

基于局部梯度和二进制模式的时间序列分类算法*

郝石磊^{1,2}, 王志海^{1,2}, 刘海洋^{1,2}



¹(北京交通大学 计算机与信息技术学院, 北京 100044)

²(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)

通信作者: 刘海洋, E-mail: haiyangliu@bjtu.edu.cn

摘要: 时间序列分类问题是时间序列数据挖掘中的一项重要任务, 近些年受到了越来越广泛的关注. 该问题的一个重要组成部分就是时间序列间的相似性度量. 在众多相似性度量算法中, 动态时间规整是一种非常有效的算法, 目前已经被广泛应用到视频、音频、手写体识别以及生物信息处理等众多领域. 动态时间规整本质上是一种在边界及时间一致性约束下的点对点的匹配算法, 能够获得两条序列间的全局最优匹配. 但该算法存在一个明显的不足, 即不一定能实现序列间的局部合理匹配. 具体的讲, 就是具有完全不同局部结构信息的时间点有可能被动态时间规整算法错误匹配. 为了解决这个问题, 提出了一种改进的基于局部梯度和二进制模式的动态时间规整算法 LGBDTW (local gradient and binary pattern based dynamic time warping), 通过考虑时间序列点的局部结构信息来强化传统动态时间规整算法. 所提算法虽然实质上是一种动态时间规整算法, 但它通过考虑序列点的局部梯度和二进制模式值来进行相似性加权度量, 有效避免了具有相异局部结构的点匹配. 为了进行全面比较, 将所提出的算法应用到了最近邻分类算法的相似性度量中, 并在多个 UCR 时间序列数据集上进行了测试. 实验结果表明, 所提出的方法能有效提高时间序列分类的准确率. 此外, 实例分析验证了所提出算法的可解释性.

关键词: 动态时间规整; 时间序列相似性; 数据挖掘; 时间序列分类

中图法分类号: TP301

中文引用格式: 郝石磊, 王志海, 刘海洋. 基于局部梯度和二进制模式的时间序列分类算法. 软件学报, 2022, 33(5): 1817-1832. <http://www.jos.org.cn/1000-9825/6208.htm>

英文引用格式: Hao SL, Wang ZH, Liu HY. Time Series Classification Algorithm Based on Local Gradient and Binary Pattern. Ruan Jian Xue Bao/Journal of Software, 2022, 33(5): 1817-1832 (in Chinese). <http://www.jos.org.cn/1000-9825/6208.htm>

Time Series Classification Algorithm Based on Local Gradient and Binary Pattern

HAO Shi-Lei^{1,2}, WANG Zhi-Hai^{1,2}, LIU Hai-Yang^{1,2}

¹(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

²(Beijing Key Lab of Traffic Data Analysis and Mining (Beijing Jiaotong University), Beijing 100044, China)

Abstract: Time series classification is an important task in time series data mining and has attracted significant attention in recent years. An important part of this problem is the similarity measurement between time series. Among many similarity measurement algorithms, dynamic time warping (DTW) is very effective, which has been widely used in many fields such as video, audio, handwriting recognition, and biological information processing. DTW is essentially a point-to-point matching algorithm under the boundary and time consistency constraints, which is able to provide the global optimal matching between two sequences. However, there is an obvious deficiency in this algorithm, that is, it does not necessarily achieve reasonable local matching between sequences. Specifically, the time points with completely different local structure information may be incorrectly matched by DTW algorithm. In order to solve this problem, an improved DTW algorithm based on local gradient and binary pattern (LGBDTW) is proposed. Although the proposed algorithm is essentially a dynamic time warping algorithm, it takes into account the local gradient and binary pattern values of sequence points to carry out similarity weighted measurement, effectively avoiding points matching with different local structures. In order to make a

* 基金项目: 中央高校基本科研业务费专项 (2019YJS041); 国家自然科学基金 (61672086, 61702030, 61771058); 北京市自然科学基金 (4182052)

收稿时间: 2020-04-12; 修改时间: 2020-08-27; 采用时间: 2020-11-18

comprehensive comparison, the algorithm is adopted as the similarity measurement of the nearest neighbor classification algorithm, and tests it on multiple UCR time series datasets. Experimental results show that the proposed method can effectively improve the accuracy of time series classification. In addition, some examples are provided to verify the interpretability of the proposed algorithm.

Key words: dynamic time warping (DTW); time series similarity; data mining; time series classification

时间序列数据广泛存在于现实世界的各个领域,目前对时间序列的分析已经成为股票价格、天气数据、生物医学测量、航空航天等许多领域研究的重要课题^[1,2].时间序列数据类型为实值型,数据维度高且数据量通常较大.此外,时间序列分类问题中的序列属性是有序的,这使其明显区别于传统的分类问题.实质上,序列中的属性排序是否按照时间进行是无关紧要的,重要的是序列中可能存在依赖于顺序的辨别性特征^[3].因此,时间序列分类问题的研究对于数据挖掘领域的发展具有重要意义.

在时间序列分类问题中,最近邻算法(one nearest neighbor, 1NN)是目前最为流行的分类算法之一,该算法较为简单,且分类效果显著.最近邻算法的核心部分就是相似性度量,而动态时间规整(dynamic time warping, DTW)就是一种非常有效的相似性度量算法,该算法能够通过局部的非线性规整找到序列间的最佳匹配路径.DTW是由 Berndt 等人^[4]提出并将其应用到了时间序列的模式发现,后来又被广泛用于人体活动识别^[5],孤立词识别^[6],文字图像匹配^[7],人体运动动画^[8],生物信息处理^[9]以及时间序列分类^[3]等多个领域中.

基于 DTW 的 1NN 算法是众多时间序列分类算法中最有效的算法之一,很难被其他算法超越^[3].DTW 本质上是一种点对点的对齐算法,它通过允许一个序列到另一个序列的非线性映射提高了匹配点对之间的时间一致性.从 DTW 提取匹配分量之后,通常是基于两点之间的欧几里得距离来进行点匹配.然而,这种使用基于点数值的欧几里得距离的相似性度量方法是不可靠的,甚至会导致匹配错误,即产生局部结构并不相似的点的对齐(如图 1(a)所示).很明显,虽然 DTW 实现了全局最优匹配,但它没有兼顾序列的局部结构信息.这也反映了为什么 DTW 相似性度量之下的最近邻分类器的可解释性会弱于基于 shapelet 的分类器^[10,11].总的来说,DTW 的确能够捕捉到时间序列的全局最优匹配,但是它没有考虑到序列的局部结构信息,这样的对齐结果是缺乏语义的.

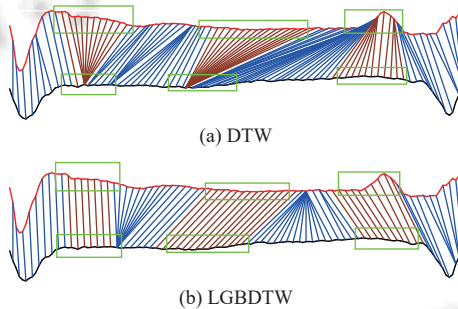


图 1 DTW 与 LGBDTW 对齐效果比较

为了解决这个问题,本文提出了一种基于局部梯度和局部二进制模式(local binary pattern, LBP)的动态时间规整算法 LGBDTW.该算法不只是考虑了时间序列点的局部梯度信息,同时还兼顾点的局部二进制模式信息,它通过将点对的这两种局部结构信息以一定的权重进行融合强化了 DTW.通过实例测试,所提算法获得了更加准确并具有较好可解释性的对齐.图 1(b)是一个基于 LGBDTW 对齐的实例,很明显时间序列间的局部形状信息被成功匹配.

本文所提出方法的灵感来自于计算机视觉领域的方向梯度直方图(histogram of oriented gradients, HOG)^[12],图像特征描述子 LBP^[13],以及一维时间序列局部特征描述子 HOG-1D^[11].众所周知,现实世界大多数数据可能含有噪声,时间序列数据也是如此,其获取甚至分类过程很容易受到噪声的影响.因此,本文首先使用滤波技术来缓解噪声对数据的影响,然后提出了一种基于局部二进制模式的时间序列局部特征描述子 LBPT(local binary pattern of time series),之后又在此基础上提出了融合 HOG-1D 和 LBPT 两种特征描述子的加权动态时间规整算法(LGBDTW),并成功将其应用到了时间序列分类任务.

