计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(1):285−293 [doi: 10.15888/j.cnki.csa.009717] [CSTR: 32024.14.csa.009717]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于角度关键点和转向点的时间序列趋势特征提取①

刘冰珂[1], 任芮彬[1], 王　溪[2]

[1](西南交通大学 数学学院, 成都 611756)
[2](西南交通大学 信息科学与技术学院, 成都 611756)
通信作者: 任芮彬, E-mail: Airy_Ren@swjtu.edu.cn

摘　要: 时间序列分段线性表示算法利用时间序列的趋势变化特征, 用序列中较少点来表示整个时间序列. 但是大多算法主要关注局部序列点信息, 很少关注全局数据, 且部分算法只关注算法在数据集上的拟合, 很少应用到分类问题中. 针对上述问题, 本文提出了基于角度关键点和转向点的时间序列趋势特征提取算法, 首先, 该算法根据序列数据的角度变化值来选择角度显著点, 然后基于角度关键点的基础上再提取转向点, 根据分段的要求, 判断是否进行插值操作, 从而得到符合要求的分段点序列. 本文在模拟数据和 40 个公开数据集上进行拟合和分类实验, 实验结果表明, 本文算法相较于分段聚合近似 PAA、自底向下 TD、自顶向上 BU、基于拐点 FFTO、基于转折点和趋势段 Trend、基于趋势转折点 ITTP 等算法, 在模拟数据集拟合效果更好; 在 UCR 公开数据集平均拟合误差为 1.165; 分类准确性同 Keogh 团队公布的 DTW-1NN 算法高出 2.8%.

关键词: 角度关键点; 转向点; 时间序列; 分段线性表示; 趋势特征

引用格式: 刘冰珂,任芮彬,王溪.基于角度关键点和转向点的时间序列趋势特征提取.计算机系统应用,2025,34(1):285–293. http://www.c-s-a.org.cn/1003-3254/9717.html

Time Series Trend Feature Extraction Based on Angular Key Points and Inflection Points

LIU Bing-Ke[1], REN Rui-Bin[1], WANG Xi[2]

[1](School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China)
[2](School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: The piecewise linear representation algorithm of the time series represents the whole series with fewer points according to trend changes in the series. However, most of these algorithms focus on the information of local sequence points and rarely pay attention to global data. Some algorithms only focus on fitting on datasets instead of being applied to classification. To solve these problems, this study proposes an algorithm for extracting trend features from time series based on angle key points and inflection points. The algorithm selects angle key points according to the angle change values of the sequence data and then extracts inflection points based on these key points. It determines whether interpolation is needed according to segmentation requirements, so as to obtain a segmentation sequence meeting the requirements. Fitting and classification experiments are conducted on simulated data and 40 public datasets. Experimental results show that the proposed algorithm exhibits better fitting on the simulated data, compared with other algorithms such as piecewise aggregate approximation (PAA), the TD algorithm, the BU algorithm, the FFTO algorithm based on inflection points, the Trend algorithm based on turning points and trend segments, and the ITTP algorithm based on trend turning points. On the UCR public datasets, the proposed algorithm achieves an average fitting error of 1.165. Its classification accuracy is 2.8% higher than the DTW-1NN algorithm published by Keogh.

① 基金项目: 中央高校基本科研业务费专项资金 (2682024ZTPY041); 四川省科技计划 (2023YFH0066); 成都市科技项目 (2023-RK00-00080-ZF)
收稿时间: 2024-05-24; 修改时间: 2024-06-26; 采用时间: 2024-07-11; csa 在线出版时间: 2024-11-15
CNKI 网络首发时间: 2024-11-18

Key words: angular key point; inflection point; time series; piecewise linear representation; trend feature

1 引言

时间序列问题一直是在数据挖掘领域备受关注的问题, 如何从时间序列中发现并挖掘其隐藏的信息或特征, 更好地进行分类任务, 一直是时间序列分类的研究热点. 在许多领域中被广泛应用, 例如生物医学[1] 、金融[2] 、工程[3]等领域.

由于时间序列有时序依赖性, 具有高维度、数据依赖等特征[4]. 若直接在原始数据上进行处理, 需要花费大量的存储和时间成本, 且容易受到噪声的影响, 导致算法的准确度下降, 因此, 对其进行趋势特征提取是具有特殊意义的. 近年来, 针对时间序列的趋势特征提取, 已有很多优秀的方法被提出. 常见的有奇异值分解[5]、基于域变换的特征提取[6]、基于符号变换的特征提取[7]和基于分段表示的特征提取[8]. 其中前面 3 类算法相对复杂, 易受到噪声值的影响, 且应用范围有限, 而分段线性表示方法形式简单、解释直观, 故而广泛使用.

