Generalised Interpretable Shapelets for Irregular Time Series

Patrick Kidger* **James Morrill*** **Terry Lyons**

Mathematical Institute, University of Oxford
The Alan Turing Institute, British Library
{kidger, morrill, tlyons}@maths.ox.ac.uk

Abstract

The shapelet transform is a form of feature extraction for time series, in which a time series is described by its similarity to each of a collection of 'shapelets'. However it has previously suffered from a number of limitations, such as being limited to regularly-spaced fully-observed time series, and having to choose between efficient training and interpretability. Here, we extend the method to continuous time, and in doing so handle the general case of irregularly-sampled partially-observed multivariate time series. This additionally then allows for learning the length of each shapelet (previously a discrete object) in a differentiable manner. Furthermore, we show that a simple regularisation penalty may be used to train efficiently without sacrificing interpretability. Finally, we generalise the measure of similarity between time series so as to be a learnt pseudometric. We validate our method by demonstrating its empirical performance on several datasets.

1 Introduction

Shaplets are a form of feature extraction for time series [1, 2, 3, 4]. Given some fixed hyperparameter K, describing how many shapelets we are willing to consider, then each time series is represented by a vector of length K describing how similar it is to each of the k selected shapelets.

We begin by recalling the classical definition of the shapelet transform [5].

1.1 Classical shapelet transform

Given N regularly sampled multivariate time series, with D observed channels, where the n-th time series is of length T_n, then the n-th time series is a matrix

$$f^n = (f^n_t)_{t\in\{0,\ldots,T_n-1\}} = (f^n_{t,d})_{t\in\{0,\ldots,T_n-1\},d\in\{1,\ldots,D\}}, \tag{1}$$

with each $f^n_{t,d} \in \mathbb{R}$.

Fix some hyperparameter $K \in \mathbb{N}$, which will describe the number of shapelets. Fix some $S \in \{0, \ldots, \min_{i\in\{1,\ldots,N\}} T_n - 1\}$, which will describe the length of each shapelet. Then the k-th shapelet is a matrix

$$w^k = (w^k_t)_{t\in\{0,\ldots,S-1\}} = (w^k_{t,d})_{t\in\{0,\ldots,S-1\},d\in\{1,\ldots,D\}},$$

with each $w^k_{t,d} \in \mathbb{R}$.

*Equal contribution.

Preprint. Under review.

Then the discrepancy between f^n and w^k is given by (sometimes without the square):

$$\sigma_S(f^n, w^k) = \min_{s \in \{0, \ldots, T_n - S\}} \sum_{t=0}^{S-1} \|f^n_{s+t} - w^k_t\|_2^2, \tag{2}$$

where $\|\cdot\|_2$ describes the L^2 norm on $\mathbb{R}^D$. A small discrepancy implies that f^n and w^k are similar to one another. This corresponds to sweeping w^k over f^n, and finding the offset s at which w^k best matches f^n. The collection of $(\sigma_S(f^n, w^1), \ldots, \sigma_S(f^n, w^K)) \in \mathbb{R}^K$ is now a feature describing f^n. This may now be passed to some model to perform classification or regression.

1.2 Limitations

The classical shapelet method suffers from a number of limitations.

1. The technique only applies to regularly spaced time series.
2. The choice of S is a hyperparameter; it is discrete, and choosing it is thus a relatively expensive optimisation procedure.
3. Learning w^k by searching [1] is expensive, whilst optimising differentiably [2] typically sacrifices interpretability [6].

Besides this, the choice of L^2 norm is ad-hoc and a general formulation should allow for other notions of similarity.

1.3 Contributions

We demonstrate how classical shapelets may be generalised in multiple ways, so as to address the collection of limitations just described.

First, we extend the method to continuous time rather than discrete time, allowing the treatment of irregularly-sampled partially-observed multivariate time series on the same footing as regular time series. Second, this continuous-time formulation means that the length of each shapelet (previously a discrete value) takes its values in a continuous range, and may now be trained differentiably.

Third, we demonstrate how simple regularisation is enough to achieve shapelets that resemble characteristic features of the data, achieving both interpretability and pattern discovery. Finally, we generalise the discrepancy between a shapelet and a time series to be a learnt pseudometric.

Our code is available at `https://github.com/jambo6/generalised_shapelets`.

2 Prior work

Shapelets may be selected as small intervals extracted from training samples [1]. However doing so is very expensive, requiring $\mathcal{O}(N^2 \cdot \max_n T_n^4)$ work. Much work on shapelets sought speedup techniques [7, 8, 9], for example via random algorithms [10, 11].

However [2] observe that the discrepancy σ_S of equation (2) is differentiable with respect to w^k, so that shapelets may be differentiably optimised jointly with the subsequent model, as part of an end-to-end optimisation of the final loss function. (Although [2] include a 'softmin' procedure which we believe to be unnecessary, as the minimum function is already almost everywhere differentiable.) This costs only $\mathcal{O}(N \cdot \max_n T_n^2)$ to train, and is the approach that we extend here.

This method is attractive for its speed and its ease of trainability via modern deep learning frameworks [12, 13, 14]. However, [6] observe that the predictive power of the distance between a shapelet and a time series need not correlate with a similarity between the two, so there is no pressure towards interpretability. [6] propose to solve this via adversarial regularisation; we will present a simpler alternative later. Without such procedures, then efficient training and interpretability become mutually exclusive.

The method may additionally be generalised by considering alternative notions of similarity between a shapelet and a time series; for example [15] replace the L^2 norm with dynamic time warping.