本文的主要贡献如下:(1)提出了一种新的基于局部二进制模式的时间序列局部特征描述子 LBPT 来更加准确的反映序列的局部结构信息.(2)提出了一种基于 HOG-1D 特征描述子和 LBPT 特征描述子的加权动态时间规

整算法。(3) 所提出的分类算法应用到真实数据集中明显提高了分类准确率, 并且具有较好的可解释性。

本文第 1 节介绍了时间序列分类, 动态时间规整算法以及局部特征提取的相关工作。在第 2 节和第 3 节介绍本文提出的算法, 然后在第 4 节给出了实验结果分析。最后, 在第 5 节进行相应总结并简单探讨未来的研究方向。

1 相关研究工作

随着时间序列分类问题在各个领域的重要性日益提高, 人们提出了许多适合不同应用的时间序列分类算法。这些算法主要可以分为基于距离的和基于特征的两类。前者是将最近邻分类器与不同的距离度量算法相结合, 比较典型的有时间扭曲编辑 (TWE)^[14]、移动-分割-合并 (MSM)^[15]等。后者是在原始序列中提取具有辨别性的特征, 以此将原始序列转换到新的特征空间。比较常见的有时间序列森林 (TSF)^[16]、时间序列特征包 (TSBF)^[17]、shapelets 转换^[10]、学习模式相似性 (LPS)^[18]、符号聚合近似 (SAX)^[19]、以及模式包 (BOP)^[20]等。此外, 一些研究者还将不可预测的卷积神经网络 (CNN) 应用到了时间序列分类问题的特征提取中。例如, Le Guennec 等人^[21]将 CNN 扩展到了时间序列分类问题, 提出了时间 LeNet (t-LeNet); Cui 等人^[22]提出了一种多尺度 CNN 方法 (MCNN), 它基于提取的子序列来训练 CNN; Wang 等人^[23]还将多层感知机 (MLP)、全卷积神经网络 (FCN) 和残差网络 (ResNet) 成功地应用于时间序列。此外, Zhao 等人^[24]提出了时间 CNN (Time-CNN) 算法, 用均方误差代替交叉熵损失函数来兼容时间序列。然而, 这类引入神经网络的方法往往复杂度过高, 同时其结果是缺乏可解释性的。

在时间序列分类问题中, DTW 是一种非常经典的相似性度量算法, 发展到现在已经有了很多关于它的改进算法。对于该算法匹配路径过于扭曲的问题, 传统来讲, 可以通过添加一些约束来缓解, 例如 Sakoe-Chiba band^[6]、局部相关约束^[25]以及有限扭曲路径长度^[26]等。此外, 一些研究者试图通过加入权重或利用导数等信息来改善序列对齐质量。例如, Keogh 等人^[27]提出了基于导数的动态时间规整 (DDTW); Jeong 等人^[28]提出了一种基于惩罚的加权动态时间规整 (WDTW) 算法; 此外, 一种复杂性-不变 DTW (CID) 被 Gustavo 等人^[29]提了出来, 这种方法通过乘以复杂性校正因子弥补了被比较序列之间的复杂性。然而, 在这些方法中, 时间点的相似性都是依赖单一点的数值或其衍生值来度量的, 这些度量方式忽略了序列点的局部结构信息。

为了解决时间序列相似性度量过程中局部信息匹配不充分的问题, 一些研究者试图先将序列的局部形状作为特征提取出来, 然后再进行相似性度量。例如, Deng 和 Baydogan 等人^[16,17]提出用直线线性拟合子序列的斜率; Zhao 等人^[11,30]将原本用于计算机视觉领域的 HOG 扩展到了时间序列数据, 提出用 HOG-1D 局部特征描述子来表示序列中点的局部结构信息。HOG-1D 继承了 HOG 的核心思想, 即通过将时间序列点局部结构进行直方图表示来反映序列的局部结构信息。时间序列中点的 HOG-1D 局部特征描述子的具体生成过程如图 2 所示, 首先是从原序列中提取该点的对应子序列 S (如图 2 中蓝色线所示), 然后将其分为若干间隔 $I = \{I_1, I_2, \dots, I_h\}$ (图 2 中矩形虚线框圈起部分)。对于每一个间隔 I_i , 统计 I_i 中所有点的梯度和方向, 可以得到一个方向梯度直方图。最后串联 h 个间隔的直方图来构成子序列 S 的特征描述子, 即 HOG-1D 局部特征描述子 (对于 HOG-1D 更为详细的信息可参考文献 [30])。方向梯度直方图的统计特性使得 HOG-1D 对观测噪声的敏感性进一步降低, 同时直方图的串联又能很好的捕捉时间信息。

类似于 HOG-1D 的局部特征提取方式是通过考虑点的梯度或斜率等来编码时间序列局部结构信息的, 之后再用这些描述子形成的新特征空间进行相似性度量。这种特征提取方式在一定程度上缓解了时间序列的局部结构信息丢失问题。然而, 受梯度或斜率等计算方式的限制, 这种类型的子序列编码方式是有缺陷的。如图 3(左) 所示为一条子序列, 在 $x=4$ 位置, 3 种形状结构下点的梯度 G 都为 0.5, 点的局部差异并没有充分体现出来。然而本文算法使用点的局部二进制值的离散特性有效的区分了点的局部差异 (如图 3(右) 所示)。可以看到, 在 $x=4$ 处 3 种形状结构下的点的二进制值编码是明显不同的 (LBP 值分别为 3、0、1), 点的局部细节信息得到了更为精准的描述。

2 基于局部二进制模式的局部特征表示

针对传统局部特征描述子对时间序列点的局部结构信息描述不充分的问题, 本文提出了一种基于局部二进制模式的时间序列的局部特征描述子 LBPT。本节将详细介绍该描述子的生成过程。

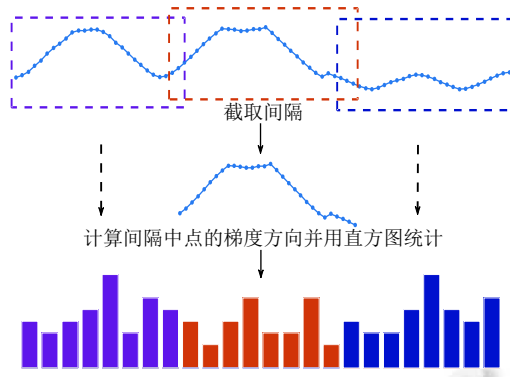


图2 HOG-1D 局部特征描述子生成过程

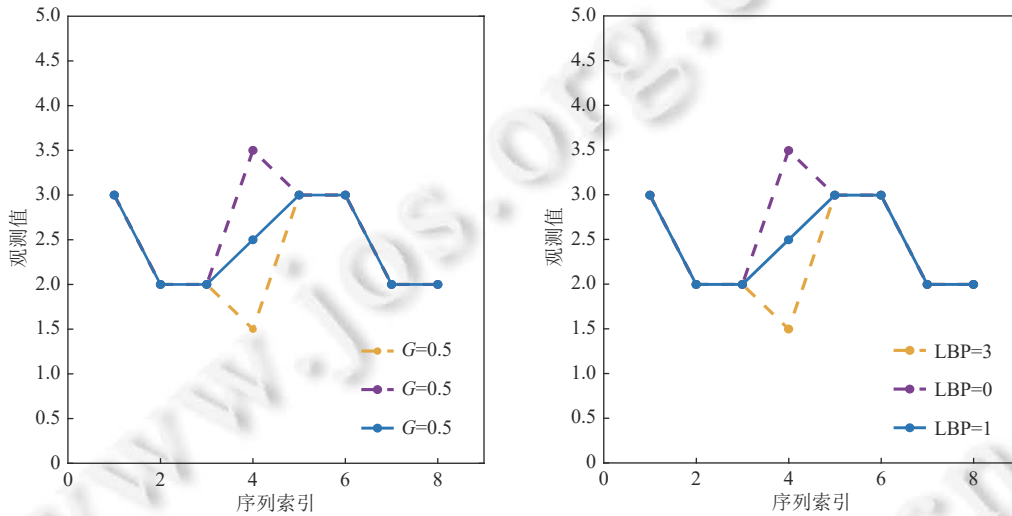


图3 时间序列点的局部信息编码效果比较

2.1 滤波技术

图像处理问题中, 为了缓解各种噪声产生的影响, 会用到各种滤波技术, 比如常见的有均值滤波, 中值滤波和高斯滤波等. 类似的, 时间序列数据也存在噪声. 因此, 为了缓解数据噪声对分类产生的影响, 借鉴图像处理中的经验, 本文将首先使用滤波技术对时间序列数据进行降噪处理. 由于均值滤波为线性滤波, 其主要采用邻域平均的思想, 相比于其他滤波技术, 该方法较为简单、速度较快. 鉴于此, 本文将采用均值滤波对序列进行处理, 即通过设置滑动窗口 w 的大小来确定每一个序列点的局部近邻子序列的长度, 然后用局部近邻子序列的均值代替原始序列点. 给定时间序列 $T=\{t_1, t_2, \dots, t_m\}$, 则其每一个点 t_i 的滤波过程如公式 (1):

$$\begin{aligned}
 t_{fi} &= \frac{1}{w} \sum_{j=i-w/2}^{i+(w-1)/2} t_j \quad w/2 < i \leq m - (w - 1)/2 \\
 t_{fi} &= \frac{1}{w} \left(\sum_{j=1}^{i+(w-1)/2} t_j + \sum_{j=1}^{w/2-i+1} t_1 \right) \quad 1 \leq i \leq w/2 \\
 t_{fi} &= \frac{1}{w} \left(\sum_{j=i-w/2}^m t_j + \sum_{j=1}^{(w-1)/2+i-m} t_m \right) \quad m - (w - 1)/2 < i \leq m
 \end{aligned}
 \tag{1}$$

其中, w 代表滑动窗口大小, t_{fi} 表示 t_i 滤波之后的序列点值, t_j 表示在滑动窗口大小为 w 时时间序列 T 中点 t_i 的近邻点, t_1 和 t_m 则表示时间序列 T 的前后两个端点.

