Classification of time series by shapelet transformation

Article *in* Data Mining and Knowledge Discovery · May 2013

DOI: 10.1007/s10618-013-0322-1

5 authors, including:

Jon Hills
University of East Anglia
8 PUBLICATIONS **1,526** CITATIONS

SEE PROFILE

Jason Lines
University of East Anglia
29 PUBLICATIONS **3,886** CITATIONS

SEE PROFILE

James Mapp
University of East Anglia
6 PUBLICATIONS **586** CITATIONS

SEE PROFILE

Anthony Bagnall
University of East Anglia
113 PUBLICATIONS **7,297** CITATIONS

SEE PROFILE

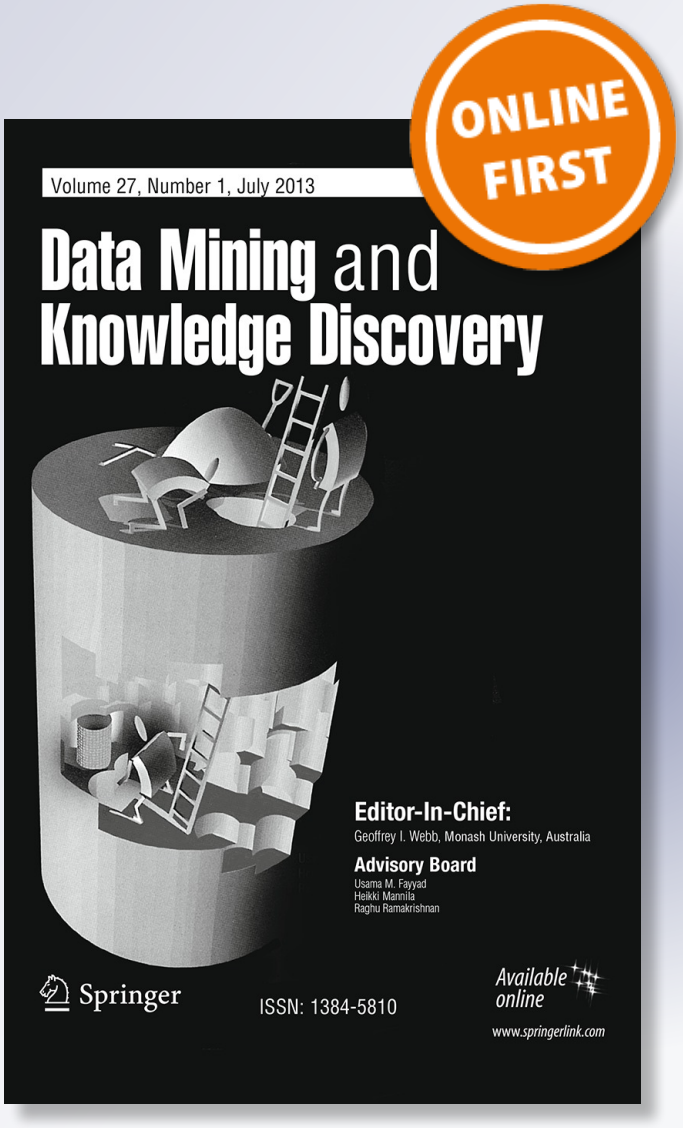
Classification of time series by shapelet transformation

Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp & Anthony Bagnall

Data Mining and Knowledge Discovery

ISSN 1384-5810

Data Min Knowl Disc
DOI 10.1007/s10618-013-0322-1



Springer

Data Min Knowl Disc
DOI 10.1007/s10618-013-0322-1

Classification of time series by shapelet transformation

Jon Hills · Jason Lines · Edgaras Baranauskas · James Mapp · Anthony Bagnall

Received: 7 February 2013 / Accepted: 6 May 2013

Abstract Time-series classification (TSC) problems present a specific challenge for classification algorithms: how to measure similarity between series. A *shapelet* is a time-series subsequence that allows for TSC based on local, phase-independent similarity in shape. Shapelet-based classification uses the similarity between a shapelet and a series as a discriminatory feature. One benefit of the shapelet approach is that shapelets are comprehensible, and can offer insight into the problem domain. The original shapelet-based classifier embeds the shapelet-discovery algorithm in a decision tree, and uses information gain to assess the quality of candidates, finding a new shapelet at each node of the tree through an enumerative search. Subsequent research has focused mainly on techniques to speed up the search. We examine how best to use the shapelet primitive to construct classifiers. We propose a single-scan shapelet algorithm that finds the best k shapelets, which are used to produce a transformed dataset, where each of the k features represent the distance between a time series and a shapelet. The primary advantages over the embedded approach are that the transformed

Responsible editor: Eamonn Keogh.

J. Hills (✉) · J. Lines · E. Baranauskas · J. Mapp · A. Bagnall
University of East Anglia, Norwich NR4 7TJ, UK
e-mail: j.hills@uea.ac.uk

J. Lines
e-mail: j.lines@uea.ac.uk

E. Baranauskas
e-mail: e.baranauskas@uea.ac.uk

J. Mapp
e-mail: j.mapp@uea.ac.uk

A. Bagnall
e-mail: anthony.bagnall@uea.ac.uk

Published online: 18 May 2013

data can be used in conjunction with any classifier, and that there is no recursive search for shapelets. We demonstrate that the transformed data, in conjunction with more complex classifiers, gives greater accuracy than the embedded shapelet tree. We also evaluate three similarity measures that produce equivalent results to information gain in less time. Finally, we show that by conducting post-transform clustering of shapelets, we can enhance the interpretability of the transformed data. We conduct our experiments on 29 datasets: 17 from the UCR repository, and 12 we provide ourselves.

1 Introduction

In time-series classification (TSC), a class label is applied to an unlabelled set of ordered data. The data need not be ordered temporally; any logical ordering is sufficient (for example, images may be represented as time series, see Fig. 1). In traditional classification problems, the order of the attributes is unimportant, and interaction between variables is considered to be independent of their relative positions. For time-series data, the order of the variables is often crucial for finding the best discriminating features. TSC research has focused on alternative distance measures for nearest neighbour (NN) classifiers, based on either the raw data, or on compressed or smoothed data (see Ding et al. 2008 for a comprehensive summary). The experimental evidence suggests that 1-NN with an elastic measure, such as dynamic time warping (DTW), is the best approach for smaller datasets; however, as the number of series increases "the accuracy of elastic measures converge with that of Euclidean distance" Ding et al. (2008). This idea has propagated through current research. For example, Batista *et al.* state that "there is a plethora of classification algorithms that can be applied to time series; however, all of the current empirical evidence suggests that simple nearest neighbor classification is very difficult to beat" Batista et al. (2011). Recently, there have been several alternative approaches, such as weighted dynamic time warping Jeong et al. (2010), support vector machines built on variable intervals Rodriguez and Alonso (2005), tree-based ensembles constructed on summary statistics Deng et al. (2011), and a fusion of alternative distance measures Buza (2011).

These approaches are focused on problems where the series from each class are observations of an underlying common curve in the time dimension. Variation around this underlying shape is caused by noise in observation, and also by noise in indexing, which may cause a slight phase shift. A classic example of this type of similarity is the cylinder-bell-funnel artificial dataset, where there is noise around the underlying shape

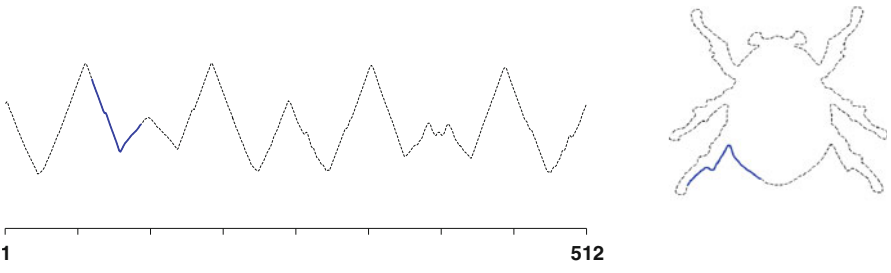


Fig. 1 A 1*D* representation (*left*) of an image (*right*). The shapelet subsequence is highlighted in *blue*

and in the index where the shape transitions. Intuitively, time-domain NN classifiers are ideal for this type of problem, as DTW can be used to mitigate noise in the indexing.

There is another set of problems involving cases where similarity in shape defines class membership. Series within a class may be distinguished by common sub-shapes that are phase independent; i.e., the defining shape may begin at any point in the series. If the underlying phase-independent shape that defines class membership is *global*, that is to say, the shape is (approximately) the length of the series, then techniques based on transformation into the frequency domain can be employed to construct classifiers Janacek et al. (2005). However, if the discriminatory shape is *local*, i.e. significantly shorter than the series as a whole, then it is unlikely that the differences between classes will be detected using spectral approaches. Ye and Keogh (2009) propose *shapelets* to address this type of problem.

A shapelet is a time-series subsequence that can be used as a primitive for TSC based on local, phase-independent similarity in shape. Shapelet-based classification involves measuring the similarity between a shapelet and each series, then using this similarity as a discriminatory feature for classification. The original shapelet-based classifier embeds the shapelet discovery algorithm in a decision tree, and uses information gain to assess the quality of candidates. A shapelet is found at each node of the tree through an enumerative search. Shapelets have been used in applications such as early classification Xing et al. (2011, 2012), gesture recognition Hartmann and Link (2010), gait recognition Sivakumar et al. (2012), and clustering Zakaria et al. (2012).

The exhaustive search for shapelets is time consuming. Thus, the majority of shapelet research has focused on techniques to speed up the search He et al. (2012), Mueen et al. (2011), Rakthanmanon and Keogh (2013), Ye and Keogh (2009, 2011). This makes the search for shapelets more tractable, but does not address the fundamental issue of how best to use shapelets to solve TSC problems. Decision trees are useful and robust classifiers, but are outperformed in many problem domains by other classifiers, such as support vector machines, Bayesian networks and random forests.

We propose a single-scan algorithm that finds the best k shapelets in a set of n time series. We use this algorithm to produce a transformed dataset, where each of the k features is the distance between the series and one shapelet. Hence, the value of the ith attribute of the jth record is the distance between the jth record and the ith shapelet. The primary advantages of this approach are that we can use the transformed data in conjunction with any classifier, and that we do not have to search sequentially for shapelets at each node. Threading the shapelet transform is an embarrassingly parallel problem.

Many of the shapelets generated by the transform are similar to one another. In terms of accuracy, this is not a problem if the classifier employed can handle correlated attributes; however, it does reduce the interpretability of the transformed dataset. To mitigate this problem, we propose a post-transform clustering procedure that groups similar shapelets. We demonstrate that this dimensionality reduction allows us to map the shapelets back to the problem domain easily, without substantially reducing the accuracy of the classifiers.

We also address how to assess shapelet quality. The similarity between each candidate shapelet and each series is measured, and this sequence of distances, with associated class membership, is used to assess shapelet quality. One implication of

constructing a recursive decision tree is that there is a need to find a split point at each node based on the sequence of distances. This requirement makes it natural to use information gain as the shapelet quality measure. However, calculating information gain for a continuous attribute requires the evaluation of all split points, and only evaluates binary splits. This introduces a time overhead, and means that, for multi-class problems, information gain may not capture fully the quality of the shapelet. In contrast, there is no need to split the data when constructing the shapelet transform. Therefore, we can base the quality measure on a hypothesis test of differences in distribution of distances between class populations. This allows us to assess multi-class problems fully, and provides some speed improvement. We experiment with measures based on the analysis of variance test for differences in means, and the non-parametric Kruskal-Wallis and Mood's median tests for difference in medians.

