Pattern Recognition 89 (2019) 55–66

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Time series feature learning with labeled and unlabeled data

Haishuai Wang[a,b,e], Qin Zhang[c], Jia Wu[d,*], Shirui Pan[f], Yixin Chen[e]

[a] Department of Computer Science and Engineering, Fairfield University, CT 06824, USA
[b] Department of Biomedical Informatics, Harvard Medical School, MA 02115, USA
[c] Cloud and Smart Industries Group (CSIG), Tencent, Shenzhen 518057, China
[d] Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
[e] Department of Computer Science and Engineering, Washington University in St Louis, MO 63130, USA
[f] Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia

ARTICLE INFO

Article history:
Received 28 September 2017
Revised 6 December 2018
Accepted 18 December 2018
Available online 19 December 2018

Keywords:
Time series
Feature selection
Semi-supervised learning
Classification

ABSTRACT

Time series classification has attracted much attention in the last two decades. However, in many real-world applications, the acquisition of sufficient amounts of labeled training data is costly, while unlabeled data is usually easily to be obtained. In this paper, we study the problem of learning discriminative features (segments) from both labeled and unlabeled time series data. The discriminative segments are often referred to as shapelets. We present a new Semi-Supervised Shapelets Learning (SSSL for short) model to efficiently learn shapelets by using both labeled and unlabeled time series data. Briefly, SSSL engages both labeled and unlabeled time series data in an integrated model that considers the least squares regression, the power of the pseudo-labels, shapelets regularization, and spectral analysis. The experimental results on real-world data demonstrate the superiority of our approach over existing methods.

1. Introduction

Time series is a set of numerical sequences with chronological order. Since time series is dynamic data and indicates the change rule of a phenomenon, it is significant to analyze time series to find the rules [1,2]. Time series analysis has been paid close attention due to the exponential growth of time-stamped data, such as economics and finance where we are continually exposed to daily stock market quotations, the research of natural phenomena based on natural gas network, power flow analysis for centralized PV plant, and intelligent fault diagnosis for electric machine. The main challenge for time series classification is to discover explainable and discriminative features that can best classify time series [3,4]. To tackle the problem, a series of research work has been proposed to explore discriminative features, referred to as shapelets [5,6], representing maximally discriminative segments of time series data. For example, Fig. 1 shows two sample shapelets extracted from the Coffee time series (available in UCR time-series repository [7]). Shapelets can capture inherent structures of time series, contributing to high prediction accuracy as explainable features. Thus, extracting shapelets from time series has given rise to increasing research interest during the last decade [8–11].

For fast *shapelet learning*, a recent work [12] proposes a regression model to extract shapelets. Compared with traditional approaches of shapelet discovery, the main merit of shapelet learning from time series is that it learns near-to-optimal shapelets directly, avoid searching exhaustively among a pool of candidates extracted from time-series segments. Therefore, shapelet learning is fast to compute and scalable to large data sets. Moreover, shapelet learning is robust to noise [12].

However, current shapelet learning approaches assume the existence of large amounts of labeled training time series data. In many real world applications, the labels of time series data are very expensive or difficult to obtain. Creating a large set of training data can be prohibitively expensive, time-consuming or even infeasible. For instance, human experts are only able to label a small portion of all available data. Thus, it is highly desired that the abundant amounts of unlabeled time series can be effectively utilized to select discriminative shapelets to improve the time series classification.

Most of the existing semi-supervised learning methods for time series classification are kernel-based methods, such as time series distance measurement [13] and probabilistic method [14]. Compared to kernel-based time series methods, shapelet-based approaches can explicitly identify segments that contribute to the classification [15–17].

In this paper, we aim to leverage both labeled and unlabeled time series data to build an effective semi-supervised shapelet learning model. The proposed optimization function treats

* Corresponding author.
E-mail addresses: Haishuai_Wang@hms.harvard.edu (H. Wang), jia.wu@mq.edu.au (J. Wu).

https://doi.org/10.1016/j.patcog.2018.12.026
0031-3203/

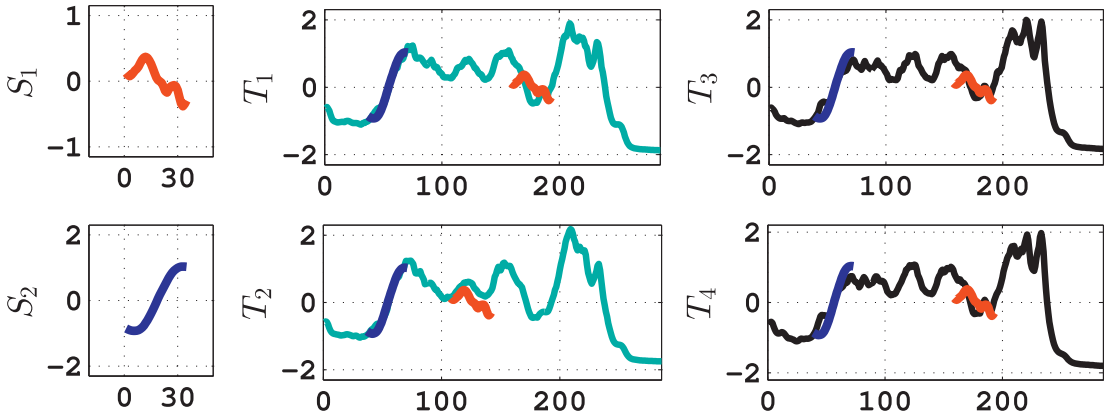


Fig. 1. An illustration of two learnt shapelets S_1 and S_2 from the Coffee data set (available in UCR time-series repository [7]). Shapelets are time series subsequences which are in some sense maximally representative of a class [12].

unlabeled samples in supervised fashion by using pseudo-labels, and then uses regularized least-square technique to learn both shapelets and classification boundaries. Meanwhile, spectral analysis is integrated in the function to preserve local structure information in the data. Moreover, a new regularization term is added to avoid selecting similar or redundant shapelets. A coordinate descent algorithm is then proposed to iteratively solve the classification boundary, pseudo-labels and shapelets, respectively.

The main contributions of our work are summarised as follows:

- To our knowledge, this is the first effort of shapelet learning that leverages both labeled and unlabeled time series data.
- A new semi-supervised shapelet learning (SSSL) model is devised that integrates least square minimization, spectral analysis, scaled pseudo labels as well as shapelet similarity regularization terms.

We also evaluate our proposed algorithm on real-world data sets and compare it with the state-of-the-art semi-supervised time series classification approaches. The experimental results demonstrate the effectiveness of the proposed model.

The remainder of this paper is organized as follows. Section 2 reviews traditional semi-supervised time series classification methods and feature extraction in time series data. Section 3 provides the preliminaries and problem definition. We introduce the proposed semi-supervised shapelet learning model in Section 4. Section 5 shows how to solve the proposed objective function in the proposed SSSL algorithm. In Section 6, we conduct the experiments on real-world data sets and compare the proposed method with benchmark approaches. Finally, we draw a conclusion and point out the future work in Section 7.

2. Related work

In this section, we review existing research related to our work in the following areas: time series classification, semi-supervised feature learning, and semi-supervised time series classification.

2.1. Time series classification

Time series classification attracts increasing interest in data mining because of its wide applications in different domains. For example, in economics and finance, we are continually exposed to daily stock market quotations [18]. Time series classification is also used in analyzing medical data [19], and moving trajectory analysis [20].

To date, existing time series classification algorithms can be generally categorized into two groups: distance-based methods, and feature-based methods. The former directly measures the similarity between two time series (e.g., dynamic time warping (DTW) [21]), while the later considers time series as feature vectors so that traditional feature-based classifier (e.g., SVM or logistic regression) can be applied. Feature-based methods rely on extracting or learning a set of feature vectors from each time series. Therefore, the main challenge of time series classification is to find discriminative features that best predict class labels. One direction of feature-based methods is feature encoding [22–25]. The feature vectors are firstly quantized into words, using a learned dictionary. Then, each time series can be represented by a histogram of word occurrences. Finally, the extracted feature is fed into a classifier, such as SVM, for time series classification. Many feature encoding approaches are used for time series classification in the literature, e.g., bag-of-words [22,23,26], sparse coding [24], fisher vector [22]. The Bag-of-Word (BoW) framework is inspired by the text mining and computer vision communities, and has been shown to be very efficient. However, the major drawback of BoW is that the quantization step is done by a fixed partitioning of the feature space, resulting in a loss of information.

To solve the challenge, a line of research has been undertaken to extract discriminative features, which are often referred to as shapelets [27], from time series. Therefore, discovering shapelets has become an important branch in time series analysis. Shapelets are maximally discriminative features of time series which enjoy the merits of high prediction capability and interpretability. The basic idea of shapelets discovery is to consider all segments from training time series data and assess them regarding a scoring function to estimate how predictive they are with respect to the given class labels [28]. However, this type of shapelets selection could be extremely inefficient because time series usually have a large number of candidate segments. Therefore, a recent work [29] proposes a new time series shapelets learning approach. Instead of searching shapelets from a candidate pool, they use regression learning 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, shapelets learning is fast to compute, scalable to large data sets, and robust to noise.

Our previous work [30] proposes an efficient unsupervised shapelets learning algorithm to classify unlabeled time series. However, this work only involves unlabeled time series and does not have function estimation on both labeled and unlabeled time series. In this paper, our approach is motivated by the fact that labeled data is often costly to generate, whereas unlabeled data is generally not. The challenge here mostly involves the technical question of how to treat time series mixed in this fashion.

2.2. Semi-supervised feature learning

Learning from both labeled and unlabeled data is called semi-supervised learning. There is a large body of research on semi-supervised learning as labeled data is usually hard to get while unlabeled data is readily available [31–33]. The most natural approach for semi-supervised learning is *self-training* [34,35]. In self-training, a classifier is first trained with a small number of labeled data. It is then used to classify the unlabeled data. By adding the most confidently classified objects into the labeled set, the classifier re-trains itself using the new labeled set. The procedure is repeated until adding newly labeled objects to the labeled set does not increase the accuracy of the classifier, or some other stopping criteria is met. Generative models [36,37] are perhaps the oldest semi-supervised learning method. It assumes a model is an identifiable mixture distribution (e.g., Gaussian mixture models). With large amounts of unlabeled data, the mixture components can be identified. Then, ideally we only need one labeled example per component to fully determine the mixture distribution. Graph-based semi-supervised methods [36,38,39] define a graph where the nodes represent labeled and unlabeled examples in the data set, and edges (may be weighted) reflect the similarity of examples.