分段线性表示的方法, 重点在于分段点的选取, 好的分段点可以降低计算成本. 早期算法中, 以分段聚合近似 (piecewise aggregate approximation, PAA)[9]、基于自顶向下策略 (top-down, TD)[10]和基于自底向上策略 (bottom-up, BU)[11]的分段线性表示方法最为经典, 但这些算法拟合误差较大、计算复杂度高, 后来又发展了基于重要点策略的分段线性表示. 将时间序列上一些特殊的点定义重要点并采取一定的评价策略进行筛选, 进而实现线性分段表示. 例如, 刘意杨等提出一种基于转折点和趋势段的时间序列线性表示方法[12], 通过相邻线段的角度变化值来提取转折点, 结合趋势段的特征来评价筛选转折点. 邢邗等提出了基于趋势拐点的时间序列提取方法[13], 利用相邻两个点做坐标旋转变换, 对变换后数据提取拐点, 但也只是关注于局部的拐点, 未有全局信息, 且只能筛选为固定长度; 廖俊等提出基于趋势转折点的时间序列分段线性表示方法[14], 根据给定距离阈值与基本转折点进行对比, 判断是否为重要转折点, 但基本转折点的筛选也是依据相邻 3 个序列点的大小; 谢婷玉等提出了基于重要点双重评价的时间序列分段线性表示方法[15], 提出距离因子与趋势因子双重评价, 但该评价局限于 3 个转折点之中. 这些算法在选取重要点时, 往往是筛选极值点, 专注局部信息, 缺乏对全局信息的掌控. 李颖等提出基于时间序列波动性的趋势分层算法[16], 将时间序列的趋势分为上下两层, 在上下两层分别提取趋势点, 但忽略了上下层点之间的关系, 易忽略连续型趋势段中的关键点. 此外, 罗宇成等提出基于二阶导数提取趋势边缘点算法[17], 并对极值点进行优化; Chen 等提出一种基于迭代端点集合算法[18], 对每个线段点提出起点, 均值和趋势的三元符号表示等.

综合上述文献, 基于重要点策略的分段线性表示方法大多是依据相邻点关系进行评价筛选, 过度关注局部变化特征, 但无法顾及序列的全局变化, 造成的趋势特征提取不充分或者过分提取趋势现象, 增加趋势提取的复杂度. 据此, 本文提出了基于角度关键点和转向点的时间序列趋势特征提取方法. 该方法利用相邻点之间的角度变化值来确定角度关键点, 并基于这些关键点提取转向点. 通过分析这些转向点, 可以确定时间序列的全局转折点, 从而提取整个时间序列的趋势特征信息. 此外, 本文还应用此方法进行了分类任务, 以评估该方法在实际应用中的价值. 通过与其他同类算法的比较, 本文提出的方法在分类准确率和稳定性方面均表现出优异的性能.

本文的主要贡献如下: (1) 提出基于角度变化值的关键点提取方法, 它可以不受非线性影响, 直接度量数据点之间的相对变化, 能够更直观地反映趋势的转变; (2) 基于角度关键点再筛选转向点的方法, 筛去关键点中部分噪声点, 该方法具有去噪能力.

2 基本定义

1997 年, Keogh 首次提出时间序列分段线性表示算法[19], 该算法用序列中若干点连线, 来近似表示原始序列, 从而实现数据压缩, 提取趋势特征的过程. 现将该算法中的基本定义[19]罗列如下.

定义 1. 时间序列. 时间序列X是将n个数值按其发生的时间先后顺序排列而成的数列: $X=\{(x_1,t_1),(x_2,t_2),\cdots,(x_i,t_i),\cdots,(x_n,t_n)\}$, 其中$(x_i,t_i)$是第$i$个时刻的观察值. 时间序列的间隔时间是相等的. 因此, 通常认为时间序列X为:

$$X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\}$$

定义 2. 分段线性表示. 设时间序列 $X=\{x_1,x_2,\cdots,x_n\}$ 分为 m 段的分段线性表示为 $X_{\mathrm{PLR}}=\{(x_{1L},x_{1R}),(x_{2L},x_{2R}),\cdots,(x_{iL},x_{iR}),\cdots,(x_{mL},x_{mR})\}$, 其中 x_{iL},x_{iR} 分别表示第 i 个分段的左右端点, $x_{iL},x_{iR}\in\{x_1,x_2,\cdots,x_n\}$, $iL,iR\in\{1,2,\cdots,n\}$. 简记为:

$$X_{\mathrm{PLR}}=\{x_1^P,x_2^P,\cdots,x_i^P,\cdots,x_m^P\}$$

定义 3. 拟合误差. 时间序列 $X=\{x_1,x_2,\cdots,x_n\}$ 经过分段线性表示之后得到 X_{PLR}, 对 X_{PLR} 线性插值之后有 $X'_{\mathrm{PLR}}=\{x'_1,x'_2,\cdots,x'_n\}$, 则两者之间的拟合误差为:

$$E=\sqrt{\sum_{i=1}^{n}(x_i-x'_i)^2} \tag{1}$$

定义 4. 时间序列的压缩率. 设时间序列为 $X=\{x_1,x_2,\cdots,x_n\}$, 线性分段之后得到 $X_{\mathrm{PLR}}=\{x_1^P,x_2^P,\cdots,x_m^P\}$, 其中 $x_1^P=x_1$, $x_m^P=x_n$, 则线性分段时间序列压缩率为:

$$R=\left(1-\frac{m}{n}\right)\times 100\% \tag{2}$$

3 基于角度关键点和转向点的时间序列趋势特征提取算法

在时间序列分析中, 序列的转折点是重要的数据特征, 因为它们可以揭示序列中的重要变化. 传统的极值点检测方法可以捕捉到这些变化, 但它们可能无法充分提取趋势或容易受到异常值的影响. 如图 1 所示, 只提取极值点无法完全得到时间序列数据的完整趋势, 仍有一些波动较大的点代表着序列的特征. 因此, 需要更精确的方法来识别这些转折点.

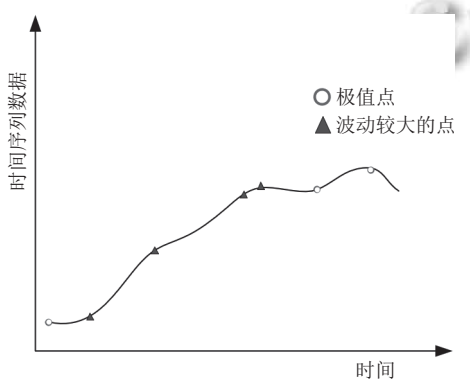

图 1 时间序列的分段点选择

经典算法中常用斜率的变化来直接反映趋势变化, 然而, 由于斜率是角度的函数, 而角度和斜率之间的关系是非线性的, 相同的变化角度可能导致不同的斜率变化. 如果仅基于斜率的变化来检测转折点, 可能会引入误判.