We conduct our experiments on 29 datasets, 17 from the UCR repository and 12 of we provide ourselves. This research is an extension of the work presented at two conferences Lines and Bagnall (2012), Lines et al. (2012). The code and datasets are available from Bagnall et al. (2012). Our contributions can be summarised as follows:

1. We describe a shapelet-discovery algorithm to find the best k shapelets in a single pass, and evaluate it with numerous experiments on benchmark time-series datasets and on new data.
2. We evaluate three alternative quality measures for use with our algorithm. Two measures were proposed for the shapelet tree in Lines and Bagnall (2012). We perform comprehensive experiments and show that the new measures offer improved speed over information gain.
3. We demonstrate that classifiers built on the shapelet-transformed data are more accurate than the tree-based shapelet classifier on a wide range of problems.
4. We compare the performance of classifiers constructed on shapelet-transformed data to classifiers constructed in the time domain.
5. We extend the original shapelet transform by using post-transform clustering to relate the shapelets to the problem domain.
6. We provide three new time-series classification problems: Beetle/Fly, Bird/Chicken, and Otoliths.

The paper is structured as follows. Section 2 provides background on time-series classification and shapelets. In Sect. 3, we define three shapelet quality measures that are an alterative to information gain. In Sect. 4, we propose a shapelet transform algorithm. We discuss the datasets used for our experiments in Sect. 5. In Sect. 6, we describe our experimental design and results, and perform qualitative analysis. Finally, in Sect. 7, we form our conclusions.

2 Background

2.1 Time-series classification

A time series is a sequence of data that is typically recorded in temporal order at fixed intervals. Suppose we have a set of n time series $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, where each time series T_i has m real-valued ordered readings $T_i =< t_{i,1}, t_{i,2}, \ldots, t_{i,m} >$,

and a class label c_i. We assume that all series in $\mathbf{T}$ are of length m, but this is not a requirement (see Hu et al. 2013 for discussion of this issue). Given a dataset $\mathbf{T}$, the time-series classification problem is to find a function that maps from the space of possible time series to the space of possible class values.

As with all time-series data mining, time-series classification relies on a similarity measure to compare data. Similarity measures can be embedded into the classifier or introduced through data transformation prior to classification. Discriminatory similarity features typically fall into one of three categories: similarity in time (correlation-based), similarity in change (autocorrelation-based), and similarity in shape (shape-based). We focus on representations that best capture similarity in shape.

If function f_c describes the common shape for class c, then a time series can be described as

$$t_{i,j} = f_{c_i}(j, s_i) + \epsilon_j,$$

where s_i is the offset for series i, and ϵ_j is some form of noise. Similarity in shape can be differentiated into global and local similarity. Global shape similarity is where the underlying shape that defines class similarity is approximately the same length as the series, but series within the same class can have different offsets. So, for example, function f_c could be sinusoidal,

$$f_{c_i}(j, s) = \sin\left(\frac{j+s}{c_i \cdot \pi}\right).$$

This form of global similarity can be detected through explicit realignment or, more commonly, through transformation into the frequency domain Janacek et al. (2005), Wu et al. (2000). However, these approaches are unlikely to work when the shape-based similarity is local. In this scenario, the discriminating shape is much smaller than the series, and can appear at any point. For example, the shape could be a sine wave embedded in noise that is triggered randomly for a short period. The function f_c would then be of the form

$$f_{c_i}(j, s) = \begin{cases} \sin\left(\frac{j}{c_i \cdot \pi}\right) & \text{if } s \leq j < s + l \\ 0 & \text{otherwise} \end{cases},$$

where l is length of the shape. It is unlikely that techniques based on Fourier transforms of the whole series will detect these embedded shapes, and spectral envelope approaches based on sliding windows are inappropriate, as we are assuming l is small. Shapelets were introduced in Ye and Keogh (2009) to measure this type of similarity.

2.2 Shapelets

A shapelet is a subsequence of one time series in a dataset $\mathbf{T}$. Every subsequence of every series in $\mathbf{T}$ is a *candidate*. Shapelets are found via an exhaustive search of every candidate between lengths min and max. Shapelets are independently normalised;

since we are interested in detecting localised shape similarity, they must be invariant to scale and offset. Shapelet discovery has three main components: candidate generation, a similarity measure between a shapelet and a time series, and some measure of shapelet quality. The generic shapelet-finding algorithm is defined in Algorithm 1.

Algorithm 1 ShapeletSelection ($\mathbf{T}$, min, max)

```
1: best ← 0
2: bestShapelet ← ∅
3: for l ← min to max do
4:    W_l ← generateCandidates(T, l)
5:    for all subsequence S in W_l do
6:       D_S ← findDistances(S, T)
7:       quality ← assessCandidate(S, D_S)
8:       if quality > best then
9:          best ← quality
10:         bestShapelet ← S
11: return bestShapelet
```

2.2.1 Generating candidates

A time series of length m contains $(m - l) + 1$ distinct candidate shapelets of length l. We denote the set of all normalised subsequences of length l for series T_i to be $W_{i,l}$ and the set of all subsequences of length l for dataset $\mathbf{T}$ to be

$$W_l = \{W_{1,l} \cup W_{2,l} \cup \cdots \cup W_{n,l}\}.$$

The set of all candidate shapelets for dataset $\mathbf{T}$ is

$$\mathbf{W} = \{W_{min} \cup W_{min+1} \cup \cdots \cup W_{max}\},$$

where $min \geq 3$ and $max \leq m$. The set $\mathbf{W}$ has $|\mathbf{W}| = \sum_{l=min}^{max} n(m - l + 1)$ candidate shapelets.

2.2.2 Shapelet distance calculations

The squared Euclidean distance between two subsequences S and R, where both are of length l, is defined as

$$dist(S, R) = \sum_{i=1}^{l} (s_i - r_i)^2$$

The distance between a time series T_i and a subsequence S of length l is the minimum distance between S and all normalised length l subsequences of T_i

$$d_{S,i} = \min_{R \in W_{i,l}} dist(S, R)$$

We compute distances between a candidate shapelet S and all series in $\mathbf{T}$ to generate a list of n distances,

$$D_S = < d_{S,1}, d_{S,2}, \ldots, d_{S,n} >$$

2.2.3 Shapelet assessment

Algorithm 1 requires a function to assess shapelet quality. Shapelet quality is based on how well the class values V are separated by the set of distances D_S. The standard approach is to use information gain (IG) Shannon et al. (1949) to determine the quality of a shapelet Mueen et al. (2011), Ye and Keogh (2009, 2011). D_S is sorted, and the IG at each possible split point sp is assessed for S, where a valid split point is the average between any two consecutive distances in D_S. For each possible split point, IG is calculated by partitioning all elements of $D_S < sp$ into A_S, and all other elements into B_S. The IG at sp is calculated as

$$IG(D_S, sp) = H(D_S) - \left(\frac{|A_S|}{|D_S|} H(A_S) + \frac{|B_S|}{|D_S|} H(B_S) \right)$$

where $|D_S|$ is the cardinality of the set D_S, and $H(D_S)$ is the entropy of D_S,

$$H(D_S) = -\sum_{v \in V} p_v \log p_v$$

The IG of shapelet S, IG_S, is calculated as

$$IG_S = \max_{sp \in D_S} IG(D_S, sp)$$

The fact that the IG calculation requires sorting D_S and then evaluating all split points introduces a time overhead of $O(n \log n)$ for each shapelet, although this is generally trivial in comparison to the time taken to calculate D_S, which is $O(nml)$.

We propose three new quality measures in Sect. 3; our experimental results (Sect. 6.1) show that the *F-stat* measure is significantly faster, and more discriminative, than IG.

2.2.4 Speed-up techniques

Since the shapelet search is enumerative, there are $n(m - l + 1)$ candidates for any given shapelet length l. Finding the distances D_S for a single candidate requires a scan along every series, performing $O(m)$ distance function calls, each of which requires $O(l)$ pointwise operations. Hence, the complexity for a single shapelet is $O(nml)$, and the full search is $O(n^2 m^4)$. It is perhaps unsurprising that the majority of research into shapelets has focused on speed-up techniques for shapelet discovery. Three types of speed-up technique have been proposed.

Early abandon of the distance calculations for shapelet S and series T_i. Since $d_{S,i}$ is a minimum of the $m - l + 1$ subsequence distances between S and T_i, individual calculations can be abandoned if they are larger than the best found so far. Further speed improvements, proposed in Rakthanmanon et al. (2012), can be achieved by normalising subsequences during the distance calculation, and by reordering the candidate S to compare the largest values first. Algorithm 2 summarises these improvements.

Algorithm 2 Similarity search with online normalisation and reordered early abandon

Input: A time series $T =< t_1, \ldots, t_m >$ and a subseries $S =< s_1, \ldots, s_l >$, where $l < m$.
Output: Minimum distance between S and all l length subseries in T.

$S' \leftarrow$normalise$(S, 1, l)$
$A \leftarrow$sortIndexes(S') {A_i *is the index of the* i^{th} *largest absolute value in* S'}
$F \leftarrow$ *normalise*$(T, 1, l)$
$p \leftarrow 0$, $q \leftarrow l$ { p stores the running sum, q the running sum of squares}
$b \leftarrow$ dist(S, F) {*find first distance, set to best so far, b*}
{scan through all subseries}
for $i \leftarrow 1$ to $m - l$ **do**
  $p \leftarrow p - t_i$ {*update running sums*}
  $q \leftarrow q - t_i^2$
  $p \leftarrow p + t_{i+l}$
  $q \leftarrow q + t_{i+l}^2$
  $\bar{x} \leftarrow \frac{p}{l}$
  $s \leftarrow \frac{q}{l} - \bar{x}^2$
  $j \leftarrow 1, d \leftarrow 0$
  {distance between S and $< t_{i+1} \ldots t_{i+l+1} >$ with early abandon}
  while $j \leq l$ & $d < b$ **do**
    $d \leftarrow d + \left(S_{A_j} - \frac{t_{i+A_j} - \bar{x}}{s} \right)^2$ { *reordered online normalisation*}
    $j \leftarrow j + 1$
  if $j = l$ & $d < b$ **then**
    $b \leftarrow d$
return b

Precalculation of distance statistics between series. Because every subsequence is compared to every other, there is duplication in the calculations. So, for example, when the subsequence starting at position a is compared to the subsequence at position b, many of the calculations that were previously performed in comparing the subsequence starting at position $a - 1$ to the one starting at $b - 1$ are duplicated. A method involving trading memory for speed is proposed in Mueen et al. (2011). For each pair of series T_i, T_j, cumulative sum, squared sum, and cross products of T_i and T_j are precalculated. With these statistics, the distance between subsequences can be calculated in constant time, making the shaplet-discovery algorithm $O(n^2m^3)$. However, precalculating of the cross products between all series prior to shapelet discovery requires $O(n^2m^2)$ memory, which is infeasible for most problems. Instead, Mueen et al. (2011) propose calculating these statistics prior to the start of the scan of each series, reducing the requirement to $O(nm^2)$ memory, but increasing the time overhead.

