计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2024/18(02)-0387-16
doi: 10.3778/j.issn.1673-9418.2211099

面向Shapelet空间的多变量时间序列密度聚类算法

盛锦超,杜明晶+,孙嘉睿,李宇蕊
江苏师范大学 计算机科学与技术学院,江苏 徐州 221100
+ 通信作者 E-mail: dumj@jsnu.edu.cn

摘　要:多变量时间序列聚类问题已经成为时间序列分析任务中重要的研究课题,相较于单变量时间序列,多变量时间序列的研究复杂性更高,难度更大。尽管当前已经提出了许多针对多变量时间序列的聚类算法,但是这些算法在精度和解释性方面仍旧不足。其一,当前大部分工作并未考虑多变量时间序列的长度冗余性和变量相关性等问题,导致最终得到的相似性矩阵具有较大误差;其二,数据在聚类过程中普遍采用划分范式,当数值空间呈现复杂分布时该思想表现不佳,并且不具备对各个变量及空间的解释力。针对上述问题,提出了一种面向Shapelet(富有高信息量的连续子序列)空间的多变量时间序列自适应权重密度聚类算法(MDCS)。算法首先对各个变量进行Shapelet搜索,通过自适应策略获取到各自的Shapelet空间,接着对各个变量产生的数值分布进行组合加权,得到了更符合数据分布特征的相似度矩阵,最后利用改进密度计算和二次分配的共享最近邻密度峰值聚类算法对数据进行最终分配。在真实数据集上的实验结果证明,与目前先进的聚类算法相比,MDCS拥有更好的聚类结果,在标准化互信息和兰德系数指标上平均提高了0.344与0.09,兼顾了性能与可解释性。

关键词:多变量时间序列;子序列;Shapelet空间;密度峰值聚类;数据挖掘
文献标志码:A　　**中图分类号:**TP181

Multivariate Time Series Density Clustering Algorithm Using Shapelet Space

SHENG Jinchao, DU Mingjing+, SUN Jiarui, LI Yurui
School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, Jiangsu 221100, China

Abstract: Multivariate time series clustering has become an important research topic in the task of time series analysis. Compared with univariate time series, the research of multivariate time series is more complex and difficult. Although many clustering algorithms for multivariate time series have been proposed, these algorithms still have difficulties in solving the accuracy and interpretation at the same time. Firstly, most of the current work does not consider the length redundancy and variable correlation of multivariable time series, resulting in large errors in the final similarity matrix. Secondly, the data are commonly used in the clustering process with the division paradigm, when the numerical space presents a complex distribution, this idea does not perform well, and it does not have the explanatory power of each variable and space. To address the above problems, this paper proposes a multivariate time series adaptive weight density clustering algorithm using Shapelet (high information-rich continuous subsequence) space (MDCS). This algorithm firstly performs a Shapelet search for each variable, and obtains its own Shapelet space through an adaptive strategy. Then, it weights the numerical distribution generated by each variable to obtain a simi-

基金项目:国家自然科学基金(62006104,61872168);江苏省高校自然科学基金(20KJB520012)。
This work was supported by the National Natural Science Foundation of China (62006104, 61872168), and the Natural Science Foundation of Jiangsu Higher Education Institutions (20KJB520012).
收稿日期:2022-11-24　　**修回日期:**2023-01-17

larity matrix that is more consistent with the characteristics of data distribution. Finally, the data are finally allocated using the shared nearest neighbor density peak clustering algorithm with improved density calculation and secondary allocation. Experimental results on several real datasets demonstrate that MDCS has better clustering results compared with current state-of-the-art clustering algorithms, with an average increase of 0.344 and 0.09 in the normalized mutual information and Rand index, balancing performance and interpretability.

Key words: multivariate time series; subseries; Shapelet space; density peak clustering; data mining

针对多变量时间序列(multivariate time series, MTS)[1]的分析已经成为时序挖掘领域的一个研究热点。例如图1所示,MTS可被认为是同一事物在多个不同状态下所产生的序列集合。相较于单变量时间序列(univariate time series, UTS),MTS拥有更多的变量个数,比如在自动驾驶状态下,需要同时分析相机、激光雷达等多种传感器信息;在捕捉脑电波数据过程中,测量仪会对不同脑部区域的数据进行记录等。

MTS不仅具有维度高、变量多等特点,而且时常伴有噪声、数据漂移等问题,这导致了传统的数据挖掘方法不能直接有效地应用到MTS上。在对MTS的学习过程中发现,虽然可以将MTS拆分为多条单独的UTS进行分析,但由于多个变量之间存在着相当重要的关联性,缺少这部分的信息会对最终结果产生影响。