2.3. Semi-supervised classification on time series

Several semi-supervised learning methods for time series classification have been proposed in the literature. Based on self-training method, Wei et al. [13] proposed a semi-supervised approach on one-nearest-neighbor with Euclidean distance classifier. Chen et al. [40] showed that Euclidean distance performed poorly in many time series classification cases and proposed a DTW-D approach by using Dynamic Time Warping (DTW). They claimed that DTW-D could perform better on some special time series data sets but not for all time series problems. Based on constrained hierarchical clustering, Marussy et al. proposed a SUCCESS method to cluster the whole set (including both labeled and unlabeled data) of time series by using single-linkage hierarchical agglomerative clustering firstly. Then, the top-level clusters were labeled by their corresponding seeds [41]. Recently, Xu and Funaya presented a graph-based semi-supervised time series classification approach and developed a probabilistic method for learning optimal graph combination to effectively capture the underlying structure of time series data [14].

Our work differs from the aforementioned works, as we introduce shapelet learning to semi-supervised time series classification, which automatically learns shapelets from both labeled and unlabeled time series. In contrast, all of the above semi-supervised approaches for time series can be considered as kernel methods. Because time series are potentially infinite, kernel-based methods often cannot identify which segments of time series are mostly discriminative for distinguishing between time series data from different classes.

3. Preliminaries

In this paper, we use lower-case bold-faced letters to represent vectors and upper-case bold-faced letters to represent matrices (e.g., $\mathbf{A}$). We use $\mathbf{A}(i,j)$ to denote the element locating at the ith row and j-column of matrix $\mathbf{A}$, and $\mathbf{A}(i,:)$ and $\mathbf{A}(:,j)$ denote vectors of the ith row and jth column of the matrix respectively. Table 1 summarizes major notations used in the paper.

Consider a set of n time series, $\mathbf{T}=\{\mathbf{T}_1,\mathbf{T}_2,\ldots,\mathbf{T}_n\}$, where each time series has q_i ordered real-valued observations $\mathbf{T}_i=<T_1^i,T_2^i,\cdots,T_{q_i}^i>$ and a class value c_i. Consider a sliding window of length ρ, when the window slides along a time series, a set of segments can be obtained. For time series $\mathbf{T}_i\in\mathbf{T}$, we can generate totally $q-\rho+1$ segments by sliding the window.

Shapelets are defined as the most discriminative time series segments. Therefore, time series segments are shapelet candidates. To represent each time series $\mathbf{T}_i\in\mathbf{T}$, we use a vector $\mathbf{S}_j$ to record $\mathbf{T}_i's$ feature values. This way, the time series data set $\mathbf{T}$ can be represented by a data matrix $\mathbf{S}=\{\mathbf{S}_1,\mathbf{S}_2,\cdots,\mathbf{S}_m\}$.

Similar to the shapelets learning model [12], 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 m_i shapelets and $m=\sum_{i=1}^{r}m_i$. The shapelets therefore will be defined as $\mathbf{S}=\{\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_m\}$, where $\mathbf{S}\in\bigcup_{i=1}^{r}\mathbb{R}^{m_i\times(i\times l_{\min})}$ and $r\times l_{\min}\ll q_i$.

In our problem setting, there are two types of time series data: labeled and unlabeled time series. We use subscript u and l of variable to represent unlabeled and labeled data respectively. For example, $\mathbf{Y}_l$ denotes the class label matrix of labeled time series while $\mathbf{Y}_u$ denotes the pseudo-class label of unlabeled time series. Suppose these n_u unlabeled time series are from c classes and denote $\mathbf{Y}_u=[y_1^u,\ldots,y_{n_u}^u]\in\{0,1\}^{c\times n_u}$, where $y_i^u\in\{0,1\}^{c\times 1}$ is the pseudo-class label vector for unlabeled time series sample T_i. Pseudo-class label is a simple and an efficient method to do semi-supervised learning. The proposed optimization function treats unlabeled samples in supervised fashion by using pseudo-labels. To obtain this pseudo-labels, we first initialized the pseudo-labels randomly, and then update them as an optimization parameter in our objective function. The scaled pseudo-class label matrix [42,43] $\mathbf{Z}$ is defined as

$$\mathbf{Z}=[z_1,\ldots,z_{n_u}]=\mathbf{Y}_u(\mathbf{Y}_u^T\mathbf{Y}_u)^{-\frac{1}{2}} \tag{1}$$

where z_i is the scaled pseudo-class indicator of unlabeled time series T_i. We thus have

$$\mathbf{Z}^T\mathbf{Z}=(\mathbf{Y}_u^T\mathbf{Y}_u)^{-\frac{1}{2}}\mathbf{Y}_u^T\mathbf{Y}_u(\mathbf{Y}_u^T\mathbf{Y}_u)^{-\frac{1}{2}}=\mathbf{I}_c \tag{2}$$

where $\mathbf{I}_c\in\mathbb{R}^{c\times c}$ is an identity matrix.

4. Semi-supervised shapelets learning

In this section, we first formulate the semi-supervised shapelets learning model in Section 4.1. After that, we propose the shapelet-transformed representation of time series in Section 4.2, and then we sequentially introduce the spectral analysis in Section 4.3, least square minimization in Section 4.4 and shapelets similarity regularization in Section 4.5.

4.1. Semi-supervised shapelets learning model

The semi-supervised shapelets learning model (SSSL) can be formulated by Eq. (3). SSSL is a joint optimization problem with respect to the classification boundary $\mathbf{W}$, scaled Pseudo-class label $\mathbf{Z}$ and candidate shapelets $\mathbf{S}$.

$$\begin{aligned}\min_{\mathbf{W},\mathbf{S},\mathbf{Z}}\ &\frac{1}{2}tr(\mathbf{Z}\mathbf{L}_G(\mathbf{S})\mathbf{Z}^\top)+\frac{\lambda_1}{2}\|\mathbf{H}(\mathbf{S})\|_F^2+\frac{\lambda_2}{2}\|\mathbf{W}\|_F^2\\&+\frac{\lambda_3}{2}\|\mathbf{W}^\top\mathbf{X}_u(\mathbf{S})-\mathbf{Z}\|_F^2+\frac{\lambda_4}{2}\|\mathbf{W}^\top\mathbf{X}_l(\mathbf{S})-\mathbf{Y}_l\|_F^2\\ s.t.\ &\mathbf{Z}\in\mathbb{R}_+^{c\times n},\mathbf{Z}^T\mathbf{Z}=\mathbf{I}_c\end{aligned} \tag{3}$$

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 last three terms are the regularized least square minimization with respect to unlabeled and labeled time series. From the objective function in Eq. (3), the discriminative shapelets can be learned from training data and used for test data classification. The objective function Eq. (3) returns classification boundary $\mathbf{W}$ and shapelets $\mathbf{S}$. Thus, we can compute the shapelet-transformed matrix $\mathbf{X}(\mathbf{S})$ for test time series. Then, the probability

Table 1
Symbols and notations.

Symbols	Descriptions
A(i,j)	the element locating at the *i*th row and *j*-column of matrix **A**
A$(i,:)$	vectors of the *i*th row of the matrix
A$(:,j)$	vectors of the *j*th column of the matrix
T	time series
$\mathbf{T}_j^i$	the *j*th value in time series $\mathbf{T}_i$
m, n	the number of shapelets, the number of time series
q_j	the length of time series $\mathbf{T}_j$
l_i	the length of shapelet $\mathbf{S}_i$
$\bar{q}$	the total number of segments with length l_i of time series $\mathbf{T}_j$
S	the set of shapelets
$\mathbf{Y}_l$	the class label matrix of labeled time series
$\mathbf{Y}_u$	the pseudo-class label of unlabeled time series
W	the weight matrix of each shapelet
Z, **I**	pseudo-class label, an identity matrix
G	the similarity matrix of time series based on the shapelet-transformed representation matrix **X**(**S**)
$\mathbf{L}_G$	Laplacian matrix
H(**S**)	the similarity matrix between each two shapelets

of belonging to each class for the test time series can be predict based on the calculated **X**(**S**), i.e.: $\mathbf{W}^T\mathbf{X}(\mathbf{S})$. We use the class that has the highest probability as the predicted label.

The parameter **S** represents the selected shapelet features from both labeled and unlabeled time series. Parameter **Z** represents scaled pseudo-class label, and parameter **W** is classification boundary that can classify time series based on the learned shapelets. For example, we first initialize the pseudo-labels **Z** and shapelets **S** with a certain length. The pseudo-labels and shapelets can be updated by minimizing the objective function in Eq. (3). Then, we can calculate the shapelet-transformed matrix **X** (Section 4.2) to discover the most discriminative shapelet patterns. **H(S)** is to ensure the selected shapelets are diversity. After that, the parameter **W** is updated based on least square minimization to classify different categories of time series.

Because matrices $\mathbf{L}_G$, **H**, $\mathbf{X}_u$ and $\mathbf{X}_l$ in Eq. (3) depend on the shapelets **S**, we explicitly write these matrices as variables with respect to shapelets **S**, i.e., $\mathbf{L}_G(\mathbf{S})$, $\mathbf{H}(\mathbf{S})$, $\mathbf{X}_u(\mathbf{S})$ and $\mathbf{X}_l(\mathbf{S})$.

Before we explain the SSSL in detail, we firstly introduce the shapelet-transformed representation of time series, which transfer time series from original space to a shapelet-based space [44]. Then, we introduce the spectral analysis, least square minimization and shapelets similarity regularization respectively.

4.2. Shapelet-transformed representation

Prior to classification, transforming a time series classification problem into an alternative data space can provide a significant improvement than performing classification directly. Lines et al. [44] proposed a shapelet transform that generates a new classification data set independently of the classifier, which can downsize a long time series into a short feature vector in the shapelets feature space.

