

# 面向复杂时间序列的 $k$ 近邻分类器<sup>\*</sup>

原继东<sup>1,2</sup>, 王志海<sup>1,2</sup>, 孙艳歌<sup>1,2</sup>, 张伟<sup>1,2</sup>

<sup>1</sup>(北京交通大学 计算机与信息技术学院, 北京 100044)

<sup>2</sup>(交通数据分析与挖掘北京市重点实验室(北京交通大学), 北京 100044)

通讯作者: 王志海, E-mail: zhhwang@bjtu.edu.cn



**摘要:** 基于时序对齐的  $k$  近邻分类器是时间序列分类的基准算法. 在实际应用中, 同类复杂时间序列经常展现出不同的全局特性. 由于传统时序对齐方法平等对待实例特征并忽略其局部辨别特性, 因此难以准确、高效地处理此类具有挑战性的时间序列. 为了有效对齐并分类复杂时间序列, 提出了一种具有辨别性的局部加权动态时间扭曲方法, 用于发现同类复杂时间序列的共同点以及异类序列间的不同点. 同时, 通过迭代学习时间序列对齐点的正例集与负例集, 获取每条复杂时间序列中每个特征的辨别性权重. 在多个人工和真实数据集上的实验结果表明了基于局部加权对齐策略的  $k$  近邻分类器所具有的可解释性与有效性, 并将所提出方法扩展至多变量时间序列分类问题中.

**关键词:** 复杂时间序列;  $k$  近邻; 局部加权; 动态时间扭曲

中图法分类号: TP311

中文引用格式: 原继东, 王志海, 孙艳歌, 张伟. 面向复杂时间序列的  $k$  近邻分类器. 软件学报, 2017, 28(11):3002–3017. <http://www.jos.org.cn/1000-9825/5331.htm>

英文引用格式: Yuan JD, Wang ZH, Sun YG, Zhang W.  $K$ -Nearest neighbor classifier for complex time series. Ruan Jian Xue Bao/Journal of Software, 2017, 28(11):3002–3017 (in Chinese). <http://www.jos.org.cn/1000-9825/5331.htm>

## K-Nearest Neighbor Classifier for Complex Time Series

YUAN Ji-Dong<sup>1,2</sup>, WANG Zhi-Hai<sup>1,2</sup>, SUN Yan-Ge<sup>1,2</sup>, ZHANG Wei<sup>1,2</sup>

<sup>1</sup>(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

<sup>2</sup>(Beijing Key Laboratory of Traffic Data Analysis and Mining (Beijing Jiaotong University), Beijing 100044, China)

**Abstract:** Temporal alignment based  $k$ -nearest neighbor classifier is a benchmark for time series classification. Since complex time series generally exhibit different global behaviors within classes in real applications, it is difficult for standard alignment, where features are treated equally while local discriminative behaviors are ignored, to handle these challenging time series correctly and efficiently. To facilitate aligning and classifying such complex time series, this paper proposes a discriminative locally weighted dynamic time warping dissimilarity measure that reveals the commonly shared subsequence within classes as well as the most differential subsequence between classes. Meanwhile, time series alignments of positive and negative subsets are employed to learning discriminative weight for each feature of each time series iteratively. Experiments performed on synthetic and real datasets demonstrate that this locally weighted, temporal alignment based  $k$ -nearest neighbor classifier is effective in differentiating time series with good interpretability. Extension of the proposed weighting strategy to multivariate time series is also discussed.

**Key words:** complex time series;  $k$ -nearest neighbor; locally weighted; dynamic time warping

时序对齐, 包括基于欧氏距离(euclidean distance, 简称 ED)或动态时间扭曲(dynamic time warpping, 简称

\* 基金项目: 国家自然科学基金(61672086, 61702030); 中央高校基本科研业务费专项资金(2016RC048, 2017YJS036)

Foundation item: National Natural Science Foundation of China (61672086, 61702030); Fundamental Research Funds for the Central Universities (2016RC048, 2017YJS036)

本文由复杂环境下的机器学习研究专刊特约编辑胡清华教授推荐.

收稿时间: 2016-12-22; 修改时间: 2017-04-11, 2017-06-05; 采用时间: 2017-08-23

DTW)的时序对齐策略,是一种根据时间序列相似性或非相似性度量来连接或匹配时间序列,提供时序度量的方法.它在时间序列分类、聚类以及分析等领域起着至关重要的作用<sup>[1-4]</sup>,被广泛应用于姿势识别<sup>[3,5-7]</sup>、手写字符识别<sup>[8]</sup>、音乐信息校准<sup>[9,10]</sup>、语音识别<sup>[11,12]</sup>、基因序列比对<sup>[13]</sup>等领域.

基于 ED 的对齐策略是一种线性映射,而 DTW 是非线性映射,它用于寻找两个序列之间的最优对齐方式,基于 DTW 的  $k$  近邻分类器( $k$ -nearest neighbor,简称 kNN)也是当前用于时间序列分类的最好算法之一,它已被成功地应用于分类那些类内具有全局相似性,并有较大范围时间延迟的时间序列.但是在某些实际应用中,DTW 不能有效处理那些类内具有不同全局特性而类间具有相似全局特性的复杂的时间序列,其原因主要在于以下两个方面:(1) 传统 DTW 是一种按照逐对时间序列来逐步比较的对齐策略(pairwise alignment),它忽视了同类或异类时间序列间的关联性;(2) DTW 赋予对齐序列的每一维度同样的权值,并且对齐过程未能有效利用其他数据分析方法(如聚类、分类等),因此削弱了其处理复杂时间序列数据的能力.

例如,图 1 展示了真实电力消费数据的样本,此数据记录了某个家庭中几乎一年(共 349 天)的电力消耗情况.其中,每条时间序列均记录了某一天的电力消耗情况(每 10min 采样一次,长度为 144).若其在耗电高峰期(如晚 6 点~8 点,图中范围[108,120])的耗电量低于年平均消耗水平,类标记为 Low;相应的类标记为 High 的时间序列表示其在用电高峰期的耗电量高于年平均水准.从图中可以很明显地观察到:对于 Low 与 High 两类复杂时间序列,同类耗电量序列数据间的全局特性有很大的不同,两个类别的主要区别在于[108,120]间的子序列,而传统 DTW 无法找到不同类别间的局部辨别性特征.

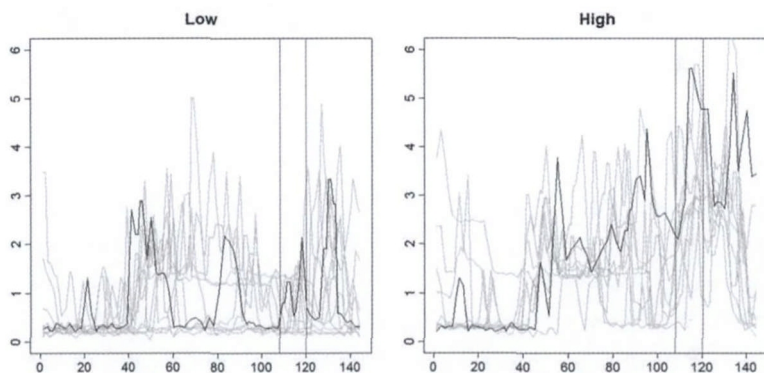


Fig.1 Electrical power consumption of Low and High classes

图 1 Low 和 High 类别的电力消耗数据

近年来,研究者已提出了多种 DTW 的改进方法,并将其应用到时间序列分类或聚类问题中.它们大多集中于定义更好的限制策略、全局的或半全局的加权方法等.例如,第 1 种改进方式主要依赖于 Sakoe-Chiba<sup>[11]</sup>或 Itakura<sup>[14]</sup>来限制 DTW 的对齐空间,或者通过预处理时间序列,比如获取高阶形状特征<sup>[15]</sup>、平均化时间序列<sup>[16]</sup>、通过多层次方法近似求解 DTW 最优路径<sup>[17]</sup>等,加快 DTW 的计算效率.第 2 种改进方式主要集中于定义全局的加权策略.比如,Jeong 等人<sup>[18]</sup>提出了一种基于 Sakoe-Chiba 限制的加权策略,其根据 DTW 代价矩阵定义了一种递增函数,用于赋予离对角线近的点较低的权值、离对角线远的点较高的权值.另一个例子来自姿势识别领域,Reyes 等人<sup>[6]</sup>提出了一种根据 DTW 均值来计算类内和类间的方差,并将所得权值赋予所有姿势类别的加权策略.第 3 种方式是一种半全局加权策略<sup>[5,19]</sup>,它也是 Reyes 等人<sup>[6]</sup>所提出的算法的扩展,其主要区别在于为每一个姿势类别的时间序列赋予不同的权值.

上述工作能够产生比较准确的时序对齐方式,但仍存在缺点,比如赋予所有类别或时间点(time stamp)相同的权值(全局加权策略);缺乏对类内以及类间局部不同行为特性的考虑;辨别性子序列通过标准 DTW 学得,而 DTW 平等地对待每一个特征;权值并没有随着时间的推进而改变等.在实际复杂时间序列中,很容易见到时间序列的某一片段明显区别于其他地方的情况,所以不同的时间点应赋予不同的权值.换言之,为发现时间序列的局部辨别性特征,需要定义一种局部加权策略,赋予每一条时间序列的每一个时间点不同的权值.Frambourg 等

人<sup>[2]</sup>对此具有挑战性的问题进行了初步讨论.它依赖于方差/协方差矩阵来增强或削弱对齐关系,但由于其需要大量的矩阵运算,导致其在面对大规模数据时效率大幅下降.