聚类[2]是机器学习无监督领域中最重要的技术之一,其本质是为了揭示数据之间的内在关联,利用数据表达、特征提取、相似度度量以及具体的算法模型将相似的数据聚成一个簇,相异的数据尽可能分离。当前聚类技术正在快速变化与发展,它也逐步成为了大量复杂数据挖掘任务的子过程与中间件,例如异常检测[3]、强化学习[4]等。时间序列聚类(time series clustering, TSC)[5]作为时间序列分析的一个重要分支,对时序的发展具有重要的研究意义和应用价值。近几年学者们对MTS聚类中相似度量和模型搭建开展了诸多探索,这些工作大致可以分为三类:基于降维的方法、基于距离的方法和基于深度学习的方法。

基于降维的方法将每条MTS数据看成一个矩阵,通过采用各种矩阵分解手段,从横向或纵向等方式将数据转变为一个有限向量或一个更小的矩阵。

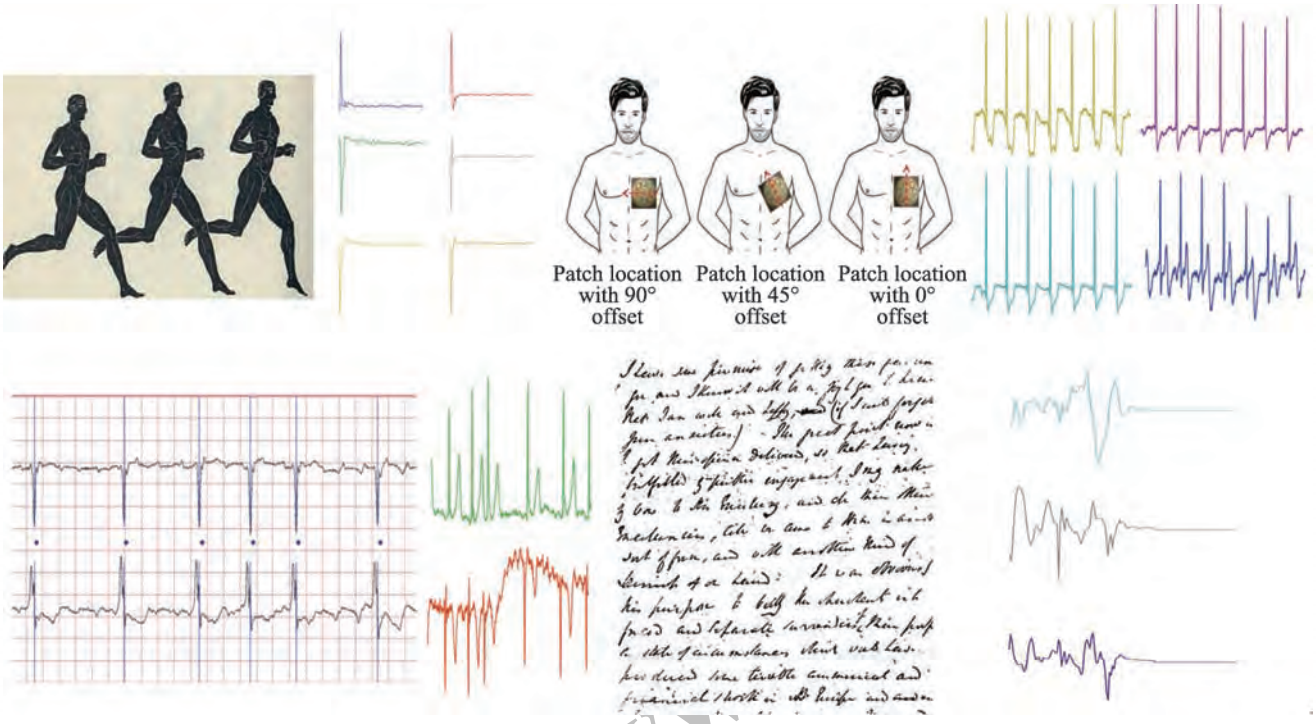


图1 多变量时间序列

Fig.1 Multivariate time series

MC2PCA（multivariate time series clustering method based on common principal component analysis）[6]和SWMDFC（spatial weighted matrix distance based fuzzy clustering）[7]是该方法的两个典型。前者通过迭代方式更新每个簇的簇划分和公共投影，并利用公共投影的重构误差重新分配簇的数目；后者则利用主成分分析方法降低时序数据的维度和变量个数，最后采用一种基于空间加权矩阵距离的模糊聚类方法进行数据标注。另外，BCNC（based on complex networks for multivariate time series clustering）[8]利用复杂网络将多变量时序映射成近邻关系图并利用社区检测技术中的鲁汶算法实现聚类；cnNMF（non-negative matrices factorization based on network construction）[9]使用KNN（K-nearest neighbor）构建组件关系网络，随后采取非负矩阵分解和模糊隶属度矩阵对数据进行聚类；cACM（component attributes based affinity propagation clustering for multivariate time series data）[10]构建数据之间的总体距离矩阵和分量近似距离矩阵，最后使用近邻传播算法实现数据划分。

基于距离的方法主要将UTS聚类方法进行了多维拓展。比如Ozer对K-Shape（shape-based time series clustering）和K-SC（K-spectral centroid clustering）进行改进，提出了MK-Shape（multidimensional K-Shape）[11]和MK-SC（multidimensional K-SC）[11]。