通过公式 (1) 可以得到滤波之后的序列 $T_f=\{t_{f1}, t_{f2}, \dots, t_{fm}\}$. 经过滤波处理之后的序列, 因为噪声引起的局部波动会得到明显的缓解. 图 (4) 是 SonyAIBORobotSurface1 数据集中类标相同的两条时间序列滤波前后的对比图,

可以看出, 跟不进行滤波 (图 4(左)) 相比, 经过滤波之后的两条序列 (图 4(右)) 更趋于重合, 因此也更有可能被匹配到同一类中去。

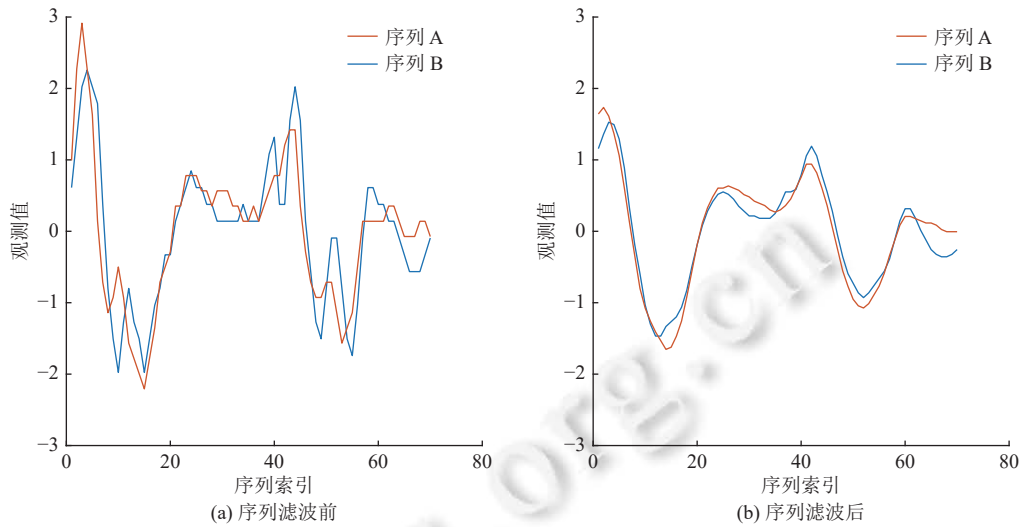


图 4 SonyAIBORobotSurface1 数据集中类标签相同的两条时间序列滤波前后比较

2.2 基于局部二进制模式的时间序列局部特征描述子

局部二进制模式 LBP 最初是由 Ojala 等人^[13]提出的一种用于纹理分类的特征描述子, 目前已被广泛应用到了图像分析^[31]、人脸识别^[32,33]、目标识别^[34]等多个领域. 该描述子只需通过简单的二进制运算就可以获得, 具有较低的计算复杂度, 更重要的是该描述子本身的离散特性使其具有较高的特征鉴别能力 (如图 3 所示).

鉴于 LBP 对局部特征较高的鉴别能力及其低复杂性, 本文提出了一种新的应用于一维时间序列的局部特征描述子 LBPT. LBPT 描述子本质上是通过将时间序列点对应子序列上的每个点作为中心点, 再将其周围近邻点进行二进制阈值化处理, 然后把得到的多位二进制数转换为 LBP 整数值, 最后对子序列中所有点的 LBP 值进行直方图统计. 图 5 是一条子序列中任一点的 LBP 值生成过程. 需要注意的是, 子序列中点的近邻点的数量是可以调整的, 但不宜过多, 否则会导致最终生成的描述子维度过高. 简单起见, 图 5 中我们选取的是两个近邻点.

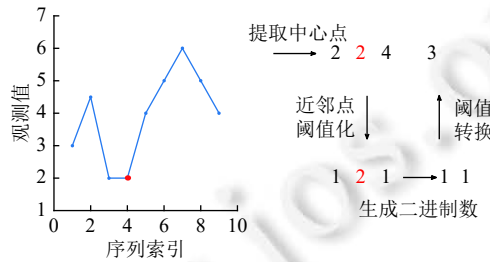


图 5 时间序列点的 LBP 值生成过程

这里给出本文的 LBPT 特征描述子的详细生成过程, 如图 6 所示. 给定一条子序列 $S = \{s_1, s_2, \dots, s_i\}$, 计算每个点 s_i 的二进制模式值 LBP_i , 其计算方式如公式 (2) 和公式 (3) 所示.

$$LBP_i = \sum_{j=0}^{N-1} b(s_{i+N-j} - s_i)2^j + \sum_{j=N+1}^{2N} b(s_{i+N-j} - s_i)2^{j-1} \quad (2)$$

$$b(\Delta s) = \begin{cases} 1, & \Delta s \geq 0 \\ 0, & \Delta s < 0 \end{cases} \quad (3)$$

其中, Δs 表示的是 s_i 近邻点与 s_i 的差值, N 表示计算点 s_i 所使用的近邻点对数. 需要注意的是, 此时的 LBPT 描述子长度为 4^N , 增长较快. 因此, 简单起见, 本文中只考虑 $N=1$ 的情况. 当求出子序列中所有点的 LBP 值之后, 用直方图统计不同 LBP 模式值的点数量. 最终用直方图统计结果来表示原始时间序列点的局部结构信息, 即 LBPT 特征描述子 (如图 6 中最后生成的特征描述子为 {1, 3, 3, 4}).

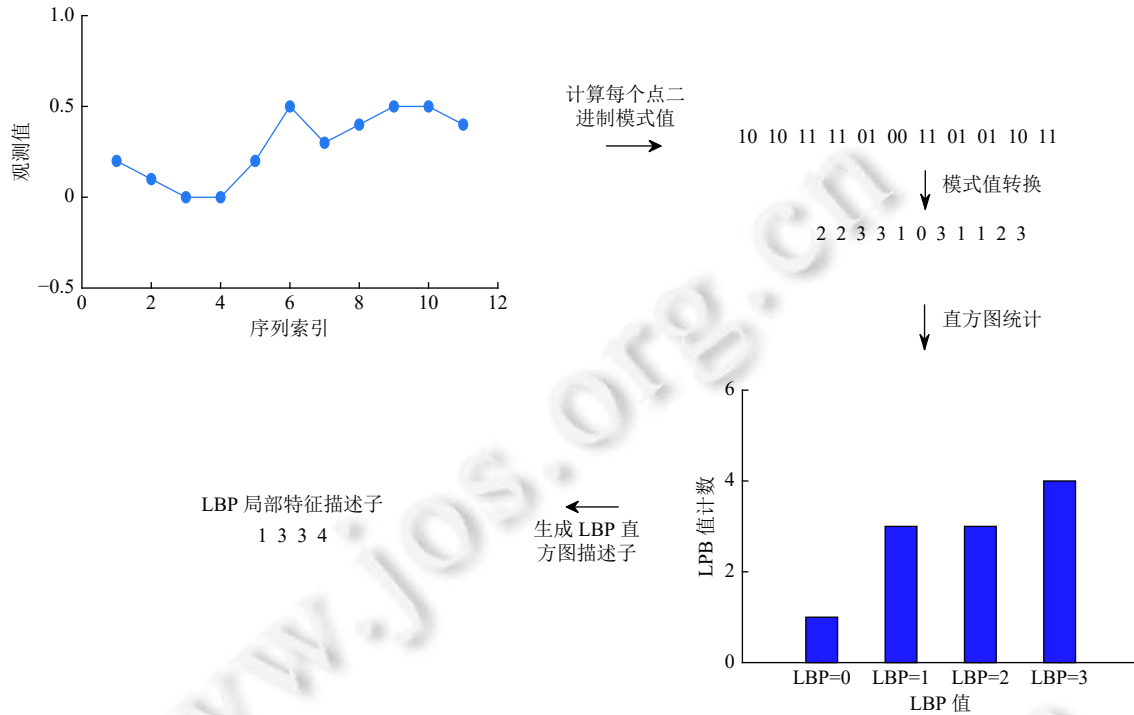


图 6 LBPT 局部特征描述子生成过程

算法 1 给出了本文的 LBPT 局部特征描述子的生成过程. 其中第 1 行是数据的初始化操作; 第 2-4 行是计算所有子序列点的 LBP 值; 最后第 5-10 行是对 LBP 值进行累加得到最终的 LBPT 局部特征描述子. 需要注意的是, 虽然该特征描述子生成过程的时间复杂度为 $O(2^N \cdot seqlen)$, 但这里 N 通常只需取相对较小的值即可, 例如 $N=1$, 因此其时间复杂度在实际情况下可以近似为 $O(seqlen)$.

算法 1. LBPT 局部特征描述子生成过程.

输入. 子序列 S , 近邻点对数 N ;

输出. LBPT 局部特征描述子.

1. 初始化: 子序列长度 $seqlen$, 描述符数组 $DS[4^N]$, LBP 值存储数组 $lbp[seqlen]$.
2. for($j=1; j \leq seqlen; j++$)
3. 用公式 (2)(3) 计算每一个子序列点的 LBP 值 $lbp[j]$.
4. end for
5. for($j=1; j \leq seqlen; j++$)
6. for($k=0; k < 4^N; k++$)
7. if($k == lbp[j]$)
8. $DS[k] \leftarrow DS[k] + 1$
9. end for

10. end for
11. return DS

3 基于局部特征加权的相似性度量算法

本节首先简单介绍传统动态时间规整算法, 然后对本文所提出的基于局部特征加权的相似性度量算法进行了详细介绍.

3.1 动态时间规整

DTW 是一种动态规划算法, 能够通过非线性扭曲找到时间序列间的最优匹配路径并进行序列间的相似性度量. 同时, DTW 适用于单变量和多变量的时间序列. 简单起见, 本文中只考虑等长单变量时间序列.