Given a set of n time series $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_n\}$ and a set of shapelet $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_m\}$, the distance between the *i*th shapelet $\mathbf{S}_i$ and the *j*th time series $\mathbf{T}_j$ is denoted as the minimum distance $\mathbf{X}(i,j)$ among the distances between the shapelet $\mathbf{S}_i$ and each time series $\mathbf{T}_j$ [44]. Hence, we use $\mathbf{X}(\mathbf{S}) \in \mathbb{R}^{m\times n}$ to denote the shapelet-transformed matrix, where each element $\mathbf{X}(i,j)$ can be calculated in Eq. (4),

$$\mathbf{X}(i,j) = \min_{g=1,\cdots,\bar{q}} \frac{1}{l_i}\sum_{h=1}^{l_i}(\mathbf{T}_{g+h-1}^j - \mathbf{S}_h^i)^2 \tag{4}$$

where q_j is the length of time series $\mathbf{T}_j$ and l_i is the length of shapelet $\mathbf{S}_i$, and $\bar{q} = q_j - l_i + 1$ denotes the total number of segments with length l_i of time series $\mathbf{T}_j$. $\mathbf{X}_u$ denotes the shapelet-transformed matrix of unlabeled time series, while $\mathbf{X}_l$ denotes the shapelet-transformed matrix of labeled time series.

Minimum distances to shapelets can be characterized as a transformation of the time series data $\mathbf{T} \in \mathbb{R}^{q\times n}$ $(q = max\{q_1, \cdots, q_n\})$ into a new representation $\mathbf{X}(\mathbf{S}) \in \mathbb{R}^{m\times n}$. As $m < q$, such a transformation reduces the original time series dimension space.

To compute the derivative of the objective function in Eq. (3), each term in Eq. (3) should be differentiable. However, the distance function in Eq. (4) is not continuous and, thus, non-differential. A differentiable approximation to the minimum function is introduced in [12], which approximates the distance function by using the *Soft Minimum* function as in Eq. (5),

$$\mathbf{X}(i,j) \approx \frac{\sum_{q=1}^{\bar{q}} d_{i,j,q}\cdot e^{\alpha d_{i,j,q}}}{\sum_{q=1}^{\bar{q}} e^{\alpha d_{i,j,q}}} \tag{5}$$

where parameter α controls the precision of the function and the soft minimum approaches the true minimum when $\alpha \to -\infty$, and $d_{i,j,q} = \frac{1}{l_i}\sum_{h=1}^{l_i}(\mathbf{T}_{q+h-1}^j - \mathbf{S}_h^i)^2$. Based on the observation in [12], $\alpha = -100$ is small enough to make the soft minimum yield exactly the same results as the true minimum. Thus, we kept this value fixed throughout all our experiments.

4.3. Spectral analysis

Spectral analysis has been widely used in feature learning on unlabeled data [45,46]. The main idea of the spectral analysis is that samples (time series) that are close to each other are likely to share the same class label. For two similar unlabeled series $\mathbf{T}_i$ and $\mathbf{T}_j$, their pseudo-class labels (i.e., $\mathbf{Z}(:,i)$ and $\mathbf{Z}(:,j)$) are needed to be same, Therefore, we can formulate a spectral regularization term as follows,

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{b}\mathbf{G}(i,j)\|\mathbf{Z}(:,i) - \mathbf{Z}(:,j)\|_2^2 \\
=&\frac{1}{2}\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{b}\mathbf{G}(i,j)(\mathbf{Z}(k,i) - \mathbf{Z}(k,j))^2 \\
=&\sum_{k=1}^{c}\mathbf{Z}(k,:)(\mathbf{D}_G - \mathbf{G})\mathbf{Z}(k,:) \\
=&tr(\mathbf{Z}\mathbf{L}_G\mathbf{Z})
\end{aligned} \tag{6}$$

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

similarity matrix of time series based on the shapelet-transformed representation matrix $\mathbf{X}(\mathbf{S})$, then $\mathbf{G}$ is calculated by Eq. (7)

$$\mathbf{G}(i,j) = e^{-\frac{\|\mathbf{X}_u(:,i)-\mathbf{X}_u(:,j)\|^2}{\sigma^2}} \tag{7}$$

where σ is the parameter of the RBF kernel.

4.4. Least square minimization

We aim to minimize the least square error based on the labeled time series $\mathbf{X}_l, \mathbf{Y}_l$ and the unlabeled ones $\mathbf{X}_u, \mathbf{Z}$. We use $\mathbf{W} \in \mathbb{R}^{m\times c}$ to represent the unified classification boundary for both labeled and unlabeled time series. Therefore, we minimize the least square errors of the predicted and the labels, and a regularization term is further added. The specific function is shown in Eq. (8),

$$\min_{\mathbf{W}} \|\mathbf{W}^\top \mathbf{X}_l - \mathbf{Y}_l\|_F^2 + \|\mathbf{W}^\top \mathbf{X}_u - \mathbf{Z}\|_F^2 + \|\mathbf{W}\|_F^2 \tag{8}$$

By combining with shapelet-transformed representation, spectral analysis, shapelet similarity minimization and least square minimization terms, we final formulate the semi-supervised shapelets learning model as in Eq. (3).

4.5. Shapelet similarity minimization

The selected most discriminative shapelets should be diverse in shape to avoid similar shapelets be selected. Assume that $\mathbf{H}(\mathbf{S}) \in \mathbb{R}^{m\times m}$ is the similarity matrix, where each element $\mathbf{H}(i,j)$ represents the similarity between two shapelets $\mathbf{S}_i$ and $\mathbf{S}_j$, and $\mathbf{H}(i,j)$ can be calculated as follows,

$$\mathbf{H}(i,j) = e^{-\frac{\|d_{i,j}\|^2}{\sigma^2}} \tag{9}$$

where $d_{i,j}$, the distance between shapelets $\mathbf{S}_i$ and $\mathbf{S}_j$, can be calculated by Eq. (5).

5. The SSSL algorithm

To solve the optimization function in SSSL, we first rewrite the optimization problem in Eq. (3) as follows,

$$\begin{aligned}\min_{\mathbf{W},\mathbf{S},\mathbf{Z}} \ & \frac{1}{2} tr(\mathbf{Z}\mathbf{L}_G(\mathbf{S})\mathbf{Z}^\top) + \frac{\lambda_1}{2}\|\mathbf{H}(\mathbf{S})\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 \\ & + \frac{\lambda_3}{2}\|\mathbf{W}^\top\mathbf{X}_u(\mathbf{S}) - \mathbf{Z}\|_F^2 + \frac{\lambda_4}{2}\|\mathbf{W}^\top\mathbf{X}_l(\mathbf{S}) - \mathbf{Y}_l\|_F^2 \\ & + \frac{\zeta_1}{2}(\mathbf{Z}^T\mathbf{Z} - \mathbf{I}_c) - \zeta_2\mathbf{Z}\end{aligned} \tag{10}$$

where ζ_1, ζ_2 are parameters to control the orthogonality and positive conditions. In practise, ζ_1 and ζ_2 should be large enough to insure the constraints satisfied. Then we resort to the coordinate descent algorithm to iteratively update one variable by fixing the remaining two variables.

1) **Solve Z (fix W and S):** With fixed $\mathbf{W}$ and $\mathbf{S}$, Eq. (10) degenerates to

$$\begin{aligned}\min_{\mathbf{Z}} \mathcal{F}(\mathbf{Z}) = & \frac{1}{2} tr(\mathbf{Z}\mathbf{L}_G\mathbf{Z}^\top) + \frac{\lambda_3}{2}\|\mathbf{W}^\top\mathbf{X}_u(\mathbf{S}) - \mathbf{Z}\|_F^2 \\ & + \frac{\zeta_1}{2}(\mathbf{Z}^T\mathbf{Z} - \mathbf{I}_c) - \zeta_2\mathbf{Z}\end{aligned} \tag{11}$$

Then by setting the derivatives of the function in Eq. (11) to 0 with respect to parameters $\mathbf{Z}$, we obtain that $\mathbf{Z}$ can be updated as follows,

$$\mathbf{Z}^{t+1} = (\lambda_3\mathbf{W}_t^\top\mathbf{X}_u^t + \zeta_2\mathbf{I})(\mathbf{L}_G^t + \lambda_3\mathbf{I} + \zeta_1\mathbf{I})^{-1} \tag{12}$$

The derivation of Eq. (12) is in Appendix A.1.

2) **Solve W (fix S and Z):** With fixed $\mathbf{S}$ and $\mathbf{Z}$, Eq. (10) degenerates to

$$\begin{aligned}\min_{\mathbf{W}} \mathcal{F}(\mathbf{W}) = & \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda_3}{2}\|\mathbf{W}^\top\mathbf{X}_u(\mathbf{S}) - \mathbf{Z}\|_F^2 \\ & + \frac{\lambda_4}{2}\|\mathbf{W}^\top\mathbf{X}_l(\mathbf{S}) - \mathbf{Y}_l\|_F^2\end{aligned} \tag{13}$$

Then by setting the derivatives of the function in Eq. (13) to 0 with respect to parameters $\mathbf{W}$, we obtain that $\mathbf{W}$ can be updated as follows,

$$\begin{aligned}\mathbf{W}_{t+1} = & (\lambda_2\mathbf{I} + \lambda_3\mathbf{X}_u^t(\mathbf{X}_u^t)^\top + \lambda_4\mathbf{X}_l^t(\mathbf{X}_l^t)^\top)^{-1} \\ & \times (\lambda_3\mathbf{X}_u^t(\mathbf{Z}^t)^\top + \lambda_4\mathbf{X}_l^t(\mathbf{Y}_l^t)^\top)\end{aligned} \tag{14}$$

The derivation of Eq. (14) is in Appendix A.2.