3.1 角度关键点和转向点

为了避免斜率引起的误判, 本文提出绝对的角度变化值, 给出时间序列角度关键点的定义.

定义 5. 给定时间序列 $X=\{x_1,x_2,\cdots,x_n\}$, 其角度关键点的定义如下: $X^s=\{x_1^s,x_2^s,\cdots,x_M^s\}$, x_i^s 表示第 i 个角度关键点, $x_i^s\in\{x_1,x_2,\cdots,x_n\}$, 且 x_i^s 满足以下条件:

$$\alpha_i=\begin{cases}\pi-|\theta_{i1}|-|\theta_{i2}|, & \theta_{i1}\times\theta_{i2}<0\\ |\theta_{i1}-\theta_{i2}|, & \theta_{i1}\times\theta_{i2}\geqslant 0\end{cases} \tag{3}$$

$$\alpha_i\geqslant\delta \tag{4}$$

其中, $\theta_{i1}=\arctan\dfrac{x_i-x_{i-1}}{\Delta t},\theta_{i2}=\arctan\dfrac{x_{i+1}-x_i}{\Delta t}$, α_i 表示点 x_i 处的角度变化值.

基于角度变化来筛选点, 可以在很多领域得到应用. 如在数据分析中, 基于角度变化可以检测异常点. 如在二维平面中, 若一个点与其临近点角度变化异常大, 那么这个点可能是异常点. 在图像处理和计算机视觉领域, 由于角度不会受到序列点旋转等影响, 如处理旋转敏感的数据集时, 基于角度提取的数据点不依赖数据的绝对位置, 具有广泛的适用性. 同时角度因其直观性与可解释性, 在可视化中也有应用空间.

根据时间序列的上升、下降、不变 3 种基本趋势, 以相邻 3 个时间序列点为例, 可以得到时间序列的 9 种趋势, 其中 6 种形态存在趋势转向, 本文以此为基础, 给出转向点相关定义.

定义 6. 给定时间序列的角度关键点 $X^s=\{x_1^s,x_2^s,\cdots,x_i^s,\cdots,x_M^s\}$, 转向点的定义如下: $X^I=\{x_{d_1}^I,x_{d_2}^I,\cdots,x_{d_j}^I,\cdots,x_{d_M}^I\}$, 其中 $x_{d_j}^I$ 表示第 j 个转向点, $x_{d_j}^I\in\{x_1^s,x_2^s,\cdots,x_i^s,\cdots,x_M^s\}$, $x_{d_i}^I$ 满足以下条件:

$$(x_{i+2}-x_{i+1})\times(x_{i+1}-x_i)=P \tag{5}$$

根据式 (5), 可得到关于转向点 X^I 的筛选, 如表 1 所示. 当转向点数低于分段数的要求时, 需要对其进行插值, 本文给出插值点的定义.

定义 7. 给定时间序列角度关键点 $X^s=\{x_1^s,x_2^s,\cdots,x_i^s,\cdots,x_M^s\}$ 及其转向点 $\{x_{d_j}^I,x_{d_{j+1}}^I\}$, 其分段提取点序列为 $\{x_{d_j}^s,x_{d_j+1}^s,\cdots,x_{d_{j+1}}^s\}$, 该分段内的插值点定义为: x_d^s, 且满足:

$$\alpha(x_d^s)=\max\left\{\alpha(x_{d_j}^s),\alpha(x_{d_j+1}^s),\cdots,\alpha(x_{d_{j+1}}^s)\right\},\ d_j\leqslant d\leqslant d_{j+1} \tag{6}$$

表 1 转向点的筛选

P	$x_{i+2}-x_{i+1}$	$x_{i+1}-x_i$	X^I
=0	=0	=0	无
	≠0	=0	x_{i+1}
	=0	≠0	x_{i+1}
<0	<0	>0	x_{i+1}
	>0	<0	x_{i+1}
>0	>0	>0	无
	<0	<0	无

3.2 时间序列趋势特征提取算法描述

本文提出基于角度关键点和转向点的时间序列分段表示方法, 通过角度变化值, 可以更精确地判断时间序列中的趋势变化. 它不受非线性影响. 这种方法通过直接度量数据点之间的相对变化, 能够更直观地反映趋势的转变. 具体来说, 如果某点的角度变化值较小, 意味着相邻 3 个点之间的连线更加接近于一条直线, 这表明趋势的变化较小. 因此, 该点作为转折点的可能性也就相对较小. 相反, 如果某点的角度变化值较大, 意味着该点在时间序列中可能是一个重要的趋势变化点, 作为转折点的可能性也就越大.

然而, 当序列数据遇见噪声干扰时, 噪声数据中会有部分干扰点也被提取角度关键点, 如若直接对此产生的角度关键点进行分析, 会使得提取的趋势特征不准确, 且相对局部, 极度造成资源浪费的同时无法达到较好的趋势特征提取.

通过其转向点可以提取趋势变化较大的角度关键点, 舍去受噪声影响的角度关键点, 但是由于本文中输入为时间序列的分段数, 输出为分段提取点序列. 受分段数的限制, 当转向点数大于分段数时, 要不断增大 δ的值来提取转向点; 当转向点个数低于分段提取点数的情况, 则需要进行插值, 以每个序列点的角度变化值为依据, 将变化值更大的角度关键点插入分段点序列中, 使其满足最后的分段数. 具体算法详见算法 1.

