软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(12):4411−4428 [doi: 10.13328/j.cnki.jos.006346]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

时间序列可变尺度的时频特征求解及其分类*

魏池璇[1,2], 王志海[1,2], 原继东[1,2], 林钱洪[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: 原继东, E-mail: yuanjd@bjtu.edu.cn

摘 要: 对于许多实际应用来说, 获取多个不同窗口尺度上的模式, 有助于发现时间序列的不同规律性特征. 同时, 通过对时间序列时域和频域两方面的分析, 有助于挖掘更多的知识. 提出了一种新的基于可变尺度的时域频域辨别性特征挖掘方法以及应用于分类的算法. 主要采用了不同尺度窗口、符号聚合近似技术以及符号傅里叶近似技术等, 以有效地发掘时间序列不同尺度时域频域模式; 与此同时, 使用统计学方法挖掘部分最具辨别性的特征用于时间序列分类, 有效地降低了算法时间复杂度. 在多个数据集上的对比实验结果, 说明了该算法具有较高的准确率; 在真实数据集上的解析, 表明了该算法具有更强的可解释性. 同时, 该算法可扩展应用到多维时间序列分类问题中.

关键词: 时间序列; 模式挖掘; 时间序列符号化; 可解释性

中图法分类号: TP301

中文引用格式: 魏池璇, 王志海, 原继东, 林钱洪. 时间序列可变尺度的时频特征求解及其分类. 软件学报, 2022, 33(12): 4411−4428. http://www.jos.org.cn/1000-9825/6346.htm

英文引用格式: Wei CX, Wang ZH, Yuan JD, Lin QH. Time Series Pattern Discovery and Classification with Variable Scales in Time-frequency Domains. Ruan Jian Xue Bao/Journal of Software, 2022, 33(12): 4411−4428 (in Chinese). http://www.jos.org.cn/1000-9825/6346.htm

Time Series Pattern Discovery and Classification with Variable Scales in Time-frequency Domains

WEI Chi-Xuan[1,2], WANG Zhi-Hai[1,2], YUAN Ji-Dong[1,2], LIN Qian-Hong[1,2]

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

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

Abstract: For many real-world applications, capturing patterns at diverse window scales can help to discover the different periodicity of time series. At the same time, it is helpful to gain more knowledge by analyzing time series from both time-domain and frequency-domain. This study proposes a novel method to detect distinctive patterns at variable scales in time-domain and frequency-domain of time series, and discuss its application on classification. This method integrates multiple scales, the symbolic approximation and symbolic Fourier approximation techniques to explore multi-scales and multi-domain patterns efficiently in time series. Meanwhile, statistical method is applied to select some of the most discriminative patterns for time series classification, which also can effectively reduce time complexity of the algorithm. The experiments performed on various datasets demonstrate that the proposed method has higher accuracy and better interpretability. In addition, it can be extended to multi-dimensional time series easily.

Key words: time series; pattern mining; time series symbolic representation; interpretability

时间序列数据是实际应用问题中重要的常见数据类型之一, 例如农作物每年销量产量数据、飞机的传感器数据以及病人的心电图等. 时间序列分类问题是数据挖掘技术解决的主要问题之一, 其存在于现实生活中的诸多领域. 例如: 在生物医学领域, 通过对正常心电图和心肌梗塞病人心电图的数据分析, 能够快速诊断

* 基金项目: 国家自然科学基金(61771058); 北京市自然科学基金(4214067)
收稿时间: 2020-08-12; 修改时间: 2020-11-16; 采用时间: 2021-04-14

患者是否心肌梗塞[1]; 在生态环境领域, 获取不同气候的雨量计观测值并进行训练, 能够协助气象人员预测天气情况[2]; 在图像声音领域, 利用时间序列分类方法来进行语音识别、动作识别[3]等. 基于这些广泛的应用, 时间序列分类问题受到了越来越多的关注. 与传统的分类方法类似, 时间序列分类是通过对给定类标的时间序列数据集进行训练学习得到不同类别的差异性特征, 进而能够为未知类标的时间序列数据指定一个类别. 但与传统方法不同的地方在于: 时间序列数据的属性之间具有关联性、次序性且其中包含了丰富的特征信息, 从而导致传统的分类方法难以有效应用在时间序列分类问题中. 因此, 如何快速有效地对时间序列数据进行分类, 成为了当前研究热点.

解决时间序列分类问题最直接的策略是, 如何利用样本间的差异性. 通常使用不同的距离度量方法, 结合最近邻分类器来构建基本框架[4]. 例如: 使用基于欧式距离(Euclidean distance, ED)、基于动态时间扭曲(dynamic time warpping, DTW)或基于局部加权 DTW[5]的距离度量方法, 根据时间序列的相似性或者非相似性度量来匹配时间序列. 由于这些方法直接使用原始时间序列数据进行分类, 因此存在训练过程中时间复杂度高以及可解释性较差的问题. 另一种策略是不直接使用全部的原始时间序列数据, 而是在原始数据的基础上做相应的转换. 这种策略能够减少噪音的影响, 从而更容易捕获到不同时间序列数据的规律性特征, 可有效提高分类准确率并具有较强的可解释性. 这类方法可分为基于模式的方法和基于字典的方法. 本文的研究重点在于将两种方法相互结合, 即使用基于字典的方法来挖掘最具辨别性的模式.

基于模式的方法是从原始时间序列数据中提取最具辨别性的子序列, 例如 shapelets[6], 可以直接使用这些 shapelets 来建立分类模型或者基于 shapelets 将原始的序列映射到新的特征空间, 从而再建立分类器. 例如: Ma 等人提出使用动态对抗 shapelet 网络(adversarial dynamic shapelet networks, ADSN)来获取 shapelets, 同时采用对抗训练策略来保证生成的 shapelets 与实际子序列相似[7]; Zhao 等人通过正则化 shapelet 学习框架来获取 shapelets, 该方法能够有效降低时间复杂度[8]; Li 等人提出使用时间序列关键点的方式来提取 shapelets 后选项, 依据这些 shapelets 来构建决策树进行时间序列分类[9]; Kramakum 等人通过信息熵来发现时间序列 shapelets[10]; Ji 等人通过快速 shapelets 选择算法来找到最优的 shapelets 候选项[11]; Rakthanmanon 等人提出的快速 shapelets(fast shapelet, FS)算法是利用符号化技术将原始序列转化到低维的符号表示, 进而在低维空间中发掘有效的候选子序列, 再到原始空间进行验证[12]. 但是这些方法不能兼顾类中辨别性模式出现的频率. 而基于字典的分类方法通过建立模式的频率计数或其他统计特征来解决这些问题.

基于字典的方法是通过滑动窗口提取原始序列中所有的子序列, 然后基于这些子序列来构造直方图统计每个子序列出现的频率, 最后将这些直方图作为新的特征空间用于分类. 例如, Lin 等人提出的模式袋(bag of patterns, BOP)模型将词袋模型应用于时间序列分类[13]. BOP 模型首先使用符号近似估计(symbolic aggregate approximation, SAX)技术将时间序列转换成为一系列时域空间的单词集合, 然后用直方图统计单词出现的频率[14]. 在此基础之上, Senin 等人提出了符号近似估计向量空间模型(symbolic aggregate approximation-vector space model, SAX-VSM). 与 BOP 模型类似, SAX-VSM 也使用 SAX 技术将时间序列转化为单词序列集合[15]. 两者的主要区别在于, SAX-VSM 对每一个类而不是对每一个序列生成单词频率向量. Nguyen 等人提出的 SAX- VFSEQL 和 SAX-VESEQL 模型也同样是基于 SAX 符号化表示来对时间序列进行分类, 同时, 该模型能够有效地降低分类算法的时间复杂度[16]. 此外, Schäfer 等人提出的 BOSS(bag of SFA symbols)模型是利用离散傅里叶变换将时间序列转化到频域进行分析[17], BOSS 使用符号傅里叶近似(symbolic Fourier approximation, SFA)技术进行单词转换, 同时使用直方图统计单词的频率作为输入的特征空间[18]. Middlehurst 等人提出了 BOSS 的改进版本 RBOSS, 该算法通过随机选择参数的方式提高了 BOSS 的运行效率[19]; Large 等人具体分析了 BOSS 算法与 BOP 算法的主要区别[20]; 其他使用 SFA 表示的还有 BOSSVS(bag-of-SFA symbols in vector space)[21], WEASEL (word extraction for time series classification)[22]算法及其改进算法[23]等. 基于字典的方法是在时间序列的符号表示上建立分类器, 因此, 这种方式能够有效地提高分类准确率.