3) **Solve S (fix W and Z)**: With fixed $\mathbf{W}$ and $\mathbf{Z}$, Eq. (10) degenerates to

$$\begin{aligned}\min_{\mathbf{S}} \mathcal{F}(\mathbf{S}) = & \frac{1}{2} tr(\mathbf{Z}\mathbf{L}_G(\mathbf{S})\mathbf{Z}^\top) + \frac{\lambda_1}{2}\|\mathbf{H}(\mathbf{S})\|_F^2 \\ & + \frac{\lambda_3}{2}\|\mathbf{W}^\top\mathbf{X}_u(\mathbf{S}) - \mathbf{Z}\|_F^2 + \frac{\lambda_4}{2}\|\mathbf{W}^\top\mathbf{X}_l(\mathbf{S}) - \mathbf{Y}_l\|_F^2\end{aligned} \tag{15}$$

We cannot solve $\mathbf{S}$ like updating $\mathbf{W}$ and $\mathbf{Z}$ because Eq. (15) is non-convex. We use an iterative algorithm by setting a learning rate η, i.e., $\mathbf{S}_{i+1} = \mathbf{S}_i - \eta\nabla\mathbf{S}_i$, where $\nabla\mathbf{S}_i = \partial\mathcal{F}(\mathbf{S}_i)/\partial\mathbf{S}$. The derivative of Eq. (15) with respect to $\mathbf{S}(k,p)$ is

$$\begin{aligned}\frac{\partial\mathcal{F}(\mathbf{S})}{\mathbf{S}(k,p)} = & \frac{1}{2}\mathbf{Z}^\top\mathbf{Z}\frac{\partial\mathbf{L}_G(\mathbf{S})}{\partial\mathbf{S}(k,p)} + \lambda_1\mathbf{H}(\mathbf{S})\frac{\partial\mathbf{H}(\mathbf{S})}{\partial\mathbf{S}(k,p)} \\ & + \lambda_3\mathbf{W}(\mathbf{W}^\top\mathbf{X}_u(\mathbf{S}) - \mathbf{Z})\frac{\partial\mathbf{X}_u(\mathbf{S})}{\partial\mathbf{S}(k,p)} \\ & + \lambda_4\mathbf{W}(\mathbf{W}^\top\mathbf{X}_l(\mathbf{S}) - \mathbf{Y}_l)\frac{\partial\mathbf{X}_l(\mathbf{S})}{\partial\mathbf{S}(k,p)}\end{aligned} \tag{16}$$

where $k = 1, \cdots, m$, and $p = 1, \cdots, l_k$.

Appendix A.3 provides the details to calculate the gradient $\nabla\mathbf{S}_i = \frac{\partial\mathcal{F}(\mathbf{S})}{\partial\mathbf{S}}$.

Algorithm 1 SSSL: Semi-supervised shapelets learning.

```
Initialize: S_0, W_0, Z_0;
Generate the labeled and unlabeled series based on τ;
repeat
    Calculate:
        X(i, j) based on Eq. (4);
        L_G^t based on Eq. (6);
        H_t based on Shapelet similarity minimization;
    for i={1, ..., l_min} do
        S_{i+1} ← S_i − η∇S_i;
        ∇S_i = ∂F(S_i)/∂S   // from Eq. (16)
    end for
    S_{t+1} = Si_max + 1
    W_{t+1} ← (λ2 I + λ3 X_u^t (X_u^t)^⊤ + λ4 X_l^t (X_l^t)^⊤)^{-1} (λ3 X_u^t (Z^t)^⊤ +
    λ4 X_l^t (Y_l^t)^⊤);
    Z^{t+1} ← λ3 W_t^⊤ X_u^t (L_G^t + λ3 I)^{-1}; //update W_{t+1}, Z^{t+1}
    t ← t + 1;
until Convergence
return S* = S_{t+1}, W* = W_{t+1};
```

After having derived the gradients of the shapelets, pseudo-labels, and the weights, we can introduce the overall learning algorithm. Our approach iteratively updates one value by fixing the remaining two variables based on the coordinate descent algorithm. Therefore, the final algorithm for optimizing Eq. (3) is presented in Algorithm 1. The algorithm expects to initialize $\mathbf{S}_0$, $\mathbf{Z}_0$, and $\mathbf{W}_0$. The algorithm performance and convergence speed depend on the

Table 2
The real-world time series data sets.

Data set	Classes	Size of dataset	Length
Coffee	2	56	286
CBF	3	930	128
ECG	2	200	96
Face four	4	112	350
Gun point	2	200	150
ItalyPow.Dem.	2	1096	24
Lighting2	2	121	637
Lighting7	7	143	319
OSU leaf	6	442	427
Trace	4	200	275
WordsSyn	25	905	270
OliveOil	4	60	570
StarLightCurves	3	9236	1024

parameters initialization. As a result, we applied clustering techniques to initialize the shapelets more efficiently. 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, $\mathbf{X}_0$ can be initialized by using the shapelet-transformed matrix of time series. Next, the centers of clusters obtained by k-means is used to initialize $\mathbf{W}_0$ and $\mathbf{Z}_0$. The initialization can achieve fast convergence. The initialization enables fast convergence. After initializing $\mathbf{S}_0, \mathbf{Z}_0$, and $\mathbf{W}_0$, we can calculate $\mathbf{X}(i, j)$ based on Eq. (4), $\mathbf{L}_G^t$ based on Eq. (6), and H_t based on shapelet similarity minimization. Then, theses matrices are used for updating $\mathbf{S}_{t+1}$, $\mathbf{W}_{t+1}$ and $\mathbf{Z}^{t+1}$. The process ends until the objective function becomes convergence.

Convergence: The convergence of Algorithm 1 depends on stepwise descents. When updating $\mathbf{S}$, a closed-form derivative is difficult to obtain as the objective function in Eq. (15) is not convex. We resort to a gradient descent algorithm as a solution to tackle this problem. We set a small learning rate η to make the objective function decrease to convergence. Due to the objective function in Eq. (3) is non-negative, it has a lower bound of 0. Therefore, Algorithm 1 converges to local optima. We run the algorithm several times under various initializations to select the best solution as output.

6. Experiments

The extensive experiments are carried out to validate the effectiveness of SSSL model compared with the state-of-the-art approaches. All experiments are conducted on a Linux Ubuntu server with 16*2.9 GHZ CPU and 64G memory.

6.1. Data sets

We evaluate our algorithm on 13 publicly available real-world data sets from UCR time-series repository [7]. Due to the space limitation, the 13 data sets from the UCR repository are randomly selected but include all the data sets used in the state-of-the-art method in [14]. The data sets consisting of time-series data sets with various numbers of instances, lengths and number of classes. The detailed information is summarized in Table 2.

6.2. Benchmark methods

- **Wei**'s approach [13] is one of the most prominent semi-supervised time-series classifiers. The method starts by training a Euclidean distance-based classifier with labeled data, by which the unlabeled data can be classified. Then, the most confident unlabeled time series are added to the training data. The classifier is retrained and the procedure repeated until convergence.
- **DTW-D** [40] uses modified Dynamic Time Warping (DTW) as distance measure. The method firstly trains a classifier on labeled data, then computes distance of each unlabeled samples to labeled samples using DTW-D. Among all the unlabeled objects, the one that is closest to the labeled samples will be added into the labeled data set. The procedure repeated until stopping criterion is met.
- **SUCCESS** [41] is based on constrained hierarchical clustering. The method clusters all of the labeled and unlabeled time series by using single-linkage hierarchical agglomerative clustering. Then the top-level clusters are labeled by their corresponding seeds.
- **Xu**'s approach [14] is a graph-based semi-supervised learning framework. This *state-of-the-art* method first constructs a graph to effectively derive the underlying structures of the whole set of time series data. Then the unlabeled time series are classified by label propagation over the constructed graph based on harmonic Gaussian fields method.
- **BoW** [26] generates the bag-of-words representation for time series classification. Firstly, a group of local segments are extracted from each time series by a sliding window with a certain length. Then, a codebook consists of several codewords is constructed by clustering all local segments from training time series. Each local segment is assigned a codeword. The time series is then represented as a histogram of codewords, each entry of which is the count of a codeword appeared in the time series. Finally, the bag-of-words representation is used as input to SVM for classification. Since the BoW method is based on supervised learning, we only used the labeled time series to construct the codebook and extract a histogram representation for each time series.

6.3. Measures

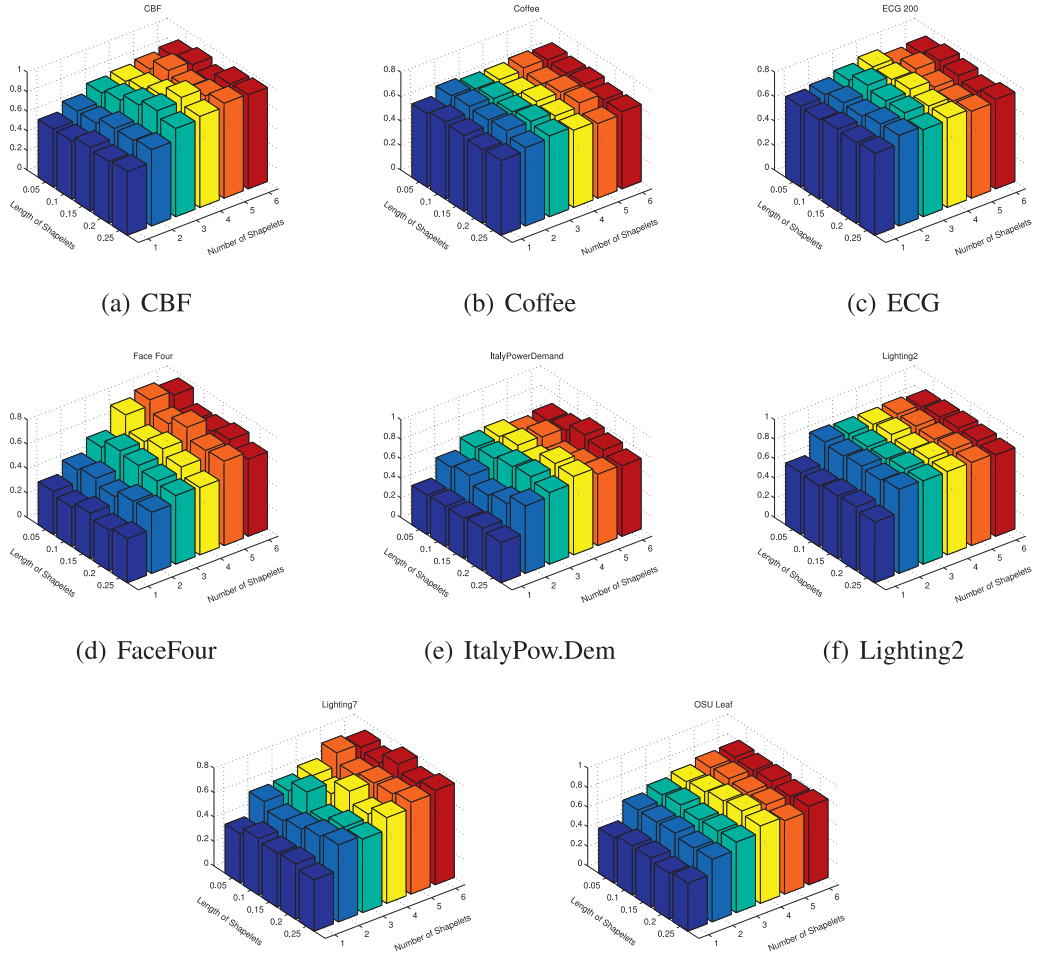
In the experiments, we use 80% for training and the remaining 20% for testing, as the same in all baselines [13,14,40,41]. In the training instances, we randomly split 10% (20%, 30% and 40%) as labeled time series while the remaining instances as unlabeled time series.