The shapelet method is attractive for its normalisation of variable-length time series and demonstratation of typically good performance [4, 16]. Additionally, it is (depending on training procedure) interpretable, as use of a particular feature corresponds to the importance of the similarity to the shapelet w^k. This may describe some shape that is characteristic of a particular class, and can discover previously unknown patterns in the data (we will see examples later).

3 Method

3.1 Continuous-time objects

We interpret a time series as a discretised sample from an underlying process, observed only through the time series. Similarly, a shapelet constructed as in Section 1.1 may be thought of as a discretisation of some underlying function. The first important step in our procedure is to construct continuous-time approximations to these underlying objects.

Continuous-time path interpolants Formally speaking, we assume that for $n \in \{1, \dots, N\}$ indexing different time series, we observe a collection of time series

$$f^n = (f^n_{t_\tau})_{\tau \in \{1,\dots,T_n\}},$$

where $t_\tau \in \mathbb{R}$ is the observation time of $f^n_{t_\tau} \in (\mathbb{R} \cup \{*\})^D$, where $*$ denotes the possiblity of a missing observation.

Next, interpolate to get a function $\iota(f^n)\colon [0, T_n - 1] \to \mathbb{R}^D$ such that $\iota(f^n)(t_\tau) = f^n_{t_\tau}$ for all $\tau \in \{0, \dots, T_n - 1\}$ such that $f^n_{t_\tau}$ is observed. There are many possible choices for interpolations, for example splines, kernel methods [17], or Gaussian processes [18, 19]. In our experiments, we use piecewise linear interpolation.

Continuous-time shapelets The shapelets themselves we are free to control, and so for $k \in \{1, \dots, K\}$ indexing different shapelets, we take each $w^{k,\rho}\colon [0, 1] \to \mathbb{R}^D$ to be some learnt function depending on learnt parameters ρ. For example, this could be an interpolated sequence of learnt points, an expansion in some basis functions, or a neural network. In our experiments we use linear interpolation of a sequence of a learnt points.

Then for some learnt length $S_k > 0$, define $w^{k,\rho,S_k}\colon [0, S_k] \to \mathbb{R}^D$ by

$$w^{k,\rho,S_k}(t) = w^{k,\rho}\left(\frac{t}{S_k}\right).$$

Taking the length S_k to be continuous is a necessary prerequisite to training it differentiably. We will discuss the training procedure in a moment.

3.2 Generalised discrepancy

The core of the shapelet method is that the similarity or discrepancy between f^n and w^{k,ρ,S_k} is important. In general, we approach this by defining a *discrepancy function* between the two, which will typically be learnt, and which we require only to be a pseudometric.

We denote this discrepancy function by π^A_S. It depends upon a length S and a learnt parameter A, consumes two paths $[0, S] \to \mathbb{R}^D$, and returns a real number describing some notion of closeness between them. We are being deliberately vague about the regularity of the domain of $\pi^A_{S_k}$, as it is a function space whose regularity will depend on ι.

Given some π^A_S, then the discrepancy between f^n and w^{k,ρ,S_k} is defined as

$$\sigma^A_{S_k}(f^n, w^{k,\rho,S_k}) = \min_{s \in [0, T_n - S_k]} \pi^A_{S_k}(\iota(f^n)|_{[s,s+S_k]}(s + \cdot), w^{k,\rho,S_k}). \tag{3}$$

The collection of discrepancies $(\sigma^A_{S_k}(f^n, w^{1,\rho,S_k}), \dots, \sigma^A_{S_k}(f^n, w^{K,\rho,S_k}))$ is now a feature describing f^n, and is invariant to the length T_n. Use of the particular feature $\sigma^A_{S_k}(f^n, w^{k,\rho,S_k})$ corresponds to the importance of the similarity between f^n and w^{k,ρ,S_k}. In this way, the choice of $\pi^A_{S_k}$ gives a great deal of flexibility, as we are about to see.

Existing shapelets fit into this framework A simple example, in analogy to the classical shapelet method of equation (2), is to take

$$\pi^A_{S_k}(f, w) = (\int_0^{S_k} \|f(t) - w(t)\|_2^2 \,\mathrm{d}t)^{\frac{1}{2}},$$

which in fact has no A dependence. If ι is taken to be a piecewise constant 'interpolation' then this will exactly correspond to (the square root of) the classical shapelet approach.

Learnt L^2 discrepancies The previous example may be generalised by taking our learnt parameter $A \in \mathbb{R}^{D\times D}$, and then letting

$$\pi^A_S(f, w) = (\int_0^S \|A(f(t) - w(t))\|_2^2 \,\mathrm{d}t)^{\frac{1}{2}}. \tag{4}$$

That is, allowing some learnt linear transformation before measuring the discrepancy. In this way, particularly informative dimensions may be emphasised. In our experiments we take A to be diagonal. Allowing a general matrix was found during initial experiments to produce slightly worse performance.

More complicated discrepancies Moving on, we consider other more general choices of discrepancy, which may be motivated by the problem at hand. In particular we will discuss discrepancies based on the logsignature transform [20], and mel-frequency cepstrums (MFC) [21].

Our exposition on these two discrepancies will be deliberately brief, as the finer details on exactly when and how to use them is domain-specific. The point is that our framework has the flexibility to consider general discrepancies motivated by other discplines, or which are known to extract information which is particular useful to the domain in question. An understanding of either logsignatures or mel-frequency cepstrums will not be necessary to follow the paper.