Early abandon of the shapelet. An early abandon of the shapelet assessment is proposed in Ye and Keogh (2011). After the calculation of each value $d_{S,i}$, an upper bound on the

IG is found by assuming the most optimistic future assignment. If this upper bound falls below the best found so far, the calculation of D_S can be abandoned. Huge potential speed up by abandoning poor shapelets comes at a small extra overhead for calculating the best split and upper bound for each new $d_{S,i}$. However, for multi-class problems, a correct upper bound can be found only through enumerating split assignments for all possible classes, which can dramatically increase the overhead.

3 Alternative shapelet quality measures

Unlike the shapelet tree, our shapelet transform does not require an explicit split point to be found by the quality measure. IG introduces extra time overhead and may not be optimal for multi-class problems, since it is restricted to binary splits. We investigate alternative shapelet quality measures based on a hypothesis tests of differences in distribution of distances between class populations. We look at three alternative ways of quantifying how well the classes can be split by the list of distances D_S.

3.1 Kruskal-Wallis

Kruskal (1952) (KW) is a non-parametric test of whether two samples originate from a distribution with the same median. The test statistic is the squared-weighted difference between ranks within a class and the global mean rank. Given a sorted list of distances D split by class membership into sets $D_1, \ldots, D_C$, and a list of ranks $R =< 1, 2, \ldots, n >$ split so that the ranks of elements in D_i in R are assigned to set R_i, the KW statistic is defined as

$$K = \frac{12}{n(n+1)} \sum_{i=1}^{C} |R_i|(\bar{R}_i - \bar{R})^2,$$

where $\bar{R}_i$ is the mean ranks for class i and $\bar{R} = \frac{\sum_{i=1}^{n} i}{n}$. This simplifies to

$$K = \frac{12}{n \cdot (n+1)} \sum_{i=1}^{c} \frac{\sum_{r_j \in R_i} r_j^2}{|R_i|} - 3(n+1).$$

3.2 Analysis of variance F-statistic

The F-statistic (F-stat) for analysis of variance is used to test the hypothesis of difference in means between a set of C samples. The null hypothesis is that the population mean from each sample is the same. The test statistic for this hypothesis is the ratio of the variability between the groups to the variability within the groups. The higher the value, the greater the between-group variability compared to the within-group variability. A high-quality shapelet has small distances to members of one class and large distances to members of other classes; hence, a high-quality shapelet yields a high

F-stat. To assess a list of distances $D = < d_1, d_2, \ldots, d_n >$, we first split them by class membership, so that D_i contains the distances of the candidate shapelet to time series of class i. The F-statistic shapelet quality measure is

$$F = \frac{\sum_i \left(\bar{D}_i - \bar{D}\right)^2/(C-1)}{\sum_{i=1}^{C} \sum_{d_j \in D_i} \left(d_j - \bar{D}_i\right)^2/(n-C)}$$

where C is the number of classes, n is the number of series, $\bar{D}_i$ is the average of distances to series of class i and $\bar{D}$ is the overall mean of D.

3.3 Mood's median

Mood et al. (1974) median (MM) is a non-parametric test to determine whether the medians of two samples originate from the same distribution. Unlike IG and KW, MM does not require D to be sorted. Only the median is required for calculating MM, which can be found in $O(n)$ time using Quickselect Hoare (1962). The median is used to create a contingency table from D, where the counts of each class above and below the median are recorded. Let o_{i1} represent the count of class i above the median and o_{i2} the count of those below the median. If the null hypothesis is true, we would expect the split above and below the median to be approximately the same. Let e_{i1} and e_{i2} denote the expected number of observations above and below the median if the null hypothesis of independence is true. The MM statistic is

$$M = \sum_{i=1}^{C} \sum_{j=1}^{2} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

4 Shapelet transform

Bagnall et al. (2012) demonstrate the importance of separating the transformation from the classification algorithm with an ensemble approach, where each member of the ensemble is constructed on a different transform of the original data. They show that, firstly, on problems where the discriminatory features are not in the time domain, operating in a different data space produces greater performance improvement than designing a more complex classifier. Secondly, a basic ensemble on transformed datasets can significantly improve simple classifiers. We apply this intuition to shapelets, and separate the transformation from the classifier.

Our transformation processes shapelets in three distinct stages. Firstly, the algorithm performs a single scan of the data to extract the best k shapelets. k is a cut-off value for the maximum number of shapelets to store, and has no effect on the quality of the individual shapelets that are extracted. Secondly, the set of k shapelets can be reduced, either by ignoring the shapelets below a cut-off point (e.g. reducing a set of

256 shapelets to 10 shapelets), or by clustering the shapelets (see Sect. 4.4). Finally, a new transformed dataset is created, where each attribute represents a shapelet, and the value of the attribute is the distance between the shapelet and the original series. Transforming the data in this way disassociates shapelet finding from classification, allowing the transformed dataset to be used in conjunction with any classifier.

4.1 Shapelet generation

The process to extract the best k shapelets is defined in Algorithm 3.

Algorithm 3 ShapeletCachedSelection(**T**, min, max, k)

```
1: kShapelets ← ∅
2: for all T_i in T do
3:    shapelets ← ∅
4:    for l ← min to max do
5:       W_{i,l} ← generateCandidates(T_i, l)
6:       for all subsequence S in W_{i,l} do
7:          D_S ← findDistances(S, T)
8:          quality ← assessCandidate(S, D_S)
9:          shapelets.add(S, quality)
10:   sortByQuality(shapelets)
11:   removeSelfSimilar(shapelets)
12:   kShapelets ← merge(k, kShapelets, shapelets)
13: return kShapelets
```

The algorithm processes data in a manner similar to the original shapelet algorithm Ye and Keogh (2011) (Algorithm 1). For each series in the dataset, all subsequences of lengths between min and max are examined. However, unlike Algorithm 1, where all candidates are assessed and the best is stored, our caching algorithm stores all candidates for a given time series, along with their associated quality measures (line 9). Once all candidates of a series have been assessed, they are sorted by quality, and self-similar shapelets are removed. Self-similar shapelets are taken from the same series and have overlapping indices. We merge the set of non-self-similar shapelets for a series with the current best shapelets and retain the top k, iterating through the data until all series have been processed. We do not store all candidates indefinitely; after processing each series, we retain only those that belong to the best k so far, and discard all other shapelets. Thus, we avoid the large space overhead required to retain all candidates.

When handling self-similarity between candidates, it is necessary to temporarily store and evaluate all candidates from a single series before removing self-similar shapelets. This prevents shapelets being rejected incorrectly. For example, in a given series, candidate A may be added to the k-best-so-far. If candidate B overlaps with A and has higher quality, A will be rejected. If a third candidate of even higher quality, C, is identified that is self-similar to B, but not to A, C would replace B, and the deleted A would be a valid candidate for the k-best. We overcome this issue by evaluating all candidates for a given series before deleting those that are self-similar (line 9 in

Algorithm 3). Once all candidates for a given series have been assessed, they are sorted into descending order of quality (line 10). The sorted set of candidates can then be assessed for self-similarity in order of quality (line 11), so that the best candidates are always retained, and self-similar candidates are safely removed.

4.1.1 Length parameter approximation

Both the original algorithm and our caching algorithm require two length parameters, min and max. These values define the range of candidate shapelet lengths. Smaller ranges improve speed, but may compromise accuracy if they prevent the most informative subsequences from being considered. To accommodate running the shapelet filter on a range of datasets without any specialised knowledge of the data, we define a simple algorithm for estimating the min and max parameters.

Algorithm 4 EstimateMinAndMax(**T**)

1: $shapelets \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to 10 **do**
3: $\quad randomiseOrder(\mathbf{T})$
4: $\quad T' \leftarrow [T_1, T_2, ..., T_{10}]$
5: $\quad currentShapelets \leftarrow ShapeletCachedSelection(T', 1, n, 10)$
6: $\quad shapelets.add(currentShapelets)$
7: $orderByLength(shapelets)$
8: $min \leftarrow shapelets_{25}.length$
9: $max \leftarrow shapelets_{75}.length$
10: **return** min, max

The procedure described in Algorithm 4 randomly selects ten series from dataset **T** and uses Algorithm 3 to find the best ten shapelets in this subset of the data. For this search, $min = 3$ and max is set to n. The selection and search procedure is repeated ten times in total, yielding a set of 100 shapelets. The shapelets are sorted by length, with the length of the 25th shapelet returned as min and the length of the 75th shapelet returned as max. While this does not necessarily result in the optimal parameters, it does provide an automatic approach to approximate min and max across a number of datasets. Hence, we can compare our filter fairly against the original shapelet-tree implementation.

4.2 Data transformation

The main motivation for our shapelet transformation is to allow shapelets to be used with a diverse range of classification algorithms, rather than the decision tree used in previous research. Our algorithm uses shapelets to transform instances of data into a new feature space; the transformed data can be viewed as a generic classification problem. The transformation process is defined in Algorithm 5.

The transformation is carried out using the subsequence distance calculation described in Sect. 2.2.2. A set of k shapelets, S, is generated from the training data **T**. For each instance of data T_i, the subsequence distance is computed between T_i

Algorithm 5 ShapeletTransform(Shapelets S, Dataset $\mathbf{T}$)

```
T' ← ∅
for all T_i in T do
    for all shapelets s_j in S do
        dist ← subsequenceDist(s_j, T_i)
        T_{i,j} = dist
    T' = T' ∪ T_i
return T'
```

and s_j, where $j = 1, 2, \ldots, k$. The resulting k distances are used to form a new instance of transformed data, where each attribute corresponds to the distance between a shapelet and the original time series. When using data partitioned into training and test sets, the shapelet extraction is carried out on the training data to avoid bias; these shapelets are used to transform each instance of the training and test data to create transformed data sets, which can be used with any traditional classification algorithm.

4.3 Shapelet selection

Using k shapelets in the filter will not necessarily yield the best data for classification. Using too few shapelets does not provide enough information to the classifier; using too many may cause overfitting, or dilute the influence of important shapelets. For our experiments, we use $\frac{n}{2}$ shapelets in the filter, where n is the length of a single series of the data.

4.4 Clustering shapelets

By definition, a shapelet that discriminates well between classes will be similar to a set of subsequences from other instances of the same class. It is common for the transform to include multiple shapelets that match one another. In some datasets, including the matches of a discriminative shapelet can mean that useful shapelets are missed; additionally, the duplication can reduce the comprehensibility of the transformed data. To mitigate these problems, we hierarchically cluster shapelets after the transform with the procedure given in Algorithm 6. A distance map is created representing the shapelet distance between each pair of shapelets. For k shapelets, this is a $k \times k$ matrix with reflective symmetry around the diagonal (which consists of zeros). The pair with the smallest shapelet distance between them are clustered, and an updated $k - 1 \times k - 1$ distance map is created with the clustered pair removed and the cluster added. The process is repeated until a user-specified number of clusters are formed. We compute the shapelet distance between two clusters, C_i and C_j, as the average of the shapelet distance between each member of C_i and each member of C_j. For any cluster of shapelets, we represent the cluster with the cluster member that is the best by the appropriate shapelet quality measure. The other members of the cluster are assumed to be matches of this shapelet.