算法 1. 基于角度关键点和转向点的时间序列趋势特征提取算法

输入: 时间序列$X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\}$分段数$m$.
输出: 时间序列的分段表示$X_{\mathrm{PLR}}=\{x_1^P,x_2^P,\cdots,x_i^P,\cdots,x_m^P\}$.

1. 初始化: 角度关键点的阈值δ_0.
2. $X^s\leftarrow FindAngularKeyPoint(X,\delta_0)$ //根据定义 5, 获取时间序列角度关键点.
3. $X^I\leftarrow FindInflectionPoint(X^s,X)$ //根据定义 6, 获取角度关键点的转向点
4. while True:
5. if $Len(X^I)>m$:
6. $\delta+=\delta_0$
7. $X^s\leftarrow FindAngularKeyPoint(X,\delta)$
8. $X^I\leftarrow FindInflectionPoint(X^s,X)$
9. else if $Len(X^I)<m$:
10. $err\leftarrow FitError(X^I,X)$
11. $X^I\leftarrow InterpolateSegment(X^I,err)$ //根据定义 7, 进行插值
12. else if $Len(X^I)=m$:
13. $X_{\mathrm{PLR}}\leftarrow X^I$
14. break
15. return X_{PLR}

本文提出的算法计算复杂度为O(m). 从算法 1 的伪代码中可以看出, 本文主要算法分为两步, 第 1 步从时间序列中筛选角度关键点, 第 2 步从角度关键点中筛选转向点作为时间序列的分段点. 经过插值等优化, 只需遍历一次时间序列点, 便可以得到时间序列的分段线性表示, 故算法复杂度为O(m), 且$m<n$. 相较于直接遍历时间序列的复杂度O(n), 本算法计算复杂度更低.

4 实验分析

为了验证本文所提出的基于角度关键点和转向点的时间序列趋势特征提取算法的表现效果, 本文采用多个数据集进行实验, 并将该算法与 PAA[9] 、TD[10] 、BU[11] 、Trend[12] 、FFTO[13]及 ITTP[14]等 6 种方法进行对比. 其中 PAA、TD、BU 算法实现来自作者提供的代码资源, Trend、FFTO、ITTP 算法参照原论文的算法表述使用 Python 编程实现. 通过对比实验结果, 能够更清晰了解该算法在数据集上的表现, 并对其实际应用效果进行评估.

4.1 模拟数据实验

(1) 模拟数据

构建序列 X 为模拟数据, 具体如下:

$$X=\begin{cases}40, & t\in[1,50]\\ 40-(t-51)\times 2, & t\in[51,90]\\ -40, & t\in[91,130]\\ -40+(t-131)\times 3, & t\in[131,160]\\ 50+(t-161)/2, & t\in[161,200]\\ 69.5, & t\in[201,250]\\ 69.5-(t-251)/2, & t\in[251,290]\\ 47-(t-291)\times 3, & t\in[291,320]\\ -40, & t\in[321,360]\\ -40+(t-361)\times 2, & t\in[361,400]\\ 40, & t\in[401,450]\end{cases}$$

其中, t 为整数, 共 450 个数据, 该组序列包含不变、上升和下降等多种序列状态, 可以全面评估本文算法的综合性能.

(2) 拟合结果与分析

根据 X 的趋势变化, 可以将序列分为 12 段. 对序列 X 分别加上$\mu=0$, $\sigma=0.5,1.0,1.5,2.0,2.5,3.0$的噪声误差, 比较不同方法其拟合误差和分段趋势提取结果. 显然, 当算法选取的重要点越接近原有的趋势变化点, 分段效果更佳, 误差也相应减小. 实验结果如表 2 和图 2 所示.

表 2 展示了不同噪声情况下, 几种算法进行趋势特征提取后的拟合误差. 从表 2 中可以看出, 随着噪声的增加, 各个算法的拟合误差均有所上升. 由于 TD 算法根据最大误差进行分段, 导致该算法的最小分段数始终大于 12, 所以其拟合误差相对较小, 在$\sigma=2.5$时甚至达到最小误差, 但本文算法的拟合误差与其相差不大. 总体来说本文提出的算法与其他算法相比, 展现出更小的拟合误差, 且拟合效果较为稳定. 这表明, 本文算法在去噪能力上具有显著优势.

表 2 不同噪声下的不同算法在模拟数据上的拟合误差

σ	PAA	BU	TD	FFTO	Trend	ITTP	Ours
0.5	350.00	739.74	175.74	153.12	136.81	155.81	**78.40**
1.0	350.28	546.60	107.87	93.48	148.03	106.91	**92.08**
1.5	351.41	453.76	156.13	140.66	159.14	155.81	**136.61**
2.0	352.96	350.84	182.68	184.33	165.02	165.88	**158.42**
2.5	356.42	495.00	**230.62**	250.75	252.57	258.51	241.78
3.0	356.41	531.94	310.89	377.29	399.27	321.17	**291.74**

