# Unsupervised Feature Learning from Time Series

**Qin Zhang**[*], **Jia Wu**[*], **Hong Yang**[§], **Yingjie Tian**[†,‡], **Chengqi Zhang**[*]
[*] Quantum Computation & Intelligent Systems Centre, University of Technology Sydney, Australia
[†] Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing, China.
[‡] Key Lab of Big Data Mining & Knowledge Management, Chinese Academy of Sciences, Beijing, China.
[§] MathWorks, Beijing, China
{qin.zhang@student.,jia.wu@, chengqi.zhang@}uts.edu.au, hong.yang@mathworks.cn, tyj@ucas.ac.cn

## Abstract

In this paper we study the problem of learning discriminative features (segments), often referred to as *shapelets* [Ye and Keogh, 2009] of time series, from unlabeled time series data. Discovering *shapelets* for time series classification has been widely studied, where many search-based algorithms are proposed to efficiently scan and select segments from a pool of candidates. However, such types of search-based algorithms may incur high time cost when the segment candidate pool is large. Alternatively, a recent work [Grabocka *et al.*, 2014] uses regression learning to directly learn, instead of searching for, shapelets from time series. Motivated by the above observations, we propose a new Unsupervised Shapelet Learning Model (USLM) to efficiently learn shapelets from unlabeled time series data. The corresponding learning function integrates the strengths of *pseudo-class label*, *spectral analysis*, *shapelets regularization term* and *regularized least-squares* to auto-learn shapelets, pseudo-class labels and classification boundaries simultaneously. A coordinate descent algorithm is used to iteratively solve the learning function. Experiments show that USLM outperforms search-based algorithms on real-world time series data.

## 1 Introduction

Time series classification has wide applications in finance [Ruiz *et al.*, 2012], medicine [Hirano and Tsumoto, 2006] and trajectory analysis [Cai and Ng, 2004]. The main challenge of time series classification is to find discriminative features that can best predict class labels. To solve the challenge, a line of works have been proposed to extract discriminative features, which are often referred to as *shapelets*, from time series. Shapelets are maximally discriminative features of time series which enjoy the merit of high prediction accuracy and are easy to explain [Ye and Keogh, 2009]. Therefore, discovering shapelets has become an important branch in time series analysis.

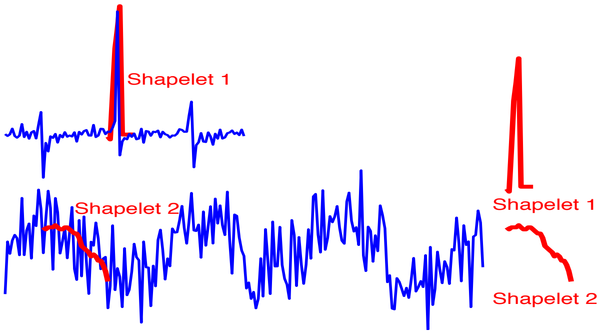


Figure 1: The two blue thin lines represent the time series data. The upper one is rectangular signals with noise and the lower one is sinusoidal signals with noise. The curves marked in bold red are learnt features(shapelets). We can observe that the learnt shapelets may differ from all candidate segments and thus are robust to noise.

The seminal work on shapelet discovery [Ye and Keogh, 2009] resorts to a full-scan of all possible time series segments where the segments are ranked according to a predefined distance metric and the segments which best predict the class labels are selected as shapelets. Based on the seminal work, a line of speed-up algorithms [Mueen *et al.*, 2011] [Rakthanmanon and Keogh, 2013] [Chang *et al.*, 2012] have been proposed to improve the performance. All these methods can be categorized as the *search-based* algorithms which scan a pool of candidate segments. For example, with the Synthetic Control dataset [Chen *et al.*, 2015] which contains 600 time series examples of length 60, the number of candidates for all lengths is $1.098 \times 10^6$.

On the other hand, a recent work [Grabocka *et al.*, 2014] proposes a new time series shapelet learning approach. Instead of searching for shapelets from a candidate pool, they use regression learning and aim to *learn shapelets from time series*. This way, shapelets are detached from candidate segments and the learnt shapelets may differ from all the candidate segments. More importantly, shapelet learning is fast to compute, scalable to large datasets, and robust to noise as shown in Fig. 1.

We present a new Unsupervised Shapelet Learning Model (USLM for short) that can auto-learn shapelets from unlabeled time series data. We first introduce *pseudo-class label* to transform unsupervised learning to supervised learning. Then, we use the popular *regularized least-squares* and *spectral analysis* approaches to learn both shapelets and classification boundaries. A new regularization term is also added to avoid similar shapelets being selected. A coordinate descent algorithm is used to iteratively solve the pseudo-class label, classification boundary and shapelets.

Compared to the search-based algorithms on unlabeled time series data [Zakaria *et al.*, 2012], our method provides a new tool that learns shapelets from unlabeled time series data. The **contributions** of the work are summarized as follows:

- We present a new Unsupervised Shapelet Learning Model (USLM) to learn shapelets from unlabeled time series data. USLM combines pseudo-class label, spectral analysis, shapelets regularization and regularized least-squares for learning.
- We empirically validate the performance of USLM on both synthetic and real-world data. The results show promising results compared to the state-of-the-art unsupervised shapelets selection models.

The remainder of the paper is organized as follows: Section 2 surveys related work. Section 3 gives the preliminaries. Section 4 introduces the proposed unsupervised shapelet learning model USLM. Section 5 introduces the learning algorithm with analysis. Section 6 conducts experiments. We conclude the paper in Section 7.