Logsignature discrepancies The logsignature transform is a transform on paths, known to characterise its input whilst extracting statistics which describe how the path controls differential equations [20, 22, 23, 24]. Let μ denote the Möbius function, and let

$$\beta_{D,R} = \sum_{r=1}^{R} \frac{1}{r} \sum_{\rho|r} \mu\left(\frac{r}{\rho}\right) D^\rho,$$

which is Witt's formula [25]. Let

$$\mathrm{LogSig}^R\colon \left\{f\colon [0,T] \to \mathbb{R}^D \;\middle|\; T \in \mathbb{R}, f \text{ is of bounded variation}\right\} \to \mathbb{R}^{\beta_{D,R}}$$

be the depth-R logsignature transform. Let $A \in \mathbb{R}^{\beta_{D,R}\times\beta_{D,R}}$ be full or diagonal as before, and let $\|\cdot\|_p$ be the L^p norm on $\mathbb{R}^{\beta_{D,R}}$. Then we define the *p-logsignature discrepancy* between two functions to be

$$\pi^A_S(f, w) = \|A(\,\mathrm{LogSig}^R(f) - \,\mathrm{LogSig}^R(w))\|_p. \tag{5}$$

MFC discrepancies The computation of an MFC is a function-to-function map derived from the short-time Fourier transform, with additional processing to focus on frequencies that are particularly relevant to human hearing [21]. Composing this with the L^2 based discrepancy of equation (4) produces

$$\pi^A_S(f, w) = (\int_0^S \|A(\mathrm{MFC}(f)(t) - \mathrm{MFC}(w)(t))\|_2^2 \,\mathrm{d}t)^{\frac{1}{2}}. \tag{6}$$

The generalised shapelet transform Whatever the choice of π^A_S, and in analogy to the classical shapelet transform [5], we call the map

$$f \mapsto (\sigma^A_{S_1}(f, w^{1,\rho,S_1}), \ldots, \sigma^A_{S_K}(f, w^{K,\rho,S_K}))$$

the *generalised shapelet transform*.

3.3 Interpretable regularisation

As previously described, learning shapelets differentiably can sacrifice interpretability [6], as the learnt shapelets need not resemble the training data. We propose a novel regularisation penalty to solve this: simply add on

$$\sum_{k=1}^{K} \min_{n \in \{1,\ldots,N\}} \sigma_S^A(f^n, w^{k,\rho,s}) \tag{7}$$

as a regularisation term, so that minimising the discrepancy between f^n and $w^{k,\rho,S}$ is also important. Note the choice of minimisation over n, rather than a sum over n. A sum over n would ask that every shapelet should look like every training sample. Taking a minimum instead asks only that every shapelet should be similar to a single training sample.

3.4 Minimisation objective and training procedure

Overall, suppose we have some differentiable model F^θ parameterised by θ, some loss function $\mathcal{L}$, and some observed time series $f^1, \ldots, f^N$ with targets $y_1, \ldots, y_N$.

Then letting $\gamma > 0$ control the amount of regularisation, we propose to minimise

$$\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(y_n, F^\theta(\sigma_{S_1}^A(f^n, w^{1,\rho,S_1}), \ldots, \sigma_{S_K}^A(f^n, w^{K,\rho,S_K}))) + \gamma \sum_{k=1}^{K} \min_{n \in \{1,\ldots,N\}} \sigma_{S_k}^A(f^n, w^{k,\rho,S_k}) \tag{8}$$

over model parameters θ, discrepancy parameters A, shapelet parameters ρ, and shapelet lengths S_k, via standard stochastic gradient descent based techniques.

Differentiability Some thought is necessary to verify that this constructions is differentiable, in particular with respect to S_k. Examining the definition of $\sigma_{S_k}^A$ in equation (3), there are two operations that may seem to pose a problem, namely the minimum over a range $\min_{s \in [0, T_n - S_k]}$, and the restriction operator $\iota(f^n) \mapsto \iota(f^n)|_{[s,s+S_k]}$.

Practically speaking, however, it is straightforward to resolve both of these issues. For the minimum over a range, this may reasonably be approximated by a minimum over some collection of points $s \in \{0, \varepsilon, 2\varepsilon, \ldots, T_n - S_k - \varepsilon, T_n - S_k\}$, for some $\varepsilon > 0$ small and dividing $T_n - S_k$. This is now a standard piece of an autodifferentiation package. The error of this approximation may be controlled by the modulus of continuity of $s \mapsto \pi_{S_k}^A(\iota(f^n)|_{[s,s+S_k]}(s + \cdot), w^{k,\rho,S_k})$, but in practice we found this to be unnecessary, and simply took ε equal to the smallest gap between observations.

Next, the continuous-time paths $\iota(f^n)$ and continuous-time shapelets w^{k,ρ,S_k} must both be represented by some parameterisation of function space, and it is thus sufficient to restrict to considering differentiability with respect to this parameterisation.

In our experiments we represent both $\iota(f^n)$ and w^{k,ρ,S_k} as a continuous piecewise linear function stored as a collection of knots. In this context, the restriction operator is clearly differentiable, as a map from the unrestricted function, represented by one collection of knots, to the restricted function, represented by another collection of knots. Each knot is either kept (the identity function), thrown away (the zero function), or interpolated between to place a new knot at the boundary (a ratio of existing knots).

Choice of F^θ Interpretability of the model will depend on an interpretable choice of F^θ. In our experiments we used a linear model on the logarithm of every feature, so that a very negative coefficient corresponds to the importance of f^n and w^{k,ρ,S_k} being similar to each other.

