Local-Shapelets for Fast Classification of Spectrographic Measurements

Daniel Gordon[a,c,*], Danny Hendler[a], Aryeh Kontorovich[a], Lior Rokach[b,c]

[a]*Department of Computer Science, Ben-Gurion University of The Negev Be'er Sheva 84105, Israel*
[b]*Department of Information Systems Engineering, Ben-Gurion University of The Negev Be'er Sheva 84105, Israel*
[c]*Telekom Innovation Laboratories, Ben-Gurion University of The Negev Be'er Sheva 84105, Israel*

Abstract

Spectroscopy is widely used in the food industry as a time-efficient alternative to chemical testing. Lightning-monitoring systems also employ spectroscopic measurements. The latter application is important as it can help predict the occurrence of severe storms, such as tornadoes.

The *shapelet* based classification method is particularly well-suited for spectroscopic data sets. This technique for classifying time series extracts patterns unique to each class. A significant downside of this approach is the time required to build the classification tree. In addition, for high throughput applications the classification time of long time series is inhibitive. Although some progress has been made in terms of reducing the time complexity of building shapelet based models, the problem of reducing classification time has remained an open challenge.

We address this challenge by introducing *local-shapelets*. This variant of the shapelet method restricts the search for a match between shapelets and time series to the vicinity of the location from which each shapelet was extracted. This significantly reduces the time required to examine each shapelet during both the learning and classification phases. Classification based on local-shapelets is well-suited for spectroscopic data sets as these are typically very tightly aligned. Our experimental results on such data sets demonstrate that the new approach reduces learning and classification time by two orders of magnitude while retaining the accuracy of regular (non-local) shapelets-based classification. In addition, we provide some theoretical justification for local-shapelets.

Keywords: Spectrography, time series, classification, shapelets, local

Research highlights

- We present an algorithm for classifying spectrographic measurements.
- The concept of locality is introduced into an established time series algorithm.
- A technique for estimating a tolerance parameter is presented.

[*]Corresponding author. Tel: +972 (0)86428782; fax: +972 (0)86477650;
Email addresses: gordonda@cs.bgu.ac.il (Daniel Gordon), hendlerd@cs.bgu.ac.il (Danny Hendler), karyeh@cs.bgu.ac.il (Aryeh Kontorovich), liorrk@bgu.ac.il (Lior Rokach)

- Learning and classification times are reduced by two orders of magnitude.
- Accuracy levels are retained.

1. Introduction

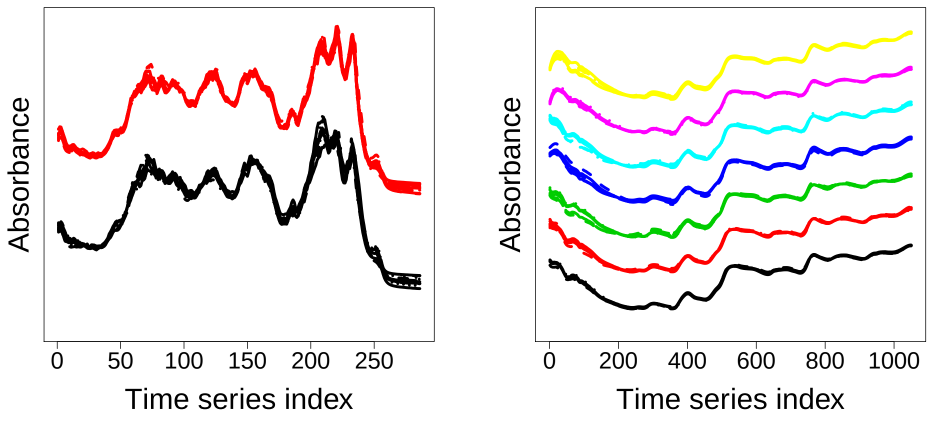
Spectroscopy is a field devoted to the study and characterization of physical systems by measuring the electromagnetic frequencies they absorb or emit (Herrmann and Onkelinx, 1986). Items differing in their chemical composition or molecular bonds absorb or emit light at different wavelengths leaving a different spectroscopic fingerprint thus enabling differentiation between them. For example, Al-Jowder et al. (2002) used mid-infrared spectroscopy to detect meat adulteration by comparing the spectra of adulterated meat with that of unadulterated meat. A study by Briandet et al. (1996) discriminated between two different types of coffee beans (Arabica and Robusta) using mid-infrared spectroscopy. Other methods for distinguishing between different types of food exist, which are based on wet chemical analysis (Bicchi et al., 1993; Lumley, 1996; Sharma et al., 1994). The advantages of spectroscopy over wet chemical analysis are in its simplicity (Briandet et al., 1996) and speed of response.

Spectroscopic measurements are also generated by systems monitoring lightning (Eads et al., 2002). This application is important as relative percentages of different types of lightning can indicate the outbreak of severe storms, such as tornadoes. In addition to laboratory research, spectroscopic equipment is starting to be mass produced for every day use allowing anyone to analyze their surroundings with the aid of spectroscopic measurements (SCIO, 2014). The measurements are uploaded to a cloud service where they are analyzed and then the results of the analysis are made available. The service is cloud based, requiring algorithms with high throughput to enable a quick response to high volumes of queries by users.

The outcome of the spectroscopic analysis of a physical system is a vector in which each index represents a frequency and each value is the measured intensity of that frequency. The representation of spectroscopic measures and time series are identical (Ye and Keogh, 2011a), as the only explicit data are the measurements and the meaning of each measurement is defined by its location in the vector. This equivalence allows the application of time series classification methods to the field of spectroscopy. A previous experimental study (Hills et al., 2013) showed that the shapelet based classification method is particularly suited for data sets from the field of spectroscopy, as it achieved a higher accuracy than other machine-learning classification methods.

Recently, Ye and Keogh (2011a) introduced the shapelets approach for classifying time series. A *shapelet* is a subsequence extracted from one of the time series in the data set. The shapelet is chosen by its ability to distinguish between time series from different classes. A test time series is classified based on its distance from the shapelet. In the case of multiple shapelets, these form the nodes of a classification tree. The intuition behind this approach is that the pattern best separating the classes is not necessarily an entire time series. Rather, a certain subsequence may best describe the discriminating pattern. Ye and Keogh's algorithm considers all possible subsequences in the training set in order to identify those shapelets that yield the optimal split. Through the rest of this paper, we will refer to this algorithm as the *YK-algorithm*.