尽管基于时序对齐的 kNN 是时间序列分类领域最成功以及最广泛使用的分类器,但 kNN 对噪音敏感, $k$  值的变化很容易引起已预测类标的变化,特别是当异类的数据间具有相同的全局特征时,此情况更易发生.另外,由于 kNN 是一个懒惰式的分类器,它不像贝叶斯网络或者决策树那样具有明显的可解释性.

为了解决上述问题,本文提出一种具有辨别性的基于局部加权 DTW 的时间对齐方式,并将其应用于 kNN 中.所提出的加权方法通过迭代学习最近的同类和异类序列的多步对齐(multiple alignments)来获取辨别性特征.它能够在增大类间方差的同时降低类内方差,并提高分类准确率.主要工作可总结为以下几点.

- (1) 提出一种有效的时间序列局部加权模型,为每一条时间序列的时间点提供权值,并通过属性局部权重发现辨别性子序列;
- (2) 通过将所提出的算法和其他基于非相似性度量的 kNN 分类器以及基于 Shapelets 的决策树相比较,展示了其可解释性;
- (3) 将所提出的模型扩展至多变量时间序列分类问题,并讨论了其特殊情况,即加权 ED 的有效性;
- (4) 通过在多个复杂时间序列上与多个方法的对比,展现所提出模型在时间序列分类问题上的准确性.

本文第 1 节介绍序列对齐的背景知识.基于 DTW 的时间序列局部加权模型及其学习算法将在第 2 节和第 3 节详细介绍.第 4 节通过实验评价所提出模型在多个数据集(包括单变量和多变量时间序列)上的分类效果.第 5 节通过在多个人工数据集和真实数据集上的实例解析,阐释所提出模型的可解释性.最后进行本文工作总结.

## 1 背景知识

定义 1(时序对齐). 对于一个包含有  $N$  个单变量离散时间序列的数据集  $X=\{x_1, x_2, \dots, x_N\}$ , 每一条时间序列  $x=\{x_1, x_2, \dots, x_T\}$  的长度为  $T$ . 两个时间序列  $x_i$  与  $x_j$  之间的一个对齐排列  $\pi$  的长度为  $|\pi|=m(T-m+1)$ , 它是由  $x_i$  与  $x_j$  的对齐点所组成的, 即  $\pi=((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), \dots, (\pi_1(m), \pi_2(m)))$ , 其中, 定义在 1 到  $m$  的成对  $\pi_1$  和  $\pi_2$  遵循以下限制边界和单调性条件.

- (1)  $1=\pi_1(1) \leq \pi_1(2) \leq \dots \leq \pi_1(m)=T$ ;
- (2)  $1=\pi_2(1) \leq \pi_2(2) \leq \dots \leq \pi_2(m)=T$ ;
- (3)  $\forall l \in \{1, \dots, m-1\}, \pi_1(l+1) \geq \pi_1(l)+1, \pi_2(l+1) \geq \pi_2(l)+1, (\pi_1(l+1)-\pi_1(l))+(\pi_2(l+1)-\pi_2(l)) \leq 1$ .

如图 2 所示, 基于 ED 的时序对齐是一对一映射, 两条时间序列  $x_i$  与  $x_j$  的 ED 时序对齐如矩阵的对角线所示. 而基于 DTW 的时序对齐可以是一种一对多的非线性映射, 比如  $x_i$  的第 13 个采样点可以与  $x_j$  的第 14 个~17 个这 4 个点相关联, 其映射的起点与终点固定, 分别为  $(1,1)$  和  $(T,T)$ .

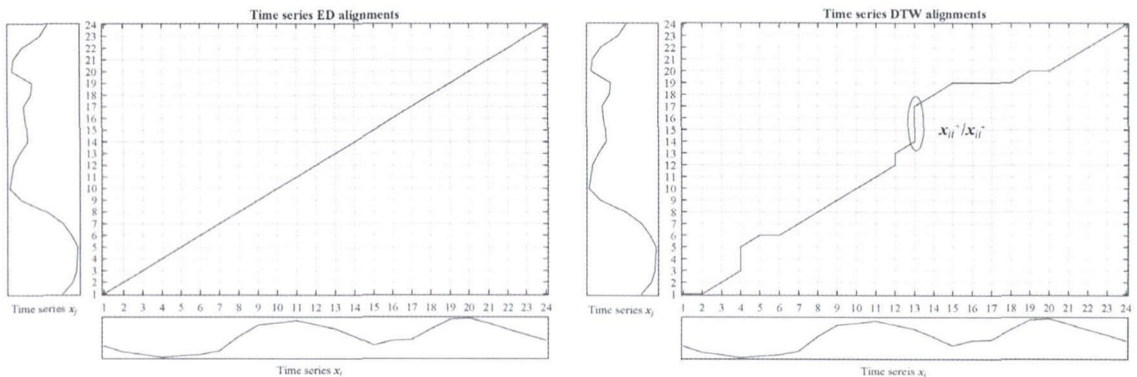


Fig.2 Time series ED/DTW alignments

图 2 时间序列 ED/DTW 对齐

另外,定义  $A$  为两个序列间所有对齐排列的组合, $y$  为时间序列  $x$  的类标.本文中所涉及的符号见表 1.

Table 1 Symbol table

表 1 符号表

符号	符号释义
$X$	时间序列数据集
$N$	数据集中时间序列的条数
$T$	时间序列的长度
$x/x_i/x_j$	一条离散时间序列
$w/w_i$	时间序列权重向量
$\pi$	两条时间序列间的对齐点集合
$\Pi$	对齐点集合 $\pi$ 的集合
$m$	对齐点的个数
$y$	时间序列 $x$ 的类标
$A$	两条时间序列间的所有对齐排列
$t/t'$	时间点
$c$	时间序列 $x$ 的最近邻的个数
$cNN^+ / cNN_i^+$	$x/x_i$ 的正例集合
$cNN^- / cNN_i^-$	$x/x_i$ 的负例集合
$x_t^+$	$x_i$ 在时间点 $t$ 的正对齐集合
$x_t^-$	$x_i$ 在时间点 $t$ 的负对齐集合
$q$	多维时间序列每个点包含的数值的个数
$\alpha$	影响加权机制的控制参数
$min$	训练集中不同类别实例的最小个数

2 时间序列局部加权模型

本文介绍的局部加权模型综合了 Frambourg 等人介绍的方差/协方差对齐模型和 Weinberger 等人提出的最大化边际度量学习算法的优点<sup>[20]</sup>.为了计算每一条时间序列每一个采样点的权值,对每一个  $x$ ,计算其两个集合  $cNN^+$  和  $cNN^-$  (分别称为正例集和负例集),它们分别由  $x$  的  $c$  个距离最近的同类别时间序列和不同类别时间序列组成.如图 3 所示, $cNN^+$  和  $cNN^-$  都是从训练集中得到的.

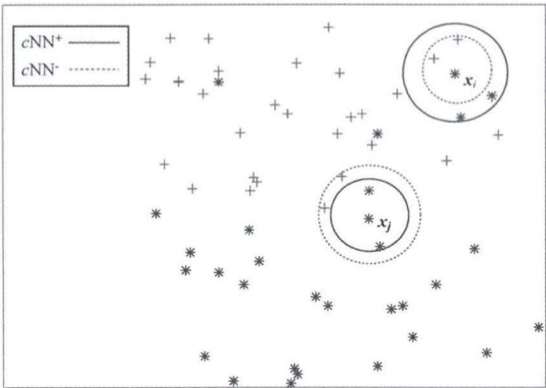


Fig.3 For time series  $x_i$  or  $x_j$  (star),  $cNN^+$  is composed of 2 ( $c=2$ ) closest series of the same class surrounded by circle, while  $cNN^-$  is composed of 2 closest series of the different class (cross) surrounded by dashed circle

图 3 对于时间序列  $x_i$  或  $x_j$ (星号), $cNN^+$  由实线圆环中 2 个( $c=2$ )最近的序列组成, 而  $cNN^-$  由虚线圆环中两个不同类别的十字型组成

注意:本文中定义的正例集和负例集不同于 Weinberger 等人提出的目标集(targets)和入侵集(impostors),其

中,入侵集指的是侵入到目标集范围内的不同类别的实例,而正例集和负例集之间不存在此限制.相应地,对时间序列  $x_i$  的每一个采样点  $t$ ,其对应于两个集合  $x_{it}^+$  和  $x_{it}^-$ , 定义为

$$x_{it}^+ = \{t'/(t, t') \in \pi_{it}, x_{i't'} \in \text{cNN}_i^+\},$$

$$x_{it}^- = \{t'/(t, t') \in \pi_{it}, x_{i't'} \in \text{cNN}_i^-\},$$

其中,  $\pi_{it}$  是  $x_i$  和  $x_{i'}$  之间的标准 DTW 对齐排列,  $\text{cNN}_i^+$  和  $\text{cNN}_i^-$  是  $x_i$  的正例集和负例集.根据图 2 所示的 DTW 时序对齐可知:若  $x_i$  与  $x_j$  的类标相同,则对于时间点  $x_{13}$ ,其  $x_{it}^+ = \{14, 15, 16, 17\}$ ;若两条时间序列类标不同,则  $x_{it}^- = \{14, 15, 16, 17\}$ .