Algorithm 6 ClusterShapelets(Shapelets S, $noClusters$)

```
C ← ∅ //C is a set of sets.
for all s_i ∈ S do
    C ← C ∪ {s_i}
while |C| > noClusters do
    M ← |C| × |C| matrix
    for all C_i ∈ C do
        for all C_j ∈ C do
            distance ← 0
            comparisons ← |C_i| × |C_j|
            for all c_l ∈ C_i do
                for all c_k ∈ C_j do
                    dist ← dist + d_S(c_l, c_k)
            M_{i,j} ← dist / comparisons //store average linkage distances in distance map.
    best ← ∞
    position ← {0, 0}
    for all M_{i,j} ∈ M do
        if M_{i,j} < best ∧ i ≠ j then
            x ← i
            y ← j
            best ← M_{i,j} //Find smallest distance in distance map.
    C' ← C_x ∪ C_y
    C ← C − {C_x} − {C_y}
    C ← C ∪ C' //Update set of clusters, merging the closest pair.
return C
```

5 Datasets

We perform experiments on 17 datasets from the UCR time-series repository. We selected these particular UCR datasets because they have relatively few cases; even with optimisation, the shapelet algorithm is time consuming.

We also provide a number of new datasets that we have donated to the UCR repository and make freely available to researchers. We have eight bone-outline problems (Sect. 5.3), synthetic data designed to be optimal for the shapelet approach (Sect. 5.1), two new image-processing outline-classification problems derived from the MPEG-7 dataset Bober (2001), and an outline-classification problem involving classifying herring based on their otoliths (see Sects. 5.1.1 and 5.2 respectively). The datasets we use are summarised in Table 1.

For all but the smallest problems, we partition the data into training and testing sets and report the accuracy on the test set. Shapelet selection, model selection, and classifier training are performed exclusively on the training set; the test set is used only with the final trained classifier.

5.1 Synthetic data

We create a number of synthetic datasets designed to be tractable to the shapelet approach. The datasets consist of 1,100 time series representing a two-class classification problem. The series are length 500, normally-distributed ($\mathcal{N}(0, 1)$) random noise. For each dataset, two shapes (see Bagnall et al. 2012), A and B, are randomly

Table 1 Summary of datasets

	Assessment	Instances (train/test)	Length	No. classes	Shapelet min/max
Adiac	Train/Test	390/391	176	37	3/10
Beef	Train/Test	30/30	470	5	8/30
Beetle/Fly	LOOCV	40	512	2	30/101
Bird/Chicken	LOOCV	40	512	2	30/101
ChlorineConcentration	Train/Test	467/3,840	166	3	7/20
Coffee	Train/Test	28/28	286	2	18/30
DiatomSizeReduction	Train/Test	16/306	345	4	7/16
DP_Little	Train/Test	400/645	250	3	9/36
DP_Middle	Train/Test	400/645	250	3	15/43
DP_Thumb	Train/Test	400/645	250	3	11/47
ECGFiveDays	Train/Test	23/861	136	2	24/76
FaceFour	Train/Test	24/88	350	4	20/120
GunPoint	Train/Test	50/150	150	2	24/55
ItalyPowerDemand	Train/Test	67/1029	24	2	7/14
Lighting7	Train/Test	70/73	319	7	20/80
MedicalImages	Train/Test	381/760	99	10	9/35
MoteStrain	Train/Test	20/1252	84	2	16/31
MP_Little	Train/Test	400/645	250	3	15/41
MP_Middle	Train/Test	400/645	250	3	20/53
Otoliths	Train/Test	64/64	512	2	30/101
PP_Little	Train/Test	400/645	250	3	13/38
PP_Middle	Train/Test	400/645	250	3	14/34
PP_Thumb	Train/Test	400/645	250	3	14/41
SonyAIBORobotSurface	Train/Test	20/601	70	2	15/36
Symbols	Train/Test	25/995	398	6	52/155
SyntheticControl	Train/Test	300/300	60	6	20/56
SyntheticData	Train/Test	100/1,000	500	2	25/35
Trace	Train/Test	100/100	275	4	62/232
TwoLeadECG	Train/Test	23/1,139	82	2	7/13

selected. One instance of A is added to the noise at random locations in 550 of the time series; an instance of B is added to the remaining 550 series. The series are split into a size 100 training set and a size 1,000 testing set. Hence, we create a classification problem where the distinguishing feature of the classes is a representative subsequence located somewhere in the series. Each dataset gives a random instance of this type of problem. This allows us to test whether one approach is significantly better for a class of problem where the shapelet approach should be optimal. All the results presented for synthetic data are averaged over 200 runs of independently generated train and test sets.

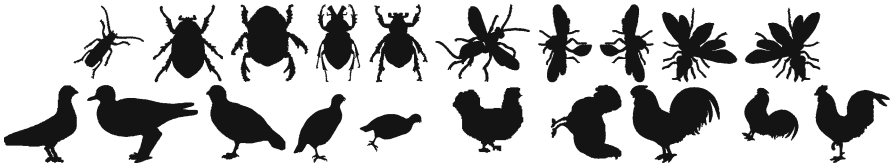


Fig. 2 Five beetle images (*top left*) and five fly images (*top right*) from the Beetle/Fly problem. Five bird images (*bottom left*) and five chicken images (*bottom right*) from the Bird/Chicken problem. There is considerable intra-class variation, as well as inconsistent size and rotation

5.1.1 MPEG-7 shapes

MPEG-7 CE Shape-1 Part B Bober (2001) is a database of binary images developed for testing MPEG-7 shape descriptors, and is available free online. It is used for testing contour/image and skeleton-based descriptors Latecki et al. (2000). Classes of images vary broadly, and include classes that are similar in shape to one another. There are 20 instances of each class, and 60 classes in total. We have extracted the outlines of these images and mapped them into 1-*D* series. We have created two time-series classification problems from the shapes, *Beetle/Fly* and *Bird/Chicken*. Figure 2 shows some of the images from the two problems.

5.2 Otoliths

Otoliths are calcium carbonate structures present in many vertebrates, found within the sacculus of the pars inferior. There are three types of otoliths: sagittae, lapilli, and asterisci. In fish, it is primarily the sagittal otoliths that are studied, as they are larger and easier to prepare and observe. Otoliths vary markedly in shape and size between species, but are of similar shape to other stocks of the same species (Fig. 3). Otoliths contain information that can be used by 'expert readers' to determine several key factors important in managing fish stock. Analysis of otolith boundaries may allow estimation of stock composition, including whether the samples are from one stock or multiple stocks Campana and Casselman (1993), De Vries et al. (2002), Duarte-Neto et al. (2008), allowing management decisions to be made Stransky (2005). We consider the problem of classifying herring stock (either North sea or Thames) based on the otolith outline (Fig. 4).

5.3 Bone outlines

The bone datasets (DP_Little, DP_Middle, DP_Thumb, MP_Little, MP_Middle, PP_Little, PP_Middle, and PP_Thumb) consist of image outlines from hand X-rays, converted into 1-*D* series. The original images can be found at (Image Processing and Informatics Lab). Each of the eight datasets represents a different bone of the hand, and is labelled as belonging to one of three classes, *Infant*, *Junior*, or *Teen*. The classification problem is to predict the class to which a bone belongs, a process that is largely performed manually by doctors. For more information, see Davis et al. (2012).

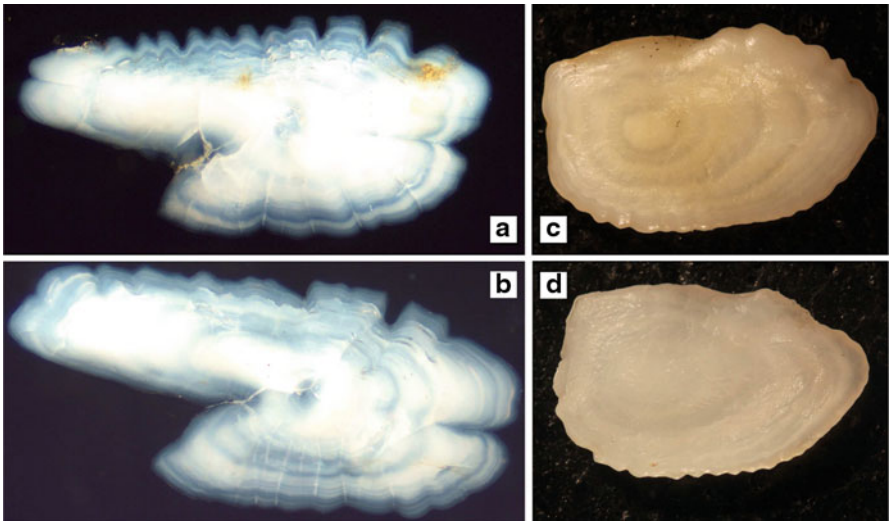


Fig. 3 Otoliths from North-Sea Herring (**a**), Thames Herring (**b**) and two distinct populations of Plaice (**c** and **d**)

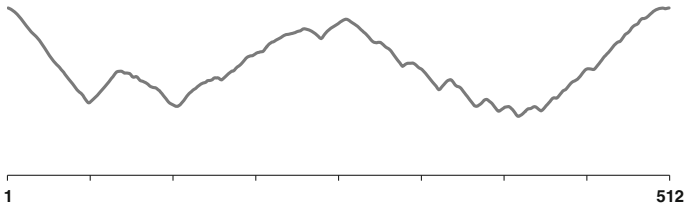


Fig. 4 A 1*D* time-series representation of the herring otolith shown in Fig. 3a

5.4 Scalability

At best, finding shapelets of a single length by exhaustive search has complexity $O(n^2m^3)$ where n is the size of the dataset and m is the length of the series Rakthanmanon and Keogh (2013). This is untenable for very large datasets. Our slowest experiments were with the Otolith dataset, which has 64 training examples of length 512. We used 71 different shapelet lengths. In the worst case, this requires 3.9×10^{13} operations. The experiments took several days to perform in our high-performance computing facility.

6 Results

We present our experimental results in three stages. In Sect. 6.1, we compare IG and the three alternative quality measures described in Sect. 3. In Sect. 6.2, we comprehensively evaluate the effectiveness of the shapelet transform. Finally, In Sect. 6.3, we

demonstrate the explanatory power of shapelets by mapping them back to the problem domain.

All algorithms and experiments are implemented in Java within the Weka Hall et al. (2009) framework; the shapelet transform is implemented as a Weka batch filter to allow for easy integration with existing classification code. The code to generate our results is available at Bagnall et al. (2012).

6.1 Shapelet quality measures

To evaluate the quality measures in isolation, we perform these experiments with our implementation of the shapelet tree described in Ye and Keogh (2011). We compare IG, KW, F-stat, and MM in terms of accuracy and speed. The alternative statistics do not explicitly split the data when searching for the best shapelet. To make the comparison to IG as fair as possible, once the best shapelet has been selected with a statistic, we use IG on the resulting set of distances to find the best split point. We do this to focus on the ability of the statistic to assess shapelets, rather than perform data splits.

6.1.1 Effect of quality measures on classification accuracy

Table 2 shows the accuracies and ranks of shapelet-tree classifiers built using the four quality measures on 29 datasets. The tree based on the F-stat is the most accurate classifier on 15 of the 29 data sets, and has the highest average rank. Furthermore, the F-stat is significantly better than the other three measures tested on 200 repetitions of the synthetic data. However, we cannot claim that F-Stat is universally better, since their is no significant difference in ranks between the four measures when we consider all 29 data sets. Figure 5 shows the a critical difference diagram for ranked accuracies (see Demšar 2006). The diagram is derived from the overall test of significance of mean ranks, where classifiers are grouped into *cliques* represented by solid bars. The diagram shows that all classifiers are part of a single clique, and therefore that they are not significantly different from one another.