模式挖掘往往和时间序列所呈现出来的规律性有一定的关系, 这种规律性可以通过某些特定子序列来反映. 在现实世界的时间序列数据中, 不同的时间序列呈现出不同长度的规律性子序列. 例如在加州交通运输

数据集中(http://pems.dot.ca.gov), 该数据集记录了加利福尼亚城市随时间变化的交通信息, 其中包含不同规律的模式, 即从几个小时的高峰时间的模式到每周 5 天的稳定模式. 在气象数据分析中[24], 以几个小时为周期的模式有利于短期的气候预报, 以几个月为周期的模式则代表某种季节性现象, 并且这些不同长度的模式可能同时出现在一个时间序列中. 因此, 挖掘不同长度的模式是很有必要的. 现有的一些方法[14,16,18]采用固定的窗口尺度挖掘模式, 但这种方式不适用于更一般的情况. 例如, 图 1 中描述了挖掘固定尺度模式的局限性. ECGFiveDays 是一组心电图时间序列数据, 它记录了患者在两天中的心电波动情况, 每一天代表一个类. 第 1 个模式(图中左侧较大方框)的长度为 40, 类别 2 在该区间相对较高具有辨别性; 第 2 个模式(图中右侧较小方框)的长度为 20, 类别 1 在该区间的凸起较延迟具有辨别性. 如果使用固定尺度的模式发现方法只能得到这些模式中的部分, 且当尺度不确定的时候, 这些方法准确率较低. 为克服这种局限性, 我们提出一种可变尺度的算法来挖掘不同长度的模式. 同时, 为了获取更多有效的规律性子序列, 本文分别从时域和频域来分析时间序列数据.

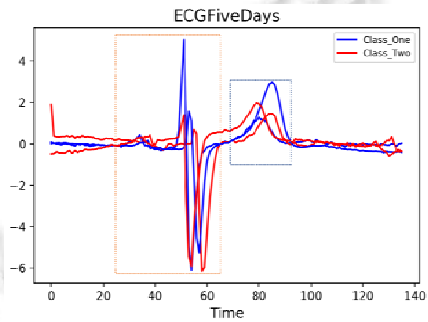

图 1 两种不同尺度的模式示例

在时域方面, SAX 符号化技术被广泛应用于时间序列分类领域来提取时间序列时域特征[14−16,25]. 一般是基于固定长度的滑动窗口来提取所有子序列, 并将子序列构建为 SAX 单词. 通过对单词出现的频率进行分析并构建分类器, 从而有效地对时间序列数据进行分类. 与此同时, 这种方式能够在保持时间序列规律性的前提下, 有效地缓解原始序列维度较高问题. 在频域方面, 基于傅里叶变换的 SFA 符号化技术也被成功应用于提取时间序列频域上的特征, 例如 WEASEL[22]和 BOSS[17]算法, 该技术使用离散傅里叶变换将时域信息映射到频谱, 通过对系数进行离散化来获取时间序列的频域信息. 这种方式能够获取时间序列的全局信息, 可以有效地提高分类准确率. 但这些方法在一定程度上存在信息丢失问题, 因而本文通过融合时域信息和频域信息来减少符号化过程中的信息丢失.

综上所述, 本文提出一种基于可变尺度的时域和频域模式挖掘方法. 首先, 考虑到不同时间序列的周期性不同, 并且每个序列可能有多个周期性特征, 在算法中采用不同尺度窗口的模式挖掘方法, 这样不仅能够挖掘不同的周期性特征, 还可以减少模型参数; 其次, 从时域和频域对时间序列进行符号表示, 使用 F 检验挖掘最具辨别性模式; 进而, 基于辨别性模式建立线性时间复杂度的时间序列分类算法, 快速准确地对时间序列进行分类; 此外, 本文模型还能够可视化地展示辨别性模式, 使得分类结果具有可解释性; 最后, 通过在多个数据集上的对比实验, 说明了其具有较高的分类准确率.

本文第 1 节给出基本定义与所采用的数学符号形式. 第 2 节阐述了本文挖掘关键特征的过程以及分类算法的构建. 第 3 节是实验设计, 并通过实验结果分析来说明本文模型的有效性. 第 4 节通过在真实数据集上的实例解析, 展示模型的可解释性. 最后进行总结.

1 基本定义与数学符号

定义 1. 设时间序列数据集 S 是由 N 个实例组成. 每一个实例 T 是由 n 个实际观测值 $t_1,t_2,\ldots,t_n$ 组成, 则 $T=\{t_1,t_2,\ldots,t_n\}$被称为一维时间序列, $t_i\in R$.

定义 2. 设时间序列中滑动窗口的长度为 w, 则序列 T 中连续的 w 个值组成的序列$\{t_i,t_{i+1},\ldots,t_{i+w-1}\}$称为时间序列 T 的子序列, 其中, $1\leqslant i\leqslant n+w-1$. 任意长度为 n 的时间序列包含 $n-w+1$ 个长度为 w 的子序列.

定义 3. 对于多维时间序列, 给定一个具有 N 个实例 H 维的多维时间序列数据集 S, 其中每一个维度长度均为 n, 则 $x_i(t)$; $[i=1,2,\ldots,H;\ t=1,2,\ldots,n]$称为多维时间序列. 其中, $x_i(t)$表示 i 索引时间点 t 上第 i 维变量 x_i 的值.

定义 4. 给定含有 N 个实例的时间序列训练集 $S=\{T_1,T_2,\ldots,T_N\}$, 其中每一个实例 T 都有 n 个观测值和 1 个类标 $T=\{t_1,t_2,\ldots,t_n\}$, 且训练集中一共有 C 个类, 那么时间序列分类任务的目标就是在训练集 S 中学习到时间序列观测值到类值的映射关系.

定义 5. 给定一个长度为 n 的时间序列 $T=\{t_1,t_2,\ldots,t_n\}$, 利用不同的时间序列符号化技术, 将 T 转换成一个维度为 $m(m<<n)$的符号串, 这个过程称之为时间序列的符号化表示.

定义 6. SAX 技术首先将序列按照单词长度 l 分割为若干个区间, 然后求每个区间的平均值, 最后, 基于高斯分布将其进行离散化, 其中, 该字符串中单词的字母来自于字母表 c. 例如, 图 2(左)所示展示了长度为 112 的序列进行 SAX 符号化的示例图, 其中, 单词长度为 7.

定义 7. SFA 技术则是对时间序列 T 经过傅里叶变换后的系数进行离散化处理. 此外, SFA 在进行离散化时会针对每个系数采用不同的离散化分割点. 例如, 图 2(右)所示, 展示了一个序列进行 SFA 离散化过程, 其单词长度为 4.

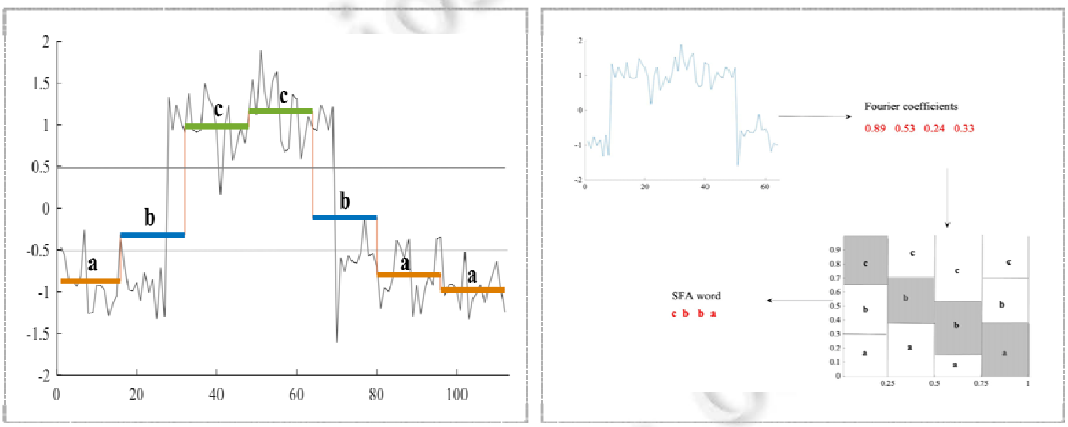

图 2 SAX 和 SFA

定义 8. 给定长度为 w 的时间序列子序列, 设单词长度为 l, 字母表大小为 c, 将子序列经过时间序列符号化表示成长度为 $l(l<<w)$的字符串, 则该字符串称之为时间序列的模式.

基于上述基本定义, 规范本文所涉及的符号见表 1.

表 1 符号表

符号	符号含义	符号	符号含义
S	时间序列数据集	r	滑动窗口个数
N	数据集中实例个数	l	单词长度
n	时间序列长度	m	所有符号的集合
T	数据集中每一个实例	D	最具辨别性模式集合
C	类属性个数	k	模式个数
H	时间序列维度	c	字母表大小
w	滑动窗口长度	p	不同的置信水平分位表

2 算法的实现

在本节, 我们将详细描述本文提出的时间序列多尺度的时域与频域模式表示过程及其最具辨别性特征挖掘过程, 并在此基础上阐述时间序列分类模型的构建.

2.1 时间序列的模式表示

本文模型将时间序列数据转换成一系列模式表示袋, 这些模式不仅能够完整表示时间序列的关键信息, 而且可以降低时间序列维度提高运行效率. 首先获取周期性信息, 时间序列不同的规律性特征往往体现在不同长度的子序列上, 并且不同类别的规律性特征也存在着明显的差异. 本模型通过使用多尺度的滑动窗口来获取时间序列不同长度的规律性子序列, 与此同时, 通过直方图统计不同的周期性子序列在时间序列上出现的频率. 其次, 获取时域信息, 通过SAX技术, 将周期性子序列转换成不同长度的单词, 获取序列丰富的局部时域信息; 在频域上, 通过 SFA 技术将子序列映射到频域的相关系数离散化成不同长度的单词, 获得序列全局的趋势信息. 另外, 本文通过融合子序列的时域与频域信息来解决符号化转换过程中的信息丢失问题.