We measure the performance of the baselines and our approach by classification accuracy: *(tp+tn) / (tp+fp+fn+tn)*, where *tp* is true positive, *fp* is false positive, *fn* is false negative, and *tn* is true negative. For each data set, we repeated all experiments 10 times, the averaged result is reported to measure the performance.

6.4. Parameter study

Fig. 2 shows the parameter tests with respect to the two key parameters in SSSL, i.e., the number of shapelets m and the length of shapelets $l_{\min}$ on the real-world data sets. The parameters are $m = 1$, $l_{\min} = 0.1$ by default. We set the learning rate $\eta = 0.01$ and the number of iterations $i_{\max} = 50$ in updating $\mathbf{S}$. We use the grid search to choose the best parameters because SSSL contains many parameters, i.e., the parameters λ_1, λ_2, λ_3 and λ_4 vary from 10^{-8} to 10^8. The number of shapelets varies in a range of $m \in \{1, 2, 3, 4, 5, 6\}$, and $l_{\min} \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$, which is a fraction of the series length, e.g. $m = 0.2$ means 20% of q_i. For the baselines, the parameter settings are based on the settings in their original publications.

Based on the results in Fig. 2, we can observe that the algorithm performs well when the parameters m and $l_{\min}$ are set to a small value. Therefore, it is not necessary to set the shapelet number and shapelet length to be very large.



(a) CBF (b) Coffee (c) ECG

(d) FaceFour (e) ItalyPow.Dem (f) Lighting2

(g) Lighting7 (h) OSU Leaf

Fig. 2. Parameter study based on the classification accuracy of SSSL with respect to the number m and the length $l_{\min}$ of the learnt shapelets.

Table 3
Comparisons of classification accuracy between SSSL and baselines.

Data set	Wei	DTW-D	SUCCESS	Xu	BoW	SSSL
Coffee	0.571	0.601	0.632	0.588	0.620	**0.792**
CBF	0.995	0.833	0.997	0.921	0.873	**1.00**
ECG	0.763	0.953	0.775	0.819	**0.955**	0.793
Face four	0.818	0.782	0.800	0.833	0.744	**0.851**
OSU leaf	0.468	0.701	0.534	0.642	0.685	**0.835**
ItalyPow.Dem.	0.934	0.664	0.924	0.772	0.813	**0.941**
Lighting2	0.658	0.641	0.683	0.698	0.721	**0.813**
Lighting7	0.464	0.503	0.471	0.511	0.677	**0.796**
Gun point	0.925	0.711	**0.955**	0.729	0.925	0.824
Trace	0.950	0.801	**1.00**	0.788	**1.00**	**1.00**
WordsSyn	0.590	0.863	0.618	0.639	0.795	**0.875**
OliveOil	0.633	0.732	0.617	0.639	0.766	**0.776**
StarLightCurves	0.860	0.743	0.800	0.755	0.851	**0.872**
Average	0.741	0.733	0.754	0.718	0.802	**0.859**

6.5. Comparisons

In this part, we compare SSSL with the state-of-the-art baselines. We report the best results by conducting experiments ten times under different parameter initializations. We set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1, \sigma = 1, I_{\max} = 50, \eta = 0.01$, and $l_{\min}$ changes over $\{0.05, 0.1, 0.15\}$, $r \in \{1, 2\}$, and $m \in \{1, 2, 3\}$.

Table 3 shows the comparisons given 10% labeled data with respect to classification accuracy. We can observe that the proposed SSSL method performs better than other four benchmarks on the most of data sets. The results validate the effectiveness of the semi-supervised learning-based shapelets discovery method. Since we repeated all experiments 10 times, Fig. 3 illustrates the boxplot of accuracies of 10 times on different data sets with 10% labeled data. From the boxplot, we can see that the proposed SSSL algorithm is more robustness and less fluctuation of the performance.

The classification accuracy of SSSL with respect to different ratios τ of labeled data (10%, 20%, 30%, 40%) are shown in Fig. 4. It reveals that the classification accuracy rises with increasing the ratios of labeled time series. This is because more labeled data can help to improve the performance of semi-supervised learning. Nevertheless, our proposed SSSL model can achieve acceptable accuracy even with low ratios of labeled data.

6.6. The learnt shapelets

Fig. 5 lists the shapelets learnt by SSSL on the real-world data sets. We vary the number of shapelets from 1 to 6 and draw the learnt shapelets in red. The results show that when increasing the number of shapelets, there will be heavy overlap of the learnt shapelets. This confirms that we do not need to set the length of shapelets too large in practice.

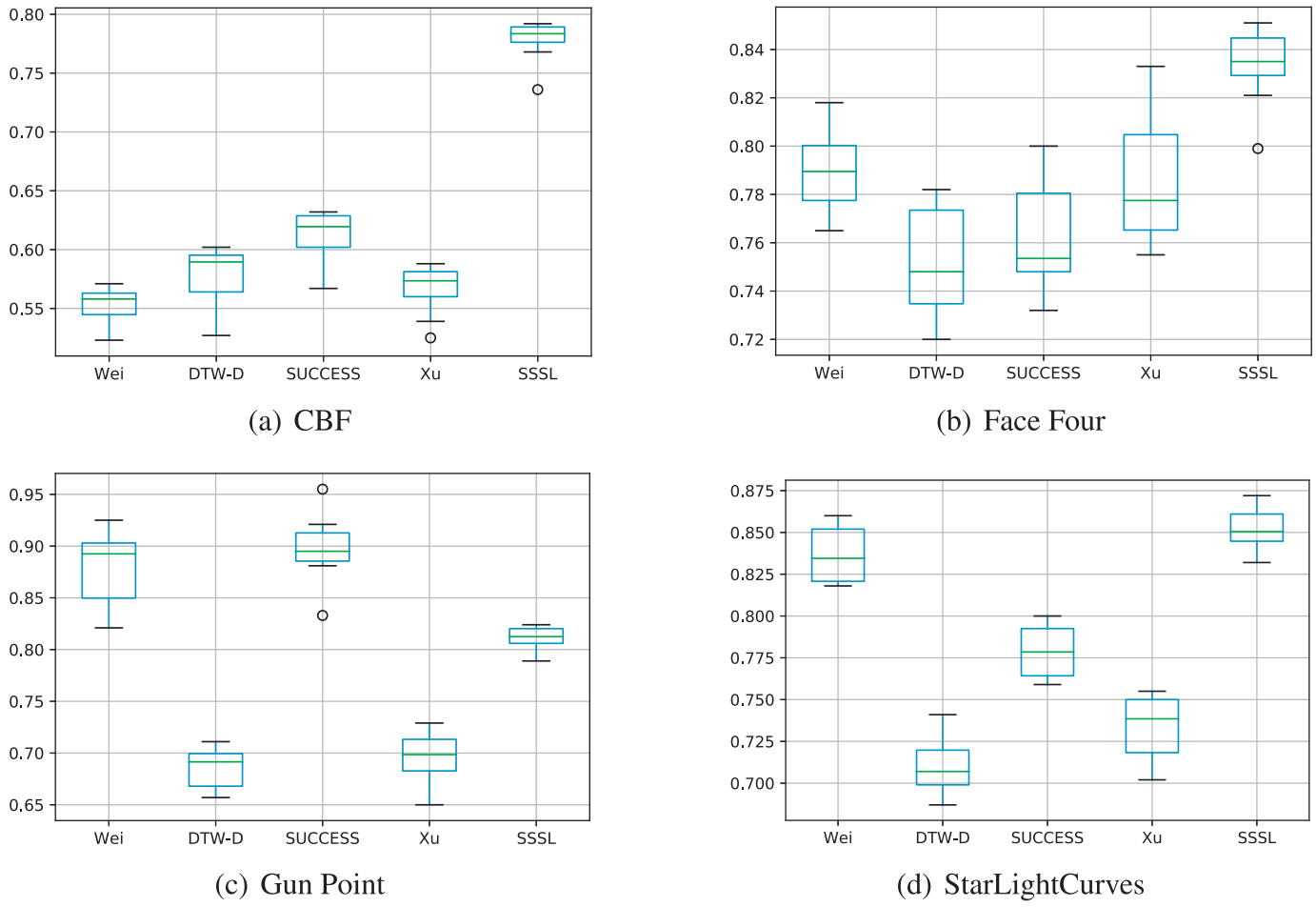


Fig. 3. The boxplot of accuracies of 10 times on the CBF, Face Four, Gun Point and StarLightCurves data sets.