给定两条时间序列 $P=\{p_1, p_2, \dots, p_n\}$ 和 $Q=\{q_1, q_2, \dots, q_n\}$. 定义 P, Q 之间的点对距离矩阵 $D \in R^{n \times n}$, 其中每个元素 $d(p_i, q_j)$, $1 \leq i, j \leq n$, 表示 p_i, q_j 之间的距离. 对于距离的度量通常使用欧几里得距离, 即 $d(p_i, q_j)=|p_i - q_j|$. 对齐 P, Q 的目的是找到一条规整路径 $W=(e_1, f_1), (e_2, f_2), \dots, (e_l, f_l)$, $n \leq l \leq (2n-1)$, 该路径将序列 P 中索引为 e_i 的点与序列 Q 中索引为 f_i 的点进行匹配. 最后求得的路径 W 满足 $\sum_{i=1}^l d(p_{e_i}, p_{f_i})$ 最小, 同时规整路径要求满足边界, 单调性, 连续性 3 个约束条件, 如公式 (4) 所示:

$$\begin{cases} (e_1, f_1) = (1, 1) \\ (e_l, f_l) = (n, n) \\ (e_{i+1}, f_{i+1}) - (e_i, f_i) \in \{(1, 0), (1, 1), (0, 1)\} \end{cases} \quad (4)$$

这里用 $\gamma(i, j)$ 表示序列 P, Q 间的累积距离, 其递归过程如公式 (5) 和公式 (6):

$$\gamma(i, j) = d(p_i, q_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \} \quad (5)$$

$$DTW(P, Q) = \gamma(n, n) \quad (6)$$

3.2 一种基于 HOG-1D 和 LBPT 加权的相似性度量算法

在一定的约束条件下, DTW 的确能够获得全局的最优对齐, 但它忽略了序列的局部结构信息. HOG-1D 这种基于梯度或斜率的局部特征描述子虽然着眼于局部信息, 但也存在对原始子序列编码不够精确的问题 (如图 3 所示). 鉴于此, 本文考虑利用 LBP 较高的局部特征鉴别能力来缓解传统的基于梯度或斜率的局部特征描述子对序列的局部结构信息表示不充分的问题, 提出了基于 HOG-1D 和 LBPT 加权的相似性度量算法. 给定两条时间序列 $P=\{p_1, p_2, \dots, p_n\}$, $P \in R^n$. 和 $Q=\{q_1, q_2, \dots, q_n\}$, $Q \in R^n$. 用 $P^G=\{p_1^G, p_2^G, \dots, p_n^G\}$, $p_i^G \in R^h$, $P^G \in R^{n \times h}$, $Q^G=\{q_1^G, q_2^G, \dots, q_n^G\}$, $q_i^G \in R^h$, $Q^G \in R^{n \times h}$ 表示 P 和 Q 的 HOG-1D 描述子序列; 用 $P^B=\{p_1^B, p_2^B, \dots, p_n^B\}$, $p_i^B \in R^{4^N}$, $P^B \in R^{n \times 4^N}$, $Q^B=\{q_1^B, q_2^B, \dots, q_n^B\}$, $q_i^B \in R^{4^N}$, $Q^B \in R^{n \times 4^N}$ 表示 P 和 Q 的 LBPT 描述子序列, 这里给出基于这两种特征描述子的相似性度量算法 (如公式 (7)–公式 (11) 所示):

$$d(p_i, q_j) = \sqrt{\delta d_{HOG-1D}(p_i, q_j) + (1-\delta) d_{LBPT}(p_i, q_j)} \quad (7)$$

$$\gamma(i, j) = d(p_i, q_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \} \quad (8)$$

$$LGBDTW(P, Q) = \gamma(n, n) \quad (9)$$

其中, $d_{HOG-1D}(p_i, q_j)$ 和 $d_{LBPT}(p_i, q_j)$ 分别表示时间序列点 i, j 的 HOG-1D 特征描述子之间距离和 LBPT 特征描述子之间的距离, 其计算方式如公式 (10) 和公式 (11):

$$d_{HOG-1D}(p_i, q_j) = \sum_{a=1}^h |p_{ia}^G - q_{ja}^G|^2 \quad (10)$$

$$d_{LBPT}(p_i, q_j) = \sum_{a=1}^{4^N} |p_{ia}^B - q_{ja}^B|^2 \quad (11)$$

其中, h 表示的是 HOG-1D 局部特征描述子的长度, 其大小随着时间序列点所对应的子序列被划分的间隔数发生变化. 在公式 (7) 中, $0 \leq \delta \leq 1$ 是一个平衡因子, 它反映的是在距离度量过程中 HOG-1D 和 LBPT 两种描述子所占比重; N 表示的是在计算子序列中点的 LBP 值时使用的近邻点对数. 由此得到基于 HOG-1D 和 LBPT 两种特征描

述子的加权动态时间规整算法 LGBDTW.

算法 2 给出了本文的相似性度量算法的伪代码. 其中第 1 行是相关数据的初始化; 第 2-4 行是对原始时间序列进行滤波; 第 5 到 10 行是基于 HOG-1D 及 LBPT 两种特征描述子的相似性度量. 需要注意的是, DTW 算法的时间复杂度为 $O(n^2)$, 而对于本文算法, 考虑到 N 、 k 等的实际设置值都要远小于 n , 所以计算 HOG-1D 及 LBPT 描述子时的复杂度可以近似表示为 $O(n)$, 同时 LGBDTW 相似性度量算法复杂度为 $O(n^2)$, 所以本文算法最终复杂度可以近似表示为 $O(n^2)$, 接近于 DTW.

算法 2. LGBDTW 相似性度量算法.

输入: P 、 Q , 滤波器窗口大小 w , 子序列长度 k , HOG-1D 描述子长度 h , 平衡因子 δ , 近邻点对数 N ;

输出: $LGBDTW(P, Q)$.

1. 初始化: 描述符序列 P^G, Q^G, P^B, Q^B 以及序列 P, Q 长度 $qlen$.
 2. for($j=1; j \leq qlen; j++$)
 3. 用公式 (1) 对 P, Q 中的点进行滤波.
 4. end for
 5. for($j=1; j \leq qlen; j++$)
 6. 以长度 k 提取 P, Q 的第 j 个点的子序列 P_j, Q_j .
 7. 用算法 1 计算 LBPT 描述子 $P^B[j], Q^B[j]$.
 8. 同时生成传统的 HOG-1D 描述子 $P^G[j], Q^G[j]$.
 9. end for
 10. 用公式 (7)–公式 (11) 计算 P, Q 的距离 $LGBDTW(P, Q)$
 11. return $LGBDTW(P, Q)$
-

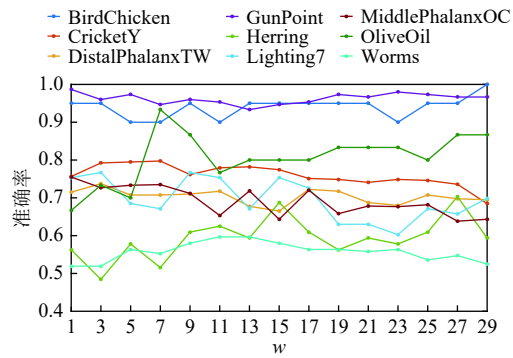
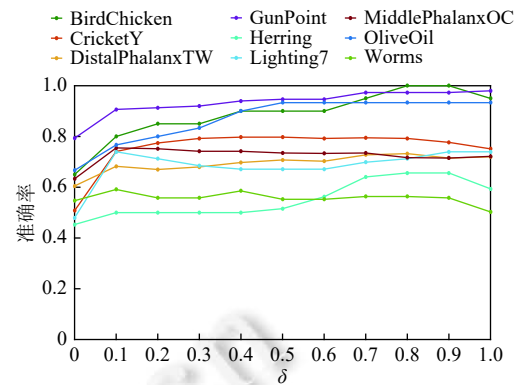
4 实验分析

本节对所提出算法的相关实验内容进行介绍, 实验中所使用的 73 个数据集均来源于 UCR 时间序列数据库^[35]. 这里首先介绍一些参数对实验的影响, 然后将本文提出的算法的分类效果与当前比较流行的分类器进行了比较.

4.1 参数分析及选取

正如前面算法中所述, 本文算法受到一些参数的影响. 简单起见, 参考文献 [11] 中的设置, 实验中将时间序列点的对应子序列长度 k 固定为 30; HOG-1D 长度 h 固定为 16; 此外, 因为 LBPT 描述子的长度随近邻点对数 N 的变化增长较快, 为方便测试, 我们将 N 固定为 1, 即每一个时间序列点的 LBPT 描述子为一个 4 维向量. 在保持这些参数不变的前提下, 本文分别测试了滤波器窗口 w 以及平衡因子 δ 对分类结果的影响. 当测试滤波器窗口参数 w 时, 将 δ 固定为 0.5; 当测试平衡因子 δ 时, 将 w 大小固定为 7.

首先我们在 9 个时间序列数据集上测试了滤波器窗口大小 w 在 [1, 29] 范围内变化时对分类准确率的影响, 测试结果如图 7 所示. 图 7 展示了随着 w 大小的变化 LGBDTW 分类准确率的变化趋势. 从图中可以看出, 在所测试的数据集中, 随着 w 的增加, 大多数数据集的准确率都有一个增减过程. 需要注意的是当 $w=1$ 时, 实质上是对原始时间序列进行滤波. 当 w 处于 11 到 23 之间时, 分类准确率相对较高且稳定, 而在 w 超过 23 之后, 部分数据集的准确率开始明显下降, 因此, 考虑到计算复杂性及准确率, 本文算法取 $w=15$. 而对于平衡因子 δ , 如图 8 所示, 在所测试的 9 个数据集上, 随着平衡因子 δ 的变化, 大多数数据集的准确率在 $\delta \in [0, 1]$ 之间有一个增减过程. 其中当 $\delta=0$ 时, 表示的是单独使用 LBPT 描述子时的分类准确率; 当 $\delta=1$ 时, 表示的是单独使用 HOG-1D 描述子时的分类准确率. 可以观察到, 在 δ 处于 0.6 到 0.9 之间时, 分类准确率相对稳定并达到最高. 因此, 简单起见, 我们在分类过程中将平衡因子 δ 固定为 0.8, 以达到较为理想的分类效果.