对于每一个  $x_i$ ,赋予其一个权值向量  $w_i = (w_{i1}, w_{i2}, \dots, w_{iT})$ ,此向量根据  $x_i$  中每个采样点对拉近正例集和远离负例集的贡献而确定.所以,此处定义  $w_{it}$  为

$$w_{it} = \frac{A_{it}}{\sum_{t'=1}^T A_{it'}}, w_{it} > 0, \sum_{t=1}^T w_{it} = 1 \quad (1)$$

其中,

$$A_{it} = \frac{\sum_{x_{i'} \in \text{cNN}_i^-} \frac{1}{|\pi_{it'}|} \left( \sum_{(t, t' \in \pi_{it'})} (x_{it} - x_{i't'})^2 \right)}{\sum_{x_{i'} \in \text{cNN}_i^+} \frac{1}{|\pi_{it'}|} \left( \sum_{(t, t' \in \pi_{it'})} (x_{it} - x_{i't'})^2 \right)} \quad (2)$$

$|\pi_{it}|$  是  $x_{it}$  对齐点的个数,除以  $|\pi_{it}|$  意味着获取正例集和负例集对齐点的均值.对于公式(2)而言,分母可认为是  $x_{it}$  的类内方差,而分子是类间方差.由于此模型的目标是在最大化类间方差的同时缩小类内方差,  $A_{it}$  或者  $w_{it}$  越大,  $x_{it}$  的辨别性就越高.

尽管每一个  $x_{it}$  都至少拥有 1 对正例集或负例集的对齐点(当正例集存在时),当所有的  $x_{i't'}$  都与  $x_{it}$  相等时,公式(2)的分母将变为 0,从而公式(2)变得无意义.为了避免这种情况的发生,公式(2)被修改为

$$A_{it} = \frac{\sum_{x_{i'} \in \text{cNN}_i^-} \frac{1}{|\pi_{it'}|} \left( \sum_{(t, t' \in \pi_{it'})} (x_{it} - x_{i't'})^2 \right)}{\sum_{x_{i'} \in \text{cNN}_i^+} \frac{1}{|\pi_{it'}|} \left( \sum_{(t, t' \in \pi_{it'})} (x_{it} - x_{i't'})^2 \right) + \varepsilon} \quad (3)$$

其中,  $\varepsilon$  是一个非常小的数,比如  $10^{-6}$ .

## 2.1 局部加权DTW

DTW 是最为广泛使用的时间序列度量方法之一.它最简单的定义形式如下:

$$DTW(x_i, x_{i'}) := \min_{\pi \in \mathcal{A}} \left( \sum_{(t, t' \in \pi_{it'})} \|x_{it} - x_{i't'}\|^2 \right) \quad (4)$$

其中,  $x_i$  的每一个点都有相同的权重  $1/T$ .

本文介绍的局部加权 DTW(locally weighted dynamic time warping,简称 LDTW)非相似性度量方法计算加权序列  $(x_i, w_i)$  与  $x_{i'}$  之间的距离:

$$LDTW((x_i, w_i), x_{i'}) := \min_{\pi \in \mathcal{A}} \left( \sum_{(t, t' \in \pi_{it'})} f(w_{it}) \|x_{it} - x_{i't'}\|^2 \right) \quad (5)$$

其中,  $\|x_{it} - x_{i't'}\|^2$  为两者间的欧式距离.尽管存在一些高级的函数来估计  $w_{it}$  的贡献,为不失一般性和简洁性,此处将  $f(w_{it})$  定义为如下所示的指数函数或者幂函数:

$$f(w) = e^{-\alpha w} \quad (6)$$

$$f(w) = w^{-\alpha} \quad (7)$$

对于公式(6)和公式(7),  $\alpha \in \mathbf{R}^+$  控制加权机制的影响,并且  $\alpha=0$  对应于标准的、未加权的、均匀的 DTW.之所以用负号,是为了保证优先选择最重要的时间点,而且 LDTW 是为了最小化所有时间序列时间点的值之间的非相似性.因此, LDTW 是 DTW 的一种更一般化的表示,其对齐排序符合定义 1 中关于时序对齐的限制边界和单

调性条件的描述,并且与图 2 中基于 DTW 的时序对齐类似。

## 2.2 多变量时间序列

对于多变量时间序列,每一条时间序列  $\mathbf{x}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{iT}\}$  包含  $T$  个  $q$  维的数值变量,其中  $\mathbf{x}_{it} \in \mathbb{R}^q$  并且  $\mathbf{x}_{it} = \{x_{it}^1, \dots, x_{it}^q\}^T$ 。尽管每条时间序列的长度  $T$  可能不同,但基于 DTW 的度量方式能够应对。为了将本文提出的 LDTW 方法应用于多变量时间序列,需要为每一条时间序列的每一个时间点的每一个数值变量计算权值,针对多变量时间序列的 DTW 被定义为

$$DTW(\mathbf{x}_i, \mathbf{x}_{i'}) := \min_{\pi \in \mathcal{A}} \left( \sum_{k=1}^q \sum_{(t^k, t'^k) \in \pi_{ii'}^k} \|x_{it}^k - x_{i't'}^k\|^2 \right) \quad (8)$$

因此,局部加权后的公式(8)可被表示为

$$LDTW((\mathbf{x}_i, \mathbf{w}_i), \mathbf{x}_{i'}) := \min_{\pi \in \mathcal{A}} \left( \sum_{k=1}^q \sum_{(t^k, t'^k) \in \pi_{ii'}^k} f(w_{it}) \|x_{it}^k - x_{i't'}^k\|^2 \right) \quad (9)$$

其中,  $\mathbf{w}_i$  从  $1 \times T$  维向量(单变量时间序列)转变为  $q \times T$  维矩阵(多变量时间序列),相似的变化也发生在  $\pi_{ii'}$  上。

## 2.3 局部加权ED

对于基于 ED 的 kNN 分类器,两个时间序列  $\mathbf{x}_i$  和  $\mathbf{x}_j$  之间存在一一对应的关系。所以,长度为  $T$  的对齐排列  $\pi$  定义为  $T$  对  $\mathbf{x}_i$  和  $\mathbf{x}_j$  之间的对齐元素:  $\pi = ((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), \dots, (\pi_1(T), \pi_2(T)))$ 。与 DTW 的定义相比,很容易发现基于 ED 的度量是 DTW 的一个特例。实际上,DTW 代价矩阵的对角线就代表了 ED 度量(如图 2 所示),因此很容易将 LDTW 的思想应用到局部加权的 ED 中(locally weighted euclidean distance,简称 LED)。

$$LED((\mathbf{x}_i, \mathbf{w}_i), \mathbf{x}_{i'}) := \sum_{t=1}^T f(w_{it}) \|x_{it} - x_{i't'}\|^2 \quad (10)$$

$w_{it}$  的定义与公式(1)相同,而  $A_{it}$  被修正为

$$A_{it} = \frac{\sum_{x_{i'} \in cNN_i} (x_{it} - x_{i't'})^2}{\sum_{x_{i'} \in cNN_i} (x_{it} - x_{i't'})^2 + \varepsilon}, \varepsilon > 0 \quad (11)$$

## 3 基于 LDTW 的 kNN 算法

为方便起见,本文只介绍在单变量时间序列中学习局部加权 LDTW 和相应的 kNN 分类器。对于基于 LDTW 的 kNN 分类器,需要找到每一条测试实例  $\mathbf{x}_j (1 \leq j \leq N_{test})$  的  $k$  个根据公式(5)所得的距离最近的时间序列。在算法 1 中,同类/异类时间序列的个数  $c$  将会在下一节进行讨论(第 1 行)。LDTW 算法和标准 DTW 算法之间的主要不同在于:前者需要根据算法 2 计算每一个  $\mathbf{x}_i$  的权值  $\mathbf{w}_i$  (第 2 行~第 4 行);后者直接寻找最近邻,不考虑权重的不同。

**算法 1.**  $LDTW(X_{train}, X_{test})$ 。

输入:训练集  $X_{train}$ ,测试集  $X_{test}$ 。

输出:错误率  $errorRate$ 。

- 1: 初始化:读取训练集  $X_{train}$ ,测试集  $X_{test}$ ;  $i=1 (1 \leq i \leq N_{train}), j=1 (1 \leq j \leq N_{test})$ ;  
最近同类或不同类序列的个数  $c$  以及最近邻的个数  $k$ 。
- 2: **for**  $\mathbf{x}_i, 1 \leq i \leq N_{train}$
- 3:   根据算法 2 更新  $\mathbf{x}_i$  的权重  $\mathbf{w}_i$
- 4: **end for**
- 5: **for**  $\mathbf{x}_j, 1 \leq j \leq N_{test}$
- 6:   根据公式(5)寻找  $\mathbf{x}_j$  的  $k$  近邻
- 7:   预测的  $\mathbf{x}_j$  类标
- 8: **end for**

9: 根据真实类标计算错误率  $errorRate$

10: **return**  $errorRate$

算法 2 用于学习每一条训练序列的辨别性权重  $w$ , 初始时  $w^0 = (1/T, \dots, 1/T)$  意味着每一个特征或者时间点的权重相同, 所以  $\Pi^{*(0)}$  是根据标准 DTW 获得的最优对齐排列 (第 1 行~第 3 行). 随着  $w^p$  的不断更新 (第 9 行), 权重变得不均匀, 并且  $\Pi^{*(p)}$  变为根据 LDTW 获得的最优对齐排列 (第 7 行). 需要注意的是, 为得到最优排列  $\Pi^{*(p)}$ , 每一次迭代都需要根据  $w^p$  和  $c$  找到  $x$  的  $cNN^+$  和  $cNN^-$  以及相应的对齐点集合. 该算法不断迭代, 直到  $w^p$  保持不变或者  $w^p$  和  $w^{p-1}$  非常接近为止 (具体停止标准将在第 4.2 节进行讨论).