6.1.2 Timing results

Table 3 shows the time required to find the best shapelet using IG, KW, F-stat, and MM. We adopt this approach to ensure fair comparisons are made between measures, as comparing the build times of whole decision trees is biased if the classifiers are of different depths. Extracting a single shapelet ensures that the same number of candidates are processed for each quality measure.

Table 3 shows that there are no datasets where IG is fastest. F-stat is the fastest measure on average, and has the fastest time on the most datasets. Figure 6 shows that, based on ranked timings, the F-stat is significantly faster than both IG and KW (using the Friedman rank order test). However, the speed up is not of an order of magnitude, and it is possible the difference could diminish with code and hardware optimisation. Nevertheless, we argue that the F-stat should be the default choice for shapelet quality.

Table 2 Classification accuracies for shapelet-tree classifiers

Dataset	Information gain	Kruskal-Wallis	F-stat	Mood's median
Adiac	**29.92 % (1)**	26.60 % (3)	15.60 % (4)	27.11 % (2)
Beef	50.00 % (2)	33.33 % (3)	**56.67 % (1)**	30.00 % (4)
Beetle/Fly	77.50 % (3)	70.00 % (4)	**90.00 % (1)**	80.00 % (2)
Bird/Chicken	85.00 % (4)	87.50 % (2.5)	**90.00 % (1)**	87.50 % (2.5)
ChlorineConcentration	**58.80 % (1)**	51.95 % (4)	53.52 % (2)	52.11 % (3)
Coffee	96.43 % (2)	85.71 % (3.5)	**100 % (1)**	85.71 % (3.5)
DiatomSizeReduction	72.22 % (2)	62.11 % (3)	**76.47 % (1)**	44.77 % (4)
DP_ Little	65.44 % (3)	68.00 % (2)	60.31 % (4)	**71.00 % (1)**
DP_ Middle	70.53 % (2)	69.33 % (3)	61.86 % (4)	**73.67 % (1)**
DP_ Thumb	58.11 % (3)	**72.00 % (1)**	55.97 % (4)	70.33 % (2)
ECGFiveDays	77.47 % (4)	87.22 % (3)	**99.00 % (1)**	92.80 % (2)
FaceFour	**84.09 % (1)**	44.32 % (3)	75.00 % (2)	40.91 % (4)
GunPoint	89.33 % (4)	94.00 % (2)	**95.33 % (1)**	92.00 % (3)
ItalyPowerDemand	89.21 % (4)	90.96 % (3)	**93.10 % (1)**	91.06 % (2)
Lighting7	**49.32 % (1)**	47.95 % (2)	41.10 % (3)	27.40 % (4)
MedicalImages	48.82 % (3)	47.11 % (4)	**50.79 % (1)**	48.95 % (2)
MoteStrain	82.51 % (4)	**83.95 % (2)**	**83.95 % (2)**	**83.95 % (2)**
MP_ Little	66.39 % (3)	69.67 % (2)	57.83 % (4)	**70.33 % (1)**
MP_ Middle	71.01 % (3)	**75.00 % (1)**	60.93 % (4)	72.00 % (2)
Otoliths	**67.19 % (1)**	60.93 % (2)	57.81 % (3)	54.69 % (4)
PP_ Little	59.64 % (3)	**72.00 % (1)**	58.60 % (4)	67.33 % (2)
PP_ Middle	61.42 % (3)	68.33 % (2)	58.14 % (4)	**69.67 % (1)**
PP_ Thumb	60.83 % (3)	71.33 % (2)	59.07 % (4)	**73.00 % (1)**
SonyAIBORobotSurface	84.53 % (2)	72.71 % (4)	**95.34 % (1)**	74.87 % (3)
Symbols	77.99 % (2)	55.68 % (4)	**80.10 % (1)**	57.39 % (3)
SyntheticControl	94.33 % (2)	90.00 % (3)	**95.67 % (1)**	85.67 % (4)
SyntheticData	93.30 % (2)	80.56 % (3.5)	**100 % (1)**	80.56 % (3.5)
Trace	98.00 % (2.5)	94.00 % (4)	98.00 % (2.5)	**100 % (1)**
TwoLeadECG	85.07 % (3)	76.38 % (4)	**97.01 % (1)**	85.34 % (2)
Average rank	2.53	2.78	2.22	2.47

This experiment can be reproduced with method DMKD_2013, Bagnall et al. (2012)
The best result for each dataset is shown in bold

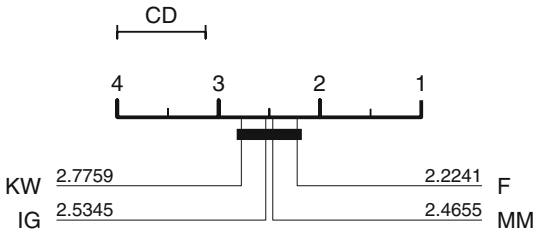


Fig. 5 Critical difference diagram of the ranked accuracies for the *four* shapelet-tree classifiers

Table 3 Time to find first shapelet for each dataset

Dataset	Information gain	Kruskal-Wallis	F-stat	Mood's median
Adiac	17758.48 (4)	**4974.50 (1)**	4509.91 (2)	4752.63 (3)
Beef	1284.46 (4)	1253.17 (3)	1251.21 (2)	**1228.51 (1)**
Beetle-fly	21707.02 (4)	21400.71 (2)	21496.51 (3)	**21133.98 (1)**
Bird-chicken	20258.90 (2)	20349.63 (3)	20465.63 (4)	**19996.78 (1)**
ChlorineConcentration	26233.51 (4)	16699.23 (3)	**15681.39 (1)**	16572.67 (2)
Coffee	261.27 (2)	264.75 (4)	**258.15 (1)**	263.82 (3)
DiatomSizeReduction	55.35 (4)	54.61 (3)	**53.91 (1)**	54.36 (2)
DP_Little	97508.12 (4)	80556.13 (2)	**78005.70 (1)**	82052.11 (3)
DP_Middle	106382.47 (4)	94081.04 (3)	**91208.52 (1)**	91664.80 (2)
DP_Thumb	149567.07 (4)	125334.92 (3)	**123766.49 (1)**	124508.41 (2)
ECGFiveDays	151.64 (3)	150.90 (2)	**149.10 (1)**	155.43 (4)
FaceFour	4695.97 (4)	4621.97 (2)	**4556.41 (1)**	4648.45 (3)
GunPoint	592.00 (4)	569.51 (2)	**569.42 (1)**	580.76 (3)
ItalyPowerDemand	3.18 (4)	1.56 (2)	1.75 (3)	**1.46 (1)**
Lighting7	15497.07 (4)	15157.93 (3)	**14912.74 (1)**	14940.20 (2)
MedicalImages	15703.95 (4)	8148.76 (3)	**7742.97 (1)**	8111.36 (2)
MoteStrain	11.55 (4)	11.02 (3)	**10.76 (1)**	11.00 (2)
MP_Little	108518.67 (4)	89849.25 (3)	**88071.50 (1)**	89634.65 (2)
MP_Middle	156750.20 (4)	135852.37 (3)	**134731.54 (1)**	134756.02 (2)
Otoliths	55090.54 (2)	56141.82 (4)	55874.19 (3)	54986.19 (1)
PP_Little	97987.27 (4)	**79285.08 (1)**	79993.31 (2)	80514.60 (3)
PP_Middle	68730.32 (4)	59579.27 (3)	**57815.02 (1)**	58389.16 (2)
PP_Thumb	110204.43 (4)	**91183.51 (1)**	91401.49 (3)	91202.87 (2)
SonyAIBORobotSurface	7.80 (4)	6.79 (2.5)	**6.73 (1)**	6.79 (2.5)
Symbols	8992.15 (4)	8941.21 (3)	**8901.28 (1)**	8922.01 (2)
SyntheticControl	2280.82 (4)	1029.95 (3)	984.36 (2)	**974.27 (1)**
SyntheticData	403.50 (4)	401.28 (2)	401.74 (3)	**399.99 (1)**
Trace	54829.06 (3)	55155.36 (4)	**54128.53 (1)**	54205.65 (2)
TwoLeadECG	3.61 (4)	3.15 (3)	3.12 (2)	**3.11 (1)**
Average rank	3.72	2.64	1.62	2.02

This experiment can be reproduced with method DMKD_2013, Bagnall et al. (2012)
The best result for each dataset is shown in bold

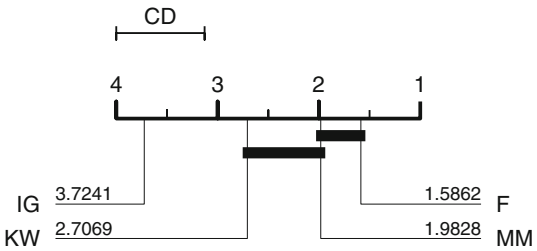


Fig. 6 Critical difference diagram of the ranked timings of the *four* shapelet-tree classifiers

Using the F-stat measure is marginally, but significantly, faster, and the accuracy is highly competitive on the 29 data sets as a whole. The F-stat is also significantly more accurate on the synthetic data.

6.2 Shapelet transformation

Despite our conclusion from the previous set of experiments, we use IG as the shapelet quality measure in all consequent experiments. Fixing the shapelet quality measure removes a source of variation in performance and allows us to focus on our key hypothesis that it is better to transform then use a more complex classifier than it is to embed the shapelet discovery in a decision tree.

Our first objective is to establish that dissociating shapelet discovery from classification does not reduce classification accuracy. We implement a shapelet decision-tree classifier as described in Ye and Keogh (2011), and compare the performance to a C4.5 decision tree trained and tested on shapelet-transformed data. The shapelet tree has greater accuracy on 15 datasets and the C4.5 tree has greater accuracy on 14 datasets. No significant difference is detected between the classifiers using a paired t-test or a Wilcoxon signed rank test. We conclude that performing the shapelet extraction prior to constructing the decision tree does not reduce the accuracy of the classifier. Figure 7 presents the findings graphically.

