 1505–1530. URL: https://doi.org/10.1007/s10618-014-0377-7. doi:10.1007/s10618-014-0377-7.

[2] M. Middlehurst, W. Vickers, A. Bagnall, Scalable dictionary classifiers for time series classification, in: Intelligent Data Engineering and Automated Learning – IDEAL 2019, Springer International Publishing, 2019, pp. 11–19.

URL: https://doi.org/10.1007/978-3-030-33607-3_2. doi:10.1007/978-3-030-33607-3_2.

[3] M. Flynn, J. Large, T. Bagnall, The contract random interval spectral ensemble (c-RISE): The effect of contracting a classifier on accuracy, in: Lecture Notes in Computer Science, Springer International Publishing, 2019, pp. 381–392. URL: https://doi.org/10.1007/978-3-030-29859-3_33. doi:10.1007/978-3-030-29859-3_33.

[4] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, G. I. Webb, Proximity forest: an effective and scalable distance-based classifier for time series, Data Mining and Knowledge Discovery 33 (2019) 607–635. URL: https://doi.org/10.1007/s10618-019-00617-3. doi:10.1007/s10618-019-00617-3.

[5] A. J. Bagnall, J. Lines, A. Bostrom, J. Large, E. J. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. 31 (2017) 606–660. URL: https://doi.org/10.1007/s10618-016-0483-9. doi:10.1007/s10618-016-0483-9.

[6] L. Ye, E. Keogh, Time series shapelets: A new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 947–956. URL: https://doi.org/10.1145/1557019.1557122. doi:10.1145/1557019.1557122.

[7] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Mining and Knowledge Discovery 28 (2013) 851–881. URL: https://doi.org/10.1007/s10618-013-0322-1. doi:10.1007/s10618-013-0322-1.

[8] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65. URL: https://doi.org/10.1016/0377-0427(87)90125-7. doi:10.1016/0377-0427(87)90125-7.

[9] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, Hexagon-ML, The ucr time series classification archive, 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018.

[10] P. Geurts, Pattern Extraction for Time Series Classification, Springer Berlin Heidelberg, 2001, p. 115–127. URL: http://dx.doi.org/10.1007/3-540-44794-6_10. doi:10.1007/3-540-44794-6_10.

[11] A. Mueen, E. Keogh, N. Young, Logical-shapelets: An expressive primitive for time series classification, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 1154–1162. URL: https://doi.org/10.1145/2020408.2020587. doi:10.1145/2020408.2020587.

[12] T. Rakthanmanon, E. Keogh, Fast shapelets: A scalable algorithm for discovering time series shapelets, in: Proceedings of the 2013 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2013. URL: http://dx.doi.org/10.1137/1.9781611972832.74. doi:10.1137/1.9781611972832.74.

[13] C. Ji, C. Zhao, L. Pan, S. Liu, C. Yang, L. Wu, A fast shapelet discovery algorithm based on important data points, International Journal of Web Services Research 14 (2017) 67–80. URL: https://doi.org/10.4018/ijwsr.2017040104. doi:10.4018/ijwsr.2017040104.

[14] X. Renard, M. Rifqi, W. Erray, M. Detyniecki, Random-shapelet: An algorithm for fast shapelet discovery, in: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2015. URL: https://doi.org/10.1109/dsaa.2015.7344782. doi:10.1109/dsaa.2015.7344782.

[15] A. Guillaume, C. Vrain, W. Elloumi, Random dilated shapelet transform: A new approach for time series shapelets, in: Pattern Recognition and Artificial Intelligence, Springer International Publishing, 2022, pp. 653–664. URL: https://doi.org/10.1007/978-3-031-09037-0_53. doi:10.1007/978-3-031-09037-0_53.

[16] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 392–401. URL: https://doi.org/10.1145/2623330.2623613. doi:10.1145/2623330.2623613.

[17] A. Bostrom, A. Bagnall, Binary shapelet transform for multiclass time series classification, in: Big Data Analytics and Knowledge Discovery, Springer International Publishing, 2015, pp. 257–269. URL: https://doi.org/10.1007/978-3-319-22729-0_20. doi:10.1007/978-3-319-22729-0_20.