4 Experiments

We compare our generalised shapelet transform to the classical shapelet transform, in terms of both performance and interpretability, on a large range of time series classification problems. Every experiment is run three times, and we report the mean and standard deviation of test accuracy. We take the interpolation scheme ι to be piecewise linear interpolation; in particular efficient algorithms for computing the logsignature transform only exist for piecewise linear paths [24].

Table 1: Test accuracy (mean $\pm$ std, computed over three runs) on UEA. A 'win' is the number of times each algorithm was within 1 standard deviation of the top performer for each dataset.

	Discrepancy		
Dataset	L^2	Logsignature	Classical
BasicMotions	90.8% $\pm$ 1.4%	80.8% $\pm$ 3.8%	**96.7% $\pm$ 5.8%**
ERing	**82.6% $\pm$ 6.3%**	43.3% $\pm$ 2.9%	67.2% $\pm$ 11.8%
Epilepsy	**88.4% $\pm$ 3.0%**	**88.6% $\pm$ 0.8%**	72.9% $\pm$ 5.4%
Handwriting	10.3% $\pm$ 2.6%	**11.8% $\pm$ 1.2%**	6.5% $\pm$ 3.7%
JapaneseVowels	**97.2% $\pm$ 1.1%**	53.9% $\pm$ 3.0%	91.5% $\pm$ 4.1%
Libras	**67.0% $\pm$ 9.4%**	**67.8% $\pm$ 5.5%**	62.2% $\pm$ 2.4%
LSST	**36.1% $\pm$ 0.2%**	35.7% $\pm$ 0.4%	33.5% $\pm$ 0.5%
PenDigits	**97.3% $\pm$ 0.1%**	96.7% $\pm$ 0.7%	**97.5% $\pm$ 0.6%**
RacketSports	**79.6% $\pm$ 0.7%**	61.2% $\pm$ 9.2%	**79.6% $\pm$ 2.4%**
Wins	7	3	3

The regularisation parameter γ is taken to be 10^{-4}. This was selected by starting at 10^{-3} and reducing the value until test accuracy no longer improved, so as to ensure that it did not compromise performance.

The loss was cross entropy, the optimiser was Adam [26] with learning rate 0.05 and batch size 1024. If validation loss stagnated for 20 epochs then the learning rate was reduced by a factor of 10 and training resumed, down to a minimum learning rate of 0.001. We note that these relatively large learning rates are proportional to the large batch size, as is standard practice. If validation loss and accuracy failed to decrease over 60 epochs then training was halted. Once training was completed then the model parameters were rolled back to those which produced the highest validation accuracy.

Precise experimental details may be found in Appendix A.

4.1 The UEA Time Series Archive

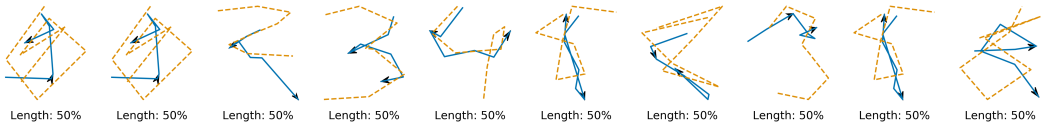
This is a collection of 30 fully-observed regularly-sampled datasets with varying properties [27]. Evaluating on the full collection of datasets would take a prohibitively long time, and so we select 9 representing a range of difficulties. Details of these (number of training samples, etc.) can be found in Appendix A.

We begin by performing hyperparameter optimisation (number of shapelets, length of shapelets) for the classical shapelet transform, separately for each dataset. We then use the same hyperparameters for the generalised shapelet transform. For the generalised shapelet transform, the length hyperparameter is used to determine the initial length of the shapelet, but this may of course vary as it is learnt.

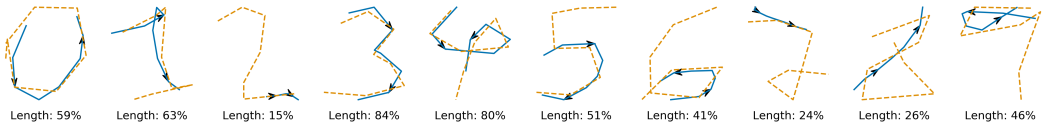
For the generalised shapelet transform, we consider two different discrepancy functions, namely the learnt L^2 discrepancy and p-logsignature discrepancies of equations (4) and (5). For the latter, we take $p = 2$ and the depth $R = 3$. We did not try to optimise p and R, as we use the logsignature discrepancy simply to highlight the possibility of using more unusual discrepancies if desired.

The results are given in Table 1. We see that the generalised shapelet transform with L^2 discrepancy function achieves within one standard deviation of the top performing algorithm on 7 of the 9 datasets, whilst the classical approach does so for only 3. As crude summary statistics, the average performance gain across all datasets was 4.6% points, the maximum performance drop was only 5.9% points (on BasicMotions), and the maximum performance gain was 15.4% points (on ERing).

Interpretability on PenDigits We demonstrate interpretability by examining the PenDigits dataset. (Chosen because of its nice visuals.) This is a dataset of handwritten digits 0–9, sampled at 8 points along their trajectory. We select the most informative shapelet for each of the ten classes (as in Section 3.4), for both the classical shapelet transform and the generalised shapelet transform, with L^2 discrepancy. We then locate the training sample that it is most similar to, and plot an overlay of the two. See Figure 1.



(a) Classical shapelet transform.



(b) Generalised shapelet transform with L^2 discrepancy.

Figure 1: The most significant shapelet for each class (blue, solid), overlaid with the most similar training example (orange, dashed). Similarity is measured with respect to the (learnt) discrepancy function.