## 2 Related Work

**Shapelets** [Ye and Keogh, 2009] are time series short segments that can best predict class labels. The basic idea of shapelets discovery is to consider all segments of training data and assess them regarding a scoring function to estimate how predictive they are with respect to the given class labels [Wistuba *et al.*, 2015]. The seminal work [Ye and Keogh, 2009] builds a decision tree classifier by recursively searching for informative shapelets measured by information gain. Based on information gain, several new measures such as F-Stat, Kruskall-Wallis and Mood's median are used in shapelets selection [Hills *et al.*, 2014] [Lines *et al.*, 2012].

Since time series data usually have a large number of candidate segments, the runtime of brute-force shapelets selection is infeasible. Therefore, a series of speed-up techniques have been proposed. On the one hand, there are smart implementations using early abandon of distance computations and entropy pruning of the information gain heuristic [Ye and Keogh, 2009]. On the other hand, many speed-ups rely on the reuse of computations and pruning of the search space [Mueen *et al.*, 2011], as well as pruning candidates by searching possibly interesting candidates on the SAX representation [Rakthanmanon and Keogh, 2013] or using infrequent shapelets [He *et al.*, 2012]. Shapelets have been applied in a series of real-world applications.

**Shapelet learning.** Instead of searching for shapelets exhaustively, a recent work [Grabocka *et al.*, 2014] proposes to learn optimal shapelets and reports statistically significant improvements in accuracy compared to other shapelet-based classifiers. Instead of restricting the pool of possible candidates to those found in the training data and simply searching them, they consider shapelets to be features that can be learnt through regression learning. This type of learning method does not consider a limited set of candidates but can obtain arbitrary shapelets.

**Unsupervised feature selection.** Many unsupervised feature selection algorithms have been proposed to select informative features from unlabeled data. A commonly used criterion in unsupervised feature learning is to select features best preserving data similarity or manifold structure constructed from the whole feature space[Zhao and Liu, 2007] [Cai *et al.*, 2010], but they fail to incorporate discriminative information implied within data, which cannot be directly applied in our shapelet learning problem. Earlier unsupervised feature selection algorithms evaluate the importance of each feature individually and select features one by one [He *et al.*, 2005] [Zhao and Liu, 2007], with a limitation that correlation among features is neglected [Cai *et al.*, 2010]c̆itezhao2010efficient [Zhang *et al.*, 2015].

State-of-the-art unsupervised feature selection algorithms perform feature selection by simultaneously exploiting discriminative information and feature correlation. Unsupervised Discriminative Feature Selection (UDFS) [Yang *et al.*, 2011] aims to select the most discriminative features for data representation, where manifold structure is also considered. Since the most discriminative information for feature selection is usually encoded in labels, it is very important to predict a good cluster indicators as pseudo labels for unsupervised feature selection.

**Shapelets for clustering.** Shapelets also have been utilized to cluster time series [Zakaria *et al.*, 2012]. Zakaria et al. [Zakaria *et al.*, 2012] have proposed a method to use unsupervised-shapelets (u-Shapelets) for time series clustering. The algorithm searches for u-Shapelets which can separate and remove a subset of time series from the rest of the dataset, then it iteratively repeats the search among the remaining data until no data remains to be separated. It is a greedy search algorithm which attempts to maximize the gap between the two groups of time series divided by a u-shapelet.

The $k$-shape algorithm is proposed in the work [Paparrizos and Gravano, 2015] to cluster time series. $k$-shape is a novel algorithm for shape-based time series clustering that is efficient and domain independent. $k$-shape is based on a scalable iterative refinement procedure which creates homogeneous and well-separated clusters. Specifically, $k$-Shape requires a distance measure that is invariant to scaling and shifting. It uses a normalized version of the cross-correlation measure as distance measure to consider the shapes of time series. Based on the normalized cross-correlation, the method computes cluster centroids in every iteration to update the assignment of time series to clusters.

Our work differs from the above research problems. We introduce a new approach for *unsupervised shapelet learning* to auto-learn shapelets from unlabeled time series by combining shapelet learning and unsupervised feature selection methods.

## 3 Preliminaries

In this paper, scalars are denoted by letters $(a, b, ...; \alpha, \beta, ...)$, vectors by lower-case bold letters $(\mathbf{a}, \mathbf{b}, ...)$, and matrices by boldfaced upper-case letters $(\mathbf{A}, \mathbf{B}, ...)$. We use $\mathbf{a}_{(k)}$ to denote the $k$-th element of vector $\mathbf{a}$, and $\mathbf{A}_{(ij)}$ to denote the element locating at the $i$th-row and $j$-th column. $\mathbf{A}_{(i,:)}$ and $\mathbf{A}_{(:,j)}$ denote vectors of the $i$-th row and $j$-th column of the matrix respectively. In a time series example, $\mathbf{t}_{a,b}$ denotes a segment starting from $a$ to $b$.

Consider a set of time series examples $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$. Each example $\mathbf{t}_i$ $(1 \leq i \leq n)$ contains an ordered set of real values denoted as $(\mathbf{t}_{i(1)}, \mathbf{t}_{i(2)}, \ldots, \mathbf{t}_{i(q_i)})$, where $q_i$ is the length of $\mathbf{t}_i$. We wish to learn a set of top-$k$ most discriminative shapelets $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$. Similar to the shapelet learning model [Grabocka *et al.*, 2014], we set the length of shapelets to expand $r$ different length scales starting at a minimum $l_{min}$, i.e. $\{l_{min}, 2 \times l_{min}, \ldots, r \times l_{min}\}$. Each length scale $i \times l_{min}$ contains $k_i$ shapelets and $k = \sum_{i=1}^{r} k_i$. Obviously, $\mathbf{S} \in \bigcup_{i=1}^{r} \mathbb{R}^{k_i \times (i \times l_{min})}$ and $r \times l_{min} \ll q_i$ to keep the shapelets compact.