具体来说, 如图 3 的“阶段一”所示, 设给定数据集中的 3 个时间序列实例 T_1, T_2 和 T_3, 这 3 个时间序列分别属于不同的类, 如“阶段一”的左边方框所示. 其次, 给定 r 个不同尺度的滑动窗口 w, 为了具体描述步骤, 我们列举了 20, 35 和 50 这 3 个不同的尺度, 当滑动窗口长度 w=20 时, 通过在时间序列 T_1, T_2 和 T_3 上分别滑动长度为 20 的窗口来提取子序列, 并将子序列转换成不同长度的 SAX 单词(小写字母)和 SFA 单词(大写字母), 将这些单词放入无序集合中, 可以获得原始时间序列 T_1, T_2 和 T_3 的部分单词表示袋. 依据这些单词表示袋, 将时间序列 T_1, T_2 和 T_3 转换成单词在对应序列上出现的频率, 为之后挖掘最具辨别性的模式做准备, 如阶段一的右边直方图所示. 与此同时, w=35 和 w=50 获取单词表示袋的过程与其类似. 由此, 当给定 r 个窗口时, 每个时间序列 T 都可以转化为 $2r$ 个单词表示袋. 接下来, 我们将针对每个词袋选择具有辨别性的单词作为辨别性模式, 并基于这些辨别性模式设计相应的分类模型.

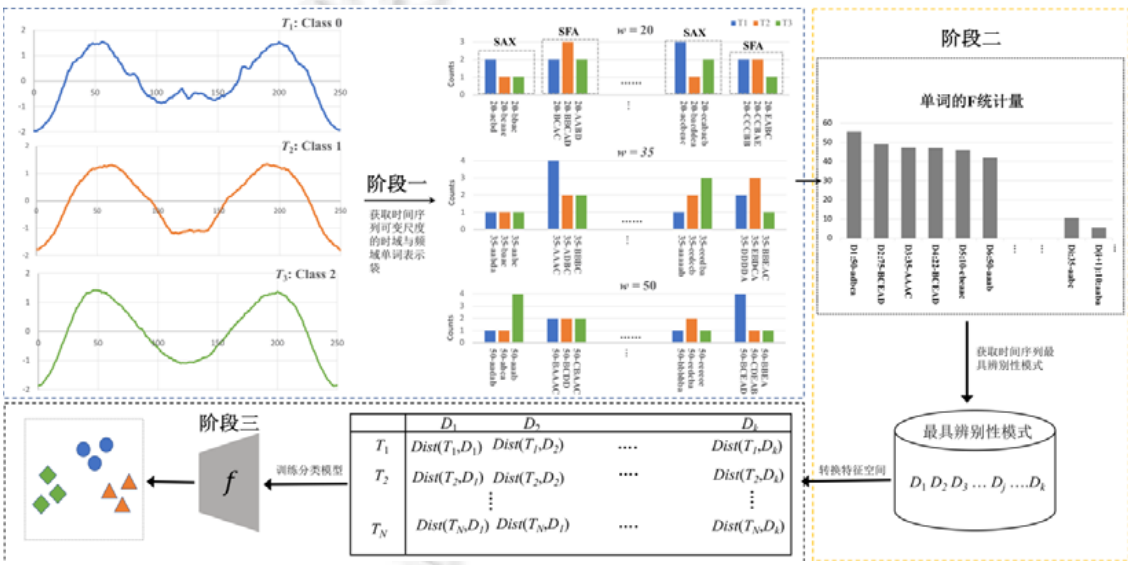

图 3 时间序列模式挖掘与分类框架

2.2 辨别性模式表示及其挖掘

辨别性模式通常指时间序列中最具辨别性的子序列, 而本文中的辨别性模式是指第 2.1 节中产生的所有单词中最具辨别性的单词集合. 由于时间序列通常具有较高的维度, 这也导致最终的生成的单词数量较大. 因此, 如何快速有效地评价每个单词的鉴别性, 是本文的研究重点之一. 首先, 本文模型使用统计学中 F 检验的方式来为每一个单词的辨别性评分; 然后, 根据设计的模式选择方法, 选择评分较高的若干单词作为时间序列辨别性模式.

F 检验是利用服从 F 分布的统计量进行假设检验的方法, 主要用来检验两个或者多个样本分布是否具有显著性差异, 其广泛应用于数据分析、统计检验以及自然语言处理[26]等领域. 其中, F 统计量是组间均方(mean

square between, MSB)与组内均方(mean square error, MSE)的比值, 如公式(1)所示.

本文算法使用 F 检验的原因在于:

- 首先, 在辨别性模式挖掘阶段, 根据 F 统计量的计算, 当 F 值较大时, 意味着单词在类间的波动较大, 那么我们可以认为该单词能够区分不同的类别, 即辨别性较强对分类结果影响较大; 反之, 当 F 值较小时, 意味着单词在类间差异较小, 其对分类结果影响较小. 因此, F 统计量能够有效地度量不同的辨别性单词对分类结果的影响程度;
- 其次, 根据 F 检验的用途, 在训练分类模型阶段, 其能够有效地检验到辨别性模式到不同类之间距离分布的显著性差异, 从而进行时间序列分类;
- 最后, F 检验要求族群是服从正态分布且每次采样都是相互独立的, 本文模型使用 SAX 技术将时间序列依据正态分布进行离散化, 且在 SFA 技术里经过傅里叶变换后的系数同样服从正态分布; 与此同时, 不同子序列转换成单词的过程都是相互独立的, 进而保证了采样的独立性.

下面我们将描述时间序列获取 F 统计量的计算过程.

给定一个有 N 个实例 C 个类标签的数据集, 对于任意一个单词 A, 我们可以计算相应的 F 统计量, 以便确定单词 A 与类标签的关联性:

$$F=\frac{MSB}{MSE} \tag{1}$$

$$MSB=\frac{\sum n_j(\bar{a}_j-\bar{a})^2}{C-1} \tag{2}$$

$$MSE=\frac{\sum_j\left(\sum_i(a_{ij}-\bar{a}_j)^2\right)}{N-C} \tag{3}$$

公式(2)中, $\bar{a}_j$ 表示为第 j 种类别的单词 A 出现次数的均值, n_j 表示第 j 种类别实例数量, $\bar{a}$ 表示单词 A 出现次数的总体均值. 公式(3)中, a_{ij} 是第 j 种类别下的第 i 个实例单词 A 出现的次数.

对于每一个词袋中的单词, 根据上述公式为其计算相应的 F 值, 并选择合适的单词作为该词袋中的辨别性模式, 如图 3“阶段二”所示. 具体来说: 在“阶段一”, 我们获取某个数据集中全部时间序列实例的模式表示袋;在“阶段二”里, 依据公式(1)–公式(3)计算每个模式的 F 统计量的值, 并将其根据 F 值大小进行排序. 这样, 排名越靠前, 则表明单词的辨别性越强. 例如, 在阶段 1, w=50 的第 1 组频率柱状图里, 单词为“aaab”的模式在类别 2 中出现频率较多; 反之, 在类别 0 和类别 1 中出现频率较少. 由于其在类间差异较大, 从而导致 F 值也相对靠前, 那么我们可以推断: 该单词具有一定的辨别性, 能够区分不同的类. 由此, 在“阶段二”, 我们挖掘得到了一系列最具辨别性模式. 但问题在于, 如何选择合适数量的辨别性模式用于分类. 为了解决这个问题, 本文设计了 3 种模式选择策略.

(1) 固定个数模式: 每个类别选定固定的 top-k 个单词作为辨别性模式;

(2) F 分位表: 基于不同置信水平的 F 分位表进行选择;

(3) 约束 F 分位表: 在 F 分位表的基础之上限制最大模式个数.

具体细节将会在第 3.1 节详细介绍.

2.3 一维时间序列分类

在第 2.1 节和第 2.2 节中, 我们将每一个时间序列转换成模式集合并找出序列的辨别性模式. 本节将在此基础上构建时间序列分类算法, 如图 3“阶段三”所示. 在训练阶段, 通过“阶段二”, 我们获取了某个数据集中前 k 个最具辨别性模式, 即 $D_1,D_2,\ldots,D_k$. 依据这些辨别性模式, 将原始时间序列数据集转换到新的特征空间, 即如“阶段三”中表格所示, 其中, 每一行表示一个时间序列 T, 每一列表示最具辨别性模式 D, 表格中内容表示时间序列 T 与对应辨别性模式 D 之间的距离. 由此, 可将每一个时间序列转换成对应的距离向量, 从而得到新的输入空间, 根据输入空间中辨别性模式到不同类之间距离分布的差异性来训练分类模型. 在测试阶段,

将待分类实例依据最具辨别性模式的尺度转换成对应尺度下模式的集合, 之后, 根据公式(4)计算待分类实例与该辨别性模式之间距离. 由此, 同样可将待分类序列转换成距离向量输入到分类模型中为其分配类标.

下面, 我们将详细介绍具体计算细节.

首先给出时间序列与模式之间的距离公式. 给定一个时间序列 T, 我们可以将其转化为 $2r$ 个字典表示, 则模式第 i 个模式 D_i 与序列 T 之间的距离为

$$Dist(T,D_i)=v*\min\{dist(W_{r,j},D_i)\} \tag{4}$$

其中, $W_{r,j}$ 指的是序列 T 符号化之后的第 r 个字典表示中的第 j 个单词, r 可以从 D_i 中获取; v 表示单词 $W_{r,j}$ 在词袋中出现的次数.