We can clearly see multiple issues with the shapelets learnt with the classical approach. The most significant shapelet for the classes 0 and 1 is the same shapelet, and for classes 1, 5, 6, 7, 9, the most significant shapelet is not even closest to a member of that class. Visually, the shapelets for 3 and 4 seem to have identified distinguishing features of those classes, but the shapelets corresponding to the other classes appear to be little more than random noise.

In contrast, the generalised shapelet approach is abundantly clear. Every class has a unique most significant shapelet, and every such shapelet is close to a member of the correct class. In the case of class 3, the shapelet has essentially reproduced the entire digit.

We see that this is a benefit of allowing learnt lengths, so that particularly distinguishing features, such as the double bend of a 3, may be resolved. Contrast with the most significant shapelet for the class 3 under the classical shapelet transform: it is almost perfectly located in the middle of that which was learnt for the generalised shapelet transform, which we speculate is a 'best fit' under the fixed length constraint.

A point of significant interest is the difference between the shapelets for the digits 5 and 6, for the generalised shapelet transform. Whilst visually very similar, we see that the difference between them is their direction. In other words, whilst a 5 and a 6 may appear visually similar on the page (with a loop in the bottom half of the digit), they may clearly be distinguished by the direction in which they tend to be written. This is a perfect example of discovering something about the data that was not necessarily already known!

Another good example of such discovery is the shapelet corresponding to the class 7, for the generalised shapelet transform. This is perhaps surprising to see as a distinguising feature of a '7'. However it turns out that no other digit features a stroke in that direction, in that place! (Figuring this out was a fun moment for the authors, sketching figures in the air.) A similar case can be made for the '2' shapelet.

4.2 Missing data and benefit of learning lengths

We now investigate the efficacy of our proposed method when there is missing data. Classical shapelet

We now demonstrate the ability of our proposed method to handle partially observed data, as well as show the effectivness of enabling learnt lengths. To test this we consider 3 datasets from the UEA archive and run experiments where we drop 10%, 30%, and 50% of the data for the cases when learnt lengths are enabled and length is left as fixed.

We first run a small hyperparameter study to determine the optimal number of shaplets and shapelet length for the fixed model. The best parameters on the validation set are then used to train the fixed model and we train the learnt model with the same number of classes but initialise the lengths of the shapelets to the worst performing length value as found in the hyperparameter study. The hope

is that the learnt model will be able to find the best length, starting at the worst length, without the need for the additional hyperparameter tuning step.

The results are given in Table 2. Firstly we note the effectiveness with which the model can deal with missing data with the accuracy being maintained well as more data is dropped. We also see that the learnt length starting at the worst shapelet length, leads to comparable performance with the model starting at the best length, demonstrating the effectiveness of this addition to the method.

Dataset	**Dropped data**	**Lengths selected by**	
		Differentiable optimisation	Hyperparameter searching
JapaneseVowels	10%	**93.2% ± 2.1%**	**93.1% ± 0.9%**
	30%	**91.2% ± 4.1%**	**91.4% ± 2.8%**
	50%	**93.5% ± 1.1%**	92.0% ± 1.1%
LSST	10%	40.2% ± 3.5%	**44.0% ± 1.0%**
	30%	**38.1% ± 0.3%**	**40.2% ± 5.6%**
	50%	41.5% ± 2.7%	**44.4% ± 0.5%**
Libras	10%	57.4% ± 4.2%	**59.3% ± 1.6%**
	30%	**81.2% ± 7.6%**	63.9% ± 8.2%
	50%	**62.5% ± 14.8%**	**65.3% ± 8.6%**
Wins		6	7

Table 2: Test accuracy (mean ± std, computed over three runs) on 3 of the UEA datasets for different proportions of dropped data and the cases where learnt length is enabled and when length is fixed. A 'win' is defined as the number of times each algorithm was within 1 standard deviation of the top performer for each dataset.

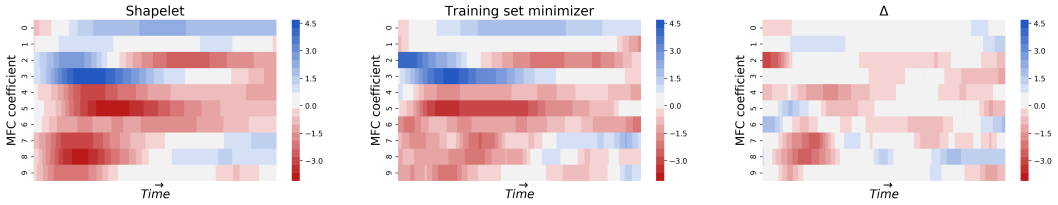
4.3 Speech Commands

Next we consider the Speech Commands dataset [28]. This is comprised of one-second audio files, corresponding to words such as 'yes', 'no', 'left', 'right', and so on. We consider 10 classes so as to create a balanced classification problem.

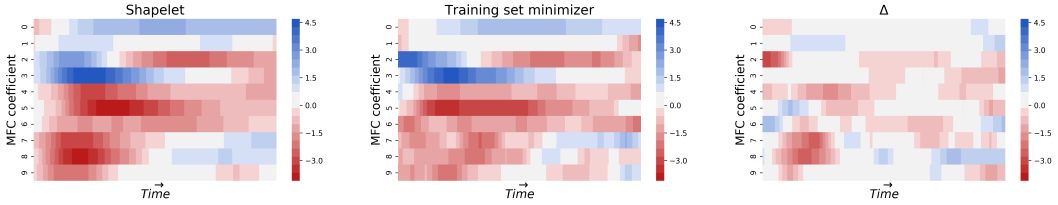
For the generalised shapelet transform, we use the MFC discrepancy described in equation (6). Knowing that this is a problem to do with spoken audio, choosing a discrepancy in this way allows us to exploit domain knowledge.