## 4 Unsupervised shapelet learning

In this section, we aim to formulate the Unsupervised Shapelet Learning Model (USLM) which is shown in Eq. (7). It combines with spectral regularization term, shapelet similarity regularization term and the regularized least square minimization term.

To introduce the USLM specifically, we introduce shapelet-transformed representation [Grabocka *et al.*, 2014] of time series first, which transfer time series from original space to a shapelet-based space. Then we introduce the pseudo-class label and the three terms of the unsupervised shapelet learning model respectively.

**Shapelet-transformed Representation** *Shapelet transformation* was introduced by the work[Lines *et al.*, 2012] to downsize a long time series into a short feature vector in the shapelets feature space. Time series are ordered sequences and shapelet-transformation can preserve the shape information for classification.

Given a set of time series examples $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$ and a set of shapelets $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$, we use $\mathbf{X} \in \mathbb{R}^{k \times n}$ to denote the shapelet-transformed matrix, where each element $\mathbf{X}_{(\mathbf{s}_i, \mathbf{t}_j)}$ denotes the distance between shapelet $\mathbf{s}_i$ and time series $\mathbf{t}_j$. For simplicity, we use $\mathbf{X}_{(ij)}$ to represent $\mathbf{X}_{(\mathbf{s}_i, \mathbf{t}_j)}$ which can be calculated as in Eq. (1),

$$\mathbf{X}_{(ij)} = \min_{g=1,\ldots,\overline{q}} \frac{1}{l_i} \sum_{h=1}^{l_i} (\mathbf{t}_{j(g+h-1)} - \mathbf{s}_{i(h)}) \tag{1}$$

where $\overline{q} = q_j - l_i + 1$ denotes the total number of segments with length $l_i$ from series $\mathbf{t}_j$, and $q_j, l_i$ are the lengths of time series $\mathbf{t}_j$ and shapelet $\mathbf{s}_i$ respectively.

Given a set of time series data $\mathbf{S}$, $\mathbf{X}_{(ij)}$ is a function with respect to all candidate shapelets $\mathbf{S}$, i.e. $\mathbf{X}(\mathbf{S})_{(ij)}$. For simplicity, we omit the variable $\mathbf{S}$ and still use $\mathbf{X}_{(ij)}$ instead.

The distance function in Eq. (1) is not continuous and thus non-differential. Based on the work [Grabocka *et al.*, 2014], we approximate the distance function using the *soft minimum function* as in Eq. (2),

$$\mathbf{X}_{(ij)} \approx \frac{\sum_{q=1}^{\overline{q}} d_{ijq} \cdot e^{\alpha d_{ijq}}}{\sum_{q=1}^{\overline{q}} e^{\alpha d_{ijq}}} \tag{2}$$

where $d_{ijq} = \frac{1}{l_i} \sum_{h=1}^{l_i} (\mathbf{t}_{j(q+h-1)} - \mathbf{s}_{i(h)})$, and $\alpha$ controls the precision of the function. The soft minimum approaches the true minimum when $\alpha \to -\infty$. In our experiments, we set $\alpha = -100$.

**Pseudo-class label** Unsupervised learning faces the challenge of unlabeled training examples. Thus, we introduce the *pseudo-class labels* for learning. Consider that we cluster a time series data set into $c$ categories, the pseudo-class label matrix $\mathbf{Y} \in \mathbb{R}^{c \times n}$ contains $c$ labels, where $\mathbf{Y}_{(ij)}$ indicates the probability of the $j$-th time series example belonging to the $i$-th category. Time series examples that share the same pseudo-class label fall into the same category. If $\mathbf{Y}_{(\bar{i}j)} > \mathbf{Y}_{(i,j)}, \forall i$, then the time series example $\mathbf{t}_j$ belong to the cluster $\bar{i}$.

**Spectral Analysis** Spectral analysis was introduced by [Donath and Hoffman, 1973] and has been widely used in unsupervised learning [Von Luxburg, 2007]. The principle behind is that examples that are close to each other are likely to share the same class label [Tang and Liu, 2014] [Von Luxburg, 2007]. Assume that $\mathbf{G} \in \mathbb{R}^{n \times n}$ is the similarity matrix of time series based on the shapelet-transformed matrix $\mathbf{X}$, then the similarity matrix can be calculated as in Eq. (3), where $\sigma$ is the parameter of the RBF kernel.

$$\mathbf{G}_{(ij)} = e^{-\frac{\|\mathbf{X}_{(:,i)} - \mathbf{X}_{(:,j)}\|^2}{\sigma^2}} \tag{3}$$

Based on $\mathbf{G}$, we expect the pseudo-class labels of similar data instances to be the same. Therefore, we can formulate a spectral regularization term as follows,

$$\begin{aligned} & \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{b} \mathbf{G}_{(ij)} \|\mathbf{Y}_{(:,i)} - \mathbf{Y}_{(:,j)}\|_2^2 \\ &= \frac{1}{2} \sum_{k=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{G}_{(ij)} (\mathbf{Y}_{(ki)} - \mathbf{Y}_{(kj)})^2 \\ &= \sum_{k=1}^{c} \mathbf{Y}_{(k,:)} (\mathbf{D}_G - \mathbf{G}) \mathbf{Y}_{(k,:)} \\ &= tr(\mathbf{Y} \mathbf{L}_G \mathbf{Y}) \end{aligned} \tag{4}$$

where $\mathbf{L}_G = \mathbf{D}_G - \mathbf{G}$ is the Laplacian matrix and $\mathbf{D}_G$ is a diagonal matrix with its elements defined as $\mathbf{D}_G(i,i) = \sum_{j=1}^{n} \mathbf{G}_{(ij)}$.