对于公式(4)中的距离函数 $dist$, 其具体计算公式如下所示:

$$dist(W_{r,j},D_i)=\sum_{q=1}^{l}|W_{r,j}(q)-D_i(q)| \tag{5}$$

其中, $W_{r,j}(q)$和 $D_i(q)$分别表示单词 $W_{r,j}$ 和 D_i 的第 q 个字符, l 表示单词长度.

简单来说, 两个单词的距离是用对应字符的字母表位置之差的绝对值之和表示. 例如: 给定单词 B=abca 和 C=cbaa, 则 $dist(B,C)$=|1−3|+|2−2|+|3−1|+|1−1|=4. 这样, 我们可以将时间序列转化为到模式的距离向量 $(Dist(T,D_1),Dist(T,D_2),\ldots,Dist(T,D_k))$, 其中, k 表示模式的个数(如图 3 所示). 在距离公式的基础上训练一个模型, 用于预测序列 T 的类标签:

$$\hat{Y}=f\left[\alpha_0+\sum_{i=1}^{k}\alpha_i*Dist(T,D_i)\right] \tag{6}$$

其中, α_0 和α_i 表示线性权重, α_i 可以表示模式 D_i 在分类过程中的重要程度; k 表示模式的个数; f 表示映射函数, 如公式(7)所示:

$$f(x)=\frac{1}{1+e^{-x}} \tag{7}$$

在二分类任务中, 我们使用 $sigmoid$ 函数来预测序列 T 属于正类标签概率: 若概率大于 0.5, 则任务序列 T 属于正类实例. 在多分类任务中, 我们采用一对多的策略训练多个二分类器来对序列 T 分类.

算法 1 中给出了本文提出的利用字典方法挖掘多个尺度的时域频域辨别性模式分类算法的伪代码(dictionary-based multi-scale and multi-domain pattern mining, DM^{2}PM), 其中: 第 2 行遍历不同尺度下的滑动窗口; 第 3 行–第 5 行将每一个序列转化为当前窗口条件下 SAX 和 SFA 单词; 第 6 行根据单词的 F 值挖掘当前条件下的辨别性模式, 并加入模式集合 D; 第 8 行将原始序列转换为序列到辨别性模式之间的距离, 并在此基础上训练分类模型; 第 9 行–第 11 行根据不同的尺度将待测实例 T 转换成单词集合, 并计算其与模式之间的距离; 第 12 行基于分类模型对 T 进行分类得到类标.

算法 1. $DM^2PM(S,T)$.

输入: 大小为 N 的训练集 S、待分类序列 T;

输出: 序列 T 类标签 C.

1: 初始化: 设置滑动窗口集合 L 和字母表大小 c、模式个数 k、模式集合 $wordMethod$、辨别性模式集合 D.

2: **for** $L[i]$, $1\leqslant i\leqslant Size(L)$

3: **for** $S[j]$, $1\leqslant j\leqslant N$

4: $wordMethod[i]\leftarrow SAXandSFA(S[j],L[i])$ //将 $S[j]$转化为 SFA 和 SAX 单词集合.

5: **end for**

6: $D\leftarrow SelectPattern(wordMethod[i],k)$ //在尺度为 $L[i]$条件下, 选择 k 个单词加入辨别性模式集合 D.

7: **end for**

8: 根据公式(4)和公式(5)转换训练集并训练分类模型.

9: **for** $L[i]$, $1\leqslant i\leqslant Size(L)$

10: T←*CreateInstance*(T,$L[i]$) //将待分类序列 T 转化为单词集合, 根据公式(4)计算距离.

11: **end for**

12: C←*classifyInstance*(T) //根据公式(6)和公式(7)预测序列 T 的类标签 C.

13: **return** C.

在辨别性模式挖掘的具体实现中, 计算给定滑动窗口大小条件下所有单词 F 值的时间复杂度为 O(NnC), 单个序列转化为字典表示的时间复杂度为 O(n), 其中, n 为序列长度, C 为类属性个数. 此外, 在选取多个滑动窗口时, 我们设置最多窗口个数不超过 10 个, 即滑动步长为(max*WinSize*−min*WinSize*)/10. 则本文的模式挖掘方法的时间复杂度为 O(Nn+NnC), 并且 C<<n. 则有 O(Nn+NnC)≈O(Nn).

2.4 多维时间序列分类

本文模型可以扩展到多维时间序列分类问题中.

给定一个具有 N 个实例的 H 维的多维时间序列数据集 S, 其中每个维度序列长度均为 n. 对于其中的每一维序列, 按照单维时间序列辨别性模式挖掘算法挖掘相应的模式. 将 H 维序列的所有模式作为该数据集上的辨别性模式, 并将数据集依据辨别性模式转换到新的特征空间, 用于训练分类模型.

算法 2 给出了本文的多维时间序列分类算法(multivariate dictionary-based multi-scale and multi-domain pattern mining, MDM^{2}PM), 其中: 第 2 行–第 9 行是分维度提取每个序列不同尺度的时域和频域辨别性模式; 第 10 行是将训练集转化为序列到辨别性模式的距离并且训练分类模型; 第 11 行–第 15 行分维度提取序列 T 并将其转化为单词集合, 根据公式(4)得到 T 与单词之间的距离; 最后, 第 16 行根据公式(6)、公式(7)所示的分类模型预测序列 T 的类标签.

算法 2. *MDM*^{2}*PM*(S,T).

输入: 维度为 H 大小为 N 的训练集 S、待分类序列 T;

输出: 序列 T 类标签 C.

1: 初始化: 设置滑动窗口集合 L 和字母表大小 c、模式集合 *wordMethodH*、辨别性模式集合 *DH*.

2: **for** 1≤h≤H

3: **for** $L[i]$, 1≤i≤*Size*(L)

4: **for** $S[j][h]$, 1≤j≤N

5: *wordMethodH*[i]←*SAXandSFA*($S[j][h]$,$L[i]$) //将 $S[j][h]$转为 SFA 和 SAX 单词集合.

6: **end for**

7: *DH*←*SelectPattern*(*wordMethodH*[i],k) // 挖掘滑动窗口为 $L[i]$时, 第 h 维序列的辨别性模式.

8: **end for**

9: **end for**

10: 根据公式(4)、公式(5)转化训练集并训练分类模型.

11: **for** 1 ≤h≤H

12: **for** $L[i]$, 1≤i≤*Size*(L)

13: T←*CreateInstance*(T,$L[i]$) //将序列 T 转化为单词集合, 且计算与模式之间的距离.

14: **end for**

15: **end for**

16: C←*classifyInstance*(T)// 根据公式(6)、公式(7)预测序列 T 的类标签 C.

17: **return** C.

3 实验与评价

本节将对本文提出模型的相关实验内容进行介绍. 实验中所选取的数据集均来源于 UEA&UCR 时间序列数据仓库(http://www.timeseriesclassification.com). 本文模型中, 除了部分需要设置的超参数之外, 不需要设

置其他参数, 超参数具体设置为: 最大窗口长度为原始序列的 36%, 字母表大小 c 设置为 4 和 5. 接下来, 我们在固定超参数的基础上对本文模型进行进一步的分析.

3.1 辨别性模式个数

本节主要讨论如何选择合适数量的辨别性模式用于时间序列分类. 在前文中, 我们提出了 3 种模式选择方法, 分别是固定个数模式选择、F 分位表选择和添加约束的 F 分位表. 为了便于区分, 将这 3 种方法分别命名为 DM^{2}PMFixed, DM^{2}PMFv 和 DM^{2}PMCFv.

对于 DM^{2}PMFixed 模型而言, 图 4 中给出了 7 个数据集在不同固定模式个数 k 条件下, 准确率的变化曲线. 从图中可以看出: 在部分数据集上, 模式个数变化几乎不影响准确率, 例如 ShapeltSim 和 TwoLeadECG 等. 这是因为在这些数据集上, 少数模式已经具有较强的辨别性; 而在 SonyAIBORS2, ItalyPowerDed 和 ProximalPhalTW 数据集上, 准确率随着模式个数的变化准确率具有一定的波动. 但当 $k\in[60,70]$时, 各个数据集上能够取得较好的准确率. 由于本文模型的时间复杂度随着 k 的增加而线性增加, 为了在保证准确率的同时尽可能地降低时间复杂度, 我们将 k 设置为 65.

DM^{2}PMFv 模型是选择固定的 F 分位表, 将 F 值大于 F 统计量的单词作为模式用于时间序列分类. 在本文中, 我们一共选取 11 个不同置信水平 p 下的 F 分位表. 图 5 中展示了 7 个数据集在不同置信水平下的分类准确率变化情况.