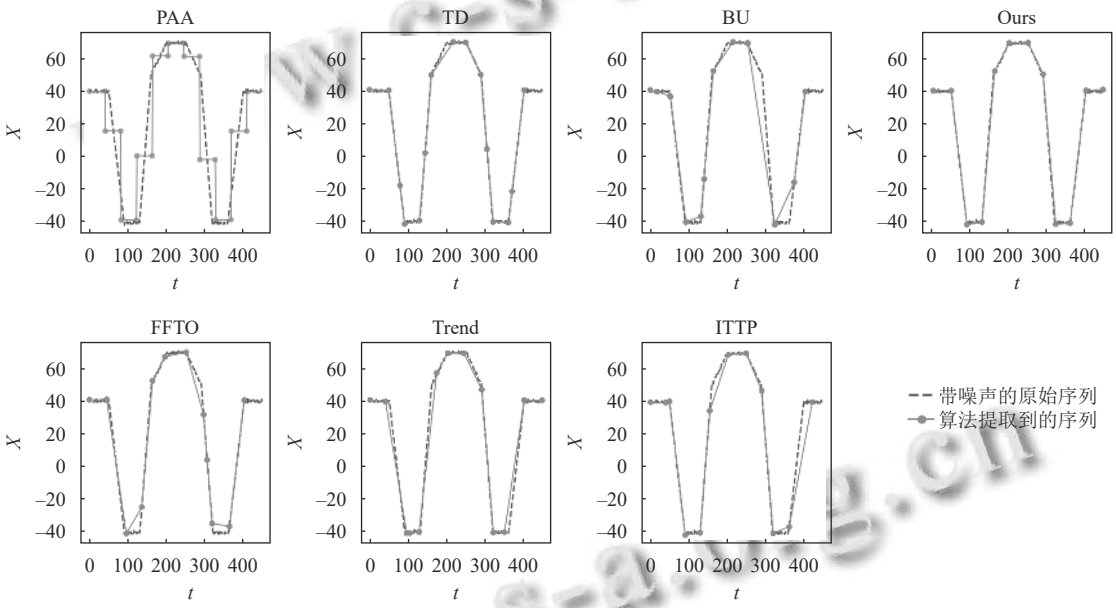

图 2 当$\sigma=0.5$时各个算法的拟合效果

图 2 展示了在噪声方差$\sigma=0.5$情况下, 不同算法的分段拟合效果图. 通过观察可以明显看出, 与其他算法相比, 本文提出的算法在提取趋势特征方面具有更高的准确性. 然而, PAA 方法由于采用提取区间内的平均值作为分段结果, 导致分段提取出现明显的错误. TD 方法在每次分段时都选择距离最远的点, 因此容易受到噪声点的影响. BU 算法和 ITTP 算法在分段时具有明显的分段错误. 而 FFTO 方法在提取分段时, 难以充分考虑序列最后部分的趋势, 导致趋势提取不够准确, 仍然存在一定的局限性. Trend 算法虽然能够较为完整的提取趋势特征, 但明显地可以看出, 与原始数据相比仍有差距, 造成较大的拟合误差.

4.2 公开数据集实验

(1) 公开数据集

采用美国加州大学河滨分校收集整理的 UCR 时间序列分类数据集[20], 本文随机选取来自不同领域的公开时间序列数据集进行实验, 数据集信息如表 3 所示.

(2) 拟合结果与分析

由于数据集来自不同的领域, 每个时间序列的取值范围差别很大, 为了方便进行各个算法的比较, 在进行拟合之前, 对数据进行标准化处理. 将数据集中的所有元素规范到[0, 1]的范围内, 消除由于数据量纲和取值范围不同而可能带来的影响.

表 3 本文数据集的详细信息

Dataset	Classes	Train	Test	Length	Dataset	Classes	Train	Test	Length
50words	50	450	455	270	Lighting7	7	70	73	319
ArrowHead	3	36	175	251	Meat	3	60	60	448
Beef	5	30	30	470	MiddlePhalanxOA	3	154	400	80
BeetleFly	2	20	20	512	MiddlePhalanxOC	2	1 800	858	80
BirdChicken	2	20	20	512	MiddlePhalanxTW	6	154	399	80
Car	4	60	60	577	OSULeaf	6	200	242	427
CBF	2	30	900	128	PhalangesOC	2	1 800	858	80
CinCECGtorso	4	40	1 380	1 639	Plane	7	105	105	144
Coffee	2	28	28	286	ProximalPhalanxOA	3	400	205	80
DistalPhalanxOC	3	139	400	80	ShapesAll	60	600	600	512
DistalPhalanxTW	3	139	400	80	Strawberry	2	370	613	235
ECG200	2	100	100	96	SwedishLeaf	15	500	625	128
ECGFiveDays	2	23	861	136	syntheticControl	6	300	300	60
FaceFour	14	560	1 690	131	ToeSegmentation1	2	40	228	277
GunPoint	2	50	150	150	ToeSegmentation2	2	36	130	343
Ham	2	109	105	431	Trace	4	100	100	275
Haptics	5	155	308	1 092	TwoPatterns	4	1 000	4 000	128
Herring	2	64	64	512	Wine	2	57	54	234
InlineSkate	7	100	550	1 882	WordsSynonyms	25	267	638	270
Lighting2	2	60	61	637	Yoga	2	300	3 000	426

在相同的压缩率情况下, 选择每个数据集中的第 1 个训练样本进行拟合实验, 几种算法的拟合误差如表 4 所示. 表 4 中, R 代表压缩率, FFTO 算法中提取拐点的阈值设为 5, PPA、TD、BU 算法压缩率会大于且最接近表 4 中的压缩率. 加粗字体表示每行中的最小拟合误差.