**Least Square Minimization** Based on the pseudo-class labels, we wish to minimize the least square error. Let $\mathbf{W} \in \mathbb{R}^{k \times c}$ be the classification boundary under the pseudo-class labels, the least square error minimizes the following objective function,

$$\min_{\mathbf{W}} \|\mathbf{W}^T \mathbf{X} - \mathbf{Y}\|_F^2 \tag{5}$$

**Shapelet Similarity Minimization** We wish to learn shapelets that are diverse in shape. Specifically, we penalize the model in case it outputs similar shapelets. Formally, we denote the shapelet similarity matrix as $\mathbf{H} \in \mathbb{R}^{k \times k}$, where each element $\mathbf{H}_{(\mathbf{s}_i,\mathbf{s}_j)}$ represents the similarity between two shapelets $\mathbf{s}_i$ and $\mathbf{s}_j$. For simplicity, we use $\mathbf{H}_{(ij)}$ to represent $\mathbf{H}_{(\mathbf{s}_i,\mathbf{s}_j)}$ which can be calculated as in Eq. (6),

$$\mathbf{H}_{(ij)} = e^{-\frac{\|d_{ij}\|^2}{\sigma^2}} \tag{6}$$

where $d_{ij}$ is the distance between shapelet $\mathbf{s}_i$ and shapelet $\mathbf{s}_j$. $d_{ij}$ can be calculated by following Eq. (2).

**Unsupervised Shapelet Learning Model** Eq. (7) gives the unsupervised shapelet learning model (USLM). It is a joint optimization problem with respect to three variables, the classification boundary $\mathbf{W}$, Pseudo-class label $\mathbf{Y}$ and candidate shapelets $\mathbf{S}$.

$$\begin{aligned}\min_{\mathbf{W},\mathbf{S},\mathbf{Y}} \; & \frac{1}{2}tr(\mathbf{Y}\mathbf{L}_G(\mathbf{S})\mathbf{Y}^\top) + \frac{\lambda_1}{2}\|\mathbf{H}(\mathbf{S})\|_F^2 \\ & + \frac{\lambda_2}{2}\|\mathbf{W}^T\mathbf{X}(\mathbf{S}) - \mathbf{Y}\|_F^2 + \frac{\lambda_3}{2}\|\mathbf{W}\|_F^2\end{aligned} \tag{7}$$

In the objective function, the first term is the spectral regularization that preserves local structure information. The second term is the shapelet similarity regularization term that prefers diverse shapelets. The third and fourth terms are the regularized least square minimization.

Note that matrix $\mathbf{L_G}$ in Eq. (4), matrix $\mathbf{H}$ in Eq. (6), and matrix $\mathbf{X}$ in Eq. (2) depend on the shapelets $\mathbf{S}$. We explicitly write these matrices as variables with respect to shapelets $\mathbf{S}$ in Eq. (7), i.e. $\mathbf{L_G}(\mathbf{S})$ , $\mathbf{H(S)}$ and $\mathbf{X(S)}$ respectively. Below, we propose a coordinate descent algorithm to solve the model.

## 5 The Algorithm

In this part, we first introduce a coordinate descent algorithm to solve the USLM, and then analyze its convergence and initialization methods.

### 5.1 Learning Algorithm

In the coordinate descent algorithm, we iteratively update one variable by fixing the remaining two variables. The steps will be repeated until convergence. Algorithm 1 summarizes the steps.

**Algorithm 1.** Unsupervised Shapelet Learning Algorithm (USLA)

```
Input:
    • Time series $\mathbf{T}$ with $c$ classes
    • Length and number of shapelets: $l_{min}, r, k$
    • Number of internal iterations $i_{max}$
    • Learning rate $\eta$ and Parameters $\lambda_1, \lambda_2, \lambda_3$ and $\alpha, \sigma$
Output: Shapelets $\mathbf{S}^*$ and class labels $\mathbf{Y}^*$
Initialize: $\mathbf{S}_0, \mathbf{W}_0, \mathbf{Y}_0$
While Not convergent do
    Calculate: $\mathbf{X}_t(\mathbf{T}, \mathbf{S}_t|\alpha)$, $\mathbf{L}_{Gt}(\mathbf{T}, \mathbf{S}_t|\alpha, \sigma)$
        and $\mathbf{H}_t(\mathbf{S}_t|\alpha)$ based on Eqs. (2), (4), and (6);
    update $\mathbf{W}_{t+1}, \mathbf{Y}_{t+1}$:
        $\mathbf{Y}_{t+1} \leftarrow \lambda_2\mathbf{W}_t^T\mathbf{X}_t(\mathbf{L}_{Gt} + \lambda_2\mathbf{I})^{-1}$
        $\mathbf{W}_{t+1} \leftarrow (\lambda_2\mathbf{X}_t\mathbf{X}_t^T + \lambda_3\mathbf{I})^{-1}(\lambda_2\mathbf{X}_t\mathbf{Y}_{t+1}^T)$.
    update $\mathbf{S}_{t+1}$:
        for $i = 1, \dots, i_{max}$ do
            $\mathbf{S}_{i+1} \leftarrow \mathbf{S}_i - \eta\nabla\mathbf{S}_i$
            $\nabla\mathbf{S}_i = \frac{\partial\mathbf{F}(\mathbf{S}_i|\mathbf{X}_{t+1},\mathbf{Y}_{t+1})}{\partial\mathbf{S}}$ is from Eq. (17)
        end for
        $\mathbf{S}_{t+1} = \mathbf{S}_{i_{max}+1}$
    $t \leftarrow t + 1$
end while
Output: $\mathbf{S}^* = \mathbf{S}_t$; $\mathbf{Y}^* = \mathbf{Y}_t$; $\mathbf{W}^* = \mathbf{W}_t$.
```