For this more complicated dataset, we found that the generalised shapelet transform substantially outperformed the classical shapelet transform. (To keep things fair, the classical shaplet transform is performed in MFC-space; the performance gap is not due to this.) We find that the classical shapelet transform produces a test accuracy of 44.8% ± 8.6%, whilst our more flexible generalised shapelet transform, with L^2 discrepancy, produces a test accuracy of 91.9% ± 2.4% (mean ± std, averaged over three runs).

Interpretability of Speech Commands To examine the interpretability of speech commands, we refer to Figure 2 which shows the first 10 MFC coefficiets for a given shapelet (left) the corresponding training set discrepancy minimizer (middle) along with the difference between the two (right). It is clear from the difference plot that the new method produces shapelets far more similar to the closest training example and thus, we expect, far easier to interpret it's meaning in terms of audio. We have converted the shapelet MFC coefficients back into audio format and you can hear the sounds at ???. Whilst it is not as easy to identify what aspect of the sound belongs to what word, clear differences can certainly be made out, we expect someone more well versed in audio analysis than ourselves would be able to better analyse these resulting sound bites. We refer to Appendix A.4 for the full plots of each MFC coefficient for the shapelet against the training set example.



(a) Classical shapelet transform.



(b) Generalised shapelet transform with L^2 discrepancy.

Figure 2: First 10 MFC coefficients for the shapelet (left), training set minimizer (middle), and the difference between them (right).

5 Conclusion

In this work we have extended and generalised the classical shapelet method in several ways. We have generalised it from discrete time to continuous time, and in doing so extend the method to the general case of irregularly-sampled partially-observed multivariate time series; furthermore this allows for the length of each shapelet to be treated as a parameter rather than a hyperparameter and optimised differentiably. We have introduced generalised discrepancies to allow for domain adaptation. Finally we have introduced a simple regularisation penalty that produces interpretable results capable of giving new insight into the data.

Broader Impact

Interpretability is important in the application of machine learning systems, so that the reason for choices made on the basis of that system can be justified, fairly, and without undue bias. Furthermore, methods which give new insight into the data are valuable for their ability to help the subequent development of theory. The generalised shapelet transform, with interpretable regularisation, is capable of supporting both of these objectives, and so it is our hope that a substantial part of the broader impact of this work will be its contributions towards these strategic goals.

Acknowledgments and Disclosure of Funding

PK was supported by the EPSRC grant EP/L015811/1. JM was supported by the EPSRC grant EP/L015803/1 in collaboration with Iterex Therapuetics. PK, JM, TL were supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

[1] L. Ye and E. Keogh, "Time Series Shapelets: A New Primitive for Data Mining," *KDD 2009*, 2009.

[2] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning Time-Series Shapelets," *KDD 2014*, 2014.

[3] L. Hou, J. Kwok, and J. Zurada, "Efficient Learning of Timeseries Shapelets," *AAAI 2016*, 2016.

[4] A. Bagnall, A. Bostrom, J. Large, and J. Lines, "The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms. Extended Version," *arXiv:1602.01711*, 2016.

[5] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery*, vol. 28, pp. 851–881, 2014.

[6] Y. Wang, R. Emonet, E. Fromont, S. Malinowski, E. Menager, L. Mosser, and R. Tavenard, "Learning Interpretable Shapelets for Time Series Classification through Adversarial Regularization," *CAP 2019*, 2019.

[7] A. Mueen, E. Keogh, and 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*, pp. 1154–1162, 2011.

[8] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme, "Scalable Discovery of Time-Series Shapelets,"

[9] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme, "Fast classification of univariate and multivariate time series through shapelet discovery," *Knowl. Inf. Syst.*, vol. 49, pp. 429–454, 2016.

[10] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proceedings of the 13th SIAM International Conference on Data Mining*, pp. 668–676, 2013.

[11] M. Wistuba, J. Grabocka, and L. Schmidt-Thieme, "Ultra-Fast Shapelets for Time Series Classification," *arXiv:1503.05018*, 2015.

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from tensorflow.org.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[14] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne, "JAX: composable transformations of Python+NumPy programs," 2018.

[15] M. Shah, J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning DTW-Shapelets for Time-Series Classification," *CODS 2016*, 2016.

[16] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," in *Big Data Analytics and Knowledge Discovery* (S. Madria and T. Hara, eds.), (Cham), pp. 257–269, Springer International Publishing, 2015.

[17] S. N. Shukla and B. Marlin, "Interpolation-prediction networks for irregularly sampled time series," in *International Conference on Learning Representations*, 2019.

[18] S. C.-X. Li and B. M. Marlin, "A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification," in *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2016.

[19] J. Futoma, S. Hariharan, and K. Heller, "Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1174–1182, 2017.

[20] S. Liao, T. Lyons, W. Yang, and H. Ni, "Learning stochastic differential equations using RNN with log signature features," *arXiv:1908.08286*, 2019.

[21] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Advances in Multimedia Information Processing - PCM 2004* (K. Aizawa, Y. Nakamura, and S. Satoh, eds.), (Berlin, Heidelberg), pp. 566–574, Springer Berlin Heidelberg, 2005.