算法 2.  $WeightLearn(c, x, X_{train})$ .

输入:  $c$ , 正、负例集中序列个数;  $x$ , 一条训练实例;  $X_{train}$ , 训练集.

输出: 局部权重  $w^p$ .

1: 根据公式 (4) 计算  $x$  与  $X_{train}$  中实例间的标准 DTW 距离, 获取每对时间序列间的 DTW 对齐方式  $\Pi^{*(0)}$

2: 根据标准 DTW 距离和  $\Pi^{*(0)}$  寻找  $cNN^+$ ,  $cNN^-$  以及相应的对齐点集合

3: 根据公式 (1) 计算  $w^0$

4:  $p=0$

5: **repeat**

6:      $p=p+1$

7:     根据公式 (5) 和  $w^{p-1}$  计算实例间的 LDTW 距离, 获取每对时间序列间的 LDTW 对齐方式  $\Pi^{*(p)}$

8:     根据 LDTW 距离和  $\Pi^{*(p)}$  寻找  $cNN^+$ ,  $cNN^-$  以及相应的对齐点集合

9:     根据公式 (1) 更新  $w^p$

10: **until**  $w^p \approx w^{p-1}$

根据动态规划思想, 计算 DTW 最优对齐方式的时间复杂度为  $O(T^2)$ . 由于本文所提出的算法 LDTW 是经典算法的加权扩展, 其时间复杂度依然为  $O(T^2)$ . 但其需要根据 DTW/LDTW 迭代地寻找每一个训练实例的正例集和负例集并更新权重, 此举需要花费较多的计算量. 因此, 在  $N$  条长度为  $T$  的时间序列上, 计算每一对序列间 LDTW 距离的时间复杂度为  $O(N^2 T^2 \times (p+1))$ ,  $p+1$  为算法的迭代次数. 在多数情况下, 权重向量  $w$  仅需要迭代 2 次就趋于稳定 (详见第 4.2 节), 因此, 其时间复杂度  $O(N^2 T^2 \times (p+1)) \approx O(2N^2 T^2) \approx O(N^2 T^2)$ . 同时, LDTW 也可以通过使用对齐排列在对角线附近的子集来加速<sup>[11,14]</sup>.

## 4 实验与评价

本节将首先描述所使用的实验数据集 (UMD, BME 和 Conseason 从 <http://ama.liglab.fr/~douzal/tools.html> 获取, Trajectories 从 <https://archive.ics.uci.edu/ml/datasets/Character+Trajectories> 获取, 其他数据从 [http://www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/) 获取), 然后介绍参数的设置方法, 最后对比不同算法 (包括 ED、DTW、基于指数函数的 LDTW/LED、基于幂函数的 LDTW/LED、基于 shapelet 的决策树等) 的实验结果.

### 4.1 实验数据集

对比实验在 40 个公共数据集上完成, 见表 2. 数据集 Trajectories 是长度不一的多变量时间序列. 数据集 Conseason 的错误率通过 10 次有放回抽样获取, 而其他数据在获取时已被区分为训练集与测试集, 所有错误率都在测试集上计算. 另外, 由于正例集与负例集均是从训练集中获取的, 表 2 的最后一行  $min$  表示训练实例中不同类别实例的最小个数. 比如, 人工数据集 CBF 由 3 个不同的类别组成, 分别为 Cylinder, Bell 和 Funnel, 每一个类别在训练集中的个数分别为 10, 12 和 8, 所以, CBF 的  $min$  被设置为 8.

Table 2 Datasets  
表 2 数据集

数据集	实例个数( $N_{train}/N_{test}$ )	长度( $T$ )	类个数( $K$ )	正例集限制( $min$ )
ItalyPowerDemand	67/1029	24	2	33
ECG200	100/100	96	2	31
Gun Point	50/150	150	2	24
MoteStrain	20/1252	84	2	10
SonyAIBORobotSurfaceII	27/953	65	2	11
CinC ECG torso	40/1380	1 639	4	5
wafer	1000/6164	152	2	97
Car	60/60	577	4	11
ECGFiveDays	23/861	136	2	9
SwedishLeaf	500/625	128	15	26
Adiac	390/391	176	37	4
ChlorineConcentration	467/3840	166	3	91
Haptics	155/308	1 092	5	18
Conseason	37/328	144	2	18
OliveOil	30/30	570	4	4
CBF	30/900	128	3	8
FacesUCR	200/2050	131	14	4
Lighting2	60/61	637	2	20
Plane	105/105	144	7	9
Symbols	25/995	398	6	3
Trace	100/100	275	4	21
TwoLeadECG	23/1139	82	2	11
CC	300/300	60	6	50
BME	30/150	128	3	10
UMD	36/144	150	3	12
MALLAT	55/2345	1 024	8	2
Two Patterns	1000/4000	128	4	237
Beef	30/30	470	5	6
Coffee	28/28	286	2	14
FaceFour	24/88	350	4	3
Lighting7	70/73	319	7	5
MedicalImages	381/760	99	10	6
SonyAIBORobotSurface	20/601	70	2	6
FISH	175/175	463	7	21
FaceAll	560/1690	131	14	40
OSULeaf	200/242	427	6	15
WordsSynonyms	267/638	270	25	2
yoga	300/3000	426	2	137
InlineSkate	100/550	1 882	7	9
Trajectories	286/2572	—	20	8

4.2 参数设置

在本文所有实验中,近邻个数  $k$  的取值范围为  $\{1,3,5,7,9\}$ .另外,正例集或负例集的个数  $c$  是很难确定的,因为  $c$  值的设定关系到辨别性特征的发现,设置不正确时将会很难发现辨别性的特征,进而影响分类准确率.为覆盖尽量多的实例并不失一般性, $c$  被设置为  $c \in \{1,2,3,5,10\}$  并且  $c < min$ .若  $c = min$ ,则有可能发现不了足够的正例集.例如,CBF 的  $min$  被设置为 8,那么  $c$  只能从  $\{1,2,3,5\}$  中取;当  $c=10$  时,类别 Cylinder 和 Funnel 将不能得到足够的正例集,此时设置其权重为初始权重  $1/T$ .具体的参数设置总结于表 3.

Table 3 Parameters  
表 3 参数表

参数	值
$\alpha_{exp}$	1
$\alpha_{pow}$	0.1
$c$	$c \in \{1,2,3,5,10\}, c < min$
$k$	$k \in \{1,3,5,7,9\}$
$w_i^p \approx w_i^{p-1}$	$\ w_i^p - w_i^{p-1}\ ^2 \quad 0.1 * \ w_i^0\ ^2$ 或者 $p=4$

对于公式(6)和公式(7),当  $w$  从 0 到 1 不断变化时( $0 < w < 1$ ),指数函数的取值范围为  $1 \sim 0$ ( $0 < f(w) < 1$ ),而幂函数



的变化范围是 $+\infty \sim 1 (1 < f(w) < +\infty)$ .随着 $\alpha$ 的增长,较大  $w$  之间的差别在逐渐减小,而较小  $w$  之间的差别在不断增大.图 4 展示了 $\alpha$ 变化时分类错误率的变化(对于指数函数, $\alpha$ 变化范围为 0~10,而幂函数中的 $\alpha$ 变化范围是 0~1,此实验中, $k=1, c=3$ ),分别在两个不同的数据集 ItalyPowerDemand 和 Beef 上进行了测试,其中,实线表示指数函数的实验结果,虚线表示幂函数的实验结果. $\alpha=0$ ,对应于标准的 DTW 算法.当 $\alpha$ 从 0 增长到 10(或者相应地从 0 增长到 1)时,尽管分类错误率曲线有些许起伏,但整体上都经历了先下降后上升(或震荡平稳)的过程,并且都在 $\alpha=1$  或 0.1 附近取得最小值.为简洁起见,本文在训练 kNN 分类器时,将指数函数的 $\alpha$ 设置为 1,幂函数的 $\alpha$ 设置为 0.1.