Two key advantages of classification with shapelets are the accuracy and interpretability of the induced classification model, as it supplies information on the patterns characteristic of the different classes (Ye and Keogh, 2011a). A significant downside of this approach is the time required for building the classification tree (Hills et al., 2013; Mueen et al., 2011; Rakthanmanon and Keogh,



(a) Coffee data set   (b) Wheat data set

Figure 1: Examples of two data sets (coffee and wheat) from the field of spectroscopy. The time series of each class are vertically separated and in different colors. As shown, the examples of each class are tightly aligned, i.e., similar patterns are exhibited at similar locations along the x-axis.

2013). The search for the best shapelet requires examining *all* subsequences of *all* lengths from *all* the time series in the training set, and for each shapelet calculating its distance to each time series at *all* possible locations. Even for small data sets, this process has a time scale of days, and for large data sets, the time scale becomes one of years. Hence, the original implementation on commonly available hardware is only practical on the smallest of data sets. Additionally, for high-throughput applications, the classification time may be prohibitively expensive for long time series. This is because at each node of the tree, all possible matches between the node's shapelet and the time series to be classified need to be examined.

1.1. Our Contributions

Our goal was to reduce both learning and classification time without impairing the accuracy of the induced model by exploiting a feature common to spectroscopic data – the localized nature of information in the time series. In the YK-algorithm, no importance is attributed to the location from which the shapelet was extracted. Hence, the best match between a shapelet and a time series is searched for anywhere in the time series. We observed that for many data sets from the field of spectroscopy, time series from the same class show similar behavior patterns at similar locations along the frequency axis. Fig. 1 presents examples of two data sets which strongly support this insight. Based on this insight, we propose a new property as part of the definition of a shapelet, derived from the location in the time series from which the shapelet was extracted. This property limits the scope of the search for the best match of a shapelet to a time series to the vicinity of the location from which the shapelet was extracted. The assumption of locality is justified as spectroscopic measurements of items with similar properties should have very similar spectroscopic fingerprints, especially in areas characteristic of a specimen which are not expected to be contaminated. Our current implementation assumes that all time series are of equal length.

Although the time series are generally aligned, some allowance for misalignment is necessary. We therefore introduce a method for learning the misalignment characteristic of a data set. We evaluate our approach on data sets from the field of spectroscopy, and show that local-shapelets can reduce learning and classification time (especially for data sets with long time series) by over two orders of magnitude without impairing accuracy. For reproducibility, we have made all our code available online (local shapelets, 2014).

The rest of the article is organized as follows: First we present basic definitions required for understanding the article and shortly describe the YK-algorithm in Sect. 2. Then we present related work (Sect. 3), followed by a description of local-shapelets and our proposed method for determining the range to examine (Sect. 4). Next we present our experimental evaluation (Sect. 5) followed by a brief statistical analysis, which provides a theoretical justification for our local-shapelets approach (Sect. 6). Finally, we summarize our results and present additional research directions to pursue (Sect. 7).

2. Background

Here we present a number of definitions necessary for the proper understanding of this article and a short description of the original shapelet algorithm as it is the basis of our work.

2.1. Definitions

Definition 1. *A time series T of length m is a series of m consecutive equally spaced measurements:*

$$T = t_0, t_1, ..., t_{m-1}.$$

Definition 2. *A subsequence S of length k extracted from time series T of length m at index i such that $k \leq m$ is a series of k consecutive measurements:*

$$S = t_i, t_{i+1}, ..., t_{i+k-1}.$$

Definition 3. *The Euclidean distance between two time series T, R of length m is:*

$$d_E(T, R) = \sqrt{\sum_{i=0}^{m-1} (t_i - r_i)^2}.$$

Definition 4. *The Euclidean distance between a time series T of length m and a subsequence S of length k such that $k \leq m$ is:*

$$d_E(T, S) = \min_i \; d_E(T[i : i + k - 1], S) \;\; {\scriptstyle i \in [0, m-k].}$$

This is the minimal distance between S and all subsequences of length k in T.

Definition 5. *Given a data set D, with c classes and n examples, each class i with n_i examples, the entropy is:*

$$Ent(D) = -\sum_{i=0}^{c-1} \frac{n_i}{n} \log \frac{n_i}{n}.$$

Intuitively, a data set's entropy is a measure of its class-homogeneity. Larger homogeneity corresponds to smaller entropy values. Specifically, the smallest entropy value (0) corresponds to a data set in which all members belong to the same class.

Definition 6. *Given a data set D with n examples, split into two subsets D_1 and D_2, containing n_1 and n_2 examples respectively, such that $D_1 \cup D_2 = D$ and $D_1 \cap D_2 = \varnothing$ the information gain (IG) is:*

$$IG(D, D_1, D_2) = Ent(D) - \left(\frac{n_1}{n} Ent(D_1) + \frac{n_2}{n} Ent(D_2)\right).$$

Intuitively, information gain is a measure of the class-homogeneity induced by a split of data set D. The larger the information gain, the larger the decrease in entropy of the split data set w.r.t. D, in turn implying better class-homogeneity. As we will soon see, each shapelet induces a data set split and its effectiveness is measured by the split's information gain.

2.2. The YK-Algorithm

For completeness, we briefly present the YK-algorithm as presented in Ye and Keogh (2011a). First, we present the algorithm for two classes; we then extend the description to a multi-class data set.

Let D be a dataset with two classes and n time series. The YK-algorithm examines all possible subsequences of every length (from a minimal length, usually 3, to a maximal length which is usually the length of the shortest time series) from every time series. For each subsequence S, the distance to each time series is calculated, as defined in Definition 4. Then, the time series are ordered by their distance from S. Using this induced order, the average distance of every two adjacent time series to S is calculated. We will refer to this average distance as the *splitting distance*. Each of the n splitting distances defines two subsets, one containing all time series with a distance to S smaller than or equal to the splitting distance, and the other containing all time series with a distance to S greater than the splitting distance. For every possible split into two subsets, the information gain is calculated (see Definition 6). If the current information gain is better than the best so far, the shapelet is kept along with the corresponding splitting distance. Tie breaking is done by keeping the shapelet which induces a larger average distance between the two subsets which is referred to as the *margin*. After checking all possible subsequences, the best shapelet and the corresponding splitting distance are returned.