表 4 不同压缩率下的算法在公开数据集上的拟合误差

Dataset	PAA	TD	BU	FFTO	Trend	ITTP	Ours	R
50words	1.154	2.110	5.042	4.009	**0.049**	2.315	1.385	0.90
ArrowHead	0.721	1.829	6.484	0.932	0.887	5.423	**0.682**	0.81
Beef	**3.030**	3.614	3.095	4.000	8.980	5.445	3.250	0.89
BeetleFly	4.540	5.507	13.941	10.899	3.940	18.266	**3.021**	0.94
BirdChicken	**2.122**	7.810	16.173	7.595	3.562	9.517	3.494	0.95
Car	1.107	7.626	9.421	1.108	6.558	7.640	**0.482**	0.89
CBF	**1.096**	1.643	5.399	4.261	4.596	4.399	1.243	0.53
CinCECGtorso	1.065	4.131	12.693	5.675	**0.291**	8.092	0.430	0.85
Coffee	**0.996**	2.953	7.675	1.537	3.289	2.947	1.056	0.91
DistalPhalanxOC	0.512	1.724	1.772	0.496	2.876	1.689	**0.155**	0.61
DistalPhalanxTW	0.732	2.239	1.778	1.964	2.184	1.997	**0.481**	0.76
ECG200	**0.745**	1.432	3.374	2.814	2.407	2.487	2.137	0.77
ECGFiveDays	**1.018**	1.791	2.288	2.654	4.067	4.564	2.257	0.74
FaceFour	**1.265**	6.586	6.930	6.916	8.624	16.416	1.811	0.57
Gun_Point	0.329	2.756	2.557	2.011	0.440	2.427	**0.022**	0.66
Ham	3.332	**2.539**	14.810	5.993	12.473	18.603	3.050	0.93
Haptics	0.873	1.704	1.708	1.826	3.831	4.680	**0.337**	0.83
Herring	**1.408**	6.925	7.993	2.889	6.189	7.161	1.694	0.92
InlineSkate	**0.200**	9.700	12.048	1.727	2.286	21.540	1.228	0.85
Lighting2	0.822	1.062	**0.409**	2.486	2.214	11.846	0.461	0.75
Lighting7	**1.281**	2.714	6.065	6.174	6.617	10.181	2.742	0.62
Meat	0.279	3.575	5.530	0.263	0.946	4.267	**0.033**	0.63
MiddlePhalanxOA	0.632	2.498	1.948	0.861	1.466	1.541	**0.264**	0.67
MiddlePhalanxOC	1.144	2.430	1.925	2.096	1.850	2.151	**0.875**	0.83
MiddlePhalanxTW	1.117	2.927	1.920	0.518	0.920	1.951	**0.260**	0.82
OSULeaf	**1.560**	8.345	6.491	6.756	4.637	15.563	3.215	0.92
PhalangesOC	1.339	1.640	2.477	1.034	3.853	1.916	**0.576**	0.82

表 4 不同压缩率下的算法在公开数据集上的拟合误差 (续)

Dataset	PAA	TD	BU	FFTO	Trend	ITTP	Ours	*R*
Plane	0.837	1.689	3.658	0.363	2.552	3.710	**0.306**	0.80
ProximalPhalanxOA	0.726	2.501	4.561	1.117	1.145	2.123	**0.456**	0.78
ShapesAll	0.266	8.076	11.517	0.826	5.786	7.362	**0.087**	0.62
Strawberry	0.296	1.070	5.496	0.723	1.849	0.980	**0.173**	0.66
SwedishLeaf	0.462	3.097	4.569	1.938	3.262	3.824	**0.122**	0.60
synthetic_control	2.122	3.030	3.669	4.280	7.698	5.672	**0.431**	0.54
ToeSegmentation1	**1.284**	3.436	6.735	4.629	3.047	11.685	1.850	0.89
ToeSegmentation2	**1.119**	5.280	5.524	3.401	6.046	15.481	1.752	0.75
Trace	2.265	**1.733**	5.399	3.305	3.793	7.552	2.968	0.81
TwoPatterns	1.550	0.812	7.353	6.955	8.227	7.928	**0.036**	0.53
Wine	0.590	5.439	6.660	0.688	8.736	4.289	**0.439**	0.82
WordsSynonyms	1.905	3.490	8.462	1.440	8.373	10.369	**0.990**	0.91
Yoga	0.818	6.316	7.396	0.972	2.060	7.433	**0.362**	0.78
Average	1.216	3.644	6.074	3.003	4.065	7.086	**1.165**	—

根据表 4, 可以观察到, 在 40 组时间序列数据中, PAA 算法在 13 组数据上取得了最小的拟合误差. 同样地, BU 在 1 组时间序列数据中拟合最小, TD 算法和 Trend 算法分别各在 2 组时间序列数据中拟合最小. 值得一提的是, 本文提出的算法在 22 组时间序列数据中表现出了最小的拟合误差. 这些结果进一步印证了本文算法的有效性和优越性.

以 ECG200 和 ECGFiveDays 数据集为例, 这两个数据集是心电数据集, 每个序列都记录了一次心跳时的电活动, 该类数据集中各个序列点之间波动较大. BU、TD、Trend 等算法受到压缩率的影响, 只能提取到部分变化较大的转折点, 而 PAA 算法可以平均地将序列点平滑化, 使得提取到的序列与原序列相比之下, 波动较为平缓, 故而可以得到最好的拟合效果, 但本文算法与其他算法相比, 仍然可以有较好的拟合效果.

(3) 分类结果与分析

为了评估本算法的性能, 应用了本文提出的算法进行分类任务. Base 算法为 Keogh 团队公布的 DTW-1NN 分类准确率. 本文运用 FFTO、Trend 和 ITTP 算法对时间序列数据进行拟合, 同样采用 DTW-1NN 评估分类准确率. 对比结果如表 5 所示. 表 5 数据集的压缩率与表 4 中各个数据的压缩率相同. 加粗字体表示每行分类准确率的最大值.

在表 5 中, 有 12 组数据显示直接使用原始数据得到的分类准确率是最大的. FFTO 算法仅有 2 组数据的分类准确率达到了最大, Trend 算法仅有 1 组数据的分类准确率达到了最大, ITTP 算法仅有 4 组数据的分类准确率达到了最大. 本文提出的算法在 27 组数据中实现了最大的分类准确率. 此外, Base 算法的分类准确率平均值为 0.756; FFTO 算法获得的分类准确率平均值为 0.684; Trend 算法获得的分类准确率平均值为 0.702; ITTP 算法获得的分类准确率平均值为 0.649; 均略低于 Base. 然而, 使用本文提出的算法获得的分类准确率平均值为 0.784, 这是所有算法中的最高值. 并且相比原始数据的平均分类准确率上升了 2.8%, 表明整体分类准确率明显上升.