Using classifiers other than decision trees can improve the accuracy of classification with shapelets. Tables 4 and 5 shows the classification test accuracy of the C4.5, 1-NN, naive Bayes, Bayesian network, Random Forest, Rotation Forest, and support vector machine classifiers, all built using the default Weka settings on shapelet-transformed data. A Bayesian network is an acyclic directed graph with associated probability distributions Friedman et al. (1997). It predicts class labels without assuming independence between variables. The Random Forest algorithm classifies examples by

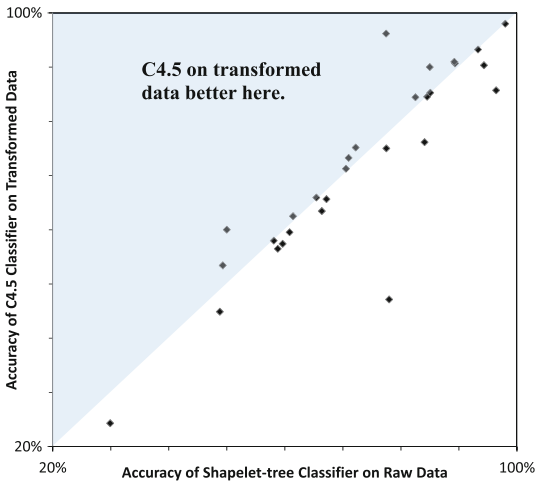


Fig. 7 Comparison of C4.5 tree on shapelet-transformed data and shapelet tree on raw data

Table 4 Testing accuracies and ranks of 1NN-DTW on raw data, and simple classifiers (C4.5, 1NN, and Naïve Bayes) constructed on the shapelet-transformed data with $\frac{n}{2}$ shapelets

Data	1 % NN-DTW raw	C4.5	1NN-Euclidean	Naive Bayes
Adiac	**49.10 % (1)**	24.30 % (7)	25.32 % (5)	28.13 % (4)
Beef	43.33 % (8)	60.00 % (6.5)	83.33 % (3)	73.33 % (4)
Beetle/Fly	65.00 % (8)	75.00 % (7)	**100.00 % (1)**	92.50 % (5)
Bird/Chicken	72.50 % (8)	90.00 % (6)	**97.50 % (1)**	87.50 % (7)
ChlorineConcentration	63.36 % (2)	56.48 % (6)	56.93 % (5)	45.96 % (8)
Coffee	46.43 % (8)	85.71 % (7)	**100.00 % (2)**	92.86 % (5)
DiatomSizeReduction	92.48 % (2)	75.16 % (8)	**93.46 % (1)**	78.76 % (7)
DP_Little	49.30 % (8)	65.92 % (7)	72.78 % (6)	73.49 % (3)
DP_Middle	54.57 % (8)	71.24 % (7)	73.73 % (6)	73.96 % (5)
DP_Thumb	53.02 % (7)	57.99 % (8)	60.71 % (6)	62.96 % (5)
ECGFiveDays	82.81 % (8)	96.17 % (6)	98.37 % (4)	96.40 % (5)
FaceFour	82.95 % (7)	76.14 % (8)	**100.00 % (1.5)**	97.73 % (4.5)
GunPoint	91.33 % (7)	90.67 % (8)	98.00 % (4)	92.00 % (6)
ItalyPowerDemand	**96.11 % (1)**	90.96 % (8)	92.13 % (5.5)	92.52 % (3)
Lighting7	**72.60 % (1)**	53.42 % (7)	49.32 % (8)	57.53 % (6)
MedicalImages	**68.95 % (1)**	44.87 % (6)	45.66 % (5)	17.37 % (8)
MoteStrain	81.55 % (8)	84.42 % (7)	**90.34 % (1)**	88.82 % (3)
MP_Little	55.81 % (8)	63.43 % (7)	68.52 % (6)	68.76 % (5)
MP_Middle	46.98 % (8)	73.25 % (4)	70.89 % (7)	71.95 % (5)
Otoliths	59.38 % (7.5)	65.63 % (3.5)	**71.88 % (1)**	68.75 % (2)
PP_Little	49.46 % (8)	57.40 % (7)	67.22 % (5)	69.23 % (4)
PP_Middle	49.92 % (8)	62.49 % (7)	68.52 % (6)	69.82 % (5)
PP_Thumb	52.56 % (8)	59.53 % (7)	67.69 % (6)	69.35 % (4)
Data	1NN-DTW Raw	C4.5	1NN-Euclidean	Naive Bayes
SonyAIBORobotSurface	69.88 % (8)	84.53 % (5)	84.03 % (6)	79.03 % (7)
Symbols	**93.37 % (1)**	47.14 % (8)	85.63 % (3)	77.99 % (7)
SyntheticControl	**97.33 % (1)**	90.33 % (4)	93.00 % (2)	78.00 % (7)
SyntheticData	70.03 % (8)	93.24 % (7)	97.66 % (5)	98.13 % (2)
Trace	99.00 % (2)	98.00 % (5.5)	98.00 % (5.5)	98.00 % (5.5)
woLeadECG	79.46 % (8)	85.25 % (7)	**99.47 % (1)**	99.12 % (3)
Average rank	5.81	6.60	4.09	5.00

The ranks include the results in Table 5. This experiment can be reproduced with method DMKD_2013 Bagnall et al. (2012)
The best result for each dataset is shown in bold

generating a large number of decision trees with controlled variation and using the modal classification decision Breiman (2001). The Rotation Forest algorithm trains a number of decision trees by applying principal components analysis on a random subset of attributes Rodriguez et al. (2006). A support vector machine finds the best separating hyperplane for a set of data by selecting the margin that maximises the

Table 5 Testing accuracies and ranks of complex classifiers (Bayesian Network, random Forest, Rotation Forest, and linear SVM) constructed on the shapelet-transformed data with $\frac{n}{2}$ shapelets

Data	Bayesian network	Random forest	Rotation forest	SVM (linear)
Adiac	25.06 % (6)	30.43 % (3)	30.69 % (2)	23.79 % (8)
Beef	**90.00 % (1)**	60.00 % (6.5)	70.00 % (5)	86.67 % (2)
Beetle/Fly	97.50 % (2.5)	90.00 % (6)	95.00 % (4)	97.50 % (2.5)
Bird/Chicken	95.00 % (3)	95.00 % (3)	92.50 % (5)	95.0 % (3)
ChlorineConcentration	57.08 % (4)	57.58 % (3)	**63.52 % (1)**	56.15 % (7)
Coffee	96.43 % (4)	**100.00 % (2)**	89.29 % (6)	**100.00 % (2)**
DiatomSizeReduction	90.20 % (4)	80.39 % (6)	83.01 % (5)	92.16 % (3)
DP_Little	72.90 % (5)	73.02 % (4)	74.67 % (2)	**75.15 % (1)**
DP_Middle	74.67 % (4)	75.50 % (3)	76.80 % (2)	**79.64 % (1)**
DP_Thumb	63.91 % (4)	64.14 % (3)	67.10 % (2)	**69.82 % (1)**
ECGFiveDays	**99.54 % (1)**	93.26 % (7)	98.61 % (3)	98.95 % (2)
FaceFour	**100.00 % (1.5)**	87.50 % (6)	98.86 % (3)	97.73 % (4.5)
GunPoint	99.33 % (2)	96.00 % (5)	98.67 % (3)	**100.00 % (1)**
ItalyPowerDemand	92.42 % (4)	93.00 % (2)	92.03 % (7)	92.13 % (5.5)
Lighting7	65.75 % (3.5)	64.38 % (5)	65.75 % (3.5)	69.86 % (2)
MedicalImages	28.16 % (7)	50.79 % (4)	51.45 % (3)	52.50 % (2)
MoteStrain	89.06 % (2)	84.58 % (6)	86.98 % (5)	88.66 % (4)
MP_Little	69.47 % (4)	71.36 % (3)	**75.15 % (1)**	75.03 % (2)
MP_Middle	71.12 % (6)	75.15 % (2)	74.67 % (3)	**76.92 % (1)**
Otoliths	64.06 % (5.5)	65.63 % (3.5)	59.38 % (7.5)	64.06 % (5.5)
PP_Little	70.06 % (2)	66.63 % (6)	69.82 % (3)	**72.07 % (1)**
PP_Middle	71.36 % (3)	70.53 % (4)	75.38 % (2)	**75.86 % (1)**
PP_Thumb	69.47 % (3)	67.81 % (5)	72.78 % (2)	**75.50 % (1)**
SonyAIBORobotSurface	**89.68 % (1)**	85.19 % (4)	89.02 % (2)	86.69 % (3)
Symbols	92.26 % (2)	84.62 % (4.5)	84.42 % (6)	84.62 % (4.5)
SyntheticControl	76.67 % (8)	89.00 % (5)	92.00 % (3)	87.33 % (6)
SyntheticData	98.00 % (3)	96.94 % (6)	97.72 % (4)	**98.40 % (1)**
Trace	**100.00 % (1)**	98.00 % (5.5)	98.00 % (5.5)	98.00 % (5.5)
TwoLeadECG	98.77 % (4)	96.14 % (6)	97.98 % (5)	99.30 % (2)
Average rank	3.48	4.45	3.64	2.93

The ranks include the results in Table 4. This experiment can be reproduced with method DMKD_2013, Bagnall et al. (2012)
The best result for each dataset is shown in bold

distance between the nearest examples of each class Cortes and Vapnik (1995). It can also transform the data into a higher dimension to make it linearly separable; we use only the linear support vector machine.

For comparison purposes, Table 4 includes the accuracy for a 1-NN-DTW classifier built on the raw data. The support vector machine is the best classifier, with an average rank of 2.93, and best performance in 10 out of 29 problems. The decision tree is the worst classifier on average; in fact, it has a worse average rank than the 1-NN-DTW

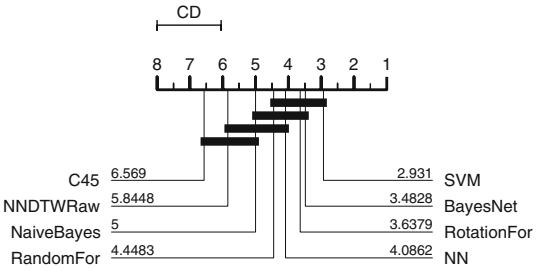


Fig. 8 Critical difference diagram for eight shapelet-based classifiers derived from the results in Tables 4 and 5

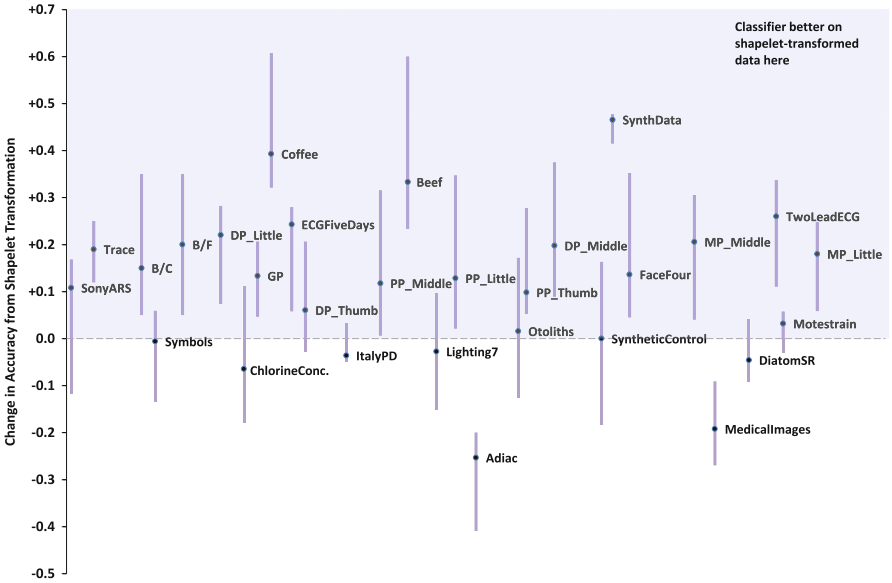


Fig. 9 Change in classification accuracy between raw data and shapelet-transformed data. Each point is the median difference; the *bar* shows the minimum and maximum change in classifier accuracy for that dataset. The results are available from Bagnall et al. (2012)

classifier based on raw data. The critical difference diagram in Fig. 8 shows that C4.5 and 1-NN-DTW are significantly worse than SVM, Bayesian network, and Rotation Forest. There may be a trade off between interpretability and accuracy (a tree is easier to understand than a support vector machine), but by separating shapelet discovery and classification, there is greater potential to explore possible solutions.