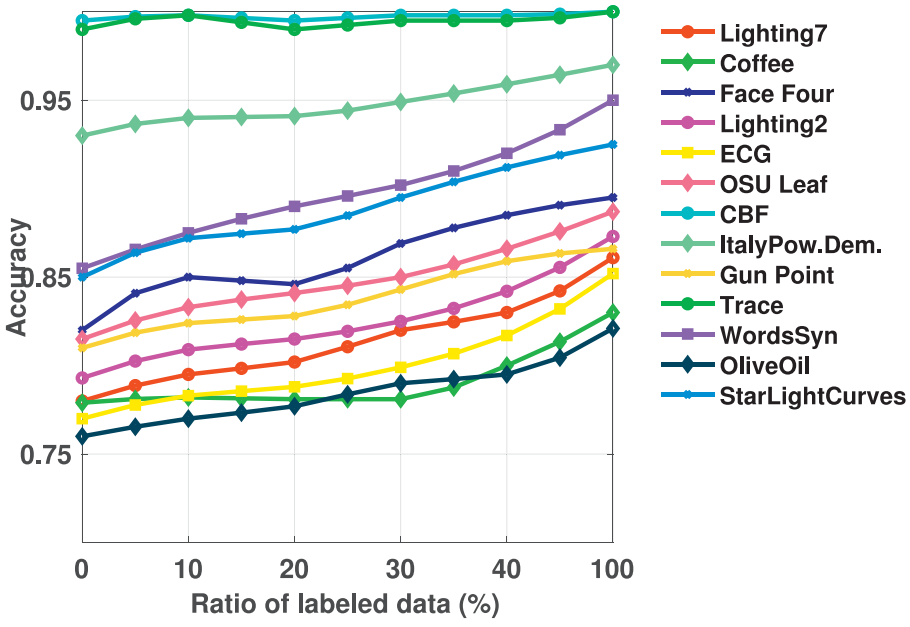


Fig. 4. Classification accuracy of SSSL with respect to various ratios of labeled data.

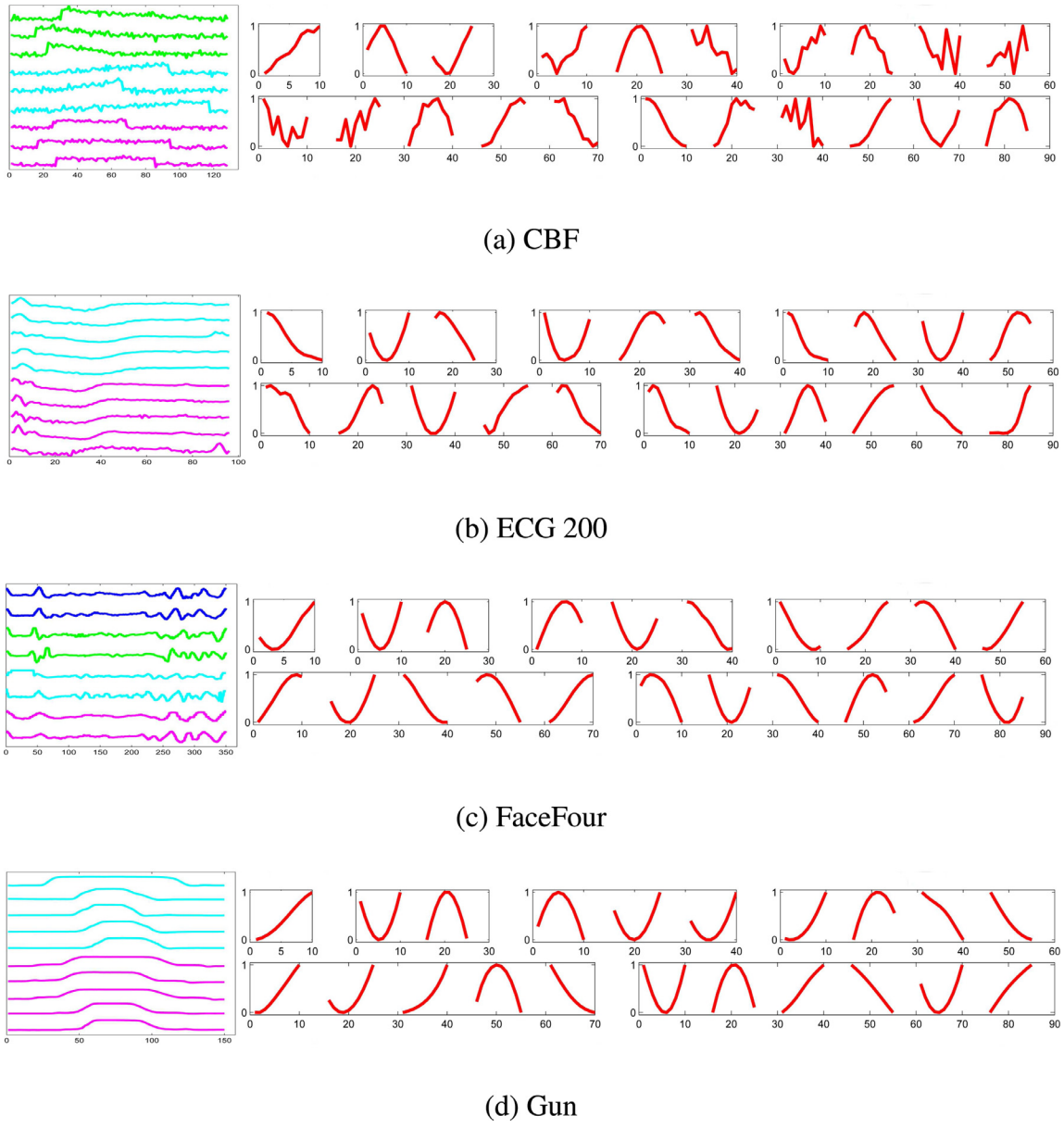
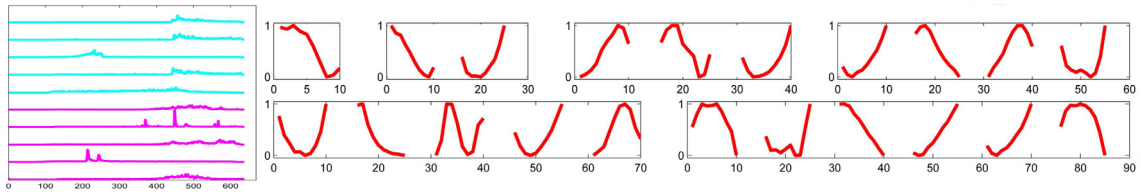


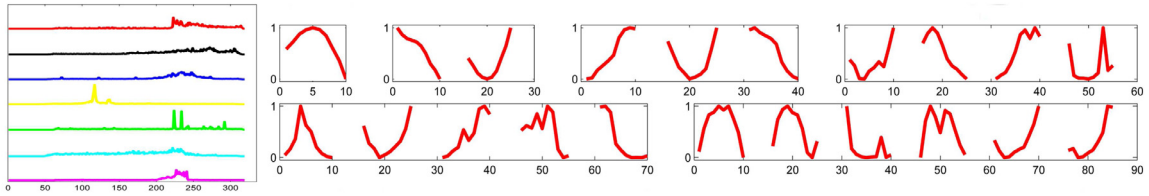
Fig. 5. An illustration of the shapelets learnt by SSSL on the real-world data sets. The left part shows a small portion of time series examples drawn from the original data sets. The right part shows the learnt shapelets. Shapelets in red color are learnt by increasing the number from 1 to 6. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Running time comparisons with respect to the number of shapelets on data of CBF, ECG, FaceFour, ItalyPow.Dem, Lighting2 and Lighting7.

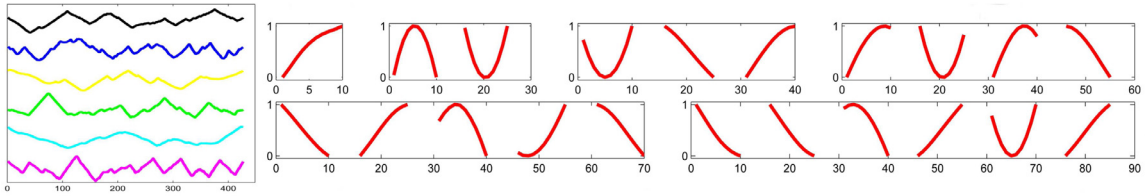
# of Shapelets	Time: mean+std. (seconds)					
	CBF	ECG	FaceFour	ItalyPow.Dem.	Lighting2	Lighting7
2	219.63 ± 28.3	7.05 ± 0.2	17.21 ± 4.1	118.74 ± 0.9	87.77 ± 11.7	23.33 ± 5.3
4	403.61 ± 30.3	11.23 ± 0.1	38.20 ± 1.5	189.90 ± 0.5	153.72 ± 2.3	42.62 ± 4.4
6	577.86 ± 42.7	16.55 ± 0.3	55.89 ± 0.5	277.33 ± 0.2	211.63 ± 7.9	69.65 ± 2.1
8	602.31 ± 31.6	18.67 ± 0.2	78.57 ± 1.2	316.17 ± 0.3	299.22 ± 11.6	85.44 ± 2.0
10	675.11 ± 8.2	20.33 ± 0.4	96.41 ± 4.8	411.90 ± 1.1	354.74 ± 13.3	101.20 ± 3.3
12	699.60 ± 12.1	22.01 ± 0.1	107.22 ± 3.8	496.44 ± 1.3	397.81 ± 8.6	121.80 ± 4.4



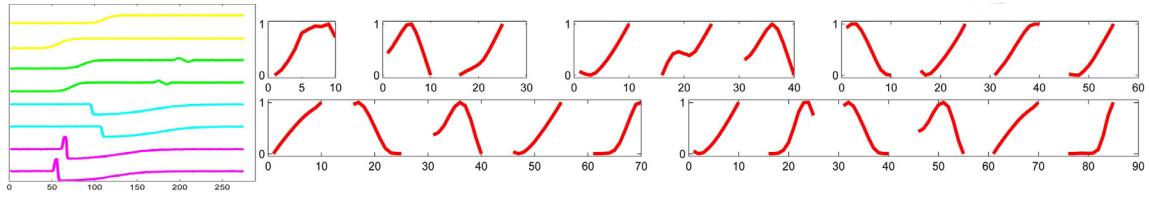
(e) Lighting2



(f) Lighting7



(h) OSU Leaf



(i) Trace

Fig. 5. Continued

Table 5
Running time comparisons with respect to the number of shapelets on data of OSU Leaf, Gun Point, Trace, WordsSyn, Coffee and OliveOil.