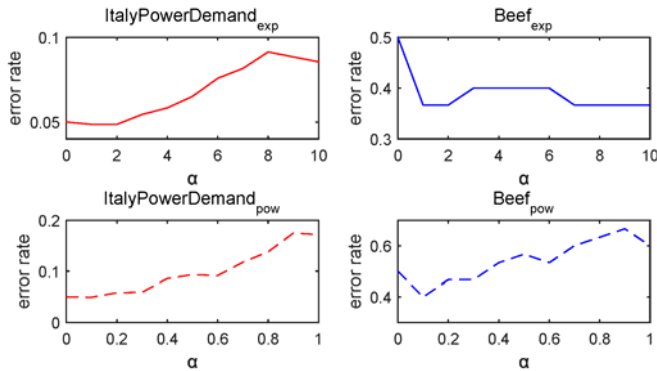


Fig.4 Error rate under different parameter  $\alpha$

图 4 不同参数 $\alpha$ 下的错误率

在算法 2 中,本文所提出的模型必须保证每一条时间序列  $x_i$  的权重  $w_i$  收敛.如图 5 所示(此实验中, $c=3, \alpha_{\text{exp}}=1, \alpha_{\text{pow}}=0.1$  保持不变),仅通过 2 次迭代,大多数的规范化欧式距离  $\|w_i^p - w_i^{p-1}\|^2$  就接近收敛于 0.为进一步保证收敛性,迭代次数被设置为 4.注意:若  $\|w_i^p - w_i^{p-1}\|^2 \leq 0.1 * \|w_i^0\|^2$  为真,则迭代停止.因为初始时  $w_i^0 = (1/T, \dots, 1/T)$ ,0.1 倍的  $\|w_i^0\|^2$  满足条件  $w_i^p \approx w_i^{p-1}$ .

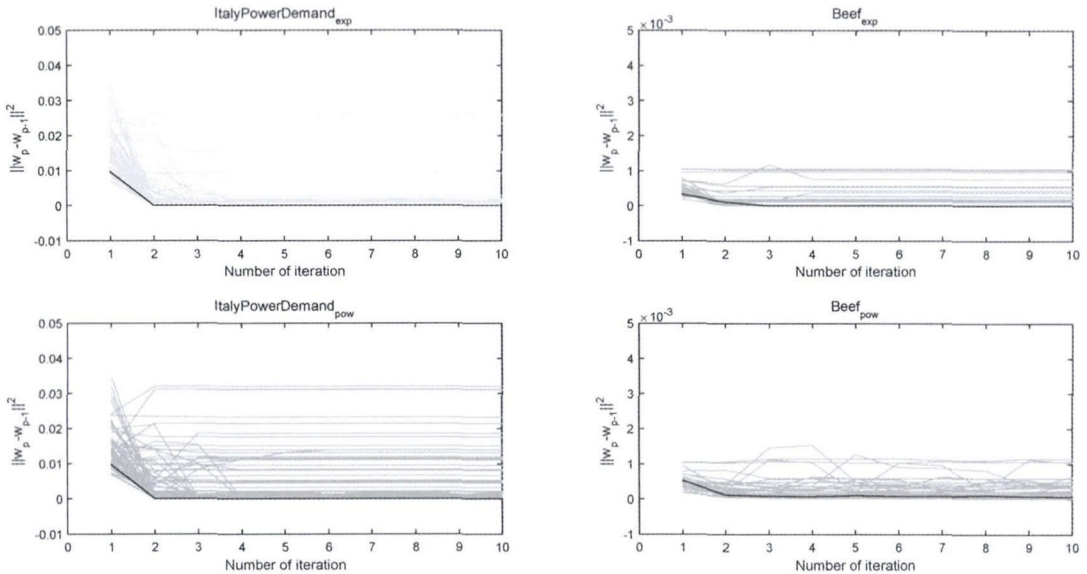


Fig.5 Examples for weights converging

图 5 权重的收敛性示例

4.3 与基准分类器对比

这一节将我们所提出的基于 LED/LDTW 的 kNN 分类器与经典的基于 ED/DTW 的 kNN 相比较.表 4 展示了所有对比算法在 40 个公共数据集上的分类错误率,其中,加粗的数值表示所对比方法中最优的结果.如果一个分类错误率与最优结果相比具有显著性差异<sup>[21]</sup>(显著性水平设置为 0.01),则用“\*”来标注.表中所有参数均通过 10 折交叉验证获得.

Table 4 Classification error rates compared with benchmarks

表 4 与基准分类器的分类错误率对比

数据集	标准度量( $k$ )		加权度量( $k, c$ )			
	ED	DTW	LED <sub>exp</sub>	LED <sub>pow</sub>	LDTW <sub>exp</sub>	LDTW <sub>pow</sub>
ItalyPower.	0.037(9)	0.049(7)	0.036(9,10)	0.035(7,1)	0.049(1,3)	0.047(7,1)
ECG200	0.100(3)	0.200*(3)	0.100(3,1)	0.070(3,2)	0.200*(3,1)	0.140(3,1)
Gun point	0.087(1)	0.093(1)	0.080(1,3)	0.087(1,5)	0.093(1,1)	0.080(1,10)
MoteStrain	0.121(1)	0.145*(9)	0.122(1,5)	0.134(1,5)	0.137(9,2)	0.138(9,1)
SonyAIBOII	0.141(1)	0.169*(1)	0.136(1,2)	0.132(1,1)	0.165*(1,1)	0.170*(1,2)
CinC ECG.	0.103(1)	0.349*(1)	0.104(1,1)	0.103(1,10)	0.347*(1,2)	0.351*(1,2)
wafer	0.005(1)	0.020*(1)	0.004(1,2)	0.005(1,1)	0.020*(1,1)	0.025*(1,1)
Car	0.267(1)	0.267(1)	0.267(1,1)	0.250(1,3)	0.267(1,1)	0.283(1,5)
ECGFive.	0.203(1)	0.232*(1)	0.190(1,2)	0.190(1,3)	0.230*(1,3)	0.196(1,5)
SwedishLeaf	0.211(1)	0.208(1)	0.213(1,1)	0.199(1,1)	0.205(1,2)	0.200(1,3)
Adiac	0.389(1)	0.396(1)	0.389(1,10)	0.394(1,5)	0.396(1,2)	0.396(1,3)
ChlorineCon.	0.350(1)	0.352(1)	0.349(1,1)	0.354(1,1)	0.353(1,2)	0.355(1,5)
Haptics	0.581(7)	0.571(3)	0.581(7,1)	0.562(7,3)	0.565(3,5)	0.575(5,3)
Conseazon	0.224(3)	0.278*(9)	0.203(9,1)	0.178(9,10)	0.254*(5,1)	0.257*(7,5)
OliveOil	0.133(1)	0.133(1)	0.133(1,1)	0.133(1,1)	0.133(3,1)	0.133(1,3)
CBF	0.148*(1)	0.003(1)	0.144*(1,1)	0.109*(1,2)	0.003(1,2)	0.000(3,2)
FacesUCR	0.231*(1)	0.095(1)	0.231*(1,2)	0.221*(1,1)	0.095(1,3)	0.101(1,3)
Lighting2	0.230*(3)	0.131(1)	0.197*(3,1)	0.213*(1,1)	0.115(1,2)	0.082(1,10)
Plane	0.038*(1)	0.000(1)	0.038*(1,1)	0.038*(1,1)	0.000(1,1)	0.000(1,1)
Symbols	0.101*(1)	0.050(1)	0.101*(1,1)	0.098*(1,5)	0.050(1,1)	0.069*(1,3)
Trace	0.240*(1)	0.000(1)	0.230*(1,2)	0.250*(1,10)	0.000(1,1)	0.000(1,1)
TwoLeadECG	0.253*(1)	0.096(1)	0.252*(1,1)	0.253*(1,5)	0.095(1,3)	0.101(1,3)
CC	0.090*(3)	0.007(1)	0.080*(3,3)	0.103*(3,5)	0.007(1,1)	0.007(1,2)
BME	0.173*(1)	0.107*(1)	0.180*(1,1)	0.220*(1,1)	0.053(3,1)	0.053(3,1)
UMD	0.194*(1)	0.118*(1)	0.181*(1,2)	0.174*(1,1)	0.007(1,2)	0.014(1,1)
MALLAT	0.080*(3)	0.066(1)	0.080*(3,1)	0.072*(3,10)	0.067(1,1)	0.060(1,1)
TwoPatterns	0.093*(1)	0.000(1)	0.089*(1,5)	0.091*(1,5)	0.000(1,1)	0.000(1,1)
Beef	0.467(1)	0.500(1)	0.467(1,1)	0.467(1,1)	0.367(1,1)	0.333(1,2)
Coffee	0.179*(3)	0.179*(1)	0.179*(3,1)	0.179*(1,3)	0.000(1,1)	0.000(1,1)
FaceFour	0.216(1)	0.170(1)	0.216(1,1)	0.193(1,1)	0.170(1,2)	0.205(1,2)
Lighting7	0.397*(3)	0.247(5)	0.370*(3,1)	0.329(3,1)	0.233(5,1)	0.219(5,2)
MedicalImag.	0.316*(1)	0.263(1)	0.311*(1,1)	0.300*(1,1)	0.262(1,2)	0.263(1,2)
SonyAIBO	0.304(1)	0.275(1)	0.301(1,2)	0.300(1,5)	0.275(1,3)	0.266(1,3)
FISH	0.217(1)	0.166(1)	0.217(1,1)	0.200(1,5)	0.166(1,1)	0.171(1,5)
FaceAll	0.286*(1)	0.190*(5)	0.280*(1,10)	0.286*(1,1)	0.130(1,5)	0.124(1,5)
OSULeaf	0.483*(1)	0.409(1)	0.480*(1,3)	0.470(1,10)	0.405(1,10)	0.397(1,10)
WordsSyn.	0.382(1)	0.351(1)	0.381(1,1)	0.386(1,1)	0.354(1,1)	0.389(1,1)
yoga	0.170(1)	0.164(1)	0.169(1,5)	0.170(1,3)	0.162(1,1)	0.161(1,3)
InlineSkate	0.658*(1)	0.616(1)	0.658*(1,1)	0.649(1,10)	0.615(1,1)	0.605(1,1)
Trajectories	—	0.159*(1)	—	—	0.068(1,1)	0.068(1,1)