Next, we compare the accuracy of seven classifiers trained on raw data with the same classifiers trained on shapelet-transformed data. Figure 9 presents the difference in classification accuracy for each classifier on each dataset. A positive value indicates that the classifier is more accurate on the transformed data, a negative value the converse. As can be seen, the change in accuracy is strongly indexed to the dataset, as well as to the classifier. There are broad similarities between classifiers, but there are cases, such as Rotation Forest on the FaceFour dataset, where the change in accuracy is very different to that of the other classifiers.

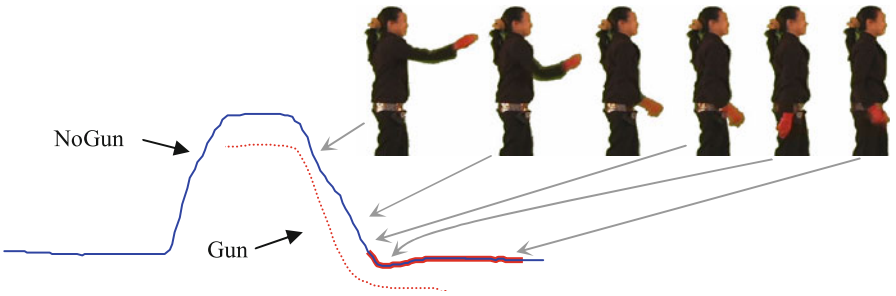


Fig. 10 An illustration of the *Gun/NoGun* problem taken from Ye and Keogh (2009). The shapelet that they extract is highlighted at the end of the series

There are a number of datasets where the shapelet-transform has been detrimental to the accuracy of the classifier, for example Adiac, where the classifiers each lose between 20 and 40 % accuracy. The image datasets and spectrograph datasets (Beef, Coffee, Beetle/Fly, Bird/Chicken) show good improvement from the shapelet transform. The synthetic data that was designed to work well with the shapelet transform shows the best and most consistent improvement, as would be expected.

The shapelet transform offers improvements in classification accuracy over several different datasets that represent a number of different types of data. This supports the shapelet approach, and suggests that it fills a classification niche that has not been covered in the literature.

6.3 Exploratory data analysis

We have shown that using shapelets to transform data can improve classification accuracy. One of the strengths of using shapelets as a classification tool is that they provide a level of interpretability that other classification approaches cannot. One of the goals of our work with shapelets is to produce accurate classification decisions that are interpretable. We demonstrate in this section that our filter retains the interpretability of the original shapelet implementation.

The GunPoint dataset consists of time series representing an actor appearing to draw a gun; the classification problem is to determine whether or not the actor is holding a prop (the *Gun/NoGun* problem). In Ye and Keogh (2009), the authors identify that the most important shapelet for classification occurs when the actor's arm is lowered; if there is no gun, a phenomenon called 'overshoot' occurs, and causes a dip in the time-series data. This is summarised in Fig. 10, taken from Ye and Keogh (2009).

The shapelet decision tree trained in Ye and Keogh (2009) contains a single shapelet at the end of the series corresponding to the arm being lowered. To demonstrate that our filter agrees with this and extracts the important information from the data, we filter the GunPoint dataset using the length parameters specified in the original paper. The top five shapelets that we extract are shown in Fig. 11, along with the shapelet reported in Ye and Keogh (2009).

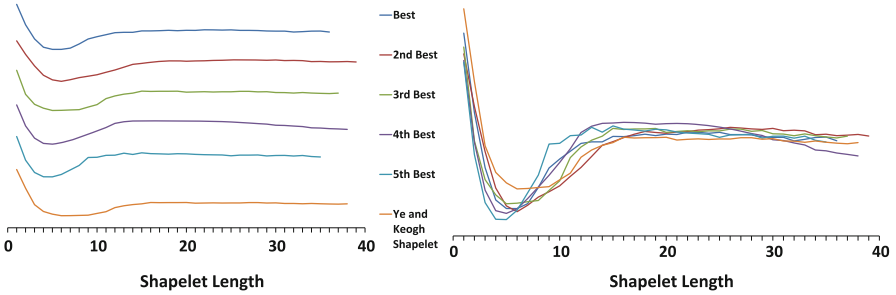


Fig. 11 An illustration of the five best shapelets extracted by our filter and the shapelet found by Ye and Keogh. The *graph* to the *right* shows how closely they match

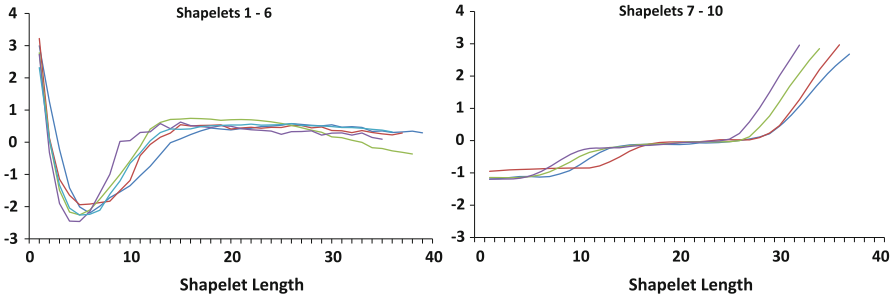


Fig. 12 The 10 best shapelets for the $Gun/NoGun$ problem. The shapelets form two distinct clusters. The graph on the left shows shapelets 1–6. They represent the 'overshoot' motion identified in Ye and Keogh (2009). The *graph* on the *right* shows shapelets 7–10. They represent the extra movement necessary to lift the prop gun when the arm is raised

Figure 11 shows that each of the top five shapelets from our filter is closely matched with the shapelet from Ye and Keogh (2009). Figure 12 shows that the best ten shapelets form two distinct clusters. Interestingly, the shapelets to the right of the figure correspond to the moments where the arm is lifted, and are instances where there is a gun. These shapelets could correspond to the subtle extra movements required to lift the prop, aiding classification by providing more information.

To explore our findings further, we hierarchically cluster the shapelets extracted from the GunPoint dataset. Our intuition is that reducing the number of shapelets will increase interpretability. The top shapelets extracted by the filter overlap to a large degree; we expect that, in cases like this, reducing k will result in a larger loss of accuracy than clustering the shapelets. In addition, multiple instances representing the same shapelet give the user less information than instances representing different shapelets.

Table 6 presents the accuracies of seven classifiers on the different transforms of the data. The transforms consist of three shapelet filters with $k = 75$, $k = 10$, and $k = 5$, and two clustered shapelet filters based on $k = 75$ shapelets clustered to 10 and 5 clusters. For the C4.5 tree, all of the transformed datasets give the same accuracy. However, for the other classifiers, the general trend is that reducing the number of shapelets results in a slight loss of accuracy (in 5 of 6 cases, the classifiers trained on the $k = 75$ shapelet-transformed data have the best, or joint best accuracy). In most

Table 6 Accuracies of seven classifiers on the GunPoint dataset

Classifier	Raw data (%)	$k = 75$ (%)	$k = 10$ (%)	10 Clusters (%)	$k = 5$ (%)	5 Clusters (%)
C4.5	77.33	**89.33**	**89.33**	**89.33**	**89.33**	**89.33**
1NN	91.33	**98.00**	**98.00**	97.33	90.00	96.00
Naive Bayes	78.67	**92.67**	90.00	90.67	87.33	89.33
Bayesian network	85.33	**99.33**	94.67	**99.33**	91.33	98.67
Random forest	91.33	**98.67**	90.00	98.00	94.00	95.33
Rotation forest	87.33	96.00	95.33	**98.67**	92.00	90.67
SVM (linear)	79.33	**100.00**	89.33	98.67	84.67	91.33

The accuracies are presented for the raw data, data transformed by filters with 75, 10, and 5 shapelets, and data transformed by a 75-shapelet filter clustered to 10 and 5 clusters
The best result for each classifier is shown in bold

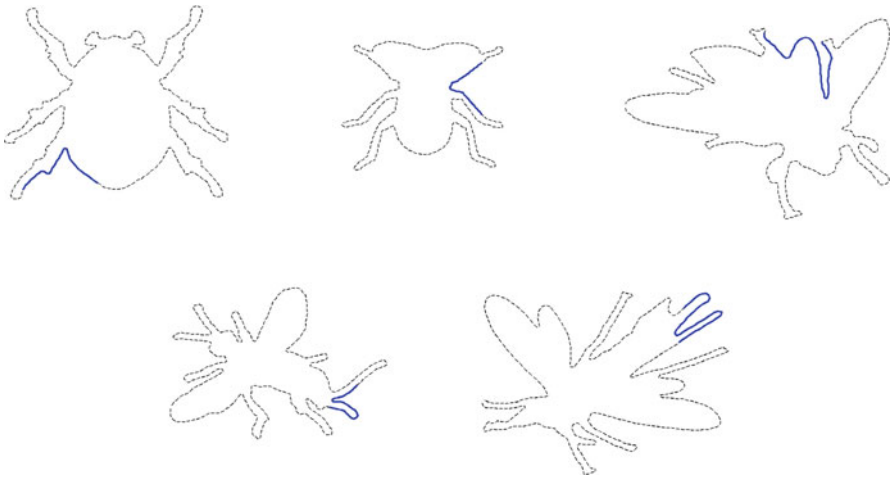


Fig. 13 The five best clustered shapelets from the Beetle/Fly dataset. The shapelets are highlighted on the outline in *blue*

cases (10 of 12), clustering the shapelets gives better accuracy than setting k to that value. This is likely to be the case because many of the top shapelets from GunPoint are very similar, which results in fewer differences between the classes in the transformed space than with the full set of shapelets. When the shapelets are clustered, the set of shapelets is likely to be more diverse, resulting in greater classification accuracy.

The main benefit of clustering the shapelets is improved interpretability. The top shapelets in the GunPoint dataset form two distinct clusters (Fig. 12). When we cluster the shapelets into 10 or 5 clusters, we find that the top two shapelets represent the two clusters found by the filter.

Other datasets show similar results when transformed using clustered shapelets. Figure 13 shows the top five clustered shapelets from the Beetle/Fly dataset. Table 7 presents the classification accuracies for the different transforms.

Clustering the shapelets for the Beetle/Fly dataset provides superior classification accuracy compared to lowering k to the same value for all but three of the fourteen

Table 7 Accuracies of seven classifiers on the Beetle/Fly dataset

Classifier	Raw data (%)	$k = 256$ (%)	$k = 10$ (%)	10 Clusters (%)	$k = 5$ (%)	5 Clusters (%)
C4.5	70.00	75.00	80.00	**82.50**	**82.50**	77.50
1NN	65.00	**100.00**	87.50	95.00	90.00	95.00
Naive Bayes	72.50	92.50	90.00	**95.00**	85.00	**95.00**
Bayesian network	72.50	**97.50**	87.50	92.50	80.00	87.50
Random forest	72.50	**90.00**	77.50	82.50	77.50	70.00
Rotation forest	77.50	**95.00**	87.50	92.50	87.50	80.00
SVM (linear)	77.50	**97.50**	87.50	**97.50**	85.00	95.00

The accuracies are presented for the raw data, data transformed by filters with 256, 10, and 5 shapelets, and data transformed by a 256-shapelet filter clustered to 10 and 5 clusters
The best result for each classifier is shown in bold

cases. There are two cases where the full shapelet transform is inferior to transforms using fewer shapelets; for the C4.5 tree, the difference is large, suggesting that the tree overfits the data when using a full set of shapelets.