**• Update Y by fixing W and S** By fixing $\mathbf{W}$ and $\mathbf{S}$, the function in Eq. (7) degenerates to Eq. (8),

$$\min_{\mathbf{Y}} \mathbf{F}(\mathbf{Y}) = \frac{1}{2}tr(\mathbf{Y}\mathbf{L}_G\mathbf{Y}^T) + \frac{\lambda_2}{2}\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2 \tag{8}$$

The derivative of Eq. (8) with respect to $\mathbf{Y}$ is,

$$\frac{\partial\mathbf{F_Y}}{\partial\mathbf{Y}} = \mathbf{Y}(\mathbf{L}_G - \lambda_2\mathbf{I}) - \lambda_2\mathbf{W}^T\mathbf{X} \tag{9}$$

Let Eq. (9) equal to 0, we can obtain the solution of $\mathbf{Y}$ as in Eq. (10),

$$\mathbf{Y} = \lambda_2\mathbf{W}^T\mathbf{X}(\mathbf{L}_G + \lambda_2\mathbf{I})^{-1} \tag{10}$$

where $\mathbf{I}$ is an identity matrix. Thus, the update of $\mathbf{Y}$ is

$$\mathbf{Y}_{t+1} = \lambda_2\mathbf{W}_t^\top\mathbf{X}_t(\mathbf{L}_{Gt} + \lambda_2\mathbf{I})^{-1} \tag{11}$$

**• Update W by fixing S and Y** By fixing $\mathbf{S}$ and $\mathbf{Y}$, Eq. (7) degenerates to Eq. (12) as below,

$$\min_{\mathbf{W}} \mathbf{F}(\mathbf{W}) = \frac{\lambda_2}{2}\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2 + \frac{\lambda_3}{2}\|\mathbf{W}\|_F^2 \tag{12}$$

The derivative of Eq. (12) with respect to $\mathbf{W}$ is,

$$\frac{\partial\mathbf{F_W}}{\partial\mathbf{W}} = (\lambda_2\mathbf{X}\mathbf{X}^T + \lambda_3\mathbf{I})\mathbf{W} - \lambda_2\mathbf{X}\mathbf{Y}^T \tag{13}$$

Let Eq. (13) equal to 0, we obtain the solution of $\mathbf{W}$ as in Eq. (14),

$$\mathbf{W} = (\lambda_2\mathbf{X}\mathbf{X}^T + \lambda_3\mathbf{I})^{-1}(\lambda_2\mathbf{X}\mathbf{Y}^T) \tag{14}$$

Thus, the update of $\mathbf{W}$ is

$$\mathbf{W}_{t+1} = (\lambda_2\mathbf{X}_t\mathbf{X}_t^\top + \lambda_3\mathbf{I})^{-1}(\lambda_2\mathbf{X}_t\mathbf{Y}_{t+1}^T) \tag{15}$$

**• Update S by fixing W and Y** By fixing $\mathbf{W}$ and $\mathbf{Y}$, Eq. (7) degenerates to Eq. (16) as below,

$$\begin{aligned}\min_{\mathbf{S}} \mathbf{F}(\mathbf{S}) = & \frac{1}{2}tr(\mathbf{Y}\mathbf{L}_G(\mathbf{S})\mathbf{Y}^T) + \frac{\lambda_1}{2}\|\mathbf{H}(\mathbf{S})\|_F^2 \\ & + \frac{\lambda_2}{2}\|\mathbf{W}^T\mathbf{X}(\mathbf{S}) - \mathbf{Y}\|_F^2\end{aligned} \tag{16}$$

Eq. (16) is non-convex and we cannot explicitly solve $\mathbf{S}$ as in finding $\mathbf{W}$ and $\mathbf{Y}$. Instead, we resort to an iterative algorithm by setting a learning rate $\eta$, i.e. $\mathbf{S}_{i+1} = \mathbf{S}_i - \eta\nabla\mathbf{S}_i$, where $\nabla\mathbf{S}_i = \frac{\partial\mathbf{F}(\mathbf{S}_i)}{\partial\mathbf{S}}$. The iterative steps will guarantee the convergence of the objective function. The derivative of Eq. (16) with respect to $\mathbf{S}_{(mp)}$ is

$$\begin{aligned}\frac{\partial\mathbf{F}(\mathbf{S})}{\partial\mathbf{S}_{(mp)}} = & \frac{1}{2}\mathbf{Y}^T\mathbf{Y}\frac{\partial\mathbf{L}_G(\mathbf{S})}{\partial\mathbf{S}_{(mp)}} + \lambda_1\mathbf{H}(\mathbf{S})\frac{\partial\mathbf{H}(\mathbf{S})}{\partial\mathbf{S}_{(mp)}} \\ & + \lambda_2\mathbf{W}(\mathbf{W}^TX - Y)\frac{\partial\mathbf{X}(\mathbf{S})}{\partial\mathbf{S}_{(mp)}}\end{aligned} \tag{17}$$

where $m = 1, \dots, k$, and $p = 1, \dots, l_m$.