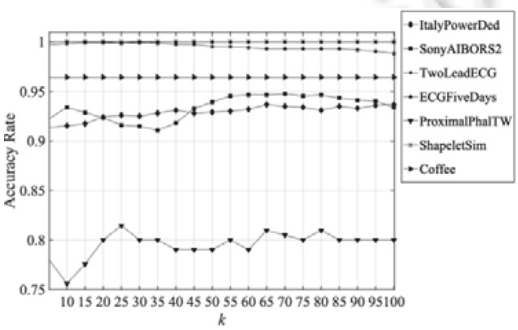

图 4 7 个数据集上不同 k 下准确率

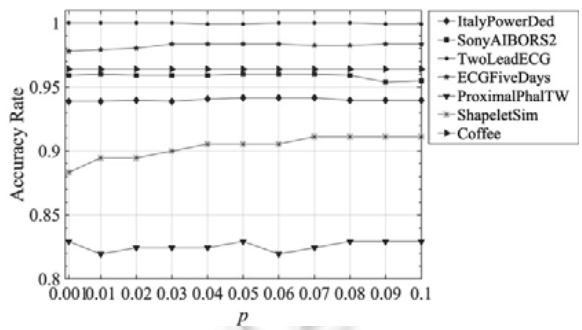

图 5 7 个数据集上不同 p 下准确率

从图 5 中可以看出: 在大部分数据集上, 随着置信水平的增加, 准确率变化不大. 对于数据集 ShapeleSim 和 ProximalPhalTW 而言, 随着置信水平的改变, 它们的分类准确率具有一定的变化. 但在 5%的置信水平下, 大部分数据集都能获得较优的分类准确率. 并且, 随着置信水平的增加, F 值逐渐缩小, 模式个数增加, 进而增加模型的时间复杂度. 所以, 我们在 DM^{2}PMFv 模型中, 将 p 值设置为 5%.

由上述结果可以看出, DM^{2}PMFv 模型在大部分数据集上都要略优于 DM^{2}PMFixed 模型. 这是由于 DM^{2}PMFv 相对于 DM^{2}PMFixed 选择了更多的模式. 表 2 展示了两个模型在各个数据集上的模式个数. 从表中我们可以看出: 即使是最小的置信水平(p=0.1%), 各个数据集上模式的个数也要远远大于 k=20 条件下各数据集上模式的个数.

但是, 表 2 中的结果也表明, DM^{2}PMFv 模型的时间复杂度要远大于 DM^{2}PMFixed 模型. 为了在保证分类准确率的同时尽可能地降低模型时间复杂度, DM^{2}PMCFv 模型在 F 统计量的基础上添加模式个数的约束. 图 6 展示了在确定 p 的条件下, 7 个数据集在不同约束条件下准确率变化情况. 其中, p 设置为 5%. 从图 6 中可以看出: 当 k 取较小值时, 除了 ECGFiveDays 数据集, 随着 k 的增大, 其他数据集上的准确率逐渐增加; 并且当 k 增大到一定值(k=70)时, 各个数据集上准确率趋于稳定. 这也表明本文的模式评价方法能够有效地衡量模式对分类的重要性.

表 2 不同条件下的模式个数

数据集	DM^{2}PMFv (p=0.1%)	DM^{2}PMFixed (k=20)
ItalyPowerDemand	1 966	**640**
SonyAIBORS2	53 260	**5 440**
TwoLeadECG	17 797	**3 560**
ECGFiveDays	33 188	**3 440**
ShapeletSim	118 214	**3 175**
Coffee	32 799	**3 200**
ProximalPhalTW	53 249	**3 560**

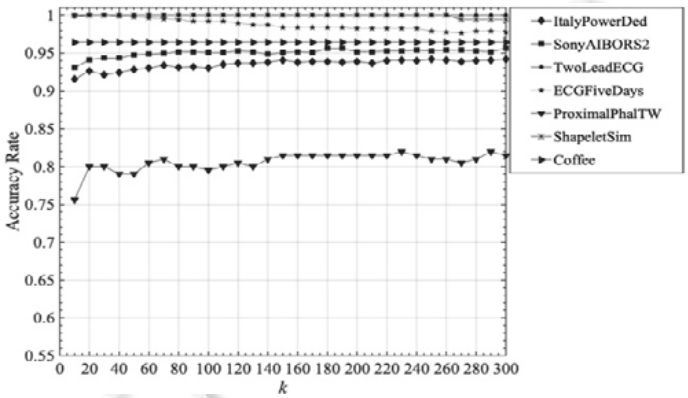

图 6 不同约束 k 条件下的准确率

表 3 总结了 3 个不同模型在 7 个数据集上产生的模式个数和准确率, 其中, 括号里表示模式个数. 从表 3 中可以看出: DM^{2}PMCFv 模型的准确率与 DM^{2}PMFixed 相当, 但 DM^{2}PMCFv 模式数量更少且具有相对较低的时间复杂度. 这表明 DM^{2}PMCFv 模型能够更加精准地选择有利于分类的模式. 为此, 本文模型将会选择带约束的 F 分位表设计策略. 在之后的实验中, 统一使用 DM^{2}PM 表示.

表 3 3 个模型的准确率和模式个数

数据集	DM^{2}PMCFv (p=5%, k=70)	DM^{2}PMFv (p=5%)	DM^{2}PMFixed (k=65)
ItalyPowerDed	0.930 (**1 301**)	**0.941** (6 046)	0.935 (2 186)
SonyAIBORS2	0.950 (**11 216**)	**0.960** (162 709)	0.950 (18 912)
TwoLeadECG	**1.000** (**7 114**)	0.999 (39 091)	**1.000** (12 117)
ECGFiveDays	**0.995** (**6 804**)	0.994 (83 169)	0.993 (11 950)
ProximalPhalTW	0.810 (**12 042**)	**0.829** (80 501)	0.805 (12 174)
ShapeletSim	**1.000** (**6 468**)	0.906 (200 073)	**1.000** (10 423)
Coffee	**0.964** (**10 353**)	**0.964** (75 659)	**0.964** (10 873)

3.2 与基于模式算法比较

本节将本文算法与基于模式的时间序列分类算法相比较, 例如快速 Shapelet(fast shapelet, FS)[12]、学习 Shapelet(learning shapelet, LS)[27]和 Shapelet 转换(shapelet transform, ST)[28]. 这些算法的结果均由 Bagnall 综述文献[29]所提供.

表 4 展示了 3 个算法在 45 个数据集上的准确率对比, 其中, 黑体表示 4 个模型中在该数据集上表现最优的模型, 最后一行表示平均准确率. 从表 3 中可以看出, LS, FS, ST 和 DM^{2}PM 模型分别在 12 个、3 个、19 个和 21 个数据集上取得最优结果. 在平均准确率方面, DM^{2}PM 模型平均准确率为 86.6%, 与 ST 模型相当, 但要优于 LS 和 FS 模型的为 82.3%和 76.6%.

为了综合衡量 4 个模型在多个数据集上的分类效果, 我们采用 Demšar[30]提出的临界差异图衡量多个模型的分类效果, 其中, 评分越小表示效果越好. 从图 7 中可以看出: DM^{2}PM, ST 和 LS 明显优于 FS; 其次, DM^{2}PM 排名也优于 ST 和 LS.

表 4 基于模式的算法在 45 个数据集上的分类结果

数据集	LS	FS	ST	DM^{2}PM	数据集	LS	FS	ST	DM^{2}PM
ArrowHead	**0.846**	0.594	0.737	0.823	MiddlePhalanxOC	0.780	0.729	0.794	**0.811**
Beef	0.867	0.567	**0.900**	0.767	MoteStrain	0.883	0.777	0.897	**0.939**
BeetleFly	0.800	0.700	**0.900**	0.850	OSULeaf	0.777	0.678	**0.967**	0.938
BirdChicken	0.800	0.750	0.800	**1.000**	PhalangesOC	0.765	0.744	0.763	**0.796**
Car	0.767	0.750	**0.917**	0.867	Plane	**1.000**	**1.000**	**1.000**	**1.000**
CBF	0.991	0.940	0.974	**1.000**	ProximalPhaOA	0.834	0.780	**0.844**	0.824
DiatomSizeR	**0.980**	0.866	0.925	0.889	ProximalPhalTW	0.776	0.702	0.805	**0.810**
DistalPhalanxOA	0.719	0.655	**0.770**	0.748	ScreenType	0.429	0.413	**0.520**	0.504
DistalPhalanxOC	**0.779**	0.750	0.775	0.764	ShapeletSim	0.950	**1.000**	0.956	**1.000**
DistalPhalanxTW	0.626	0.626	0.662	**0.727**	SmallKitchenApps	0.664	0.333	**0.792**	**0.792**
ECG5000	0.932	0.923	0.944	**0.945**	SonyAIBORS1	0.810	0.686	0.844	**0.892**
ECGFiveDays	**1.000**	0.998	0.984	0.995	SonyAIBORS2	0.875	0.790	0.934	**0.950**
FaceFour	0.966	0.909	0.852	**1.000**	Strawberry	0.911	0.903	0.962	**0.973**
Fish	0.960	0.783	**0.989**	0.966	SwedishLeaf	0.907	0.768	0.928	**0.938**
GunPoint	**1.000**	0.947	**1.000**	**1.000**	SyntheticControl	**0.997**	0.910	0.983	0.977
Ham	0.667	0.648	0.686	**0.781**	ToeSegment1	0.934	0.956	**0.965**	0.952
HandOutlines	0.481	0.811	**0.932**	0.927	Trace	**1.000**	**1.000**	**1.000**	**1.000**
Haptics	0.468	0.393	**0.523**	0.506	TwoLeadECG	0.996	0.924	0.997	**1.000**
Herring	0.625	0.531	0.672	**0.703**	TwoPatterns	**0.993**	0.908	0.955	0.989
ItalyPowerDemand	**0.960**	0.917	0.948	0.930	Wafer	0.996	0.997	**1.000**	0.999
LargeKitchenApp	0.701	0.560	**0.859**	0.800	Wine	0.500	0.759	**0.796**	0.688
Lightning2	**0.820**	0.705	0.738	0.738	WormsTwoClass	0.727	0.727	**0.831**	0.740
Lightning7	**0.795**	0.644	0.726	0.726	Average	0.823	0.766	0.861	**0.866**

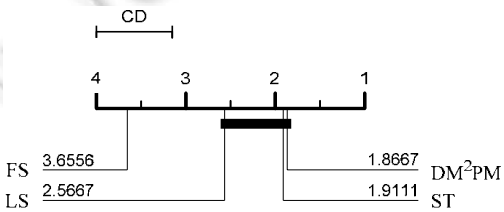

图 7 基于模式的算法在 45 个数据集上的临界差异图

另外, 为了更加形象地展示本文模型与这 3 个模型在多个数据集上的分类效果, 我们展示了本文模型与这 3 个模型在 45 个数据集上的分类准确率对比图(如图 8 和图 9 所示), 图中每个点表示一个数据集, 点落在斜线下方表示该数据集上本文模型表现较好.