In Fig. 13, the first two shapelets distinguish members of the beetle class, and the remaining three distinguish members of the fly class. By clustering down to five shapelets, we gain insight into the problem that would be less obvious from the original 256 shapelets. The beetle class is distinguished by a relatively simple angle between the legs and body; the only feature is a knee joint on the leg. The fly class is distinguished by a more complex shapelet, related to the more intricate features of the fly images.

Using only these five shapelets, three of the classifiers were able to achieve 95 % accuracy. In contrast, because of the high intraclass variation, classifiers trained on the whole outline achieved at best 77.5 % accuracy. In this case, using shapelets provides a considerable increase in classifier accuracy, and clustering down to five shapelets gives great interpretability.

For the Bird/Chicken data, the clustered shapelets (Fig. 14) are outperformed by the k best shapelets where the number of clusters is equal to k (Table 8). This shows that it is not necessarily the case that clustering improves accuracy. For different datasets, it may be worth using a validation set, or cross-validation if the dataset is small, to determine which method provides better accuracy.

The accuracies on the transformed data are all superior to the accuracies on the raw data. In many cases, using a smaller value of k results in better classification than using the full set of shapelets, perhaps because the full set overfits the data. If this is the case, it may also explain why clustering the full set is inferior to using fewer shapelets.

7 Conclusions

We investigate quality measures for shapelets, and propose a shapelet-transform algorithm.

We demonstrate the effectiveness of the Kruskal-Wallis, F-statistic, and MM statistics as quality measures for shapelet discovery by training shapelet decision-tree classifiers in the style of Ye and Keogh (2011), with these statistics in place of Information

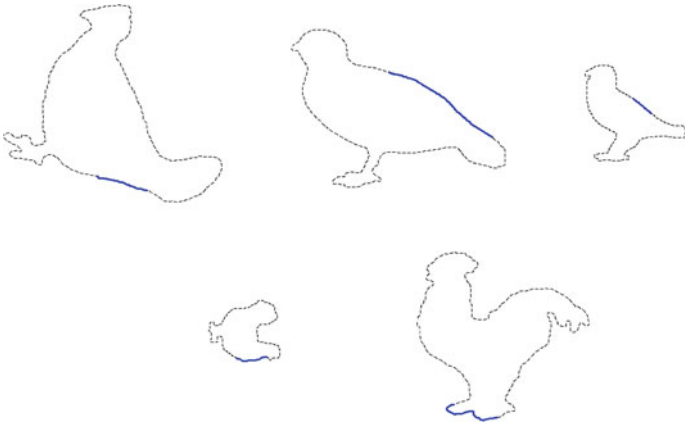


Fig. 14 The five best clustered shapelets from the Bird/Chicken dataset. The shapelets are highlighted on the *outline* in *blue*

Table 8 Accuracies of seven classifiers on the Bird/Chicken dataset

Classifier	Raw data (%)	$k = 256$ (%)	$k = 10$ (%)	10 Clusters (%)	$k = 5$ (%)	5 Clusters (%)
C4.5	75.00	**90.00**	**90.00**	87.50	87.50	87.50
1NN	85.00	**97.50**	92.50	90.00	92.50	90.00
Naive Bayes	60.00	87.50	**95.00**	92.50	92.50	90.00
Bayesian Network	60.00	95.00	**97.50**	**97.50**	95.00	82.50
Random Forest	80.00	**95.00**	87.50	90.0	87.50	85.00
Rotation Forest	87.50	92.50	**95.00**	90.00	90.00	90.00
SVM (linear)	77.50	**95.00**	**95.00**	85.00	85.00	82.50

The accuracies are presented for the raw data, data transformed by filters with 256, 10, and 5 shapelets, and data transformed by a 256-shapelet filter clustered to 10 and 5 clusters
The best result for each classifier is shown in bold

Gain. Our results show that Information Gain is the slowest measure to compute, and is no better in terms of accuracy than the other measures. The F-statistic is significantly more accurate on the synthetic data designed to be optimal for shapelets, and is the highest ranked measure overall. We believe that it should be the default measure of choice for future work with shapelet-transformed data.

We propose a shapelet-transform algorithm for time-series classification that extracts the k-best shapelets from a dataset in a single pass using a caching algorithm, and allows the shapelets to be clustered to enhance interpretability. We transform 29 datasets with our filter, and demonstrate that a C4.5 decision-tree classifier trained with transformed data is competitive with the original shapelet tree of Ye and Keogh (2009). Our transformed data can be used with other classifiers, which achieve improved accuracy while maintaining the interpretability of the shapelet approach. We provide exploratory data analysis of the shapelets extracted by our filter on the Gun/NoGun, Beetle/Fly, and Bird/Chicken problems, and show that shapelets can give insight into the problem domain.

There are limitations to the approach. First, the shapelet transform will not be optimal for all problems. Figure 9 shows that there are some datasets where the transform improves all classifiers, but others where the classifiers in the shapelet domain are less accurate than those in the time domain. Our results suggest that shapelets may be a good approach for image-outline classification, spectrograms, and ECG measurements, but these are qualitative observations. An obvious area of future work is to investigate whether there are problem domains where the shapelet approach is the best on average. Second, though the transform is faster on average than the shapelet decision tree, finding the best shapelets is slow. Exact techniques for shapelet discovery are unsuitable for large problems. Approximate techniques, such as the SAX compression described in Rakthanmanon and Keogh (2013), can mitigate this problem. Finally, the post clustering we perform improves intelligibility, but it also introduces the risk of removing important discriminatory features, and introduces a further parameter into the transform.

Despite these limitations, we believe shapelets are an important new approach for solving time-series classification problems where localised similarity in shape defines class membership. We have demonstrated that the most flexible and accurate way of using shapelets for time-series classification is as a transformation performed prior to classifier construction.

References

Bagnall A, Hills J, Lines J (2012) Shapelet based time series classification. http://www.uea.ac.uk/computing/machine-learning/Shapelets. Accessed 14 May 2013

Bagnall A, Davis L, Hills J, Lines J (2012) Transformation based ensembles for time series classification. Proceedings of the twelfth SIAM conference on data mining (SDM)

Batista G, Wang X, Keogh E (2011) A complexity-invariant distance measure for time series. Proceedings of the eleventh SIAM conference on data mining (SDM)

Bober M (2001) Mpeg-7 visual shape descriptors. IEEE Trans Circ Syst Video Technol 11(6):716–719

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Buza K (2011) Fusion methods for time-series classification. Ph.D. thesis, University of Hildesheim, Germany

Campana S, Casselman J (1993) Stock discrimination using otolith shape analysis. Can J Fish Aquat Sci 50(5):1062–1083

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Davis LM, Theobald B-J, Lines J, Toms A, Bagnall A (2012) On the segmentation and classification of hand radiographs. Int J Neural Syst 22(5):1250020

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Deng H, Runger G, Tuv E, Vladimir M (2011) A time series forest for classification and feature extraction. Tech. rep., Arizona State University

De Vries D, Grimes C, Prager M (2002) Using otolith shape analysis to distinguish eastern gulf of mexico and atlantic ocean stocks of king mackerel. Fish Res 57(1):51–62

Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E (2008) Querying and mining of time series data: experimental comparison of representations and distance measures. Proc VLDB Endow 1(2):1542–1552

Duarte-Neto P, Lessa R, Stosic B, Morize E (2008) The use of sagittal otoliths in discriminating stocks of common dolphinfish (coryphaena hippurus) off northeastern brazil using multishape descriptors. ICES J Mar Sci J du Conseil 65(7):1144–1152

Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2–3):131–163

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Hartmann B, Link N (2010) Gesture recognition with inertial sensors and optimized dtw prototypes. Systems man and cybernetics (SMC), 2010 IEEE international conference on. IEEE pp 2102–2109

He Q, Dong Z, Zhuang F, Shi Z (2012) Fast Time Series Classification based on infrequent shapelets. Machine learning and applications (ICMLA), 2012 11th international conference on. IEEE, pp 215–219

Hoare C (1962) Quicksort. Comput J 5(1):10–16

Image Processing and Informatics Lab, University of Southern California, The Digital Hand Atlas Database System. http://www.ipilab.org/BAAweb

Janacek G, Bagnall A, Powell M (2005) A likelihood ratio distance measure for the similarity between the fourier transform of time series. Proceedings of the Ninth Pacific-Asia Conference on knowledge discovery and data mining (PAKDD)

Jeong Y, Jeong M, Omitaomu O (2010) Weighted dynamic time warping for time series classification. Pattern Recognit 44:2231–2240

Hu B, Chen Y, Keogh E (2013) Time series classification under more realistic assumptions. Proceedings of the thirteenth SIAM conference on data mining (SDM)

Kruskal W (1952) A nonparametric test for the several sample problem. Ann Math Stat 23(4):525–540

Latecki L, Lakamper R, Eckhardt T (2000) Shape descriptors for non-rigid shapes with a single closed contour. Computer vision and pattern recognition, 2000. Proceedings. IEEE conference on vol. 1. IEEE, pp 424–429

Lines J, Bagnall A (2012) Alternative quality measures for time series shapelets. Intelligent data engineering and automated learning (IDEAL). Lect Notes Comput Sci 7435:475–483

Lines J, Davis L, Hills J, Bagnall A (2012) A shapelet transform for time series classification. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 289–297

Mood AM, Graybill FA, Boes DC (1974) Introduction to the theory of statistics, 3rd edn. McGraw-Hill, New York

Mueen A, Keogh E, Young N (2011) Logical-shapelets: an expressive primitive for time series classification. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1154–1162

Rakthanmanon T, Campana B, Mueen A, Batista G, Westover B, Zhu Q, Zakaria J, Keogh E (2012) Searching and mining trillions of time series subsequences under dynamic time warping. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 262–270

Rakthanmanon T, Keogh E (2013) Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets. Proceedings of the thirteenth SIAM conference on data mining (SDM)

Rodriguez J, Alonso C (2005) Support vector machines of interval-based features for time series classification. Knowl-Based Syst 18:171–178

Rodriguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. Pattern Anal Mach Intell IEEE Trans 28(10):1619–1630

Shannon C, Weaver W, Blahut R, Hajek B (1949) The mathematical theory of communication, vol 117. University of Illinois press, Urbana

Sivakumar P et al. (2012) Human gait recognition and classification using time series shapelets. International conference on advances in computing and communications (ICACC)

Stransky C (2005) Geographic variation of golden redfish (sebastes marinus) and deep-sea redfish (s. mentella) in the north atlantic based on otolith shape analysis. ICES J Mar Sci J du Conseil 62(8):1691–1698

Wu Y, Agrawal D, Abbadi AE (2000) A comparison of dft and dwt based similarity search in time-series databases. Proceedings of the ninth international conference on information and knowledge management (ACM CIKM)

Xing Z, Pei J, Yu P (2012) Early classification on time series. Knowl Inform syst 31(1):105–127

Xing Z, Pei J, Yu P, Wang K (2011) Extracting interpretable features for early classification on time series. Proceedings of the eleventh SIAM conference on data mining (SDM)

Ye L, Keogh E (2009) Time series shapelets: a new primitive for data mining. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 947–956

Ye L, Keogh E (2011) Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min Knowl Discov 22(1):149–182

Zakaria J, Mueen A, Keogh E (2012) Clustering time series using unsupervised-shapelets. Data mining (ICDM), 2012 IEEE 12th international conference. pp. 785–794