This method can be easily extended to a multi-class problem by building a tree, with a shapelet and splitting distance in each node. A new node receives one of the two subsets created by the shapelet found by the node above it, and learns the best shapelet and splitting distance for this subset of time series. The stopping criteria for this recursive algorithm is that all the time series in the subset be of one class.

Two important implementation issues are that all distance calculations are computed after local normalization (Goldin and Kanellakis, 1995) and that the margin is normalized, by dividing it by the length of the subsequence.

Classification of a time series T is accomplished by traveling down the tree. At each node the distance of the shapelet S associated with the node to T is calculated. The node decides to which of its child nodes T should be directed, depending on whether its distance from S is smaller or greater than the splitting distance. When T reaches a leaf, it is assigned the class associated with this leaf.

2.2.1. *Time Complexity of the YK-Algorithm*

Let m denote the length of a time series and let us assume all time series are of equal length. Assuming all shapelet lengths from 3 to m are examined, the number of different shapelets to examine in a single time series is $\sum_{i=3}^{m} i = O(m^2)$. Let n denote the number of time series in data set D. The number of shapelets to examine in the entire data set is $O(nm^2)$.

When searching for the minimal distance between a shapelet S of length k and a time series T, the distance of S to all subsequences of length k in T needs to be calculated (see Definition 4). The time complexity of this operation is $O(m^2)$. Calculating the distance of S to all time series requires $O(nm^2)$ calculations. The total number of calculations for all shapelets and time series is $O(n^2m^4)$ which explains the formidable time required for learning a model even for small data sets.

3. Related Work

The time complexity of the YK-algorithm is formidable (see Sect. 2.2.1) leading to a large number of attempts to reduce it. As we will show in this section, none of the previous approaches utilized the location from which the shapelet was extracted to reduce the time required to learn a model. In addition, most of these approaches do not reduce the time required to classify a time series.

The first attempt to reduce the time complexity of the YK-algorithm was introduced in the paper first presenting shapelets (Ye and Keogh, 2011a). Two optimizations were suggested. The first optimizes the distance calculation of a shapelet to a time series. The distance calculation is terminated if it exceeds the minimum distance found so far between the current shapelet and time series. This optimization was coined *early-abandon*. The second optimization (named *entropy-pruning*) checks whether the most optimistic IG possible, given the distances of a shapelet to time series already computed, can be better than the best IG found so far. If the IG cannot be improved, the shapelet under examination is discarded. As pointed out by Lines et al. (2012) this optimization requires testing O(2^c) different possibilities (c is the number of classes in the data set), which can greatly reduce the effectiveness of this optimization when the number of classes is large.

Later, Mueen et al. (2011) introduced additional optimizations. One optimization manages to compute the distance of a shapelet to a time series in constant time by precomputing necessary statistics. This optimization manages to reduce the time complexity to $O(n^2m^3)$. A major downside is that for each two time series, a matrix of size m^2 needs to be maintained. This leads to a total space complexity of $O((nm)^2)$ which is untenable for large data sets (Gordon et al., 2015; Rakthanmanon and Keogh, 2013). A second optimization discards shapelets similar to shapelets that were already discarded. A disadvantage of this approach is the large time overhead when applied to data sets with a large number of classes.

Two recent attempts managed to dramatically reduce the time complexity for learning a shapelet based model. The first method (Rakthanmanon and Keogh, 2013) quickly picks out a small number of shapelets from each shapelet length which seem able to effectively divide the data set into its classes. Then only this subset of shapelets are fully analyzed. This approach manages to reduce the time complexity to $O(nm^3)$ but requires a considerable amount of space to accommodate this reduction in time complexity. We will refer to this solution as the *hashing-algorithm*. A second approach (Gordon et al., 2015), named SALSA-R, randomly samples a constant number of shapelets (10,000) which are examined, reducing the time complexity to $O(10,000 \times nm^2)$ with no excess space requirements.

As shown, none of the aforementioned optimizations utilize the location from which the shapelet was extracted to reduce the time complexity of the learning process. In addition none of them significantly reduces classification time.

Xing et al. (2011) introduced an optimization for fast classification of streaming time series with the aid of shapelets. Their main idea was to prefer shapelets which appear early in a time series over shapelets which appear later. This ensures that time series can be classified quickly once initial measurements have arrived with no need to wait for further measurements. They coined this new type of shapelets as local-shapelets. Although we both name our shapelets similarly, the concepts differ. The motivation of Xing et al. was to classify streaming time series as early on as possible while ours is to optimize learning and classification time of non-streaming data sets. Also, the implementations differ. Xing et al. did not preserve the location from which the shapelet was extracted. Conversely, with our method, the location from which the shapelet was extracted is exploited with no limitation on the location from which to extract a shapelet.

4. Local-Shapelets

In the material that follows, the basic idea of local-shapelets is described as well as modifications which make it useful in practice.

Definition 7. *A shapelet is a tuple* $<\vec{S}, d>$. $\vec{S}$ *is a series of consecutive measurements extracted from one of the time series in the data set and* d *is a cutoff distance. Time series with a distance to* S *smaller or equal to* d *traverse one side of the tree while time series with a distance to* S *greater than* d *traverse the other side of the tree.*

Definition 8. *A local-shapelet is a tuple* $<\vec{S}, d, i>$. $\vec{S}$ *and* d *are as in Definition 7.* i *is the location from which the local-shapelet was extracted.*

Unlike a shapelet, a local-shapelet contains information regarding the location from which it was extracted, which is utilized when calculating the distance of the local-shapelet to a time series.