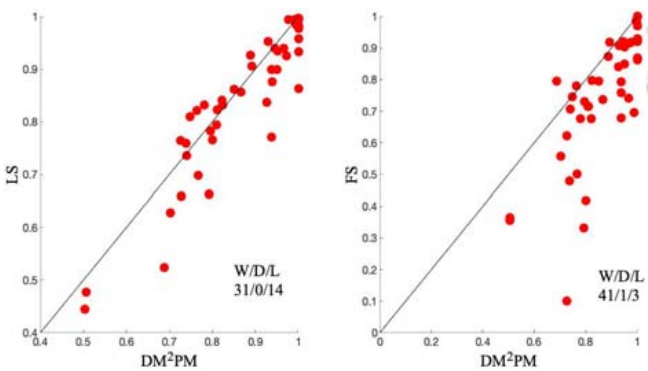

图 8 DM^{2}PM 与 LS 和 FS 在 45 个数据集上的比较

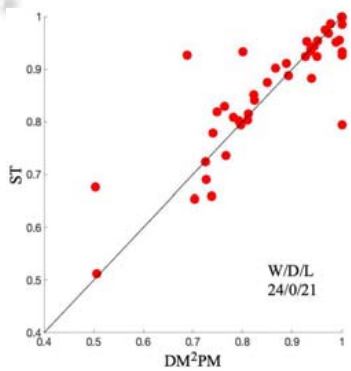

图 9 DM^{2}PM 与 ST 在 45 个数据集比较

由于 LS 和 FS 都为单分类器模型, 从图 8 中可以看出, DM^{2}PM 模型分别在 31 个和 41 个数据集上要优于 LS 和 FS 模型.

从图 9 中可以看出: DM^{2}PM 模型与 ST 模型分类效果相当, 两者分别在 24 个和 21 个数据集上表现较优. 但是 ST 模型本质上是一个集成分类器, 其时间复杂度要远远大于 DM^{2}PM 模型, 因此, DM^{2}PM 模型更具有实

际应用价值.

3.3 与基于字典算法比较

本节将本文模型与目前流行的基于字典的时间序列分类器相比较, 例如模式袋(BOP)[13]、符号聚合进行向量空间模型(SAXVSM)[15]、WEASEL[22]、SAX-VFSEQL[16]、SAX-VSEQL[16]和 mm-SEQL+LR[31]模型. 其中, mm-SEQL+LR 模型使用作者提供的结果, 其他模型则采用 Bagnall 综述文献[29]中所提供的结果.

我们同样采用临界差异图衡量这 7 个模型在多个数据集上的综合性能. 从图 10 中可以看出: DM^{2}PM, WEASEL 和 mm-SEQL+LR 要优于其他 4 个模型, 而这 3 个模型没有显著性差异. 但相对于 WEASEL 和 mm-SEQL+LR 模型而言, 本文模型具有更低的时间复杂度和内存消耗, 并且更具有可解释性.

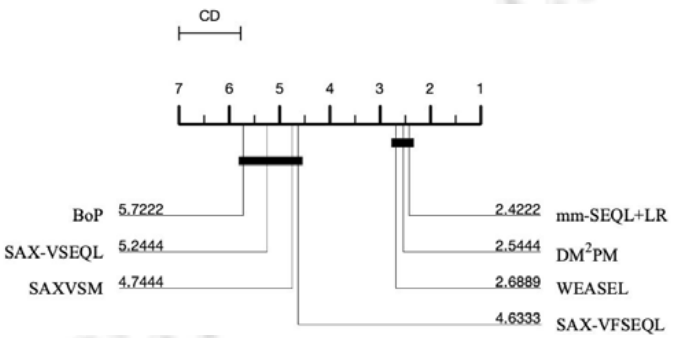

图 10 DM^{2}PM 与基于字典的方法比较

此外, 从图 11 中可以看出: DM^{2}PM 模型要明显优于 BOP, SAXVSM, SAX-VFSEQL 和 SAX-VSEQL 模型, 分别在 26 个、32 个、29 个和 27 个数据集上优于上述 4 个模型; 在平均准确率上, DM^{2}PM 模型也要远优于这 4 个模型. 而对于 WEASEL 模型和 mmSEQL+LR 模型而言, DM^{2}PM 分别在 22 个和 24 个数据集上占优, 在 1 个和 4 个数据集上准确率相同.

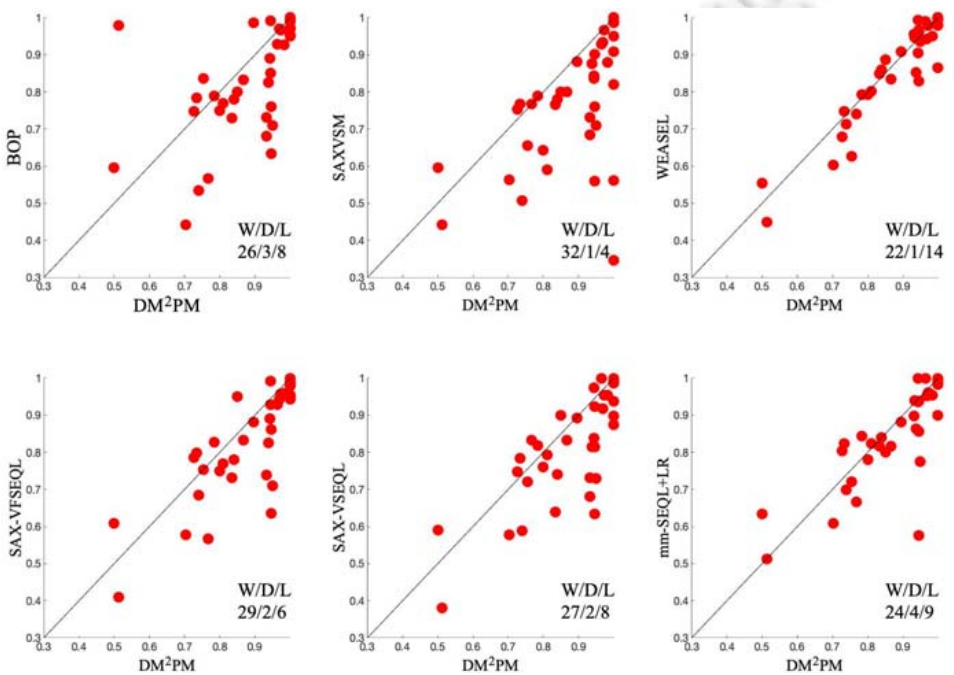

图 11 DM^{2}PM 与基于字典的方法比较

3.4 与集成算法和深度学习算法比较

本节将 DM^{2}PM 模型与现有流行的深度学习算法和表现最优的几个集成算法比较[32−37]. 由于篇幅有限, 在本节中仅仅使用临界差异图衡量多个模型在多个数据集上的分类性能.

从图 12 可知: 现有的结果并不能明显区分 DM^{2}PM 算法与 FCN, Hive-Cote, ResNet 和 Flat-Cote 以及 EE 之间的差异, 但其明显优于其余 5 个深度学习模型. 同时, 相对于神经网络和集成模型, DM^{2}PM 具有更低的时间复杂度和更好的可解释性.

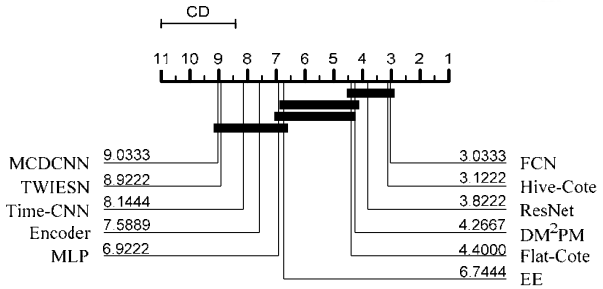

图 12 基于深度学习和集成算法的临界差异图

3.5 多维时间序列分类

本节将在多维时间序列数据集上进行实验, 并且将其与现有的时间序列基准算法 ED1NN 和 DTW1NN 比较. 表 5 展示了 3 个算法在 8 个多维数据集上的分类准确率, 其中, 括号中表示数据集的维度.