图7 分类准确率随滤波器窗口大小 w 的变化趋势图8 分类准确率随平衡因子 δ 的变化趋势

4.2 分类器性能分析

在本节中,我们将本文算法 LGBDTW 与 1NN 结合在 73 个 UCR 数据集上进行了分类测试,并将分类效果与现在比较流行的分类器进行了比较.这些算法的结果均采用的是 UCR 时间序列数据库或者是原论文中作者提供的结果.

4.2.1 与基于距离的分类器进行比较

本节将 LGBDTW 与其他几个当前比较流行的基于距离的分类方法进行了比较.例如,比较典型的有欧氏距离 (ED)、最长公共子序列 (LCSS)、DTW、DDTW、WDTW、复杂性-不变 DTW (CID)、MSM、TWE 以及 shapeDTW.表 1 展示的是以上分类器以及 LGBDTW 在 73 个数据集上的准确率对比 (由于篇幅原因,这里只展示 DTW 及其衍生类型算法的准确率).其中加粗表示 6 个分类器中在该数据集上表现最优的分类器,从表 1 可以看出,DTW、DDTW、WDTW、CID、shapeDTW、LGBDTW 分别在 8、10、9、18、16 和 25 个数据集上表现最好.在平均准确率方面,LGBDTW 为 80.2%,也优于其他的分类器.为了更加形象的展示所提算法与其他分类算法的分类效果,本文绘制了 LGBDTW 与其他 9 个算法在 73 个数据集上的分类准确率对比图 (如图 9,其中 x, y 坐标轴数值代表的是分类准确率).图中的每个红色点表示一个数据集,点落在斜线下方表示该数据集上 LGBDTW 的分类效果更好.

从图 9 可以看到,在分类准确率方面,LGBDTW 在大多数数据集上优于其他的基于距离的分类方法.在被测试的 73 个数据集中,同 DTW 比较,LGBDTW 在 49 个 (2 个等于) 数据集上优于 DTW,其中有 5 个数据集的准确率提升超过 20%.此外,同 DDTW、WDTW、CID、以及 shapeDTW 这些 DTW 的衍生算法相比,LGBDTW 分别在 49 (1 个等于)、42 (3 个等于)、43 (1 个等于)、47 (2 个等于) 个数据集上表现更好.而对于 ED、LCSS、MSM 以及 TWE 这几个方法的分类结果,LGBDTW 也分别在 48、47、36 (1 个等于)、42 (2 个等于) 个数据集上优于它们.需要注意的是,在 LGBDTW 与 MSM 的比较中,虽然 MSM 在准确率占优的数据集数量上比 LGBDTW 多 2 个,但如表 1 所示,LGBDTW 的最优准确率数量以及平均准确率都要高于 MSM.

为了更全面评价这 10 个分类器在多个数据集上的分类效果,本文还采用了 Demšar 所提出的临界差异图^[36] (如图 10) 来对多个分类器进行评分,其中评分越小表示分类效果越好.从图 10 可以看出,LGBDTW 所表现出来的分类效果跟 MSM 没有显著性差异,基本都是属于最好的.

此外,为了进一步分析不同数据特性 (即数据集类型) 对本文算法 LGBDTW 分类效果所产生的影响,我们利用表 1 分别对不同数据集类型下的 LGBDTW 分类准确率进行了比较分析.本文实验中所涉及到的数据集主要包括 8 种类型,即 device、audio、ecg、image、motion、sensor、simulated 以及 spectro.对表 1 的分析结果表明,LGBDTW 在 DEVICE、AUDIO 等类型数据集下的分类准确率大多数情况下相对其他类型数据比较低.具体地,DEVICE 类型数据集的准确率大多低于 0.7,例如 RefrigerationDevices 为 0.496,Computers 为 0.676.相反的,LGBDTW 在 simulated 以及 ECG 等类型数据集上的准确率多数超过了 0.9,例如 ECG 类型下的 ECGFiveDays 为 0.998,TwoLeadECG 为 0.970.

表 1 LGBDTW 与其他分类器分类准确率比较结果

数据集	DTW	DDTW	WDTW	CID	shapeDTW	LGBDTW
Adiac	0.603	0.583	0.617	0.627	0.731	0.696
ArrowHead	0.720	0.868	0.811	0.829	0.823	0.806
Beef	0.497	0.533	0.524	0.531	0.733	0.767
BeetleFly	0.779	0.812	0.804	0.806	0.800	0.700
BirdChicken	0.835	0.878	0.831	0.849	0.950	1.000
Car	0.666	0.735	0.720	0.714	0.867	0.917
CBF	0.993	0.564	0.993	0.984	0.920	0.991
ChlorineConcentration	0.646	0.681	0.648	0.649	0.645	0.617
CinCECGtorso	0.674	0.717	0.908	0.954	0.651	0.853
Coffee	0.989	0.958	0.986	0.989	0.964	0.964
Computers	0.700	0.700	0.687	0.707	0.644	0.676
CricketX	0.765	0.603	0.779	0.770	0.792	0.787
CricketY	0.738	0.545	0.750	0.725	0.774	0.792
CricketZ	0.772	0.582	0.784	0.776	0.792	0.800
DiatomSizeReduction	0.958	0.913	0.958	0.944	0.931	0.938
DistalPhalanxOAG	0.745	0.713	0.734	0.727	0.767	0.798
DistalPhalanxOC	0.761	0.759	0.754	0.753	0.772	0.765
DistalPhalanxTW	0.605	0.612	0.619	0.623	0.710	0.733
Earthquakes	0.703	0.715	0.695	0.694	0.742	0.761
ECG200	0.799	0.812	0.864	0.870	0.900	0.870
ECG5000	0.926	0.918	0.927	0.927	0.929	0.933
ECGFiveDays	0.760	0.667	0.824	0.823	0.943	0.998
FaceAll	0.943	0.917	0.960	0.956	0.762	0.794
FaceFour	0.849	0.719	0.860	0.833	0.909	0.921
FacesUCR	0.899	0.845	0.923	0.909	0.919	0.945
Fiftywords	0.681	0.694	0.765	0.774	0.758	0.798
FISH	0.763	0.891	0.817	0.813	0.949	0.943
GunPoint	0.876	0.982	0.956	0.951	0.993	0.973
Ham	0.692	0.657	0.747	0.715	0.543	0.591
Haptics	0.390	0.308	0.406	0.415	0.377	0.383
Herring	0.566	0.537	0.550	0.544	0.500	0.656
InlineSkate	0.395	0.470	0.404	0.428	0.384	0.418
InsectWingbeatSound	0.349	0.244	0.553	0.554	0.416	0.463
ItalyPowerDemand	0.923	0.886	0.934	0.950	0.897	0.943
LargeKitchenApps	0.797	0.778	0.795	0.783	0.840	0.853
Lighting2	0.823	0.662	0.837	0.826	0.885	0.869
Lighting7	0.698	0.550	0.754	0.719	0.767	0.712
MALLAT	0.942	0.918	0.945	0.954	0.938	0.838
Meat	0.971	0.759	0.971	0.980	0.900	0.850
MedicalImages	0.741	0.664	0.751	0.743	0.736	0.751
MiddlePhalanxOAG	0.569	0.575	0.566	0.572	0.740	0.738
MiddlePhalanxOC	0.714	0.724	0.753	0.777	0.750	0.717
MiddlePhalanxTW	0.505	0.505	0.509	0.499	0.571	0.566
MoteStrain	0.832	0.704	0.848	0.787	0.890	0.896
OliveOil	0.861	0.783	0.868	0.876	0.900	0.933
OSULeaf	0.617	0.869	0.643	0.660	0.868	0.826
PhalangesOC	0.752	0.752	0.763	0.765	0.739	0.754
Plane	1.000	0.999	1.000	0.995	1.000	1.000
ProximalPhalanxOAG	0.775	0.790	0.773	0.766	0.790	0.766
ProximalPhalanxOC	0.812	0.819	0.814	0.817	0.794	0.808

表 1 LGBDTW 与其他分类器分类准确率比较结果 (续)

数据集	DTW	DDTW	WDTW	CID	shapeDTW	LGBDTW
ProximalPhalanxTW	0.735	0.722	0.731	0.729	0.725	0.703
RefrigerationDevices	0.573	0.567	0.570	0.587	0.493	0.496
ScreenType	0.469	0.546	0.465	0.506	0.475	0.408
ShapeletSim	0.652	0.562	0.682	0.754	0.972	0.950
ShapesAll	0.764	0.849	0.811	0.818	0.888	0.872
SmallKitchenApps	0.640	0.636	0.679	0.680	0.699	0.757
SonyAIBORSurface1	0.801	0.769	0.811	0.915	0.807	0.724
SonyAIBORSurface2	0.846	0.862	0.853	0.893	0.826	0.833
Strawberry	0.955	0.960	0.954	0.955	0.949	0.951
Symbols	0.941	0.966	0.942	0.930	0.961	0.972
syntheticcontrol	0.991	0.567	0.989	0.979	0.847	0.980
ToeSegmentation1	0.738	0.739	0.728	0.718	0.899	0.864
ToeSegmentation2	0.818	0.829	0.862	0.844	0.862	0.877
Trace	1.000	1.000	1.000	0.995	1.000	1.000
TwoLeadECG	1.000	0.998	1.000	0.999	0.994	0.970
TwoPatterns	0.918	0.971	0.910	0.885	0.999	1.000
UWaveGestuLibraryY	0.728	0.676	0.775	0.787	0.642	0.647
wafer	0.984	0.975	0.996	0.994	0.990	0.994
Wine	0.884	0.848	0.885	0.891	0.463	0.833
WordsSynonyms	0.649	0.662	0.731	0.738	0.740	0.745
Worms	0.584	0.638	0.579	0.633	0.525	0.564
WormsTwoClass	0.682	0.709	0.677	0.736	0.713	0.696
yoga	0.849	0.831	0.858	0.858	0.883	0.881
Avg	0.761	0.736	0.783	0.787	0.790	0.802