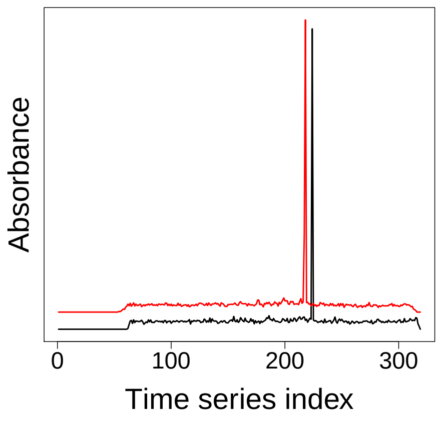
Definition 9. *The Euclidean distance between a time series* T *of length* m *and a local-shapelet* $<\vec{S}, d, i>$ *of length* k *such that* $k \leq m$ *given* x *which defines a range around* i *is:*

$$d_E(T, S) = \min_j \; d_E(T[j : j + k - 1], S) \quad {}_{j \in [max(0, i-x), min(i+x, m-k)]}.$$

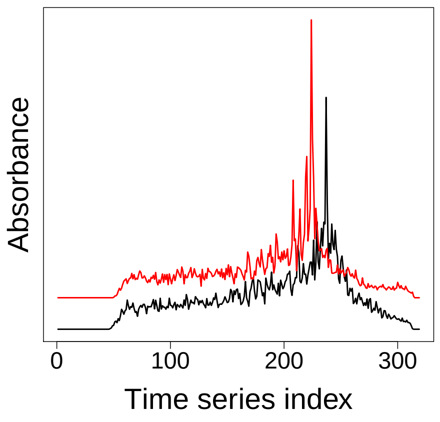
The distance between T and a local-shapelet $<\vec{S}, d, i>$ adds a constraint on the subsequences of T to which the distance of subsequence $\vec{S}$ is calculated. Instead of calculating the distance of $\vec{S}$ to all subsequences of length k in T, the distance of $\vec{S}$ is calculated only to subsequences in the vicinity of the location i from which $\vec{S}$ was extracted. This vicinity is defined by the constant x. Thus, the time required for calculating the distance of a shapelet to a time series is reduced.

4.1. The Tolerance Range

A naïve implementation of local-shapelets is to calculate the distance of a shapelet only to the single subsequence in the time series at the exact location i from which it was extracted. This approach may have detrimental impact on the learning process as exemplified in Fig. 2 which presents two time series from the same class of the data set Lightning7. As is clearly shown, the characteristic spike may appear at slightly different locations. Restricting the distance calculation



(a) Example from class 2



(b) Example from class 3

Figure 2: Two examples from different classes of the Lightning7 data set illustrating the need for a tolerance range. Without a tolerance range the spike characteristic of each class cannot be utilized as its location is different in different time series.

of a shapelet to the exact location from which it was extracted would cause characteristic patterns to be overlooked. To accommodate this issue some tolerance, which we will refer to as the *radius*, needs to be added to the index i, such that the distance of a local-shapelet to a time series T is the minimum distance to all subsequences starting in the range $[i - radius, i + radius]$ (see definition 9). We will refer to this range as the *tolerance range*.

In food spectroscopy there are many sources of noise, such as differences in the residual water content of the freeze-dried samples (Briandet et al., 1996). This noise may cause distortions in the pattern generated but will not cause a shift in the frequencies emitted, therefore we set the tolerance range to 0. In lightning spectroscopy, interferences such as frequency-dependent dispersion induced by the ionosphere (Moore et al., 1995) may lead to a shift in the frequencies measured requiring the introduction of a tolerance range greater than 0.

We present a method for computing the value of the tolerance required, as described in Procedure 1. This is achieved by experimentally observing the tolerance required for a small number of subsequences. Procedure 1 splits each time series into ten equal consecutive and disjoint subsequences (line 1). From each class (C) and for each subsequence location, a single subsequence (*subseq*) is extracted from a randomly selected time series (*rand-ts*) (lines 2-6). Then the distance of *subseq* to all time series in C is calculated using the global method for calculating distances (see Definition 4). The location of the best match of *subseq* to each time series is recorded in *locations* (line 8).

Before utilizing the information available in the list of locations, it is necessary to filter out values which are obviously non-characteristic of the data set (line 9). A major motivation is that a larger tolerance leads to an increase in learning and classification time as more distance calculations are required for each shapelet. Filtering out locations which are atypical should not impair accuracy significantly. For example, let us suppose we receive the following list of locations: 1,30,30,31,32,34,37,38,40,40,41,81. It is quite clear that the values 1,81 are outliers

Procedure 1 Algorithm for computing the tolerance characteristic of a data set

```
Compute-Tolerance($D$)  {Input is a data set}
 1: start-indices ← indices, s.t. time series will be split into 10 subsequences which are as equal
    in length as possible
 2: for each class $C$ in $D$ do
 3:     for each start-index $i$ in start-indices do
 4:         subseq-length ← length of subsequence
 5:         rand-ts ← randomly selected time series from $C$
 6:         subseq ← rand-ts[$i$ : $i$+subseq-length-1]
 7:         for each time series $t$ in class $C$ do
 8:             locations ← append(location-of-best-distance($t$, subseq))
 9:         locations ← outlier-filter(locations) {Using IQR-filtering}
10:         $Radius_{c,i}$ ← max(locations) - min(locations)
11:     $Radius_c$ ← min($Radius_{c,i}$)
12: tolerance ← max($Radius_c$)
13: return tolerance
```

and should be filtered out. Without filtering, the size of the range is 81, while after filtering, the size of the range is only 12.

Our method for outlier filtering is based on the interquartile range (IQR). Using this method, the first (Q_1) and third (Q_3) quartiles are calculated and the IQR is calculated as $IQR = Q_3 - Q_1$. All values greater than $Q_3 + 3 \times IQR$ or smaller than $Q_1 - 3 \times IQR$ are filtered out. Although it is customary to use 1.5 as the multiplication factor, we chose a multiplication factor of 3 so as to not filter out too many values which may lead to overfitting. Our tuning of the multiplication factor to a value of 3 is a heuristic as it cannot guarantee total avoidance of overfitting, because other parameters such as the model complexity also play a part. Three advantages of IQR filtering are that it is simple, that it is a-parametric and that it does not automatically drop extreme values if they are similar to the rest.