[22] T. Lyons, M. Caruana, and T. Levy, *Differential equations driven by rough paths*. Springer, 2004. École d'Été de Probabilités de Saint-Flour XXXIV - 2004.

[23] P. Bonnier, P. Kidger, I. Perez Arribas, C. Salvi, and T. Lyons, "Deep Signature Transforms," in *Advances in Neural Information Processing Systems*, pp. 3099–3109, 2019.

[24] P. Kidger and T. Lyons, "Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU," *arXiv:2001.00706*, 2020. `https://github.com/patrick-kidger/signatory`.

[25] M. Lothaire, "Combinatorics on words," 1997.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[27] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[28] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

A Experimental details

A.1 General notes

Many details of the experiments are already specified in Section 4, and we do not repeat those details here.

Code Code to reproduce every experimental can found at `https://github.com/jambo6/generalised_shapelets`.

Computer infrastructure Experiments were run on the CPU of a variety of different machines, all using Ubuntu 18.04 LTS, and running PyTorch 1.3.1.

A.2 UEA

The datasets can be downloaded from `https://timeseriesclassification.com`.

All UEA datasets were used unnormalised. Those samples which were shorter than the maximum length of the sequence were padded to the maximum length by repeating their final entry.

Hyperparameters were found by performing a grid search over 2, 3, 5 shapelets *per class*, with a maximum total number of shapelets of 30, and shapelets being set to (classical shapelet transform) / initialised at (generalised shapelet transform) 0.15, 0.3, 0.5, 1.0 times the maximum length of the time series.

The dataset comes with default train/test splits, which we respect here. The training data is split 80%/20% into train and validation sets, stratified by class label.

The maximum number of epochs allowed for training was 250.

The details of each dataset are as below. We note that the train/test splits are sometimes of unusual proportion; we do not know the reason for this odd choice.

Table 3

Dataset	Train size	Test size	Dimensions	Length	Classes	Shapelets per class	Shapelet length fraction
BasicMotions	40	40	6	100	4	3	0.5
ERing	30	30	4	65	6	2	0.5
Epilepsy	137	138	3	206	4	5	0.5
Handwriting	150	850	3	152	26	3	0.5
JapaneseVowels	270	370	12	29	9	2	0.5
Libras	180	180	2	45	15	5	1.0
LSST	2459	2466	6	36	14	2	1.0
PenDigits	7494	3498	2	8	10	5	0.5
RacketSports	151	152	6	30	4	3	0.5

We also give the hyperparameters used for Section **??** below. These were chosen to optimize the validation score for the generalised L^2 discrepancy rather than on classical shapelets and as such are different to those noted above.

Table 4

Dataset	Shapelets per class	Best length fraction	Worst length fraction
JapaneseVowels	3	0.15	0.5
Libras	2	1.0	0.15
LSST	2	0.3	1.0

For completeness, we also give the full results from both hyperparamters in Tables **??**.

	2				3				5			
Dataset	0.15	0.3	0.5	1.0	0.15	0.3	0.5	1.0	0.15	0.3	0.5	1.0
BasicMotions	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**	100.0%	75.0%	100.0%	100.0%	100.0%	100.0%
ERing	83.3%	100.0%	**100.0%**	100.0%	100.0%	83.3%	100.0%	100.0%	100.0%	83.3%	83.3%	100.0%
Epilepsy	82.1%	75.0%	82.1%	82.1%	82.1%	85.7%	82.1%	71.4%	82.1%	82.1%	**89.3%**	-
Handwriting	13.3%	16.7%	23.3%	23.3%	16.7%	23.3%	**26.7%**	23.3%	16.7%	20.0%	16.7%	23.3%
JapaneseVowels	90.7%	90.7%	**96.3%**	92.6%	92.6%	92.6%	94.4%	92.6%	90.7%	92.6%	96.3%	90.7%
Libras	83.3%	80.6%	88.9%	83.3%	77.8%	88.9%	86.1%	77.8%	77.8%	86.1%	80.6%	**91.7%**
LSST	34.8%	33.7%	33.3%	**35.2%**	33.7%	33.9%	35.0%	34.1%	33.7%	33.7%	35.0%	34.6%
PenDigits	97.6%	98.0%	98.7%	96.7%	98.0%	97.9%	98.4%	96.5%	97.6%	98.5%	**98.9%**	96.3%
RacketSports	61.3%	74.2%	83.9%	80.6%	58.1%	67.7%	**87.1%**	77.4%	71.0%	77.4%	64.5%	80.6%

Table 5: Accuracy on the validation set for the hyperparmeter runs performed to determine the hyperparameters used in Section 4.1. The top column value represents the number of classes per shapelet for the run, with the lower values being the shapelet length proportion for that number of classes. The best run is given in bold. When multiple options achieved the highest score, the hyperparameters were chosen randomly from that top performing set.

	2				3				5			
Dataset	0.15	0.3	0.5	1.0	0.15	0.3	0.5	1.0	0.15	0.3	0.5	1.0
JapaneseVowels	94.4%	94.4%	94.4%	92.6%	**96.3%***	**94.4%**$_{*}$	94.4%	94.4%	96.3%	94.4%	92.6%	92.6%
Libras	**63.9%**$_{*}$	77.8%	77.8%	**88.9%***	75.0%	80.6%	83.3%	83.3%	73.2%	69.4%	83.3%	86.1%
LSST	45.7%	**46.5%***	42.5%	**37.0%**$_{*}$	41.9%	43.3%	42.6%	39.0%	45.5%	43.5%	41.4%	39.4%

Table 6: Accuracy on the validation set for the hyperparmeter runs performed to determine the hyperparameters used in Section 4.2. The top column value represents the number of classes per shapelet for the run, with the lower values being the shapelet length proportion for that number of classes. The best hyperparameters are denoted in bold with a * and the worst length hyperperameter for the same number of shapelets per class denoted as bold followed by a $_{*}$. When multiple options achieved the highest score, the hyperparameters were chosen randomly from that top performing set.