对比与标准的度量算法,表 4 中的分类错误率展现了我们所提出的局部加权度量方法 LED 和 LDTW 在区分复杂时间序列时的有效性.尽管标准度量方式在两个数据集 MoteStrain 和 WordsSynonyms 上具有较好的分类错误率,但它与我们所提出的方法之间并没有显著性差异,也就是说,它们在预测下一个测试实例的类标时会得到相同的结果.我们所提出的算法在 40 个数据集集中的 38 个上表现较好(包括平局情况,另外需要注意:因为序列长度不相等,ED/LED 不能处理多变量数据集 Trajectories).对于没有时间延迟或具有较少延迟的序列,LED 算法能够提高其分类准确率,原因在于这些序列不需要或只有少量的时间扭曲,而 LDTW 能够很好地处理具有大量时间延迟或时间扭曲的序列.从表中也可以看到,基于 1NN 的度量算法在多于 80%的数据集上优于其他  $k$  近邻算法,这也和 Wang 等人<sup>[22]</sup>以及 Petitjean 等人<sup>[16]</sup>的描述相符.对于最近邻个数  $k$  与正例集或负例集个数  $c$  之

间的关系, $k$  大于  $c$  的情况只在 158 次测试中出现了 24 次,所以为分类准确率考虑, $c$  应该大于  $k$ .

另外,通过使用多维尺度分析(multidimensional scaling,简称 MDS)<sup>[23]</sup>,能够可视化数据集在不同度量下的潜在结构.图 6 在二维空间中展示了 Coffee 和 BME 数据集在 ED/DTW/LDTW 度量下的潜在结构(不同颜色的圆圈表示训练集的不同类别,不同颜色的十字表示测试集的不同类别,相同的颜色表示相同的类别),当括号中的参数  $stress$  小于 20%时,所获取的图像为数据集潜在结构的精确表示( $stress$  的大小用于度量 MDS 算法的契合度, $stress$  值越小,该表示图像就越契合原数据的潜在结构).从图 6 中可以观察到,LDTW 展现了较强的区分能力.换句话说,LDTW 在增大类间方差的同时降低了类内方差(如 BME 数据集).对于数据集 Coffee,从基于 LDTW 的 MDS 表示中可以发现,不同类别序列之间存在一条较为明显的分割线.

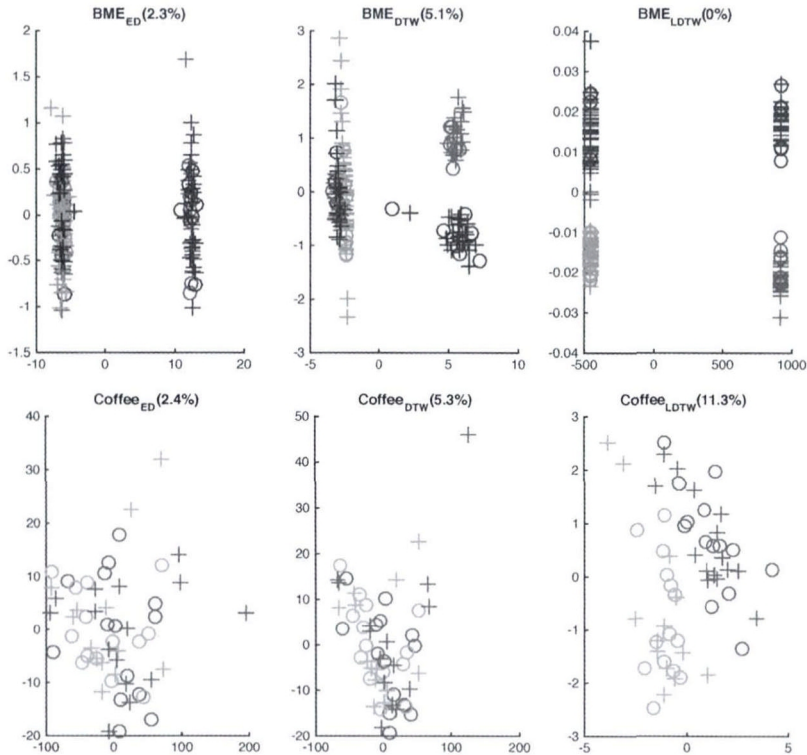


Fig.6 Underlying structures for different metrics on different datasets

图 6 不同度量方法在不同数据集上的潜在结构

#### 4.4 与基于shapelet的决策树对比

时间序列 shapelet 是时间序列中能够最大限度地表示一个类别的子序列<sup>[24,25]</sup>,基于时间序列 shapelet 的决策树具有分类精确率高、可解释性强等优点.为了进一步验证本文所提出算法的有效性,我们将其与快速 shapelet 决策树算法<sup>[26]</sup>(FS)和近似最优 shapelet 决策树算法<sup>[27]</sup>(LTS)做对比.

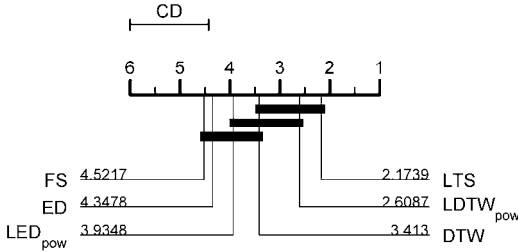
发现最好 shapelet 的时间复杂度为  $O(N^2T^4)$ ,即使经过优化,LTS 算法发现 shapelet 的时间复杂度仍为  $O(N^2T^2 \times \maxIter \times S)$ ,远超 LDTW 算法的时间复杂度,其中, $\maxIter$  为 LTS 的最大迭代次数, $S$  为 shapelet 的取值范围.因此,FS 算法和 LTS 算法只在部分较小数据集上完成了实验.

为简洁起见,我们直接选用基于幂函数的  $LED_{pow}$  和  $LDTW_{pow}$  以及基于 ED 和 DTW 的基准分类器与 LTS 和 FS 做对比实验,其中 LTS 和 FS 的参数设置参照了 Bagnall 等人<sup>[28]</sup>的设置方法.具体分类错误率和分类器排名见表 5.我们在 23 个数据集上完成了实验,表现最好的两个分类器为 LTS 和  $LDTW_{pow}$ ,其中,LTS 在 13 个数据集上的表现最佳, $LDTW_{pow}$  也在 10 个数据集上取得了最低的分类错误率.

**Table 5** Classification error rates compared with shapelet trees  
**表 5** 与 shapelet 决策树的分类错误率对比

	ED	DTW	LED <sub>pow</sub>	LDTW <sub>pow</sub>	LTS	FS
Adiac	0.389 (1)	0.396 (3.5)	0.394 (2)	0.396 (3.5)	0.437 (5)	0.514 (6)
Beef	0.467 (4.5)	0.500 (6)	0.467 (4.5)	0.333 (2)	0.240 (1)	0.447 (3)
BME	0.173 (4)	0.107 (2)	0.220 (5)	0.053 (1)	0.120 (3)	0.320 (6)
CBF	0.148 (6)	0.003 (2)	0.109 (5)	0.000 (1)	0.006 (3)	0.053 (4)
CC	0.090 (5)	0.007 (2)	0.103 (6)	0.007 (2)	0.007 (2)	0.081 (4)
Coffee	0.179 (5)	0.179 (5)	0.179 (5)	0.000 (1.5)	0.000 (1.5)	0.068 (3)
ECG200	0.100 (2)	0.200 (5)	0.070 (1)	0.140 (3)	0.150 (4)	0.250 (6)
ECGFive.	0.203 (5)	0.232 (6)	0.190 (3)	0.196 (4)	0.000 (1)	0.004 (2)
FaceAll	0.286 (4.5)	0.190 (2)	0.286 (4.5)	0.124 (1)	0.218 (3)	0.411 (6)
FaceFour	0.216 (6)	0.170 (3)	0.193 (4)	0.205 (5)	0.048 (1)	0.090 (2)
Gun point	0.087 (4.5)	0.093 (6)	0.087 (4.5)	0.080 (3)	0.000 (1)	0.061 (2)
ItalyPower.	0.037 (2.5)	0.049 (5)	0.035 (1)	0.047 (4)	0.037 (2.5)	0.095 (6)
Lighting2	0.230 (5)	0.131 (2)	0.213 (4)	0.082 (1)	0.177 (3)	0.295 (6)
Lighting7	0.397 (5)	0.247 (3)	0.329 (4)	0.219 (2)	0.197 (1)	0.403 (6)
MedicalImag.	0.316 (5)	0.263 (1.5)	0.300 (4)	0.263 (1.5)	0.271 (3)	0.433 (6)
MoteStrain	0.121 (2)	0.145 (5)	0.134 (3)	0.138 (4)	0.087 (1)	0.217 (6)
OliveOil	0.133 (2.5)	0.133 (2.5)	0.133 (2.5)	0.133 (2.5)	0.560 (6)	0.213 (5)
SonyAIBO	0.304 (5)	0.275 (3)	0.300 (4)	0.266 (2)	0.103 (1)	0.314 (6)
SonyAIBOH	0.141 (3)	0.169 (4)	0.132 (2)	0.170 (5)	0.082 (1)	0.215 (6)
Symbols	0.101 (6)	0.050 (2)	0.098 (5)	0.069 (4)	0.036 (1)	0.068 (3)
Trace	0.240 (5)	0.000 (2)	0.250 (6)	0.000 (2)	0.000 (2)	0.002 (4)
TwoLeadECG	0.253 (5.5)	0.096 (3)	0.253 (5.5)	0.101 (4)	0.003 (1)	0.090 (2)
UMD	0.194 (6)	0.118 (3)	0.174 (5)	0.014 (1)	0.056 (2)	0.139 (4)
Average rank	4.348	3.413	3.935	2.609	2.174	4.522