Once we have filtered out the outliers, the characteristic radius as reflected by this subsequence of class C is calculated and recorded (line 10). After a radius for each of the subsequences of a class has been calculated, the minimum of all these radii is selected to represent the locality of the class (line 11). We chose the minimal radius as this leads to the smallest number of distance calculations of a shapelet to a time series, which promises the best possible speedup in runtime. The last stage is the selection of a single radius as the tolerance for the data set. We chose the maximum radius from all classes (line 12) as the tolerance must accommodate the most loosely localized class. Otherwise, for some of the classes, the ultimate matches may reside outside of the recommended range and will not be examined.

The time complexity of this phase is negligible as only 10 subsequences per class are examined and the distance of each subsequence is calculated only to time series of its class.

4.2. Random Selection of Shapelets

For completeness, we present the procedure used by SALSA-R for randomly selecting shapelets in Procedure 2. We chose a distribution similar to the uniform distribution but simpler to implement.

Procedure 2 describes the method for randomly selecting the next shapelet to examine. First (line 2), the time series from which to extract the shapelet is chosen. Then (lines 3-4), the

Procedure 2 Algorithm for randomly selecting shapelets

```
extract-shapelet(D,min-sh-length)  {Input is a data set and the minimum length of a shapelet}
1: num-ts ← number-of-time-series-in-data-set(D)
2: ts-index ← random-selection(0,num-ts)
3: highest-index ← times-series-length - min-sh-length + 1
4: sh-index ← random-selection(0,highest-index)
5: longest-possible-shapelet ← times-series-length - sh-index +1
6: sh-length ← random-selection(min-sh-length,longest-possible-shapelet)
7: sh ← extract-shapelet(D,ts-index,sh-index,sh-length)
8: return sh
```

index in the time series from which to extract the shapelet is randomly generated. The range of possible indices is between 0 and the last index from which the shortest possible shapelet can be extracted. Last (lines 5-6), the length of the shapelet is randomly selected. The upper limit on the length of the shapelet (longest-possible-shapelet) is calculated based on the location from which the shapelet is to be extracted (line 5). The function *random-selection(a,b)* randomly selects an integer from the range [a,b-1] with a uniform distribution.

5. Experimental Results

Our goal is to show that for data sets from the field of spectroscopy, the usage of local-shapelets reduces the time complexity during both training and classification phases in comparison with global-shapelets (i.e., non-local shapelets) without degrading accuracy. First, we present the data sets from the field of spectroscopy with which we evaluated local-shapelets. Then, we establish the utility of our method (Procedure 1) for calculating the tolerance range. In the next phase of our evaluation we compare local-shapelets vs. global-shapelets within the YK-algorithm. In the last phase, we re-implement SALSA-R to utilize the locality of shapelets. We compare accuracy and run-time with the original implementation of SALSA-R and the hashing-algorithm. We ran all experiments on an Intel Xeon E5620 computer comprising two 2.40GHz processors with 24GB of RAM and with a 64-bit version of Windows Server 2008 R2 as the operating system.

5.1. Description of Data Sets

Our experiments were conducted on a collection of 6 data sets from the field of spectroscopy, available online (Ye and Keogh, 2011b; Keogh et al., 2014). The collection contains four data sets from the field of food spectroscopy (Beef, Coffee, OliveOil, Wheat) and two from the field of lightning spectroscopy (Lightning2, Lightning7). Table 1 contains information on the number of examples in the training and test sets, the number of classes, the length of the time series and the tolerance range used. All data sets were already split into train and test sets. We preserved the original division to train and test sets to allow easy reproduction of our results, as well as a fair comparison with other published results.

As the test sets are very small, our initial measurements of classification times were inaccurate due to minor overheads which dampened the effect of locality on the outcome and due to the inaccuracy of computer time measurements at small time scales. We solved this by enlarging each test set to a size of 1GB. This was done by duplicating examples. To ensure that the accuracy obtained would be identical to that on the original data set, we duplicated each example an equal number of times. As the classifier is deterministic, the classification of an example will always be identical no matter how many times it appears. Therefore, the proportion of correct classifications out of all classification examples will not change and the accuracy will remain the same.

Table 1: Description of the data sets

dataset	train set size	test set size	num. classes	time series length	tolerance
Beef	30	252,510	5	470	0
Coffee	28	383,264	2	286	0
Lighting2	60	106,140	2	637	0
Lighting7	70	209,510	7	319	78
OliveOil	30	208,770	4	570	0
Wheat	49	115,434	7	1,050	0

5.1.1. Food Spectrographs

Beef. The beef data set (Al-Jowder et al., 2002) contains the spectral absorbance of one type of beef cut (silverside). One class contains the spectral absorbance of the beef cut without any contaminates. Each of the other four classes contains the spectral absorbance of the beef cut contaminated with a different type of offal (kidney, liver, heart and tripe) which is cheaper than the declared beef cut.

Coffee. The coffee data set (Briandet et al., 1996) contains the spectral absorbance of instant coffee of two different types of coffee beans *Arabica* and *Robusta*. Coffee from *Arabica* beans is more highly estimated as it has a finer and more pronounced taste. Approximately 90% of world coffee production is from *Arabica* and another 9% is from *Robusta*. As the price of *Arabica* is higher than that of *Robusta*, it is important to be able to distinguish between them even after the long process that is required to produce instant coffee.

OliveOil. Olive oil samples from four different European countries (Greece, Italy, Portugal and Spain) were collected (Tapp et al., 2003). The classification task is to be able to discern the country of origin using the spectrograph of the olive oil sample.

Wheat. This data set (Ye and Keogh, 2011a) consists of spectrographs of wheat grown during the years 1998-2005. There are a number of different types of wheat in the data set but the class was assigned based only on the year in which the wheat was grown and not on the type of wheat.

5.1.2. Lightning Spectrographs

Data on frequencies emitted during lightning events were collected and then a Fourier transform was applied to produce spectrographs (Eads et al., 2002). The lightning events were categorized into 7 different classes differing in the charge of the lightning (positive or negative), whether the event was gradual or abrupt and whether the event was intra-cloud or from cloud to ground. The original authors of this data set (Eads et al., 2002) note that there is a large inter-class variation and intra-class similarity. The data set Lightning7 contains examples of all 7 classes, while Lightning2 is a simpler binary problem of distinguishing between cloud to ground events and intra-cloud events.