从表 5 中可以看出, MDM^{2}PM 要远优于 DTW1NN 和 ED1NN. 在 Handwriting, Libras 和 JapaneseVowels 数据集上, MDM^{2}PM 比最优的基准算法分别提升 11.8%, 28.3%和 49.5%. 这表明, 我们的方法在多维数据集上同样有着很好的分类效果.

表 5 多维时间序列分类结果

DataSet (dim)	MDM^{2}PM	DTW1NN	ED1NN
Ering (4)	**0.826**	0.722	0.689
Handwriting (3)	**0.459**	0.341	0.241
Libras (2)	**0.833**	0.544	0.550
JapaneseVowels (12)	**0.638**	0.084	0.143
BasicMotions (6)	**0.925**	0.850	0.400
AtrialFibrillation (2)	**0.267**	0.067	0.267
RacketSports (6)	**0.822**	0.763	0.737
Epilepsy (3)	**0.993**	0.957	0.659

3.6 时间复杂度分析

本节将本文模型的时间复杂度与现有的表现较好的时间序列分类算法进行比较. 由于深度学习算法通常使用 GPU 加速运算, 本文暂不进行比较. 由第 2.3 节可知: 本文模型中辨别性模式挖掘的时间复杂度为 O(Nn), N 为训练集实例个数, n 为序列长度. DM^{2}PM 模型使用的是带约束的 F 分位表, 假设其选择模式的个数为 k, 则特征转化的时间复杂度为 O(Nnk), 所以 DM^{2}PM 的模式挖掘时间复杂度近似为 O(Nnk). 逻辑回归的时间复杂度为 O(Nd), 其中, d 为特征维度.

图 13(左)显示了 DM^{2}PM, WEASEL, ST 和 COTE 在 20 个数据集上的 10 次平均运行时间的比较(WEASEL, ST 和 COTE 的 Java 代码来自于 https://github.com/uea-machine-learning/tsml/, 由于 mm-SEQL+LR 源代码是 C++, 与其他模型不一致, 因此本节中没有给出 mm-SEQL+LR 的运行时间). 从图中可以看出, DM^{2}PM 要明显优于另外 3 种算法. 此外, 我们选择一个相对较大的数据集 SwedishLeaf, 该数据集含有 500 个训练实例, 我们依次选择 50,100,150,…,500 个实例来训练这些模型, 如图 13(右)所示. 实验结果显示: DM^{2}PM, WEASEL 相对于 ST 和 COTE 要明显低一个数量级, 本文模型的运行时间同样少于 WEASEL.

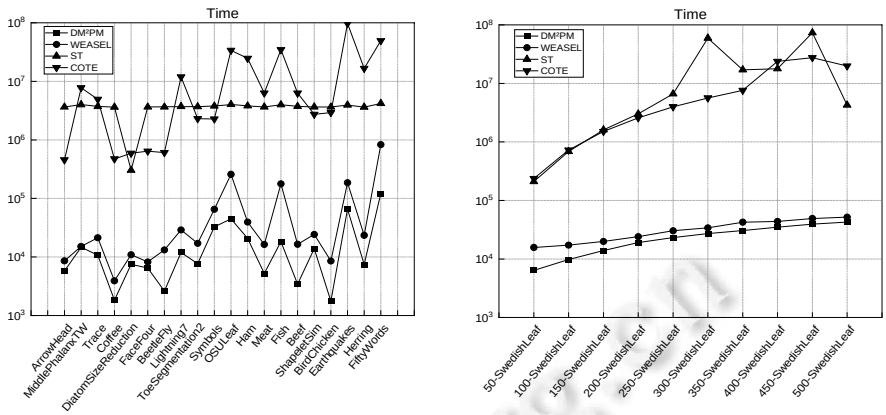

图 13 分类器运行时间比较

从表 6 中可以看出, DM^{2}PM, WEASEL 和 mm-SEQL+LR^2 的时间复杂度也远低于 COTE 和 ST 模型. 对于 DM^{2}PM, WEASEL 和 mm-SEQL+LR 模型而言, 它们在特征转化过程的时间复杂度几乎处于相同水平. 但 DM^{2}PM 模型最终的特征维度要远小于其余两个模型, 这也促使 DM^{2}PM 模型的整体时间和空间复杂度低于 WEASEL 和 mm-SEQL+LR 模型.

表 6 时间复杂度分析

算法	时间复杂度
DM^{2}PM	O(***Nnk***+***Nd***)
COTE	O(N^2n^4)
ST	O(N^2n^4)
WEASEL	O(Nn^2+Nd)
mm-SEQL+LR	O($Nn^{3/2}$log$n$+Nd)

4 可解释性

本节将关注算法的可解性, 即如何将挖掘的辨别性模式反映在原始序列中. 为此, 我们将符号化后的单词近似地映射到原始序列, 以便能够可视化地展示原始序列中最具辨别性区域.

4.1 Gun/Point数据集

Gun/Point 数据集从动作视频中捕获时间序列数据, 它被广泛应用到时间序列分类问题中. 该数据集分为两大类: Gun 类表示表演者手中拿枪执行一系列动作(拔枪、瞄准目标并将手枪放回枪套); Point 类表示表演者手中没有拿枪执行一系列动作(拔枪、瞄准目标并将手枪放回枪套). Gun/Point 数据集中含有 50 个实例作为训练集, 150 个实例作为测试集, 每个实例长度为 150.

由第 2 节可知: DM^{2}PM 模型是通过 F 值来衡量模式的重要程度, F 值越大, 则对应的模式越有利于区分序列. 对于一个模式而言, 其不仅仅由序列本身确定, 还与符号化参数有关. 表 7 举例展示了针对于每一个类, DM^{2}PM 模型选择的辨别性模式. 其中, l 表示窗口大小, w 表示单词长度, c 表示字母表大小, min 表示该模型出现在实例中的最小下标值, max 表示该模型出现在实例中的最大下标值. 根据表 7 中相关参数的值, 可以将 DM^{2}PM 选择的模式映射到原始序列, 从而得到不同类的最具辨别性区域. 如图 14 所示.

表 7 GunPoint 数据集上部分辨别性模式

	l	w	c	min	max	type	F	Pattern
Class: Gun	54	5	5	6	22	SAX	574.08	abbee
	19	8	4	5	65	SFA	299.52	bacdacbc
	54	7	5	71	91	SAX	174.72	eecbbba
Class: Point	24	6	4	0	25	SFA	114.8	cbadda
	49	6	4	73	91	SAX	76.8	dcabb

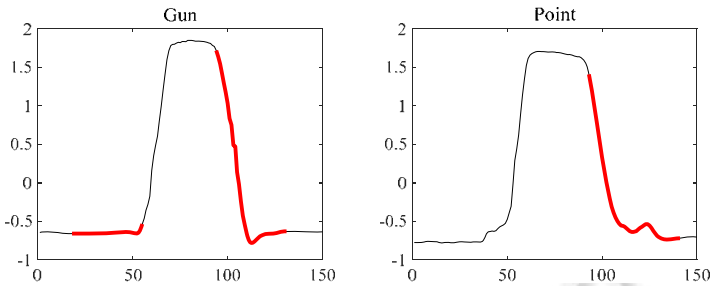

图 14 GunPoint 数据集案例(辨别性区域用粗体表示)

图 14 展示了 GunPoint 数据集中的两个不同类别的最具辨别性区域, 图中粗体部分即是本文模型发掘的辨别性模式所对应的子序列片段, 两类时间序列之间的区别可以通过开始区域以及结束区域观察到: 开始区域的小凹陷用来描述拔枪动作(gun), 结束区域的小凹凸则表明没有枪时手会移过枪套(point).

4.2 FaceFour数据集

FaceFour 数据集是从头骨轮廓图像中捕获时间序列数据, 其中有 4 种类别的头骨轮廓, 即含有 4 个类. 该数据集包括 24 个训练实例, 88 个测试实例, 每个实例都长度为 350. 表 8 展示了针对于数据集中的 4 个类别, 本文算法得到的辨别性模式, 其中, 符号的含义与表 7 相同.

表 8 FaceFour 数据集上部分辨别性模式

	l	*w*	*c*	min	max	type	*F*	Pattern
Class: One	100	5	5	117	131	SAX	51.67	dcabe
	64	6	4	16	105	SFA	51.43	baaaba
Class: Two	64	5	4	91	153	SAX	183.6	cbcbc
	76	5	4	201	227	SAX	55.16	bcbcb
	124	8	4	21	38	SFA	51.43	aaadbcad
Class: Three	52	7	4	79	138	SAX	72.92	cdccaba
	124	5	5	30	121	SAX	70.77	bdbce
	28	7	4	219	308	SAX	46.67	ccccdba
	16	6	4	118	271	SFA	46.67	dadbac
Class: Four	76	6	4	200	281	SAX	114.92	bcdcab
	124	6	4	43	50	SFA	47.5	daaaaa

图 15 展示了 FaceFour 数据集中 4 个不同类别的序列, 其中, 粗体标注部分亦能展现出 4 种类别间的不同. 另一方面, DM^{2}PM 模型在 GunPoint 和 FaceFour 数据集上均能达到目前最优准确率 100%.