Because $\mathbf{L}_G = \mathbf{D}_G - \mathbf{G}$ and $\mathbf{D}_G(i,i) = \sum_{j=1}^{n} \mathbf{G}_{(ij)}$, the first term in Eq. (17) turns to calculating $\partial \mathbf{G}_{(ij)} / \partial \mathbf{S}_{(mp)}$ as shown in Eq. (18),

$$\frac{\partial \mathbf{G}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = -\frac{2\mathbf{G}_{(ij)}}{\sigma^2} \cdot (\sum_{q=1}^{k} (\mathbf{X}_{(qi)} - \mathbf{X}_{(qj)})) \cdot (\frac{\partial \mathbf{X}_{(qi)}}{\partial \mathbf{S}_{(mp)}} - \frac{\partial \mathbf{X}_{(qj)}}{\partial \mathbf{S}_{(mp)}}) \tag{18}$$

and

$$\frac{\partial \mathbf{X}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = \frac{1}{E_1^2} \sum_{q=1}^{\overline{q}_{ij}} e^{\alpha d_{ijq}} ((1 + \alpha d_{ijq}) E_1 - \alpha E_2) \frac{\partial d_{ijq}}{\partial \mathbf{S}_{(mp)}} \tag{19}$$

where $E_1 = \sum_{q=1}^{\overline{q}_{ij}} e^{\alpha d_{ijq}}$, $E_2 = \sum_{q=1}^{\overline{q}_{ij}} d_{ijq} e^{\alpha d_{ijq}}$ and $\overline{q}_{ij} = q_j - l_i + 1$ and $d_{ijq} = \frac{1}{l_i} \sum_{h=1}^{l_i} (\mathbf{t}_{j(q+h-1)} - \mathbf{s}_{i(h)})$.

$$\frac{\partial d_{ijq}}{\partial \mathbf{S}_{(mp)}} = \left\{ \begin{array}{ll} 0 & if \;\; i \neq m \\ \frac{2}{l_m} (\mathbf{S}_{(mp)} - T_{j,q+p-1}) & if \;\; i = m \end{array} \right\} \tag{20}$$

The second term in Eq. (17) turns to calculating Eq. (21),

$$\frac{\partial \mathbf{H}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = -\frac{2}{\sigma^2} \tilde{d}_{ij} e^{-\frac{1}{\sigma^2} \tilde{d}_{ij}^2} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{S}_{(mp)}} \tag{21}$$

where $\tilde{d}_{ij}$ is the distance between shapelets $\mathbf{s}_i$ and $\mathbf{s}_j$. The calculation of $\tilde{d}_{ij}$ and $\frac{\partial \tilde{d}_{ij}}{\partial \mathbf{S}_{(mp)}}$ is similar to $\mathbf{X}_{(ij)}$ and $\frac{\partial \mathbf{X}_{(ij)}}{\partial \mathbf{S}_{(mp)}}$ respectively.

To sum up, we can calculate the gradient $\nabla \mathbf{S}_i = \frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}_i}$ by following Eqs. (17)-(21). In the following, we discuss the convergence of the coordinate descent algorithm in solving Eq. (7).

### 5.2 Convergence

The convergence of Algorithm 1 depends on stepwise descents. When updating $\mathbf{Y}$ or $\mathbf{W}$, we know that $\mathbf{Y}_{t+1} = \mathbf{Y}^*(\mathbf{W}_t, \mathbf{S}_t)$ and $\mathbf{W}_{t+1} = \mathbf{W}^*(\mathbf{Y}_{t+1}, \mathbf{S}_t)$. When updating $\mathbf{S}$, the objective function in Eq. (16) is not convex and a closed-form derivative is difficult to obtain. Instead, we use a gradient descent algorithm. In the iterations, as long as we set an appropriate learning rate $\eta$ that is usually very small, the objective function will decrease to convergence.

In addition, the objective function in Eq. (7) is non-convex but have a lower bound of 0 due to being non-negative, so Algorithm 1 converges to local optima. In our experiments we run the algorithm several times under different initializations and choose the best solution as output.

### 5.3 Initialization

Because the algorithm only outputs local optima, we discuss how to initialize the variables to improve the performance of the algorithm. The algorithm expects to initialize $\mathbf{S}_0$, $\mathbf{Y}_0$ and $\mathbf{W}_0$. We first initialize $\mathbf{S}_0$ by using the centroids of the segments having the same length with the shapelets length, because centroids represent typical patterns behind the data. Then, we use the results of the shapelet-transformed matrix of time series based on $\mathbf{S}_0$ to obtain $\mathbf{X}_0$. Next, the results obtained by k-means based on $\mathbf{X}_0$ is used to initialize $\mathbf{W}_0$ and $\mathbf{Y}_0$. The initialization enables fast convergence.

## 6 Experiments

We conduct experiments to validate the performance of USLM. All experiments are conducted on a Windows 8 machine with 3.00GHz CPU and 8GB memory. The Matlab source codes and data are available online[1].

### 6.1 Datasets

Synthetic data: This dataset is generated by following the work [Shariat and Pavlovic, 2011] and [Zakaria *et al.*, 2012]. The dataset consists of ten examples from two classes. The first class contains sinusoidal signals of length 200. The second class contains rectangular signals of length 100. We randomly embed Gaussian noise in the data. Heterogeneous noise is generated from five Gaussian processes with means $\nu \in [0, 2]$ and variances $\sigma^2 \in [0, 10]$ chosen uniformly at random.