为了验证多个分类器在多个数据集上的表现,本文采用 Demšar 所提出的方法<sup>[29]</sup>对表 5 中的分类器进行显著性检验.如图 7 中给出的 Critical Diffierence 图所示,尽管 LTS 的分类结果更好,但 LTS,LDTW<sub>pow</sub> 与 DTW 之间不存在显著性差异,而且 LTS 和 LDTW<sub>pow</sub> 明显优于 FS 算法,这也重新验证了基于 DTW 的 kNN 分类器是难以击败的<sup>[28]</sup>这个已被多个学者验证的观点.另外,由于时间复杂度或分类效果的限制,LTS 和 FS 算法较难应用于大规模的数据集中,而本文所提出的 LDTW 算法在具有较低时间复杂度的同时,保持了较好的分类效果.



**Fig.7** Critical Diffierence diagram for multiple algorithms  
**图 7** 多种算法之间的 Critical Diffierence 图

5 实例解析

本节将通过实例解析,展示我们所提出的模型在人工数据集以及实际应用中的有效性与可解释性.

5.1 人工数据集

为展示我们所提出的算法的有效性,首先考虑两个人工数据集 BME 和 UMD,这两个数据集均由类间相似而类内不同的时间序列组成<sup>[2]</sup>.BME 数据集的长度为 128,分别由 3 个类别 Begin,Middle 和 End 组成.图 8 展示了 BME 中不同类别的时间序列,其中,黑色曲线为代表性序列,灰色曲线为该类别中其他不同形状的序列.在 Begin 类别中,所有时间序列的共同点在于开始阶段会出现一个小钟型子序列;而对于 End 类别,钟型子序列会出现在序列快结束时.但是在 Beign 或 End 序列中,大钟型子序列有可能朝上或者朝下出现,其与类别的相关性

不大.另外,具有朝上方向大钟型子序列的 Beign 或 End 序列在整体上是与 Middle 序列相似的.

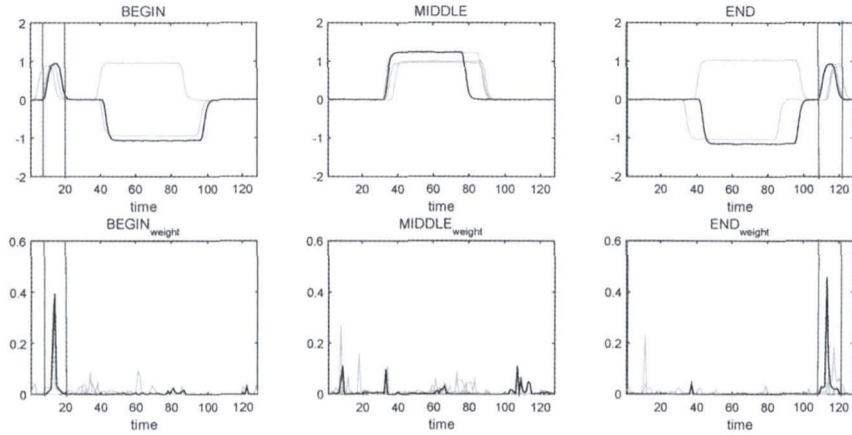


Fig.8 Discriminative LDTW weights for BME dataset

图 8 BME 数据集的判别性 LDTW 权重

图 9 展示了数据集 UMD 不同类别的轮廓,它长度为 150,由 3 个相似的类别 Up,Middle 和 Down 组成.其相对于 BME 数据集更加复杂,主要区别在于小钟型子序列的方向是朝上还是朝下的,与其出现位置或时间点并没有直接联系.从图 8 中可以观察到,当小钟型子序列方向朝上时,无论其出现在序列开头还是末尾,均定义为 Up 类别;当小钟型子序列方向朝下时,无论其出现在序列开头还是末尾,均定义为 Down 类别;当 Up/Down 中的大钟型子序列方向朝上时,它们与 Middle 序列具有全局相似性.如图 8 或图 9 的方框区域所示,最具判别性的权重刚好对应于小钟型子序列出现的地方.

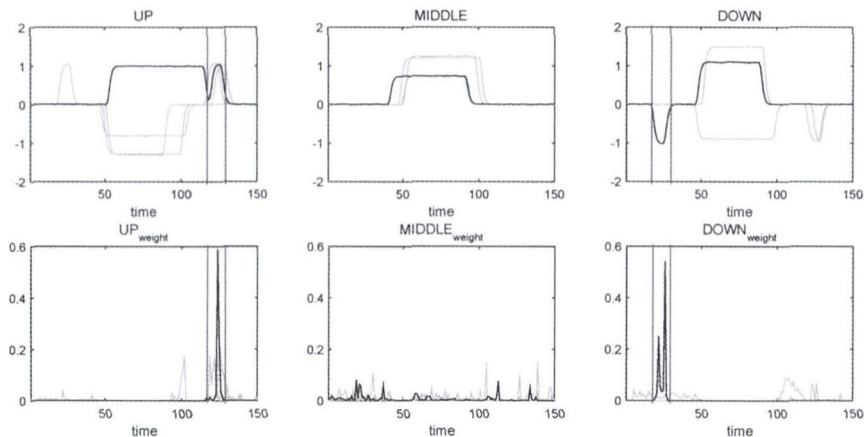


Fig.9 Discriminative LDTW weights for UMD dataset

图 9 UMD 数据集的判别性 LDTW 权重

## 5.2 真实数据集

Gun/NoGun 数据集是一组根据动作捕捉产生的时间序列,它也是时间序列中最为广泛应用的经典数据集之一.此数据集包含 200 条实例,其中 50 条作为训练集,另外 150 条作为测试集,数据集中的每一条实例的长度均为 150(不包含类属性).图 10 上半部分展示了该数据集中 Gun 与 NoGun 类别的代表性序列,其中,Gun 序列记录了表演者举起手枪并放回原位的动作序列;NoGun 序列记录了表演者做同样动作,但手中并没有手枪时的序

列.两种类别具有明显的全局相似性,其主要区别在于:若表演者手中没有枪,记录序列中就会出现如图 10 所示的方框中类似于波谷的子序列.图 10 下半部分给出了两种不同类别序列的局部权重,黑色权重曲线分别对应于上方 Gun 与 NoGun 加粗标注序列的权重.很明显,通过将学习到的权重映射到原始时间序列,LDTW 算法能够准确地发现该经典数据集的辨别性子序列(即方框中所示的片段),所发现子序列也与时间序列 shapelets<sup>[24,25]</sup>以及可解释性的序列规则<sup>[30]</sup>相符.

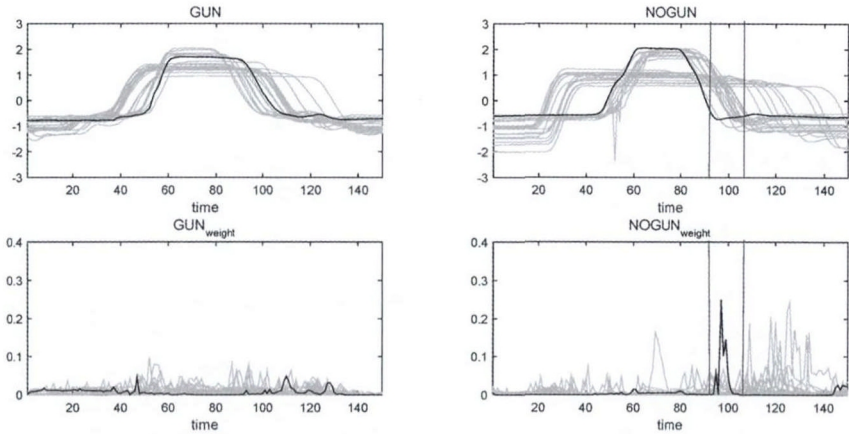


Fig.10 Discriminative LDTW weights for Gun/NoGun dataset

图 10 Gun/NoGun 数据集的辨别性 LDTW 权重

5.3 多变量时间序列数据

Trajectories 数据集是一个多变量的、长度不等的手写字符集,它记录了某人在书写 20 个不同字母(分别为 a,b,c,d,e,g,h,l,m,n,o,p,q,r,s,u,v,w,y,z)时笔尖的行走轨迹.该数据集包含 2 858 个不同长度的手写字母,每一个字母由 3 条长短不一的时间序列组成.通过将此多变量时间序列映射到二维空间,即可得到图 11 所示的手写字符.