5.2. Utility of Tolerance Range Calculation

The last column of Table 1 presents the tolerance range used during our experiments. For the four data sets of food spectroscopy, the value of the tolerance range was not calculated using Procedure 1; rather we set it to 0 based on prior knowledge in this domain. A comparison of the tolerance range based on prior knowledge with those recommended by Procedure 1 shows large agreement. For three data sets (Coffee, OliveOil and wheat) values are identical. For the Beef data set, although the calculated tolerance range was not exactly 0, it was very close with a value of 2. Our procedure also succeeds when a tolerance range greater than 0 is required. When applied to the Lightning7 data set for which it is clear a tolerance range larger than 0 is required, as shown in Fig. 2, the calculated tolerance range is 78. These findings show that our method for predicting the tolerance range manages to successfully estimate the required range.

5.3. Local YK-Algorithm

Here we show that local-shapelets reduce the time consumption without impairing accuracy. We compare local and global shapelets within the YK-algorithm framework which is the initial implementation of the shapelet algorithm which examines all possible shapelets. Due to the large time and space requirements of the YK-algorithm we could not collect results for the Lightning2 data set.

For these experiments, we used the original code used by Ye and Keogh (2011a). As pointed out by Hills et al. (2013), the entropy-pruning optimization (see Sec. 3) has an overhead which grows exponentially with the increase in the number of classes in the data set. We encountered this experimentally with the wheat data set which has 7 classes. The original implementation did not finish examining all shapelets for the first node of the tree after 5 days, while the same implementation without the entropy-pruning optimization finished learning the whole tree in this period of time. Therefore we conducted the experiments using the original code without the entropy-pruning optimization.

Results of our experiments are presented in Table 2. Each two columns compare results of global-shapelets vs. local-shapelets. The first comparison is the accuracy achieved, the second is the time required to learn a model and the third is the time required to classify the test set. In all measures, the local approach outperforms the global approach. The average improvement in accuracy is 8%, the average speedup during the learning phase is 9.5 and during the classification phase it is 80.

Table 2: Global YK-algorithm vs. Local YK-algorithm

data set	accuracy (%)		learning time (sec)		classification time (sec)	
	global	local	global	local	global	local
Beef	46.67	56.67	24,307	1,364	76.03	1.10
Coffee	96.43	92.86	1,466	407	36.79	1.54
Lighting7	43.84	53.42	98,636	67,660	109.75	59.58
OliveOil	33.33	50.00	17,448	2,287	46.85	1.01
Wheat	57.71	65.43	433,630	25,221	158.36	0.59

5.4. Local-SALSA-R

In this set of experiments we re-implemented SALSA-R to use local-shapelets instead of global-shapelets. The number of shapelets randomly selected was set to 10,000 which was found to be an optimal number by Gordon et al. (2015). We compared local-SALSA-R with global-SALSA-R and the hashing-algorithm. We repeated our experiments thirty times as each of the three methods includes an element of randomness.

Table 3 presents a comparison of the average accuracy of local-SALSA-R with global-SALSA-R and the hashing-algorithm. We applied a Friedman test (Friedman, 1937) to test if there is a significant difference between accuracy attained by the different methods. The p-value was 0.85, which clearly shows that there is no significant difference in the accuracy achieved by any of the methods.

Table 3: Comparison of accuracy of local-SALSA-R with that of global-SALSA-R and the hashing-algorithm

data set	global-SALSA-R	hashing-algorithm	local-SALSA-R
Beef	51.78	50.78	62.11
Coffee	94.17	92.74	97.02
Lighting2	67.98	67.22	66.17
Lighting7	58.45	61.62	59.13
OliveOil	73.11	73.33	73.33
Wheat	66.72	69.94	66.46

A comparison of average learning and classification times is presented in Table 4. A Friedman test on the learning and classification times shows that there is a significant difference between the methods during both the learning (p-value = 0.0057) and classification phases (p-value = 0.030). A one sided Wilcoxon-test (Wilcoxon, 1945) affirms our claim that local-shapelets are significantly faster than global-shapelets during the learning phase (p-value = 0.016 for both global-SALSA-R and the hashing-algorithm) and the classification phase (p-value = 0.016 for global-SALSA-R and p-value = 0.031 for the hashing-algorithm). On average, local-SALSA-R reduces the time required to learn a model by a factor of 120 and 440 in comparison with global-SALSA-R and the hashing-algorithm, respectively. During classification, local-SALSA-R is 180 times faster than global-SALSA-R and 110 times faster than the hashing-algorithm, on average.

Table 4: Comparison of learning and classification time of local-SALSA-R with that of global-SALSA-R and the hashing-algorithm

data set	learning time (sec)			classification time (sec)		
	global-SALSA-R	hashing-algorithm	local-SALSA-R	global-SALSA-R	hashing-algorithm	local-SALSA-R
Beef	44.98	169.77	0.47	87.46	104.87	0.51
Coffee	16.15	12.81	0.26	31.61	34.06	0.47
Lighting2	164.09	539.11	1.00	144.78	86.10	0.85
Lighting7	56.51	205.17	35.39	152.61	95.97	111.30
OliveOil	59.39	114.37	0.50	65.52	45.27	0.51
Wheat	315.19	1693.48	1.17	231.46	78.71	0.42

Analytically, it is easy to argue that the longer the time series, the greater the time saved by using local-shapelets, as the number of distance calculations avoided increases. We confirmed this experimentally as can be seen in Fig. 3. The figure shows the ratio between the time required by global-SALSA-R and local-SALSA-R as a function of the length of the time series for all data sets for which the tolerance range used was 0 (all data sets apart from Lightning7). This figure clearly confirms that the ratio increases with the length of the time series. We chose to compare local-SALSA-R with global-SALSA-R and not with the hashing-algorithm as their implementation apart from the aspect of locality is identical, allowing isolation of locality as the only parameter influencing the outcome.