Real-world data: We use seven time series benchmark datasets download from the UCR time series archive [Chen *et al.*, 2015] [Cetin *et al.*, 2015]. The datasets are summarized in Table 1. More details of the datasets can resort to their Website.

Table 1: Statistics of the benchmark time series datasets

| Dataset | Train/Test | Length | ♯ classes |
|---|---|---|---|
| CBF | 30/900(930) | 128 | 3 |
| ECG 200 | 100/100(200) | 96 | 2 |
| Face Four | 24/88(112) | 350 | 4 |
| Ita.Pow.Dem. | 67/1029(1096) | 24 | 2 |
| Lighting2 | 60/61(121) | 637 | 2 |
| Lighting7 | 70/73(143) | 319 | 7 |
| OSU Leaf | 200/242(442) | 427 | 6 |

### 6.2 Measures

Existing measures used to evaluate the performance of time series clustering include Jaccard Score, Rand Index, Folkes and Mallow index [Halkidi *et al.*, 2001] [Zakaria *et al.*, 2012]. Among them, Rand index [Rand, 1971] is the most popular one, while the remaining measures can be taken as variants of Rand index. Therefore, we use *Rand Index* as the evaluation method.

To calculate Rand index, we compare the cluster labels $\mathbf{Y}^*$ obtained by the clustering algorithm with the genuine class labels $\mathbf{L}_{true}$ as in Eq. (22),

$$Rand\ index = \frac{TP + TN}{TP + TN + FP + FN}, \tag{22}$$

where $TP$ is the number of time series pairs which belong to the same class in $\mathbf{L}_{true}$ and are assigned to the same cluster in $\mathbf{Y}^*$, $TN$ is the number of time series pairs which belong to different classes in $\mathbf{L}_{true}$ and are assigned to different clusters in $\mathbf{Y}^*$, $FP$ is the number of time series pairs which belong to different classes in $\mathbf{L}_{true}$ but are assigned to the same cluster in $\mathbf{Y}^*$, and $FN$ is the number of time series pairs which

[1] https://github.com/BlindReview/shapelet

belong to the same class in $\mathbf{L}_{true}$ but are assigned to different clusters in $\mathbf{Y}^*$. If $Rand\ index$ is close to 1, it indicates a high quality clustering [Zakaria *et al.*, 2012] [Paparrizos and Gravano, 2015].

### 6.3 Time series with unequal lengths

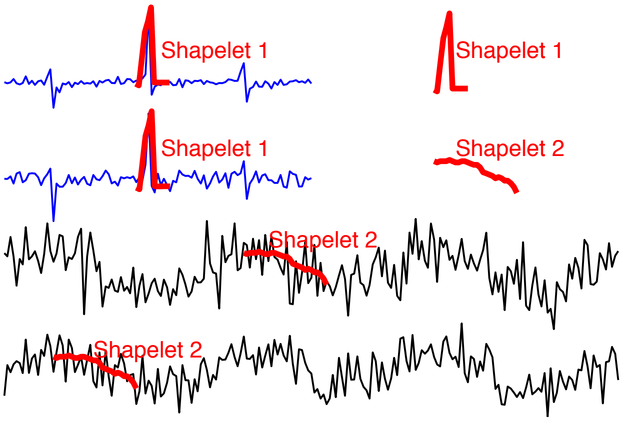


Figure 2: An example of the rectangular signals(short blue thin curves) and the sinusoidal signals (long black thin curves). The two learned shapelets are marked with bold red curves.

Fig. 2 shows an example of two shapelets learnt by USLM. *Shapelet 1* is a sharp spike of length 12 which matches the most prominent spikes of the rectangular signals. *Shapelet 2* is a subsequence of length 30 which is very similar to the standard shape of the sinusoidal signals. From the results, we can observe that USLM can auto-learn representative shapelets from unlabeled time series data.

The results in Fig. 3 also show that USLM can handle time series and shapelets of unequal lengths, where we can obtain the best Rand index value of 1. In contrast, the work [Shariat and Pavlovic, 2011] obtains only 0.9 on the dataset even if they use class label information during training.

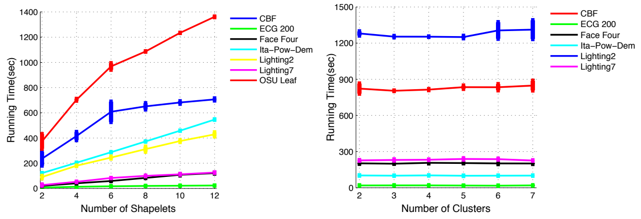
### 6.4 Running time

We test the running time of USLM by changing the number of shapelets $k$ and the number of clusters $c$. The results are given in Fig. 3.

First, we vary the parameter $k$ from 2 to 12 with a step size of 2. The remaining parameters are fixed as follows, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1, \sigma = 1, I_{max} = 50, \eta = 0.01$ and the length of the shapelets is set to 10. The running time is the average of ten executions.

From Fig. 3(a), we can observe that the running time generally increases linearly with respect to the number of shapelets. Thus, our approach scales well to large datasets.

Then, we let the number of clusters $c$ change from 2 to 7. The length of shapelets is set to be 10% of the time series length. We set $k = 2$ which means that we only learn two shapelets of equal length. The remaining parameters are the same as above. Fig. 3(b)



(a) # of shapelets (b) # of clusters

Figure 3: USLM running time *w.r.t.* (a) the number of shapelets $k$ and (b) the number of clusters $c$. The running time increases linearly *w.r.t.* the number of features and remains stable *w.r.t.* the number of clusters $c$. Thus, USLM scales well to large datasets.