为了分析造成以上差异的原因,我们分别选取 device 以及 ECG 类型数据集中的数据对序列结构进行了比较. DEVICE 类型数据集主要反映的是不同家用电器一天时间内的用电记录而多数家用电器在一天时间内的用电状态改变次数是非常有限的,其用电数据序列大多数时间是处于稳定的状态; ECG 类型数据集主要反映的是人体的心率活动的心电图,具有明显的局部波动情况. 图 11 展示的是 DEVICE 类型数据集中的 ScreenType 数据序列(序列类型包括 3 类,分别表示 3 种家用电器的用电记录)与 ECG 类型数据集中的 ECGFiveDays 数据序列(序列类型包括 2 类,表示同一个人不同两天中的心电图记录)的对齐结果比较,其中绿色箭头表示序列的局部上下波动. 图的上半部分表示的是 ECGFiveDays 数据集中两条同类序列的对齐,下半部分表示的是 ScreenType 数据集中两条同类序列的对齐. 从图中可以看出, ECGFiveDays 数据序列包含有相当数量的波动,即含有丰富的局部特征. 而 ScreenType 数据序列则含有极少量的波动,即含有极少的局部特征. 同时可以看出 ECGFiveDays 序列的对齐结果明显优于 ScreenType, 其中 ScreenType 的对齐出现了较为严重的局部错误对齐,这在分类过程中很容易导致序列距离度量出现偏差,最终造成错误分类. 由此可知,当待分类序列中包含的局部特征信息不足时, LGBDTW 算法的局部辨别能力很有可能会被削弱. 相反的,当待分类序列中包含丰富的局部特征信息时, LGBDTW 能够发挥出更好的分类性能.

4.2.2 与基于特征的分类器进行比较

在本节中,将对本文 LGBDTW 算法与目前流行的基于特征的时间序列分类器进行比较,例如,快速 shapelet (FS)^[37], 学习 shapelet (LS)^[38], DTW 特征 (DTW_F)^[39], 模式包 (BOP)^[40], 符号聚合近似向量空间模型 (SAXVSM)^[41] 等. 这里我们同样根据分类准确率对这些分类器进行了评分(如图 12), 从图中可以看到, LGBDTW 的评分结果基本上与 DTW_F 和 LS 相同, 并且明显优于 FS、BOP 和 SAXVSM 这 3 个分类器.

4.2.3 与基于深度学习的分类器进行比较

近些年深度学习也开始被广泛应用到时间序列分类问题中来. 本节中,我们将本文 LGBDTW 算法与现有的比较流行的基于深度学习的分类器进行比较,例如, MLP、FCN、ResNet^[23]、Encoder^[42]、MCNN^[22]、t-LeNet^[23]、

MCDCNN^[43]、Time-CNN^[24]以及 TWIESN^[44]等, 比较结果如图 13 所示. 从图 13 可以看到, LGBDTW 与 ResNet 和 FCN 模型的评分结果相近, 这表明它们的分类效果并没有显著性差异. 而这些分类器是目前基于深度学习的时间序列分类模型中分类效果最好的, 也就说明 LGBDTW 优于大多数的基于深度学习的时间序列分类模型, 从图 13 中其他分类模型的评分也可以看出这一点. 更重要的是, 深度学习模型往往伴随大量的参数调整, 在时间消耗方面明显高于 LGBDTW. 此外, LGBDTW 在可解释性方面跟这些基于深度学习的分类器相比是占有绝对优势的, 因为大多数此类模型可解释性较弱甚至不具有可解释性.

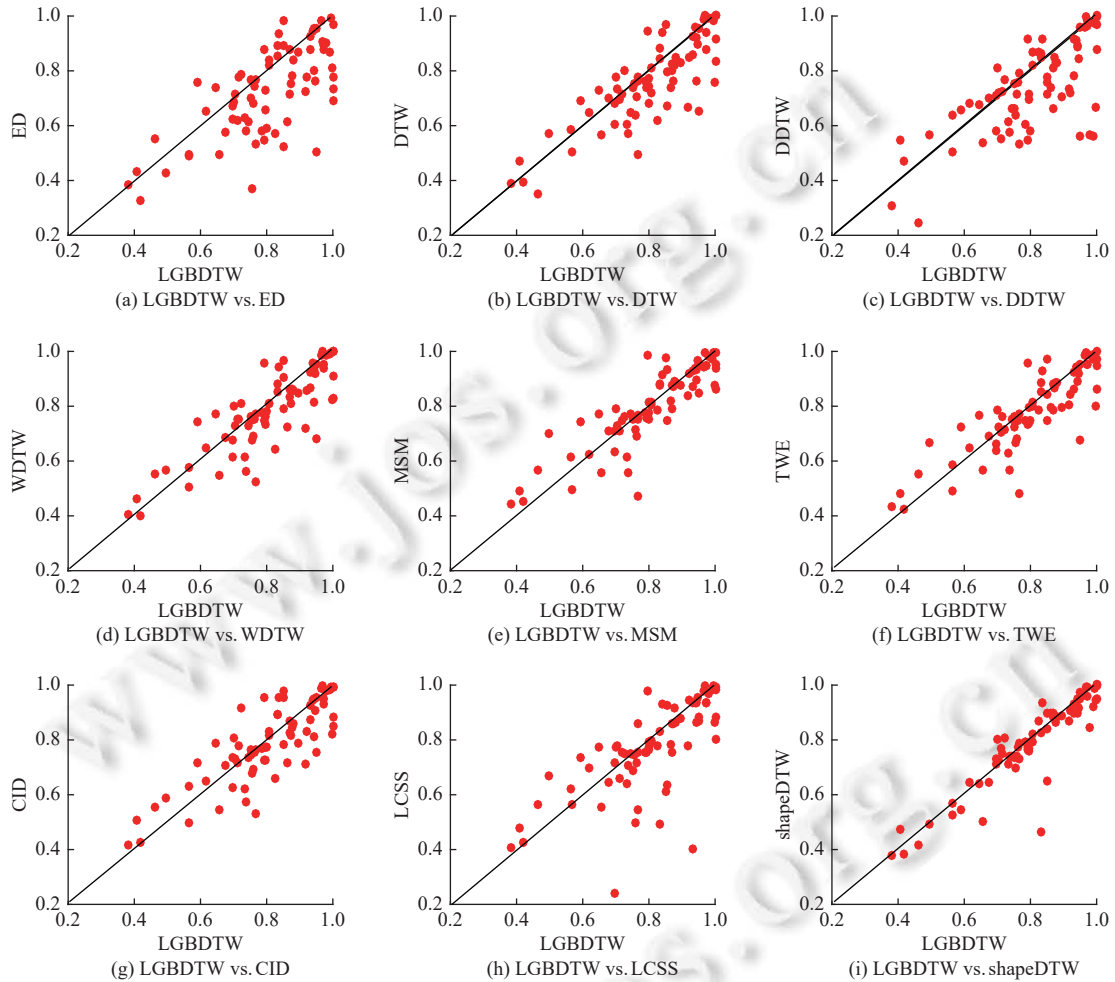


图 9 LGBDTW 与其他基于距离的分类器的分类正确率比较结果

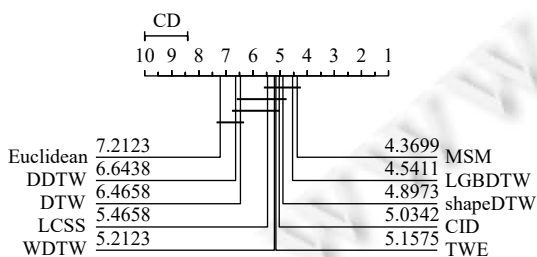


图 10 基于距离的分类器评分结果

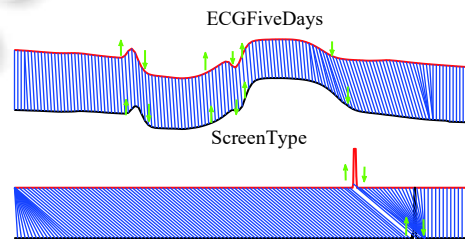


图 11 ECGFiveDays 与 ScreenType 数据集序列局部特征对比

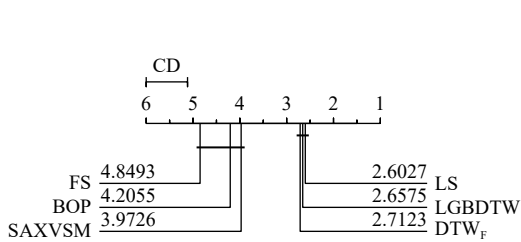


图 12 基于特征的分类器评分结果

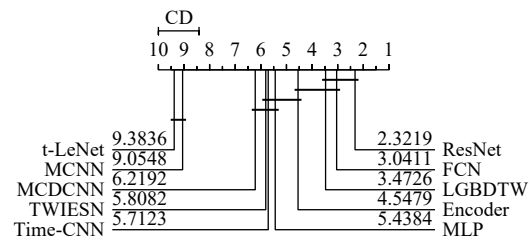


图 13 基于深度学习的分类器评分结果

4.2.4 与基于集成的分类器进行比较

在本节中,我们将本文的 LGBDTW 算法与当前比较流行的集成分类器进行比较. 集成分类器是通过集成一系列性能较弱的分类器,以达到更好的分类性能. 在时间序列领域,比较常见的集成分类器有很多,例如,时间序列森林 (TSF)^[16], 时间序列特征包 (TSBF)^[17], 学习模式相似性 (LPS)^[18], 动态集成 (EE) 以及 COTE^[38]等. 这里同样以临界差异图的方式将它们与 LGBDTW 进行了比较,结果如图 14 所示. 从图中可以看出, LGBDTW 与 EE 的分类效果没有显著性差异,但优于 TSF、TSBF 和 LPS. 此外,虽然 LGBDTW 比 COTE 和 EE 以及 BOSS 稍弱一些,但就运行速度上来讲,它比 BOSS, EE 或者 COTE 这种集成几种甚至几百种不同算法的分类器要快得多. 同时, LGBDTW 还可以集成到这些算法中来进一步提高它们的分类准确率.