表 5 不同算法在公开数据集上的分类准确率

Dataset	Base	FFTO	Trend	ITTP	Ours	*R*
50words	**0.690**	0.464	0.585	0.310	0.681	0.90
ArrowHead	0.703	0.663	0.646	0.503	**0.737**	0.81
Beef	0.633	0.633	0.500	0.667	**0.733**	0.89
BeetleFly	0.700	0.800	0.700	0.700	**0.850**	0.94
BirdChicken	0.750	0.650	0.600	0.650	**0.800**	0.95
Car	0.733	0.583	0.517	0.450	**0.817**	0.89
CBF	**0.997**	0.996	0.993	0.794	**0.997**	0.53
CinCECGtorso	0.651	0.688	0.647	0.571	**0.715**	0.85
Coffee	**1.000**	0.643	0.571	0.964	**1.000**	0.91
DistalPhalanxOC	**0.768**	0.717	0.730	0.728	0.765	0.61
DistalPhalanxTW	0.710	0.668	0.705	0.648	**0.733**	0.76
ECG200	0.770	0.760	0.760	0.740	**0.820**	0.77
ECGFiveDays	0.768	0.698	0.706	**0.780**	0.736	0.74
FaceFour	0.830	0.784	0.773	0.693	**0.841**	0.57
Gun_Point	0.907	0.927	0.900	0.693	**0.960**	0.66
Ham	0.533	**0.638**	0.533	0.571	0.629	0.93
Haptics	0.377	0.328	0.312	0.331	**0.396**	0.83
Herring	0.531	0.547	0.563	0.641	**0.656**	0.92
InlineSkate	0.384	0.382	0.340	0.253	**0.420**	0.85
Lighting2	0.869	0.787	0.770	**0.885**	0.869	0.75
Lighting7	**0.726**	0.658	0.699	0.644	**0.726**	0.62
Meat	0.933	0.667	0.767	**0.967**	0.817	0.63
MiddlePhalanxOA	0.750	0.730	0.728	**0.765**	0.760	0.67
MiddlePhalanxOC	0.648	0.558	0.667	0.585	**0.712**	0.83
MiddlePhalanxTW	0.584	0.536	0.566	0.511	**0.622**	0.82
OSULeaf	0.591	0.384	0.603	0.351	**0.707**	0.92
PhalangesOC	**0.728**	0.622	0.706	0.611	0.719	0.82
Plane	**1.000**	0.971	0.990	0.962	0.990	0.80

表 5 不同算法在公开数据集上的分类准确率 (续)

Dataset	Base	FFTO	Trend	ITTP	Ours	*R*
ProximalPhalanxOA	0.805	0.785	0.785	0.839	**0.849**	0.78
ShapesAll	**0.768**	0.703	0.698	0.228	0.762	0.62
Strawberry	**0.940**	0.900	0.928	0.896	**0.940**	0.66
SwedishLeaf	0.792	**0.805**	0.688	0.485	0.776	0.60
synthetic_control	**0.993**	0.987	0.970	0.863	0.980	0.54
ToeSegmentation1	0.772	0.671	0.772	0.662	**0.816**	0.89
ToeSegmentation2	0.838	0.792	0.831	0.715	**0.885**	0.75
Trace	**1.000**	0.860	**1.000**	0.980	**1.000**	0.81
TwoPatterns	**1.000**	0.505	0.982	0.892	0.983	0.53
Wine	0.574	0.593	0.537	0.593	**0.648**	0.82
WordsSynonyms	0.649	0.458	0.530	0.215	**0.671**	0.91
Yoga	0.836	0.813	0.776	0.623	**0.848**	0.78
Average	0.756	0.684	0.702	0.649	**0.784**	—

为了综合评估本文算法与其他几个算法在多个数据集上的效果, 采用了 Demsar[21]提出的临界差异图衡量算法的分类效果, 其中, 评分越小表示分类效果越好, 从图 3 中可以明显看出: 本文算法稍优于 Base, 但明显优于剩余 3 个算法.

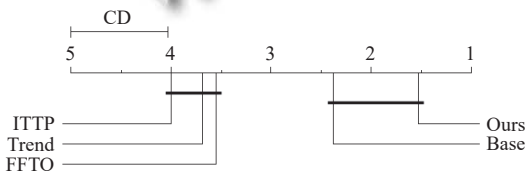

图 3 不同算法在 40 个数据集上的临界差异图

此外, 为了更直观地展现本文算法相较于其他算法在数据集分类性能方面的优势, 特别绘制了分类准确率对比图. 图 4 中, 每一个点对应一个数据集, 当点位于斜线下方时, 就意味着在该数据集上, 本文算法的分类效果更佳.