A.3 Speech Commands

The dataset can be downloaded from
`http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz`.

We began by selecting every sample from the 'yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop' and 'go' categories, and discarding the samples which were not of the maximum length (16000; not many samples were shorter than this). This gives a total of 34975 samples.

The samples were preprocessed by computing the MFC with a Hann window of length 400, hop length 200, and 400 frequency bins. Every sample is then of length 81 with 40 channels. Every channel was then normalised to have mean zero and variance one.

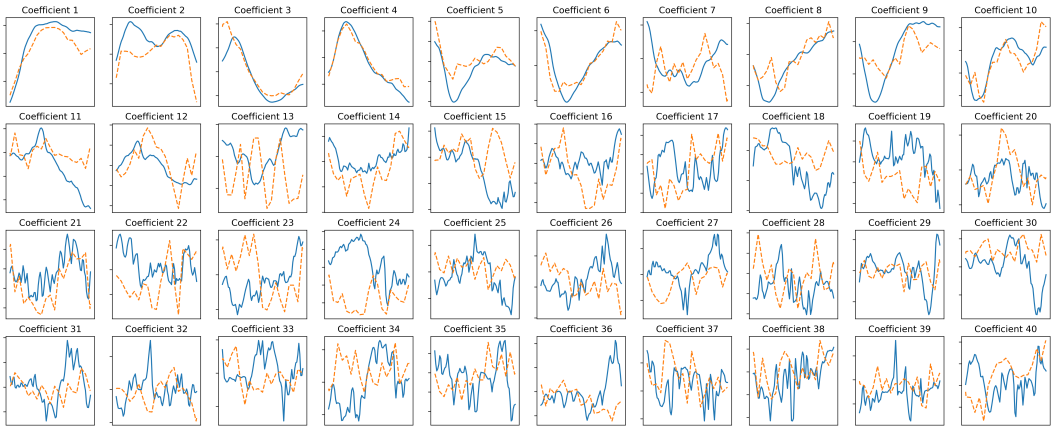
No hyperparameter searching was performed, due to the inordinately high cost of doing so - shapelets are an expensive algorithm that is primarily a 'small data' technique, and this represented the upper limit of problem size that we could consider! This is at least partly an implementation issue, and we believe that more efficient GPU implementations should be possible, but the need for different behaviour of different batch elements (which will in general have minimisers at different sections of the time series, making it necessary to keep track of the union of these points for every batch element) makes this difficult to express with typical machine learning frameworks [12, 13, 14]; this is an embarrasingly parallel problem that is best handled via naïve parallelism at the top level, rather than vectorising every operation. We note that not needing to perform hyperparameter optimisation on the length is an advantage of our generalised shapelet transform, thus reducing this kind of computaiton burden.

The maximum number of epochs allowed for training was 1000. The number of shapelets used per class was 4, for a total of 40 shapelets. The length of each shapelet (set to for the classical shapelet transform; initialised at for the generalised shapelet transform) was taken to be 0.3 of the full length of the dataset.

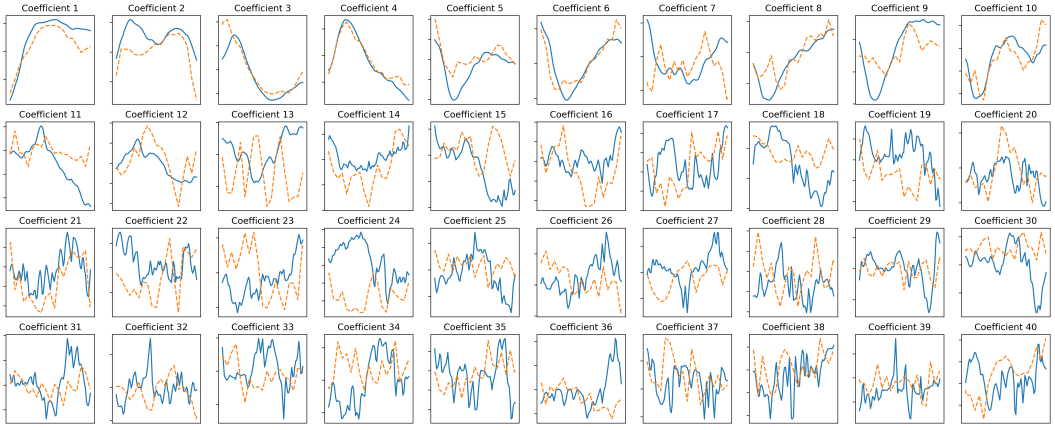
The data is combined into a single dataset and a 70%/15%/15% training/validation/test split taken, stratified by class label.

A.4 A further look at speech commands interpretability.

In Figure 3 we show all 40 MFC coefficients for the shapelet (blue) and the training set minimizer (orange, dashed) for the classical and generalised L^2 methods. This was omitted from the body of the text due to the size of the figures and also that the MFC plots are easier to digest.



(a) Classical shapelet transform.



(b) Generalised shapelet transform with L^2 discrepancy.

Figure 3: All MFC coefficients for the shapelet (blue) and the training set minimizer (orange, dashed).