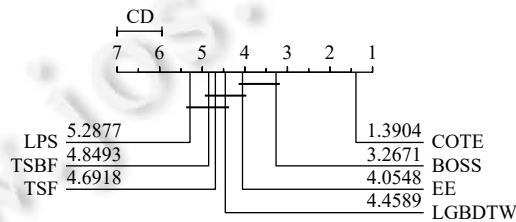


图 14 基于集成的分类器评分结果

4.3 运行时间

通过前面对分类准确率的比较可以看出 LGBDTW 优于大多数分类器. 虽然结果上比个别分类器要弱一些,但这并不意味着 LGBDTW 就整体弱于这些分类器. 因此,我们分别对 LGBDTW 以及几个在临界差异图中跟其评分差异较大的分类器进行了运行时间测试,这些分类器包括集成分类器 BOSS, EE 和 COTE. 对于基于深度学习的分类器,由于涉及大量的参数调整,这里将不再进行测试. 我们分别在 18 个相对较小的数据集上对分类器的运行时间进行了测试,测试结果如图 15 所示,我们对它们在每个数据集上的十次运行时间均值进行了比较. 从图中可以看出,与 EE 和 COTE 相比, LGBDTW 在运行时间上至少占有一个数量级的优势,而对于 BOSS 分类器, LGBDTW 也在多数数据集上的运行时间少于它.

4.4 实例分析

BirdChicken 是在文献 [10] 中作者提供的数据集,该数据集中的序列来分别提取自小鸟和小鸡两种动物的图像轮廓. 对该数据集的分类目的主要是通过图像轮廓序列分类区分小鸟和小鸡. 现有文献中基于距离的分类模型中在该数据集上分类准确率比较高的是 shapeDTW 模型,如表 1 所示,该模型准确率达到 95%. 然而, LGBDTW 在这个数据集上的准确率达到 100%. 图 16 中展示了 BirdChicken 数据集中两个同类序列分别使用 DTW、shapeDTW 以及 LGBDTW 对齐的效果. 其中每对绿色方框区域表示的是两条序列中具有相似的局部形状结构的部分. 从图中可以看出, DTW、shapeDTW 以及 LGBDTW 在序列的局部形状匹配效果上存在明显的差异. 在绿色方框表示的局部结构匹配过程中, DTW 和 shapeDTW 出现了不同程度的局部对齐错误,而 LGBDTW 则实现了较为准确的局部结构对齐,这表明在同样都能准确分类前提下,相比于 DTW 和 shapeDTW, LGBDTW 具有更好的可解释性,其分类结果更加可靠.

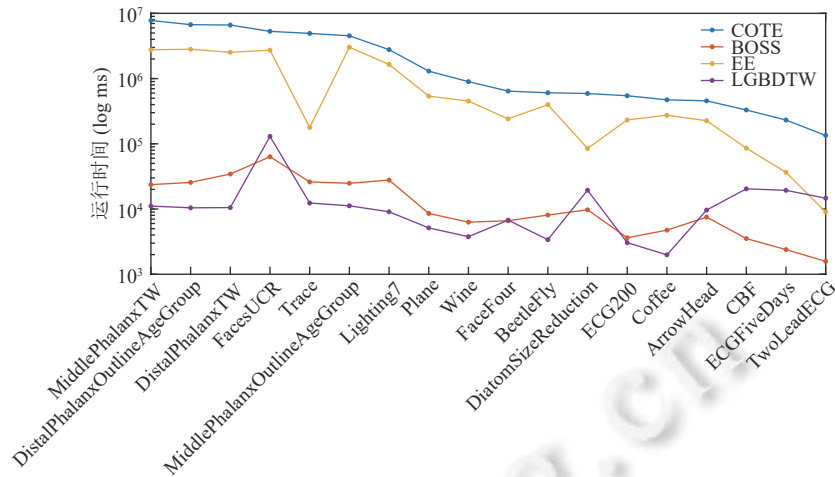


图 15 分类器运行时间比较

类似的, FacesUCR 数据集选取自 UCR 时间序列数据库, 该数据集序列记录的是不同人的脸部图像轮廓, 其序列分类目的是区分不同人的脸. 该数据集的特点是其数据序列中包含了多个起伏明显的峰、谷结构, 局部信息丰富. 这里我们同样用该数据集对 DTW、shapeDTW 和 LGBDTW 的对齐效果进行了测试. 图 17 表示的是该数据集中来自同一人的两条脸部轮廓序列采用这 3 个模型时的对齐结果. 如图中所示, 在绿色方框标注的序列局部结构中, 可以看到, 使用 DTW 和 shapeDTW 的序列对齐过程中存在明显的对齐错误. 相反的, 类似于 BirdChicken 数据集, LGBDTW 实现了该部分局部结构的准确对齐, 同样表现出了较好的可解释性, 这表明 LGBDTW 在该数据集上可以实现更加可靠的分类.

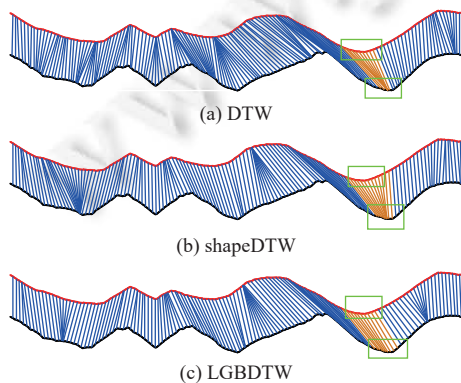


图 16 BirdChicken 数据集使用 DTW、shapeDTW 和 LGBDTW 的序列对齐比较

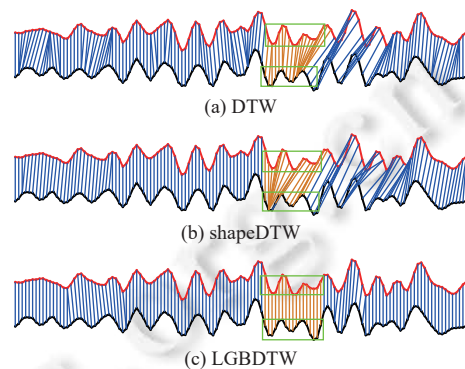


图 17 FacesUCR 数据集使用 DTW、shapeDTW 和 LGBDTW 的序列对齐比较

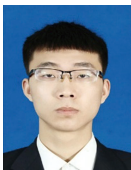
5 总结

本文提出了一种基于二进制模式的时间序列局部特征描述子 LBPT, 并在此基础上设计了基于 LBPT 和 HOG-1D 的时间序列相似性加权度量算法 LGBDTW. 相比于传统的基于距离和特征的相似性度量方法, LGBDTW 在不明显增加计算复杂性的前提下, 实现了更加精准的时间序列局部结构信息匹配. 此外, 多个数据集上的实验结果表明, 本文所提出的相似性度量算法有效的提升了时间序列的分类准确率, 分类效果较其他的基于距离和基于特征分类器有明显改善. 同时, 实例分析表明本文算法在可解释性方面也要优于大多数传统分类算法. 关于未来的工作, 我们的研究方向可能包括如何进行 LBPT 描述子与其他类型特征描述子之间的整合以及探索具有较强时间序列局部信息表示能力的新特征. 同时我们也希望本文所提出的分类算法可以扩展到更多的问题领域.

References:

- [1] Zakaria J, Mueen A, Keogh E, Young N. Accelerating the discovery of unsupervised-shapelets. *Data Mining and Knowledge Discovery*, 2016, 30: 243–281. [doi: [10.1007/s10618-015-0411-4](https://doi.org/10.1007/s10618-015-0411-4)]
- [2] Brill M, Fluschnik T, Froese V, Jain B, Niedermeier R, Schultz D. Exact mean computation in dynamic time warping spaces. *Data Mining and Knowledge Discovery*, 2019, 33: 252–291. [doi: [10.1007/s10618-018-0604-8](https://doi.org/10.1007/s10618-018-0604-8)]
- [3] 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. *Data Mining and Knowledge Discovery*, 2017, 31(3): 606–660. [doi: [10.1007/s10618-016-0483-9](https://doi.org/10.1007/s10618-016-0483-9)]
- [4] Berndt D, Clifford J. Using dynamic time warping to find patterns in time series. In: *KDD94: AAAI Workshop on Knowledge Discovery in Databases*. Seattle: KDD Workshop, 1994. 359–370.
- [5] Gavrilu DM, Davis LS. 3-D model-based tracking of human upper body movement: A multi-view approach. In: *Proc. of Int'l Symp. on Computer Vision-ISCV*. Coral Gables: IEEE Press, 1995. 272–277. [doi: [10.1109/ISCV.1995.477010](https://doi.org/10.1109/ISCV.1995.477010)]
- [6] 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](https://doi.org/10.1109/TASSP.1978.1163055)]
- [7] Rath TM, Manmatha R. Word image matching using dynamic time warping. In: *Proc. of the 2003 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2003. Madison: IEEE Press, 2003. 521–527. [doi: [10.1109/CVPR.2003.1211511](https://doi.org/10.1109/CVPR.2003.1211511)]
- [8] Hsu E, Pulli K, Popović J. Style translation for human motion. In: *Proc. of the 32nd ACM Int'l Conf. on Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH 2005)*. Los Angeles: ACM Press, 2005. 1082–1089. [doi: [10.1145/1186822.1073315](https://doi.org/10.1145/1186822.1073315)]
- [9] Boulnemour I, Boucheham B, Benloucif S. Improved dynamic time warping for abnormality detection in ECG time series. In: *Proc. of the 4th Int'l Conf. on Bioinformatics and Biomedical Engineering*. Granada: Springer Press, 2016. 242–253. [doi: [10.1007/978-3-319-31744-1_22](https://doi.org/10.1007/978-3-319-31744-1_22)]
- [10] Hills J, Lines J, Baranauskas E, Mapp J, Bagnall A. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 2014, 28(4): 851–881. [doi: [10.1007/s10618-013-0322-1](https://doi.org/10.1007/s10618-013-0322-1)]
- [11] Zhao JP, Itti L. ShapeDTW: Shape dynamic time warping. *Pattern Recognition*, 2018, 74: 171–184. [doi: [10.1016/j.patcog.2017.09.020](https://doi.org/10.1016/j.patcog.2017.09.020)]
- [12] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: *Proc. of the 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego: IEEE Press, 2005. 886–893. [doi: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177)]
- [13] Ojala T, Pietikäinen M, Maenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002, 24(7): 971–987. [doi: [10.1109/tpami.2002.1017623](https://doi.org/10.1109/tpami.2002.1017623)]
- [14] Marteau PF. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2009, 31(2): 306–318. [doi: [10.1109/TPAMI.2008.76](https://doi.org/10.1109/TPAMI.2008.76)]
- [15] Stefan A, Athitsos V, Das G. The move-split-merge metric for time series. *IEEE Trans. on Knowledge and Data Engineering*, 2013, 25(6): 1425–1438. [doi: [10.1109/TKDE.2012.88](https://doi.org/10.1109/TKDE.2012.88)]
- [16] Deng HT, Runger G, Tuv E, Vladimir M. A time series forest for classification and feature extraction. *Information Sciences*, 2013, 239: 142–153. [doi: [10.1016/j.ins.2013.02.030](https://doi.org/10.1016/j.ins.2013.02.030)]
- [17] Baydogan MG, Runger G, Tuv E. A bag-of-features framework to classify time series. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2013, 35(11): 2796–2802. [doi: [10.1109/TPAMI.2013.72](https://doi.org/10.1109/TPAMI.2013.72)]
- [18] Baydogan MG, Runger G. Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery*, 2016, 30(2): 476–509. [doi: [10.1007/s10618-015-0425-y](https://doi.org/10.1007/s10618-015-0425-y)]
- [19] Nguyen TL, Gsponer S, Ilie I, Ifrim G. Interpretable time series classification using all-subsequence learning and symbolic representations in time and frequency domains. *arXiv: 1808.04022*, 2018.
- [20] Schäfer P. The BOSS is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 2015, 29(6): 1505–1530. [doi: [10.1007/s10618-014-0377-7](https://doi.org/10.1007/s10618-014-0377-7)]
- [21] Guennec AL, Malinowski S, Tavenard R. Data augmentation for time series classification using convolutional neural networks. In: *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*. Riva Del Garda, 2016.
- [22] Cui ZC, Chen WL, Chen YX. Multi-scale convolutional neural networks for time series classification. *arXiv: 1603.06995*, 2016.
- [23] Wang ZG, Yan WZ, Oates T. Time series classification from scratch with deep neural networks: A strong baseline. In: *Proc. of the 2017 Int'l Joint Conf. on Neural Networks (IJCNN)*. Anchorage: IEEE Press, 2017. 1578–1585. [doi: [10.1109/IJCNN.2017.7966039](https://doi.org/10.1109/IJCNN.2017.7966039)]
- [24] Zhao BD, Lu HZ, Chen SF, Liu JL, Wu DY. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 2017, 28(1): 162–169. [doi: [10.21629/JSEE.2017.01.18](https://doi.org/10.21629/JSEE.2017.01.18)]
- [25] Candan KS, Rossini R, Wang XL, Sapino ML. SDTW: Computing DTW distances using locally relevant constraints based on salient

- feature alignments. Proc. of the VLDB Endowment, 2012, 5(11): 1519–1530. [doi: [10.14778/2350229.2350266](https://doi.org/10.14778/2350229.2350266)]
- [26] Zhang Z, Tavenard R, Bailly A, Tang XT, Tang P, Corpetti T. Dynamic time warping under limited warping path length. Information Sciences, 2017, 393: 91–107. [doi: [10.1016/j.ins.2017.02.018](https://doi.org/10.1016/j.ins.2017.02.018)]
- [27] Keogh EJ, Pazzani J. Derivative dynamic time warping. In: Proc. of the 2001 SIAM Int'l Conf. on Data Mining (SDM). Chicago: SDM Press, 2001. 1–11. [doi: [10.1137/1.9781611972719.1](https://doi.org/10.1137/1.9781611972719.1)]
- [28] 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](https://doi.org/10.1016/j.patcog.2010.09.022)]
- [29] Batista GEAPA, Keogh EJ, Tawaw OM, De Souza VMA. CID: An efficient complexity-invariant distance for time series. Data Mining and Knowledge Discovery, 2014, 28(3): 634–669. [doi: [10.1007/s10618-013-0312-3](https://doi.org/10.1007/s10618-013-0312-3)]
- [30] Zhao JP, Itti L. Classifying time series using local descriptors with hybrid sampling. IEEE Trans. on Knowledge and Data Engineering, 2016, 28(3): 623–637. [doi: [10.1109/TKDE.2015.2492558](https://doi.org/10.1109/TKDE.2015.2492558)]
- [31] Xu Q, Yang J, Ding SY. Texture segmentation using LBP embedded region competition. Electronic Letters on Computer Vision and Image Analysis, 2005, 5(1): 41–47. [doi: [10.5565/rev/elcvia.83](https://doi.org/10.5565/rev/elcvia.83)]
- [32] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006, 28(12): 2037–2041. [doi: [10.1109/TPAMI.2006.244](https://doi.org/10.1109/TPAMI.2006.244)]
- [33] Li SZ, Zhang L, Liao SC, Zhu XX, Chu RF, Ao M, He R. A near-infrared image based face recognition system. In: Proc. of the 7th Int'l Conf. on Automatic Face and Gesture Recognition (FGR06). Southampton: IEEE Press, 2006. 455–460. [doi: [10.1109/FGR.2006.13](https://doi.org/10.1109/FGR.2006.13)]
- [34] Satpathy A, Jiang XD, Eng HL. LBP-based edge-texture features for object recognition. IEEE Trans. on Image Processing, 2014, 23(5): 1953–1964. [doi: [10.1109/TIP.2014.2310123](https://doi.org/10.1109/TIP.2014.2310123)]
- [35] <http://www.timeseriesclassification.com>
- [36] Demšar J. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 2006, 7: 1–30.
- [37] Rakthanmanon T, Keogh E. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: Proc. of the 2013 SIAM Int'l Conf. on Data Mining (SDM). Austin: SIAM Press, 2013. 668–676. [doi: [10.1137/1.9781611972832.74](https://doi.org/10.1137/1.9781611972832.74)]
- [38] Grabocka J, Schilling N, Wistuba M, Thieme LS. Learning time-series shapelets. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2014. 392–401. [doi: [10.1145/2623330.2623613](https://doi.org/10.1145/2623330.2623613)]
- [39] Kate RJ. Using dynamic time warping distances as features for improved time series classification. Data Mining and Knowledge Discovery, 2016, 30(2): 283–312. [doi: [10.1007/s10618-015-0418-x](https://doi.org/10.1007/s10618-015-0418-x)]
- [40] Lin J, Khade R, Li Y. Rotation-invariant similarity in time series using bag-of-patterns representation. Journal of Intelligent Information Systems, 2012, 39(2): 287–315. [doi: [10.1007/s10844-012-0196-5](https://doi.org/10.1007/s10844-012-0196-5)]
- [41] Senin P, Malinchi S. SAX-VSM: Interpretable time series classification using SAX and vector space model. In: Proc. of the 13th IEEE Int'l Conf. on Data Mining. Dallas: IEEE Press, 2013. 1175–1180. [doi: [10.1109/ICDM.2013.52](https://doi.org/10.1109/ICDM.2013.52)]
- [42] Serrà J, Pascual S, Karatzoglou A. Towards a universal neural network encoder for time series. arXiv: 1805.03908, 2018.
- [43] Zheng Y, Liu Q, Chen EH, Ge Y, Zhao JL. Time series classification using multi-channels deep convolutional neural networks. In: Proc. of the 15th Int'l Conf. on Web-Age Information Management. Springer, 2014. 298–310. [doi: [10.1007/978-3-319-08010-9_33](https://doi.org/10.1007/978-3-319-08010-9_33)]
- [44] Tanisaro P, Heidemann G. Time Series classification using time warping invariant echo state networks. In: Proc. of the 15th IEEE Int'l Conf. on Machine Learning and Applications (ICMLA). Anaheim: IEEE Press, 2016. 831–836. [doi: [10.1109/ICMLA.2016.0149](https://doi.org/10.1109/ICMLA.2016.0149)]



郝石磊(1995–), 男, 博士生, 主要研究领域是数据挖掘, 机器学习.



刘海洋(1987–), 男, 博士, 讲师, CCF 专业会员, 主要研究领域是数据挖掘, 机器学习.



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