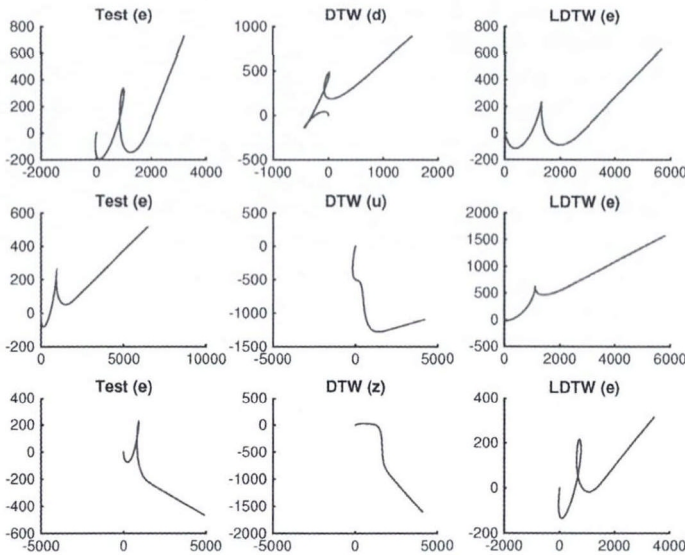


Fig.11 Learned nearest neighbor by algorithm DTW and LDTW for character “e”

图 11 DTW 和 LDTW 算法得到的与字母“e”最近的邻居

虽然所有实例均来自于同一个书写者,不同字母间却具有明显的全局相似性.基于 DTW 的 kNN 分类器在该数据集上的分类错误率为 0.159,而 LDTW 所获取的分类错误率为 0.068,两者之间具有显著性差异.从图中可以观察到,对于待分类字母“e”,基于 DTW 的度量容易将字母“e”与字母“d”“u”“z”混淆,而 LDTW 能够很好地找到它们的最近邻.

## 6 总结与展望

本文介绍了一种用于处理复杂时间序列数据的局部加权 DTW 模型.它根据训练实例采样点的正例集和负例集来发现辨别性子序列.通过在多个数据集上的测试实验,表明其无论在单变量时间序列还是多变量时间序列上都具有较强的分类能力.另外,通过实例解析阐明了基于 LDTW 的 kNN 分类器具有可解释性.但由于该模型需要迭代地学习权值,因此需要消耗大量的时间.未来的研究方向包括使用优化学习方法加速权重的学习过程,以及将该模型扩展到基于核函数的 kNN 或支持向量机分类模型等.

## References:

- [1] Soheily-Khah S, Douzal-Chouakria A, Gaussier E. Generalized  $k$ -means-based clustering for temporal data under weighted and kernel time warp. *Pattern Recognition Letters*, 2016,75:63–69. [doi: 10.1016/j.patrec.2016.03.007]
- [2] Frambourg C, Douzal-Chouakria A, Gaussier E. Learning multiple temporal matching for time series classification. In: *Proc. of the Advances in Intelligent Data Analysis XII*. Springer-Verlag, 2013. 198–209. [doi: 10.1007/978-3-642-41398-8\_18]
- [3] Zhou F, De La Torre F, Hodgins JK. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2013,35(3):582–596. [doi: 10.1109/TPAMI.2012.137]
- [4] Yuan J, Wang Z. Review of time series representation and classification techniques. *Computer Science*, 2015,42(3):1–7 (in Chinese with English abstract).
- [5] Arici T, Celebi S, Aydin AS, Temiz TT. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications*, 2014,72(3):3045–3062. [doi: 10.1007/s11042-013-1591-9]
- [6] Reyes M, Domínguez G, Escalera S. Feature weighting in dynamic timewarping for gesture recognition in depth data. In: *Proc. of the 13th Int'l Conf. on Computer Vision Workshops (ICCV Workshops)*. Barcelona: IEEE Computer Society, 2011. 1182–1188. [doi: 10.1109/ICCVW.2011.6130384]
- [7] Zhou F, Torre F. Canonical time warping for alignment of human behavior. In: *Proc. of the Advances in Neural Information Processing Systems 22*. MIT Press, 2009. 2286–2294.
- [8] Rath TM, Manmatha R. Word image matching using dynamic time warping. In: *Proc. of the 2003 Conf. on Computer Vision and Pattern Recognition (CVPR 2003)*. Washington: IEEE Computer Society, 2003. 521–527. [doi: 10.1109/CVPR.2003.1211511]
- [9] Adams NH, Bartsch MA, Shifrin JB, Wakefield GH. Time series alignment for music information retrieval. In: *Proc. of the 5th ISMIR*. 2004. 303–311.
- [10] Garreau D, Lajugie R, Arlot S, Bach F. Metric learning for temporal sequence alignment. In: *Proc. of the Advances in Neural Information Processing Systems 27*. Montréal: MIT Press, 2014. 1817–1825.
- [11] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1978,26(1):43–49. [doi: 10.1109/TASSP.1978.1163055]
- [12] Zhang XL, Luo ZG, Li M. Merge-Weighted dynamic time warping for speech recognition. *Journal of Computer Science and Technology*, 2014,29(6):1072–1082. [doi: 10.1007/s11390-014-1491-0]
- [13] Aach J, Church GM. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 2001,17(6):495–508. [doi: 10.1093/bioinformatics/17.6.495]
- [14] Itakura F. Minimum prediction residual principle applied to speech recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1975,23(1):67–72. [doi: 10.1109/TASSP.1975.1162641]
- [15] Keogh EJ, Pazzani MJ. Derivative dynamic time warping. In: *Proc. of the 1st SIAM Int'l Conf. on Data Mining (SDM 2001)*. Chicago: Society for Industrial and Applied Mathematics, 2001. 1–11.
- [16] Petitjean F, Forestier G, Webb G, Nicholson AE, Chen YP, Keogh E. Dynamic time warping averaging of time series allows faster and more accurate classification. In: *Proc. of the 14th IEEE Int'l Conf. on Data Mining (ICDM 2014)*. Shenzhen: IEEE Computer Society, 2014. 470–479. [doi: 10.1109/ICDM.2014.27]



- [17] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 2007,11(5): 561–580.
- [18] Jeong YS, Jeong MK, Omiaomu OA. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 2011, 44(9):2231–2240. [doi: 10.1016/j.patcog.2010.09.022]
- [19] Celebi S, Aydin AS, Temiz TT, Arici T. Gesture recognition using skeleton data with weighted dynamic time warping. In: *Proc. of the 8th Int'l Conf. on Computer Vision Theory and Applications (VISAPP 2013)*. Barcelona: Springer-Verlag, 2013. 620–625.
- [20] Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 2009,10(2):207–244.
- [21] Dietterich TG. Statistical tests for comparing supervised classification learning algorithms. Technical Report, Oregon State University, 1996. 1–24.
- [22] Wang X, Mueen A, Ding H, Trajcevski G, Scheuermann P, Keogh E. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 2013,26(2):275–309. [doi: 10.1007/s10618-012-0250-5]
- [23] Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 1964,29(1):1–27. [doi: 10.1007/BF02289565]
- [24] Ye L, Keogh E. Time series shapelets: A new primitive for data mining. In: *Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2009)*. Paris: ACM Press, 2009. 947–955. [doi: 10.1145/1557019.1557122]
- [25] Yuan JD, Wang ZH, Han M. A discriminative shapelets transformation for time series classification. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 2014,28(6):No.1450014. [doi: 10.1142/S0218001414500141]
- [26] Rakthanmanon T, Keogh E. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: *Proc. of the 13th SIAM Int'l Conf. on Data Mining (SDM 2013)*. Austin: Society for Industrial and Applied Mathematics, 2013. 668–676. [doi: 10.1137/1.9781611972832.74]
- [27] Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L. Learning time-series shapelets. In: *Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2014)*. New York: ACM Press, 2014. 392–401. [doi: 10.1145/2623330.2623613]
- [28] Bagnall A, Lines J, Bostrom A, Large J, Keogh E. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. In: *Proc. of the Data Mining and Knowledge Discovery*. 2016. 1–55. [doi: 10.1007/s10618-016-0483-9]
- [29] Demšar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 2006,7(1):1–30.
- [30] Yuan J, Wang Z, Han M, Sun Y. A lazy associative classifier for time series. *Intelligent Data Analysis*, 2015,19(5):983–1002. [doi: 10.3233/IDA-150754]

#### 附中文参考文献:

- [4] 原继东,王志海.时间序列的表示与分类算法综述. *计算机科学*, 2015,42(3):1–7.



原继东(1989 - ),男,河南焦作人,博士,讲师,CCF 专业会员,主要研究领域为机器学习,数据挖掘,时间序列分类.



孙艳歌(1982 - ),女,博士生,讲师,CCF 专业会员,主要研究领域为机器学习,数据挖掘.



王志海(1963 - ),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为机器学习,数据挖掘.



张伟(1987 - ),男,博士生,主要研究领域为数据挖掘,机器学习.