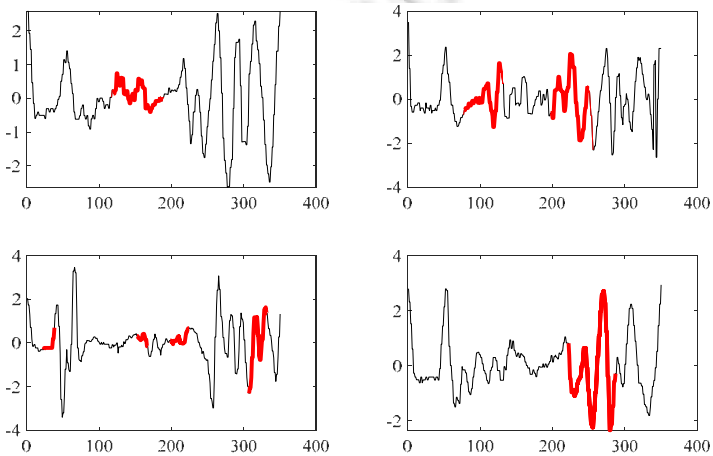

图 15 FaceFour 数据集案例(辨别性区域用粗体表示)

5 总 结

本文提出了一种基于可变尺度的时域和频域模式挖掘框架，基于此框架，使用了统计学方法来挖掘最具辨别性模式. 基于这些最具辨别性模式，我们将原始序列转换成序列到模式之间的距离，并在此新的特征空间中训练分类模型. 此外，本文模型可以扩展应用到多维时间序列分类问题中. 通过在多个数据集上的对比实验分析，说明了本文算法相对于已有的基于模式的算法具有更好的分类效果和更低的时间复杂度. 在具体的 Gun/Point 数据集、FaceFour 数据集上的实验结果解析，说明了本文算法具有较强的可解释性. 未来的研究问题可能包括处理多维时间序列空间信息丢失问题、优化多维时间序列分类模型以及优化时间序列符号化技术等.

References:

[1] Rajkomar A, Oren E, Chen K, *et al*. Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 2018, 1(1): 1–10.

[2] Dilmi M, Barthes L, Mallet C, *et al*. Iterative multiscale dynamic time warping (IMs-DTW): A tool for rainfall time series comparison. Int'l Journal of Data Science and Analytics, 2019, 1–15.

[3] Wang JD, Chen YQ, Hao SJ, *et al*. Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, 2019, 119: 3–11.

[4] Abanda A, Mori U, Lozano J. A review on distance based time series classification. Data Mining and Knowledge Discovery, 2019, 33(2): 378–412.

[5] Yuan JD, Wang ZH, Sun YG, *et al*. *K*-nearest neighbor classifier for complex time series. Ruan Jian Xue Bao/Journal of Software, 2017, 28(11): 3002−3017 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5331.htm [doi: 10.13328/j.cnki.jos.005331]

[6] Yuan JD, Wang ZH, Han M. Shapelet pruning and shapelet coverage for time series classification. Ruan Jian Xue Bao/Journal of Software, 2015, 26(9): 2311−2325 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4702.htm [doi: 10.13328/j.cnki.jos.004702]

[7] Ma QL, Zhuang WQ, Li S, *et al*. Adversarial dynamic shapelet networks. In: Proc. of the AAAI Conf. on Artificial Intelligence. New York: AAAI, 2020. 5069–5076.

[8] Zhao HY, Pan ZS, Tao W. Regularized shapelet learning for scalable time series classification. Computer Networks, 2020, 173: 107171.

[9] Li GL, Yan WH, Wu ZD. Discovering shapelets with key points in time series classification. Expert Systems with Applications, 2019, 132: 76–86.

[10] Kramakum C, Rathanmanon T, Waiyamai K. Information gain aggregation-based approach for time series shapelets discovery. In: Proc. of the 10th Int'l Conf. on Knowledge and Systems Engineering. Vietnam: IEEE, 2018. 97–101.

[11] Ji C, Zhao C, Liu SJ, *et al*. A fast shapelet selection algorithm for time series classification. Computer Networks, 2019, 148: 231–240.

[12] 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. Society for Industrial and Applied Mathematics, 2013. 668–676.

[13] 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.

[14] Lin J, Keogh E, Wei L, *et al*. Experiencing SAX: A novel symbolic representation of time series. Data Mining and Knowledge Discovery, 2007, 15(2): 107−144. [doi: 10.1007/s10618-007-0064-z]

[15] Senin P, Malinchik S. SAX-VSM: Interpretable time series classification using sax and vector space model. In: Proc. of the 13th Int'l Conf. on Data Mining. IEEE, 2013. 1175–1180. [doi: 10.1109/ICDM.2013.52]

[16] Le Nguyen T, Gsponer S, Ifrim G. Time series classification by sequence learning in all-subsequence space. In: Proc. of the 2017 IEEE Int'l Conf. on Data Engineering. San Diego: IEEE, 2017. 947–958. [doi: 10.1109/ICDE.2017.142]

[17] 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]

[18] Schäfer P, Högqvist M. SFA: A symbolic Fourier approximation and index for similarity search in high dimensional datasets. In: Proc. of the 15th Int'l Conf. on Extending Database Technology. Berlin: ACM, 2012. 516−527.

[19] Middlehurst M, Vickers W, Bagnall A. Scalable dictionary classifiers for time series classification. In: Proc. of the Int'l Conf. on Intelligent Data Engineering and Automated Learning. Cham: Springer, 2019. 11−19.

[20] Large J, Bagnall A, Malinowski S, *et al*. On time series classification with dictionary-based classifiers. Intelligent Data Analysis, 2019, 23(5): 1073−1089.

[21] Schäfer P. Scalable time series classification. Data Mining and Knowledge Discovery, 2016, 30(5): 1273−1298.

[22] Schäfer P, Leser U. Fast and accurate time series classification with weasel. In: Proc. of the 2017 ACM on Conf. on Information and Knowledge Management. New York: ACM, 2017. 637−646.

[23] Zhang W, Wang ZH, Yuan JD, *et al*. Time series discriminative feature dictionary construction algorithm. Ruan Jian Xue Bao/ Journal of Software, 2020, 31(10): 3216−3237 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5852.htm [doi: 10.13328/j.cnki.jos.005852]

[24] Yeh CCM, Kavantzas N, Keogh E. Matrix profile VI: Meaningful multidimensional motif discovery. In: Proc. of the 2017 IEEE Int'l Conf. on Data Engineering. San Diego: IEEE, 2017. 565−574.

[25] Li X, Lin J. Linear time complexity time series classification with bag-of-pattern-features. In: Proc. of the 2017 IEEE Int'l Conf. on Data Mining. IEEE, 2017. 277−286.

[26] Qamar AM, Alassaf M. Improving sentiment analysis of Arabic tweets by one-way ANOVA. Journal of King Saud University-Computer and Information Sciences, 2020.

[27] Grabocka J, Schilling N, Wistuba M, *et al*. Learning time-series shapelets. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 392−401.

[28] Bostrom A, Bagnall A. Binary shapelet transform for multiclass time series classification. In: Proc. of the Int'l Conf. on Big Data Analytics and Knowledge Discovery. Cham: Springer, 2015. 257−269.

[29] Bagnall A, Lines J, Bostrom A, *et al*. The great time series classification bakeoff: A review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 2017, 31(3): 606−660.

[30] Demsar J. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research, 2006, 7(1): 1−30.

[31] Le Nguyen T, Gsponer S, Ilie I, *et al*. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. Data Mining and Knowledge Discovery, 2019, 33(4): 1183−1222.

[32] Zheng Y, Liu Q, Chen E, *et al*. Time series classification using multi-channels deep convolutional neural networks. In: Proc. of the Int'l Conf. on Web-Age Information Management. Cham: Springer, 2014. 298−310.

[33] Wang Z, Yan W, 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. Anchorage: IEEE, 2017. 1578−1585.

[34] Cui Z, Chen W, Chen Y. Multi-scale convolutional neural networks for time series classification. arXiv:1603.06995, 2016.

[35] Zhao BD, Lu HZ, Chen SF, *et al*. Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics, 2017, 28(1): 162−169.

[36] 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. IEEE, 2016. 831−836.

[37] Bagnall A, Lines J, Hills J, *et al*. Time-series classification with COTE: The collective of transformation-based ensembles. IEEE Trans. on Knowledge and Data Engineering, 2015, 27(9): 2522−253.

附中文参考文献:

[5] 原继东, 王志海, 孙艳歌, 等. 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 [doi: 10.13328/j.cnki.jos.005331]

[6] 原继东, 王志海, 韩萌. 基于 Shapelet 剪枝和覆盖的时间序列分类算法. 软件学报, 2015, 26(9): 2311−2325. http://www.jos.org.cn/1000-9825/4702.htm [doi: 10.13328/j.cnki.jos.004702]

[23] 张伟, 王志海, 原继东, 等. 一种时间序列鉴别性特征字典构建算法. 软件学报, 2020, 31(10): 3216−3237. http://www.jos.org.cn/1000-9825/5852.htm [doi: 10.13328/j.cnki.jos.005852]

魏池璇(1997－), 女, 博士生, CCF 学生会员, 主要研究领域为数据挖掘, 时间序列分类.

王志海(1963－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为数据挖掘, 时间序列.

原继东(1989－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为数据挖掘, 时间序列分类.

林钱洪(1996－), 男, 硕士, 主要研究领域为机器学习, 时间序列分类, 广告投放优化.