### 6.5 Comparisons with $k$-Shape

The $k$-Shape algorithm is proposed in the work [Paparrizos and Gravano, 2015] to cluster unlabeled time series. In this part, we compare our algorithm with the $k$-Shape algorithm. The source codes for $k$-Shape are downloaded from the website of the original authors. Table 2 lists the results. We select the best results obtained by $k$-Shape after 100 times of repeats. We can observe that the results obtained by USLM are better.

Table 2: Comparison between $k$-Shape and USLM.

| Rand Index | $k$-Shape | USLM |
|---|---|---|
| CBF | 0.74 | **1.00** |
| ECG200 | 0.70 | **0.76** |
| Fac.F. | 0.64 | **0.79** |
| Ita.Pow | 0.70 | **0.82** |
| Lig.2 | 0.65 | **0.80** |
| Lig.7 | 0.74 | **0.79** |
| OSU L. | 0.66 | **0.82** |
| Average | 0.69 | **0.83** |

Form Table 2, we can observe that USLM outperforms the $k$-Shape algorithm on all the seven datasets. The average improvement on the seven datasets is 14% and the best improvement is 26% on the 'CBF' dataset. To sum up, the results show that USLM gains higher accuracy than $k$-Shape.

## 7 Conclusions

In this paper, we investigated a new problem of feature learning from unlabeled time series data. To solve the problem, we proposed a new learning model USLM by combining the pseudo-class label, spectral analysis, shapelets regularization and regularized least-squares minimization. USLM can auto-learn the most discriminative features from unlabeled times series data. Experiments on real-world time series data have shown that USLM can obtain an average improvement of 14% compared to $k$-Shape.

## Acknowledgments

This work was supported by the Australian Research Council (ARC) Discovery Projects under Grant No. DP140100545 and DP140102206 and also been partially supported by grants from National Natural Science Foundation of China (Nos. 61472390 and 11271361). Y. Tian is the corresponding author.

## References

[Cai and Ng, 2004] Yuhan Cai and Raymond Ng. Indexing spatio-temporal trajectories with chebyshev polynomials. In *SIGMOD*, pages 599–610, 2004.

[Cai *et al.*, 2010] Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.

[Cetin *et al.*, 2015] Mustafa S Cetin, Abdullah Mueen, and Vince D Calhoun. Shapelet ensemble for multi-dimensional time series. *SDM*, pages 307–315, 2015.

[Chang *et al.*, 2012] Kai-Wei Chang, Bikash Deka, Wen-Mei W Hwu, and Dan Roth. Efficient pattern-based time series classification on gpu. In *ICDM*, pages 131–140, 2012.

[Chen *et al.*, 2015] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, www.cs.ucr.edu/∼eamonn/time_series_data/. 2015.

[Donath and Hoffman, 1973] William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[Grabocka *et al.*, 2014] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *KDD*, pages 392–401, 2014.

[Halkidi *et al.*, 2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17(2):107–145, 2001.

[He *et al.*, 2005] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *NIPS*, pages 507–514, 2005.

[He *et al.*, 2012] Qing He, Zhi Dong, Fuzhen Zhuang, Tianfeng Shang, and Zhongzhi Shi. Fast time series classification based on infrequent shapelets. In *ICMLA*, pages 215–219, 2012.

[Hills *et al.*, 2014] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.

[Hirano and Tsumoto, 2006] Shoji Hirano and Shusaku Tsumoto. Cluster analysis of time-series medical data based on the trajectory representation and multiscale comparison techniques. In *ICDM*, pages 896–901, 2006.

[Lines *et al.*, 2012] Jason Lines, Luke M Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In *KDD*, pages 289–297, 2012.

[Mueen *et al.*, 2011] Abdullah Mueen, Eamonn Keogh, and Neal Young. Logical-shapelets: an expressive primitive for time series classification. In *KDD*, pages 1154–1162, 2011.

[Paparrizos and Gravano, 2015] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *SIGMOD*, pages 1855–1870, 2015.

[Rakthanmanon and Keogh, 2013] Thanawin Rakthanmanon and Eamonn Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SDM*, pages 668–676, 2013.

[Rand, 1971] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[Ruiz *et al.*, 2012] Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Correlating financial time series with micro-blogging activity. In *WSDM*, pages 513–522, 2012.

[Shariat and Pavlovic, 2011] Shahriar Shariat and Vladimir Pavlovic. Isotonic cca for sequence alignment and activity recognition. In *ICCV*, pages 2572–2578, 2011.

[Tang and Liu, 2014] Jiliang Tang and Huan Liu. An unsupervised feature selection framework for social media data. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2914–2927, 2014.

[Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[Wistuba *et al.*, 2015] Martin Wistuba, Josif Grabocka, and Lars Schmidt-Thieme. Ultra-fast shapelets for time series classification. *Journal of Data and Knowledge Engineering*, 2015.

[Yang *et al.*, 2011] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. l2,1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, pages 1589–1594, 2011.

[Ye and Keogh, 2009] Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *KDD*, pages 947–956, 2009.

[Zakaria *et al.*, 2012] Jamaluddin Zakaria, Abdullah Mueen, and Eamonn Keogh. Clustering time series using unsupervised-shapelets. In *ICDM*, pages 785–794, 2012.

[Zhang *et al.*, 2015] Peng Zhang, Chuan Zhou, Peng Wang, Byron J Gao, Xingquan Zhu, and Li Guo. E-tree: An efficient indexing structure for ensemble models on data streams. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):461–474, 2015.

[Zhao and Liu, 2007] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, pages 1151–1157, 2007.