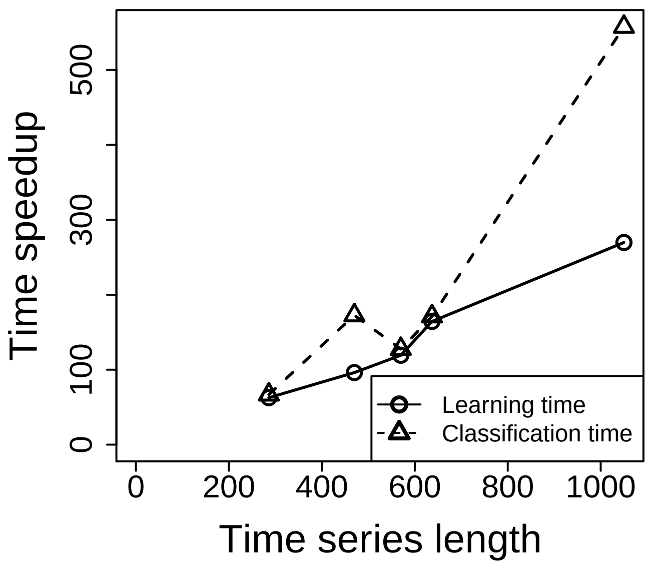


Figure 3: Each point is the ratio of the time required by global-SALSA-R and local-SALSA-R. One plot is for learning times and the second for classification times. As can be seen the ratio increases with the length of the time series.

6. Statistical analysis

The approach proposed in this paper was mainly motivated by algorithmic considerations: restricting the search to a small subset of the possible shapelet locations significantly speeds up both training and classification. In this section, we will argue that as a by-product, our approach offers statistical advantages as well. By restricting the number of features, we are constraining the complexity of the hypothesis class. As we show below, hypothesis classes of low complexity require fewer training examples to attain a certain accuracy level.

The argument is made precise in the language of learning theory. A learner faced with a *classification* task is trying to learn a function $g : \mathcal{X} \to \{-1, 1\}$, where $\mathcal{X}$ is the instance space (in our case, it is the set of all possible time series). The learner gets to observe example-label pairs $(X_i, Y_i) \in \mathcal{X} \times \{-1, 1\}$ generated iid from some unknown distribution P over $\mathcal{X} \times \{-1, 1\}$. This corresponds to the intuition that the training labels may be noisy, and indeed, there may be no "correct" classifier $g : \mathcal{X} \to \{-1, 1\}$ that achieves perfect accuracy. Although there are universal approximators capable of fitting arbitrary labeled samples, if unconstrained they will necessarily overfit (Devroye et al., 1996). Hence, when choosing the learning model, its richness (i.e., hypothesis complexity) must be taken into account.

The learner's n observed labeled examples (X_i, Y_i) constitute the *training set*, based on which it will produce a *hypothesis* $g : \mathcal{X} \to \{-1, 1\}$. We will denote by $\mathcal{H}$ the collection of all admissible hypotheses (formally, $\mathcal{H} \subset 2^{\mathcal{X}}$) and associate with every $h \in \mathcal{H}$ two key quantities: its *sample* (or training) error,

$$\widehat{\mathrm{err}}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{h(X_i)\neq Y_i\}}$$

and *generalization error*,

$$\mathrm{err}(h) = \mathbb{E}[\mathbf{1}_{\{h(X_i)\neq Y_i\}}] = \mathbb{P}(h(X) \neq Y).$$

In words, $\widehat{\mathrm{err}}(h)$ is the relative fraction of mistakes that h makes on the training set while $\mathrm{err}(h)$ is the probability that h makes a mistake on a freshly drawn (X, Y) pair — crucially, drawn from the same distribution used to generate the training set. Note that while the typical goal is to guarantee a small generalization error, the latter quantity cannot be computed without knowledge of the sampling distribution. Instead, the readily computable $\widehat{\mathrm{err}}(h)$ may be used (under certain conditions, detailed below) as a proxy for $\mathrm{err}(h)$.

In this setting, the learner's task is twofold: (i) **algorithmic**: efficiently find an $h \in \mathcal{H}$ for which $\widehat{\mathrm{err}}(h)$ is small, and (ii) **statistical**: guarantee that, with high probability, $\mathrm{err}(h)$ will not be much greater than $\widehat{\mathrm{err}}(h)$, regardless of which $h \in \mathcal{H}$ the learner chooses. The foregoing sections were devoted to the algorithmic aspects, and we shall focus on the statistical one here. In the case of finite $\mathcal{H}$, a particularly simple connection exists between $\mathrm{err}(h)$ and $\widehat{\mathrm{err}}(h)$:

Theorem 1 (Mohri et al. (2012))**.** *Suppose that $|\mathcal{H}| < \infty$ and the learner observes a training set consisting of n examples. Then, for any $\delta > 0$, we have that*

$$\mathrm{err}(h) \quad \leq \quad \widehat{\mathrm{err}}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log(1/\delta)}{2n}} \tag{1}$$

holds with probability at least $1 - \delta$, uniformly over all $h \in \mathcal{H}$.

For simplicity, let us consider the case where $\mathcal{H} = 2^{\mathcal{X}}$ (i.e., $\mathcal{H}$ consists of all possible binary functions). In this case, $|\mathcal{H}| = 2^{|\mathcal{X}|}$, and hence even a modest reduction of the instance space — by reducing the feature set, for example — can have a noticeable effect on the second term in the right-hand side of (1), and hence yield a faster convergence rate. The basic features used in this paper are distances from a shapelet to a time series. It is precisely this feature set that gets reduced when our algorithm considers only a subset of the possible locations. This observation provides a statistical justification to our local-shapelets approach, in addition to the algorithmic speedup.

7. Conclusions

The objective of our investigation was to utilize the localization of characteristic patterns in spectrographic measurements using the shapelet algorithm, which has previously been found to be suited (Hills et al., 2013) for this domain. Our adaption to the shapelet algorithm reduces the number of distance calculations and thus shortens the time required to train a model and classify examples.

As pointed out by Ye and Keogh (2011a), one important advantage of the shapelet approach is its interpretability, i.e., the process extracts informative subsequences representative of each class. The chosen shapelets provide insights into patterns characteristic of each class. One such example can be seen in Fig. 4. The shapelet, shown as a dashed red line, is overlaid on each of the two time series at the location from which it was extracted and represents a pattern characteristic of class 2. In addition to identifying the discriminative pattern, local-shapelets also identify the discriminative frequencies. We compared the frequencies found to be discriminative by the shapelet with those found to be discriminative by Briandet et al. (1996) and found that they coincide.