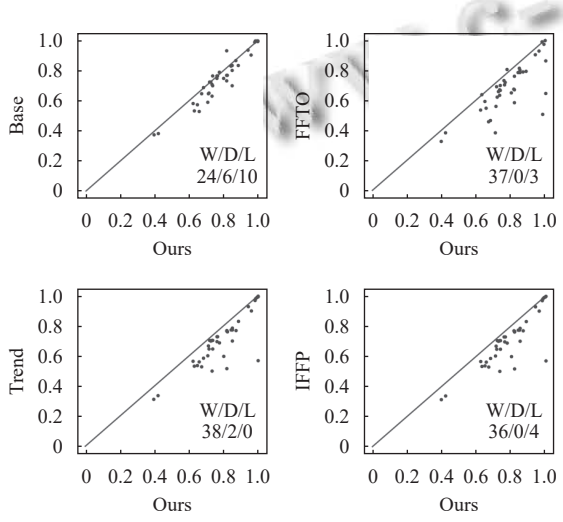

图 4 本文算法与其他算法在 40 个数据集上的分类比较

从图 4 中可以看出, 本文算法在 37 个数据集上明显优于 FFTO, 在 38 个数据集上明显优于 Trend, 在 36 个数据集上明显优于 ITTP. 本文算法与 Base 两者分别在 24 和 10 数据集表现更优, 但本文算法使用数据中更少的点, 却得到了和 Base 差不多的分类准确率, 相比之下, 本文的算法更有可解释性和实际应用价值.

5 结论与展望

本文在时间序列分段线性表示的基础上, 提出了利用角度变化值来提取时间序列的角度关键点, 充分利用时间序列的趋势特征, 在角度关键点序列的基础上, 提取转向点, 进而根据压缩率的要求, 在全局上判断是否需要进行插值, 从而得到符合压缩率要求的分段点序列. 该算法理解简单, 且容易实现. 通过在模拟数据和公开数据集上进行拟合实验可以看出, 本文算法有一定的抗噪能力, 且拟合效果相对较好. 在公开数据集上进行分类可以看出, 本文算法可以用较少的点达到较好的分类效果, 且相较于 Keogh 团队公布的原始数据分类准确率相比, 本文算法的分类准确度能够提高 2.8%. 接下来的工作, 结合趋势特征来改进时间序列的相似性度量, 从而进行更好的分类任务也是一个很好的研究方向.

参考文献

1 Kunakorntham P, Pattanaprateep O, Dejthevaporn C. Detection of statin-induced rhabdomyolysis and muscular related adverse events through data mining technique. BMC Med Inform Decis Mak, 2022, 22(1): 233. [doi: 10.1186/s12911-022-01978-4]

2 Li T, Kou G, Peng Y, *et al*. An integrated cluster detection, optimization, and interpretation approach for financial data. IEEE Transactions on Cybernetics, 2022, 52(12): 13848–13861. [doi: 10.1109/TCYB.2021.3109066]

3 Xu JJ, Guo YH, Wang J. Data mining technology and its application in multi-sensor data processing. Proceedings of the 2nd IEEE International Conference on Power, Electronics and Computer Applications (ICPECA). Shenyang: IEEE, 2022. 492–495.

4 袁慧, 谭章禄, 王福浩. 一种高效的相似性度量方法及其分类效果研究. 中国科学: 技术科学, 2022, 52(7): 1096–1110.

5 Redford J, Li XJ. Exploration of SVD for image compression and time series processing. Proceedings of the 2020 IEEE MIT Undergraduate Research Technology Conference

(URTC). Cambridge: IEEE, 2020. 1–4.
6 魏池璇, 王志海, 原继东, 等. 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. [doi: 10.13328/j.cnki.jos.006346]
7 Yang SJ, Wang Y, Zhang J. A similarity measure for time series based on symbolic aggregate approximation and trend feature. Proceedings of the 39th Chinese Control Conference (CCC). Shenyang: IEEE, 2020. 6386–6390.
8 Yang XY, Zhai CX, Li F, *et al*. A genetic algorithm based piecewise linear representation of time series. Proceedings of the 14th IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE). Dalian: IEEE, 2019. 1199–1204.
9 Keogh E, Chakrabarti K, Pazzani M, *et al*. Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems, 2001, 3(3): 263–286. [doi: 10.1007/PL00011669]
10 Keogh E J, Pazzani MJ. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining. New York: AAAI Press, 1998. 239–243.
11 Park S, Lee D, Chu WW. Fast retrieval of similar subsequences in long sequence databases. Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange. Chicago: IEEE, 1999. 60–67.
12 刘意杨, 李俊朋, 白洪飞, 等. 基于转折点和趋势段的时间序列趋势特征提取. 计算机应用, 2020, 40(S1): 92–97. [doi: 10.11772/j.issn.1001-9081.2019111985]
13 邢邗, 石晓达, 孙连英, 等. 时间序列数据趋势转折点提取算法. 计算机工程, 2018, 44(1): 56–61, 68. [doi: 10.3969/j.issn.1000-3428.2018.01.009]
14 廖俊, 于雷, 罗寰, 等. 基于趋势转折点的时间序列分段线性表示. 计算机工程与应用, 2010, 46(30): 50–53, 81.
15 谢婷玉, 徐德刚, 阳春华, 等. 基于重要点双重评价的时间序列趋势提取. 信息与控制, 2018, 47(6): 730–736, 744.
16 李颖, 于东, 胡毅, 等. 基于时间序列波动性的分段线性表示方法. 计算机系统应用, 2021, 30(6): 300–305. [doi: 10.15888/j.cnki.csa.007978]
17 罗宇成, 张明恩, 刘飞, 等. 基于分段特征提取的仿真模型结果验证方法. 系统仿真学报, 2024, 36(1): 272–281.
18 Chen HY, Du JH, Zhang WN. An iterative end point fitting based trend segmentation representation of time series and its distance measure. Multimedia Tools and Applications, 2020, 79(19): 13481–13499.
19 Keogh E. Fast similarity search in the presence of longitudinal scaling in time series databases. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence. Newport Beach: IEEE, 1997. 578–584.
20 Bagnall A, Lines J, Bostrom A, *et al*. 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]
21 Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 2006, 7(1): 1–30.

(校对责编: 孙君艳)