# of Shapelets	Time: mean+std. (seconds)					
	OSU Leaf	Gun Point	Trace	WordsSyn	Coffee	OliveOil
2	366.25 ± 33.3	38.20 ± 17.5	10.33 ± 1.3	401.75 ± 22.0	5.22 ± 0.2	6.65 ± 1.0
4	643.60 ± 8.0	51.03 ± 10.33	19.85 ± 1.0	688.55 ± 15.4	9.33 ± 0.1	15.30 ± 2.0
6	876.56 ± 22.0	93.13 ± 2.8	22.60 ± 3.3	902.30 ± 10.33	14.10 ± 0.3	19.23 ± 2.1
8	956.60 ± 3.3	155.63 ± 8.0	25.85 ± 2.1	997.66 ± 11.5	17.75 ± 0.6	23.43 ± 0.9
10	1125.66 ± 3.9	233.33 ± 11.3	28.76 ± 2.0	1112.45 ± 8.6	19.95 ± 0.2	27.66 ± 1.3
12	1228.87 ± 2.0	301.20 ± 1.1	31.41 ± 3.3	1356.43 ± 9.5	21.11 ± 1.0	29.89 ± 2.3

6.7. Run time

We test the run time of SSSL with respect to the number of shapelets m. We vary the parameter m from 2 to 12 with a step size of 2 and repeat experiments ten times. From Tables 4 and 5, we can see that the run time generally increases linearly with respect to the number of shapelets, which means it scales well to large data sets.

7. Conclusion

Time series classification has been a long-standing problem with a large scope of real-world applications [47,48] such as

biomedical engineering and clinical prediction. However, in real world applications, it can be expensive or time-consuming to label data as it may require access to domain experts, whereas unlabeled data is cheap and easy to collect and store. In this paper, we explored a new problem of semi-supervised shapelets learning, where the data contain both labeled and unlabeled time series. An optimization model SSSL by integrating the strength of regularized least-square, shapelets regularization, spectral analysis, and pseudo-label to auto-learn the most discriminative shapelets from labeled and unlabeled time series data. The experiments and comparisons on real-world time series data sets demonstrate that SSSL outperforms state-of-the-art semi-supervised time series classification algorithms on most of the data sets.

Although our algorithm of SSSL over labeled and unlabeled time series achieves high accuracy and meets the need of using both labeled and unlabeled data, some limitations exist that need improvement in future work: 1) we will compare SSSL with other shapelet learning models using advanced time series representation and alternative distance measures instead only using Euclidean distance as a measure; and 2) the proposed algorithm, based on a gradient descent algorithm, is a straightforward solution to solve the objective function, however more state-of-the-art gradient descent algorithms could be applied to this semi-supervised time series learning problem. This work inspires some interesting directions for future research: 1) the problem could be further extended by using deep learning framework, such as tri-training or co-training deep learning framework can be used for semi-supervised time series learning; and 2) the idea could also be used for multivariate time series. For example, the objective function could be extended to learn shapelet feature from both labeled and unlabeled multivariate time series.

Acknowledgements

This work was supported in part by the MQNS under Grant 9201701203, in part by the MQEPS under Grant 9201701455, in part by the MQRSG under Grant 95109718, in part by the National Natural Science Foundation of China under Grant 61702355, in part by the Soft Science Research Project of Anhui Science and Technology Plan under Grant 1607a0202071, and in part by the 2018 Collaborative Research Project between Macquarie University and Data 61.

Appendix A

A1. The derivation of Eq. (12)

The derivatives of Eq. (11) with respect to $\mathbf{Z}$ is

$$\begin{aligned}\nabla\mathcal{F}(\mathbf{Z}) &= \mathbf{Z}\mathbf{L}_G - \lambda_3(\mathbf{W}^\top\mathbf{X}_u - \mathbf{Z}) + \zeta_1\mathbf{Z} - \zeta_2\mathbf{I} \\ &= \mathbf{Z}(\mathbf{L}_G + \lambda_3\mathbf{I} + \zeta_1\mathbf{I}) - \lambda_3\mathbf{W}^\top\mathbf{X}_u - \zeta_2\mathbf{I}\end{aligned} \tag{A.1}$$

where $\mathbf{I}$ is an identity matrix. By making the derivatives equal to 0, we obtain,

$$\mathbf{Z} = (\lambda_3\mathbf{W}^\top\mathbf{X}_u + \zeta_2\mathbf{I})(\mathbf{L}_G + \lambda_3\mathbf{I} + \zeta_1\mathbf{I})^{-1} \tag{A.2}$$

Thus, $\mathbf{Z}$ can be updated as follows,

$$\mathbf{Z}^{t+1} = (\lambda_3\mathbf{W}_t^\top\mathbf{X}_u^t + \zeta_2\mathbf{I})(\mathbf{L}_G^t + \lambda_3\mathbf{I} + \zeta_1\mathbf{I})^{-1} \tag{A.3}$$

A2. The derivation of Eq. (12)

The derivatives of Eq. (13) with respect to $\mathbf{W}$ is

$$\begin{aligned}&\nabla\mathcal{F}(\mathbf{W}) = \lambda_2\mathbf{W} + \lambda_3\mathbf{X}_u(\mathbf{W}^\top\mathbf{X}_u - \mathbf{Z}) + \lambda_4\mathbf{X}_l(\mathbf{W}^\top\mathbf{X}_l - \mathbf{Y}_l) \\ &= \lambda_2\mathbf{W} + \lambda_3\mathbf{X}_u\mathbf{X}_u^\top\mathbf{W} - \lambda_3\mathbf{X}_u\mathbf{Z}^\top + \lambda_4\mathbf{X}_l\mathbf{X}_l^\top\mathbf{W} - \lambda_4\mathbf{X}_l\mathbf{Y}_l^\top \\ &= (\lambda_2\mathbf{I} + \lambda_3\mathbf{X}_u\mathbf{X}_u^\top + \lambda_4\mathbf{X}_l\mathbf{X}_l^\top)\mathbf{W} - \lambda_3\mathbf{X}_u\mathbf{Z}^\top - \lambda_4\mathbf{X}_l\mathbf{Y}_l^\top\end{aligned} \tag{A.4}$$

When setting the derivatives equal to 0, we have,

$$\mathbf{W} = (\lambda_2\mathbf{I} + \lambda_3\mathbf{X}_u\mathbf{X}_u^\top + \lambda_4\mathbf{X}_l\mathbf{X}_l^\top)^{-1}(\lambda_3\mathbf{X}_u\mathbf{Z}^\top + \lambda_4\mathbf{X}_l\mathbf{Y}_l^\top) \tag{A.5}$$

Thus, $\mathbf{W}$ can be updated as follows,

$$\begin{aligned}\mathbf{W}_{t+1} =&(\lambda_2\mathbf{I} + \lambda_3\mathbf{X}_u^t(\mathbf{X}_u^t)^\top + \lambda_4\mathbf{X}_l^t(\mathbf{X}_l^t)^\top)^{-1} \\ &\times(\lambda_3\mathbf{X}_u^t(\mathbf{Z}^t)^\top + \lambda_4\mathbf{X}_l^t(\mathbf{Y}_l^t)^\top)\end{aligned} \tag{A.6}$$

A3. Details of calculating $\nabla\mathbf{S}_i$ in Eq. (16)

As $\mathbf{L}_G = \mathbf{D}_G - \mathbf{G}$ and $\mathbf{D}_G(i,i) = \sum_{j=1}^n \mathbf{G}(i,j)$, the first term in Eq. (16) turns to calculating $\partial\mathbf{G}(i,j)/\partial\mathbf{S}(k,p)$ as in Eq. (A.7),

$$\begin{aligned}\frac{\partial\mathbf{G}(i,j)}{\partial\mathbf{S}(k,p)} =& -\frac{2\mathbf{G}(i,j)}{\sigma^2}\left(\sum_{q=1}^m\left(\mathbf{X}_u(q,i) - \mathbf{X}_u(q,j)\right)\right) \\ &\times\left(\frac{\partial\mathbf{X}_u(q,i)}{\partial\mathbf{S}(k,p)} - \frac{\partial\mathbf{X}_u(q,j)}{\partial\mathbf{S}(k,p)}\right)\end{aligned} \tag{A.7}$$

and

$$\frac{\partial\mathbf{X}(i,j)}{\partial\mathbf{S}(k,p)} = \frac{1}{\Theta^2}\sum_{q=1}^{\bar{q}_{i,j}} e^{\alpha d_{i,j,q}}((1+\alpha d_{i,j,q})\Theta - \alpha\Lambda)\frac{\partial d_{i,j,q}}{\partial\mathbf{S}(k,p)} \tag{A.8}$$

where $\Theta = \sum_{q=1}^{\bar{q}_{i,j}} e^{\alpha d_{i,j,q}}$, $\Lambda = \sum_{q=1}^{\bar{q}_{i,j}} d_{i,j,q}e^{\alpha d_{i,j,q}}$ and $\bar{q}_{i,j} = q_j - l_i + 1$, $d_{i,j,q} = \frac{1}{l_i}\sum_{h=1}^{l_i}(\mathbf{T}_{q+h-1}^j - \mathbf{S}_h^i)^2$.

$$\frac{\partial d_{i,j,q}}{\partial\mathbf{S}(k,p)} = \begin{cases}0 & if \quad i \neq k \\ \frac{2}{l_k}(\mathbf{S}(k,p) - \mathbf{T}_{q+p-1}^j) & if \quad i = k\end{cases} \tag{A.9}$$

The second term in Eq. (16) turns to calculating Eq. (A.10)

$$\frac{\partial\mathbf{H}(i,j)}{\partial\mathbf{S}(k,p)} = -\frac{1}{\sigma^2}\tilde{d}_{i,j}e^{-\frac{1}{\sigma^2}\tilde{d}_{i,j}^2}\frac{\partial\tilde{d}_{i,j}}{\partial\mathbf{S}(k,p)} \tag{A.10}$$

where $\tilde{d}_{i,j}$ is the distance between shapelets $\mathbf{S}_i$ and $\mathbf{S}_j$.

Based on Eqs. (A.7)–(A.10), we can calculate the gradient $\nabla\mathbf{S}_i = \frac{\partial\mathcal{F}(\mathbf{S})}{\partial\mathbf{S}_i}$.