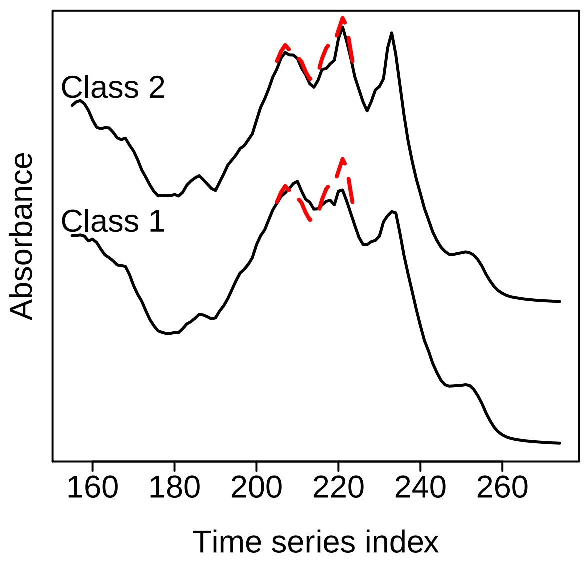


Figure 4: Interpretability of local-shapelets. Two time series from the coffee data set are presented in black. Each time series is from a different class. Only part of each time series is presented so as to focus on the important details. Overlaid on each of the time series is the shapelet selected by local-SALSA-R as most discriminative appearing as a dashed red line. As can be seen the shapelet represents time series from class 2 (Robusta coffee beans).

The algorithm we presented searches for matches of a shapelet on a time series only in the vicinity of the location from which the shapelet was originally extracted. We proposed an algorithm (Procedure 1) for calculating the exact vicinity based on properties of each class in the data set. Our main result is that local-shapelets can indeed speedup both the learning and classification methods 100-fold when using SALSA-R without impairing accuracy. We also show that our estimation of the vicinity to examine is quite accurate.

This research can be extended in many ways. It may be interesting to explore possible trade-offs between speedup and accuracy as a function of the vicinity to examine. Another research direction is the application of local-shapelets to domains other than spectroscopy which may also require the adaption of our algorithm for time series of different lengths.

References

R. Herrmann, C. Onkelinx, Quantities and units in clinical chemistry: Nebulizer and flame properties in flame emission and absorption spectrometry (Recommendations 1986), Pure and Applied Chemistry 58 (12) (1986) 1737–1742.

O. Al-Jowder, E. K. Kemsley, R. H. Wilson, Detection of adulteration in cooked meat products by mid-infrared spectroscopy, Journal of Agricultural and Food Chemistry 50 (6) (2002) 1325–1329.

R. Briandet, E. K. Kemsley, R. H. Wilson, Discrimination of Arabica and Robusta in instant coffee by Fourier transform infrared spectroscopy and chemometrics, Journal of Agricultural and Food Chemistry 44 (1) (1996) 170–174.

C. P. Bicchi, A. E. Binello, M. M. Legovich, G. M. Pellegrino, A. C. Vanni, Characterization of roasted coffee by S-HSGC and HPLC-UV and principal component analysis, Journal of Agricultural and Food Chemistry 41 (12) (1993) 2324–2328.

I. Lumley, Authenticity of meat and meat products, in: Food Authentication, Springer, 108–139, 1996.

N. Sharma, A. Srivastava, J. Gill, D. Joshi, Differentiation of meat from food animals by enzyme assay, Food Control 5 (4) (1994) 219–221.

D. R. Eads, D. Hill, S. Davis, S. J. Perkins, J. Ma, R. B. Porter, J. P. Theiler, Genetic algorithms and support vector machines for time series classification, in: International Symposium on Optical Science and Technology, International Society for Optics and Photonics, 74–85, 2002.

SCIO, `http://www.consumerphysics.com/myscio/`, 2014.

L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data Mining and Knowledge Discovery (2011a) 1–34.

J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Mining and Knowledge Discovery (2013) 1–31.

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, ACM, 1154–1162, 2011.

T. Rakthanmanon, E. Keogh, Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets, in: Proceedings of the Thirteenth SIAM Conference on Data Mining (SDM), SIAM, 668–676, 2013.

local shapelets, `ftp://www.ise.bgu.ac.il/`, 2014.

D. Goldin, P. Kanellakis, On similarity queries for time-series data: Constraint specification and implementation, in: Principles and Practice of Constraint Programming CP'95, Springer, 137–153, 1995.

J. Lines, L. M. Davis, J. Hills, A. Bagnall, A shapelet transform for time series classification, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 289–297, 2012.

D. Gordon, D. Hendler, L. Rokach, (in press) Fast and Space-Efficient Shapelets-Based Time-Series Classification, Intelligent Data Analysis 19 (5).

Z. Xing, J. Pei, S. Y. Philip, K. Wang, Extracting Interpretable Features for Early Classification on Time Series., in: Eleventh SIAM International Conference on Data Mining (SDM), SIAM, 247–258, 2011.

K. R. Moore, P. C. Blain, S. D. Briles, R. G. Jones, Classification of RF transients in space using digital signal processing and neural network techniques, in: SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics, International Society for Optics and Photonics, 995–1006, 1995.

L. Ye, E. Keogh, shapelet data sets, `http://alumni.cs.ucr.edu/~lexiangy/shapelet.html`, 2011b.

E. Keogh, Q. Zhu, B. Hu, H. Y., X. Xi, L. Wei, C. A. Ratanamahatana, The UCR Time Series Classification/Clustering Homepage, `www.cs.ucr.edu/~eamonn/time_series_data/`, 2014.

H. S. Tapp, M. Defernez, E. K. Kemsley, FTIR spectroscopy and multivariate analysis can distinguish the geographic origin of extra virgin olive oils, Journal of Agricultural and Food Chemistry 51 (21) (2003) 6110–6115.

M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (200) (1937) 675–701.

F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.

L. Devroye, L. Györfi, G. Lugosi, A probabilistic theory of pattern recognition, vol. 31 of *Applications of Mathematics (New York)*, Springer-Verlag, New York, ISBN 0-387-94618-7, 1996.

M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of machine learning, MIT press, 2012.