References

[1] H. Wang, J. Wu, P. Zhang, C. Zhang, Temporal feature selection on networked time series, arXiv:1612.06856 (2016).
[2] J. Zhao, L. Itti, Shapedtw: shape dynamic time warping, Pattern Recognit. 74 (2018) 171–184.
[3] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Min. Knowl. Discov. 28 (4) (2014) 851–881.
[4] M. Wistuba, J. Grabocka, L. Schmidt-Thieme, Ultra-fast shapelets for time series classification, arXiv:1503.05018 (2015).
[5] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 947–956.
[6] A. Mueen, E. Keogh, N. Young, Logical-shapelets: an expressive primitive for time series classification, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2011, pp. 1154–1162.
[7] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR Time Series Classification Archive, 2015.
[8] L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data Min. Knowl. Discov. 22 (1–2) (2011) 149–182.
[9] T. Rakthanmanon, E. Keogh, Fast shapelets: A scalable algorithm for discovering time series shapelets, in: Proceedings of the SIAM Conference on Data Mining (SDM), 2013, pp. 668–676.
[10] D. Gordon, D. Hendler, L. Rokach, Fast and space-efficient shapelets-based time-series classification, Intell. Data Anal. 19 (5) (2015) 953–981.
[11] L. Ulanova, N. Begum, E. Keogh, Scalable clustering of time series with u-shapelets, in: Proceedings of the SIAM International Conference on Data Mining (SDM), 192, 2015.
[12] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2014, pp. 392–401.
[13] L. Wei, E. Keogh, Semi-supervised time series classification, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2006, pp. 748–753.
[14] Z. Xu, K. Funaya, Time series analysis with graph-based semi-supervised learning, in: Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA), 2015, pp. 1–6.

[15] H. Wang, J. Wu, Boosting for real-time multivariate time series classification, AAAI, 2017, pp. 4999–5000.
[16] A. Raza, S. Kramer, Ensembles of randomized time series shapelets provide improved accuracy while reducing computational costs, arXiv:1702.06712 (2017).
[17] A. Bostrom, A. Bagnall, Binary Shapelet Transform for Multiclass Time Series Classification, in: Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII, Springer, 2017, pp. 24–46.
[18] E.J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, A. Jaimes, Correlating financial time series with micro-blogging activity, in: Proceedings of the International Conference on Web Search and Data Mining (WSDM), 2012, pp. 513–522.
[19] S. Hirano, S. Tsumoto, Cluster analysis of time-series medical data based on the trajectory representation and multiscale comparison techniques, in: Proceedings of the Sixth IEEE International Conference on the Data Mining (ICDM), 2006, pp. 896–901.
[20] Y. Cai, R. Ng, Indexing spatio-temporal trajectories with chebyshev polynomials, in: Proceedings of the International Conference on Management of Data (SIGMOD), 2004, pp. 599–610.
[21] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7 (3) (2005) 358–386.
[22] J. Zhao, L. Itti, Classifying time series using local descriptors with hybrid sampling, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 623–637.
[23] M.G. Baydogan, G. Runger, E. Tuv, A bag-of-features framework to classify time series, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2796–2802.
[24] M.W. Fakhr, Online nonstationary time series prediction using sparse coding with dictionary update, in: Information and Communication Technology Research (ICTRC), 2015 International Conference on, IEEE, 2015, pp. 112–115.
[25] R. Tavenard, S. Malinowski, L. Chapel, A. Bailly, H. Sanchez, B. Bustos, Efficient temporal kernels between feature sets for time series classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 528–543.
[26] J. Wang, P. Liu, M.F. She, S. Nahavandi, A. Kouzani, Bag-of-words representation for biomedical time series classification, Biomed. Signal Process. Control 8 (6) (2013) 634–644.
[27] L. Ye, E. Keogh, Time series shapelets: A new primitive for data mining, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 947–956.
[28] K.-W. Chang, B. Deka, W.-M. W. Hwu, D. Roth, Efficient pattern-based time series classification on GPU, in: Proceedings of the International Conference on Data Mining (ICDM), 2012, pp. 131–140.
[29] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2014, pp. 392–401.
[30] Q. Zhang, J. Wu, H. Yang, Y. Tian, C. Zhang, Unsupervised feature learning from time series, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2016, pp. 2322–2328.
[31] K. Avrachenkov, P. Chebotarev, A. Mishenin, Semi-supervised learning with regularized laplacian, arXiv:1508.04906 (2015).
[32] Z. Lu, L. Wang, Noise-robust semi-supervised learning via fast sparse coding, Pattern Recognit. 48 (2) (2015) 605–612.
[33] P.K. Mallapragada, R. Jin, A.K. Jain, Y. Liu, Semiboost: boosting for semi–supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2000–2014.
[34] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV), 1, 2005, pp. 29–36.
[35] J. Tanha, M. van Someren, H. Afsarmanesh, Semi-supervised self-training for decision tree classifiers, Int. J. Mach. Learn. Cybern. (2015) 1–16.
[36] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3581–3589.
[37] P. Fox-Roberts, E. Rosten, Unbiased generative semi-supervised learning, J. Mach. Learn. Res. 15 (1) (2014) 367–443.
[38] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning (ICML), 3, 2003, pp. 912–919.
[39] J. Wang, T. Jebara, S.-F. Chang, Semi-supervised learning using greedy max-cut, J. Mach. Learn. Res. 14 (1) (2013) 771–800.
[40] Y. Chen, B. Hu, E. Keogh, G.E. Batista, Dtw-d: time series semi-supervised learning from a single example, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2013, pp. 383–391.
[41] K. Marussy, K. Buza, Success: a new approach for semi-supervised classification of time-series, in: Artificial Intelligence and Soft Computing, 2013, pp. 437–447.
[42] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, l2, 1-norm regularized discriminative feature selection for unsupervised learning, in: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 22, 2011, p. 1589.
[43] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, et al., Unsupervised feature selection using nonnegative spectral analysis., AAAI, 2012.
[44] J. Lines, L.M. Davis, J. Hills, A. Bagnall, A shapelet transform for time series classification, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2012, pp. 289–297.
[45] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[46] Z. Zhao, H. Liu, Semi-supervised feature selection via spectral analysis., in: Proceedings of the SIAM Conference on Data Mining (SDM), SIAM, 2007, pp. 641–646.
[47] K.Ø. Mikalsen, F.M. Bianchi, C. Soguero-Ruiz, R. Jenssen, Time series cluster kernel for learning similarities between multivariate time series with missing data, Pattern Recognit. 76 (2018) 569–581.
[48] K.S. Tuncel, M.G. Baydogan, Autoregressive forests for multivariate time series modeling, Pattern Recognit. 73 (2018) 202–215.

Haishuai Wang received the Ph.D. degree in computer science from the Center of Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Fairfield University, Fairfield, CT, USA. He is also a Visiting Assistant Professor with the Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. His research interests include data mining, machine learning, and applications on bioinformatics and social networks.

Qin Zhang is currently a senior researcher with Cloud and Smart Industries Group (CSIG), Tencent, China. She got her doctoral degree from University of Technology Sydney, Australia in 2018. And she received her master's degree from the University of Chinese Academy of Sciences, China in 2014. Her main research interests include sequence data learning, natural language processing, and network analysis by using various deep learning and optimization methods. She has published several qualified research papers in top journals and top conferences, including IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Knowledge and Data Engineering (TKDE) etc., and International Joint Conferences on Artificial Intelligence (IJCAI), IEEE International Conference on Data Mining (ICDM) and so on. She also served as a reviewer (sub-reviewer) for KDD, NIPS, ICDM, IJCAI, AAAI and SDM etc.

Jia Wu received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Lecturer in the Department of Computing, Macquarie University, Sydney. Prior to that, he was with the Centre for Artificial Intelligence, University of Technology Sydney. His current research interests include data mining and machine learning. Since 2009, he has published 100+ refereed journal and conference papers, including IEEE *Transactions on Pattern Analysis and Machine Intelligence* (TPAMI), IEEE *Transactions on Knowledge and Data Engineering* (TKDE), IEEE *Transactions on Neural Networks and Learning Systems* (TNNLS), ACM *Transactions on Knowledge Discovery from Data* (TKDD), *International Joint Conference on Artificial Intelligence* (IJCAI), AAAI *Conference on Artificial Intelligence* (AAAI), IEEE *International Conference on Data Mining* (ICDM), and SIAM *International Conference on Data Mining* (SDM). Dr Wu was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM14 Best Paper Candidate Award. He is the Associate Editor of the ACM *Transactions on Knowledge Discovery from Data* (TKDD), *Journal of Network and Computer Applications* (JNCA) and *Neural Networks* (NN).

Shirui Pan received the Ph.D. degree in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is currently a Lecturer with the Faculty of Information Technology, Monash University, Australia. Prior to that, he was a Lecturer with the School of Software, University of Technology Sydney. His research interests include data mining and machine learning. To date, Dr Pan has published over 50 research papers in top-tier journals and conferences, including the IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Cybernetics (TCYB), ICDE, AAAI, IJCAI, and ICDM.

Yixin Chen received his Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign (UIUC) in 2005. He is a Professor of Computer Science and Engineering at the WashingtonUniversity in St. Louis. His research interestsinclude data mining, machine learning, artificial intelligence, and optimization. He received the Best Paper Award at the IDEAL Conference (2016), Distinguished Paper Award at the AMIA Conference (2015), Best Student Paper Runner-up Award at the ACM SIGKDD Conference (2014), Best Paper Award at the AAAI Conference on Artificial Intelligence (2010), and the IEEE International Conference on Tools for AI (2005). He also received Best Paper Award nominations at IEEE ICDM (2013), IEEE RTAS (2012), and KDD (2009). His work on planning has won First Prizes in the InternationalPlanning Competitions (2004 & 2006). He received an Early Career Principal Investigator Award from the Department of Energy (2006) and a Microsoft Research New Faculty Fellowship (2007). His research has been funded by NSF, NIH, DOE, Microsoft, and Memorial Sloan-Kettering Cancer Center. He is an Associate Editor for ACM Transactions of Intelligent Systems and Technology, Annals of Mathematics and Artificial Intelligence, and Journal of Artificial Intelligence Research. He was an Associate Editor for IEEE Transactions on Knowledge and Data Engineering from 2008 to 2012.