FCFW（fuzzy clustering based on feature weights）[12]和RP-VWKM（variable-weighted K-medoids clustering algorithm based on a reverse nearest neighborhood-based density peaks approach）[13]在*K*-means[14]思想基础上采用了相近的优化思路，两者都利用拓展后的基于形状的距离度量（shape-based distance measure，SBD）作为相似计算方法，并利用密度峰值聚类[15]算法初始化簇中心，但在变量加权和聚类过程方面存在不同；FCFW根据不同变量的最大与最小值之差作为衡量权重的方法并使用K模糊算法[16]聚类，RP-VWKM则使用了类似于WOCIL（subspace clustering of categorical and numerical data with an unknown number of clusters）[17]的加权聚类相似度学习方法，将同簇和非同簇中各自数据的均值和方差作为权重，最终采用迭代方式归类。

目前，基于深度学习的MTS聚类方法较少，但其也逐渐成为一个重要的研究分支。USRL（unsupervised scalable representation learning）[18]利用编码器架构和三元组损失来训练模型，该模型可以处理可变长度的输入并获得稳定和高质量的特征；DeTSEC（deep time series embedding clustering）[19]则采用注意力和门控机制来获得时序数据的嵌入表示，最终用*K*-means获得集群结果。

从上述分析可见，大部分MTS聚类算法在执行过程中并没有捕捉关键变量与信息的能力，可解释性较弱。并且这些方法大多都是基于划分范式的，随着数据量和变量个数的增大，其应对复杂数据空间的能力会变得非常有限。基于U-Shapelet的UTS聚类算法[20]因其具有可解释性能力和高质量的聚类性能受到了国内外学者的广泛研究，但由于U-Shapelet只面向UTS，因此研究将Shapelet（提供显著区分能力的连续子序列）思想推广到MTS领域很有现实意义。然而如何获取MTS中的U-Shapelet集合以及各个变量之间权重如何与U-Shapelet关联成为了该项工作的研究难点。同时由于当前面向多变量U-Shapelet工作相对较少，特别是U-Shapelet选择算法，给与研究的相应参考与指导有限。

本文从U-Shapelet的基本概念出发，在多变量加权距离度量和复杂高维空间聚类方面进行了创新与改进，提出并设计了一种面向Shapelet空间的MTS自适应权重密度聚类算法（multivariate time series adaptive weight density clustering algorithm using Shapelet space，MDCS）。如图2所示，算法首先对各个变量进行了自适应Shapelet搜索，利用空间分布距离统计指标寻找符合当前变量最优分布的距离矩阵；接着利用组合权重的方式获取各个变量的相关性，主要包括使用KL散度获取互补性和相关系数获取一致性，进而对产生的多个矩阵进行融合；最后，利用改进密度计算和二次分配方式的共享最近邻密度峰值聚类算法[21]对数据进行最终分配。总结本文的贡献如下：

（1）提出了一种面向U-Shapelet的MTS聚类选择算法，算法充分发挥显著子序列的优势，舍弃了大部分平凡和高噪声的数据，得到了各个变量中最重要的局部结构，增强了可解释性。

（2）引入一种组合权重的方式对各个变量产生的距离矩阵进行加权，这种融合过程类似于联合子视角的方法，因而有助于得到有价值数据蕴含的一致性与互补性信息。

（3）考虑了矩阵计算过程中由于数值线性叠加而可能产生的空间密度分布不均问题，进而利用改进共享最近邻密度计算方式与加快二次分配的方法对数据进行归类。最终与11个目前流行且重要的

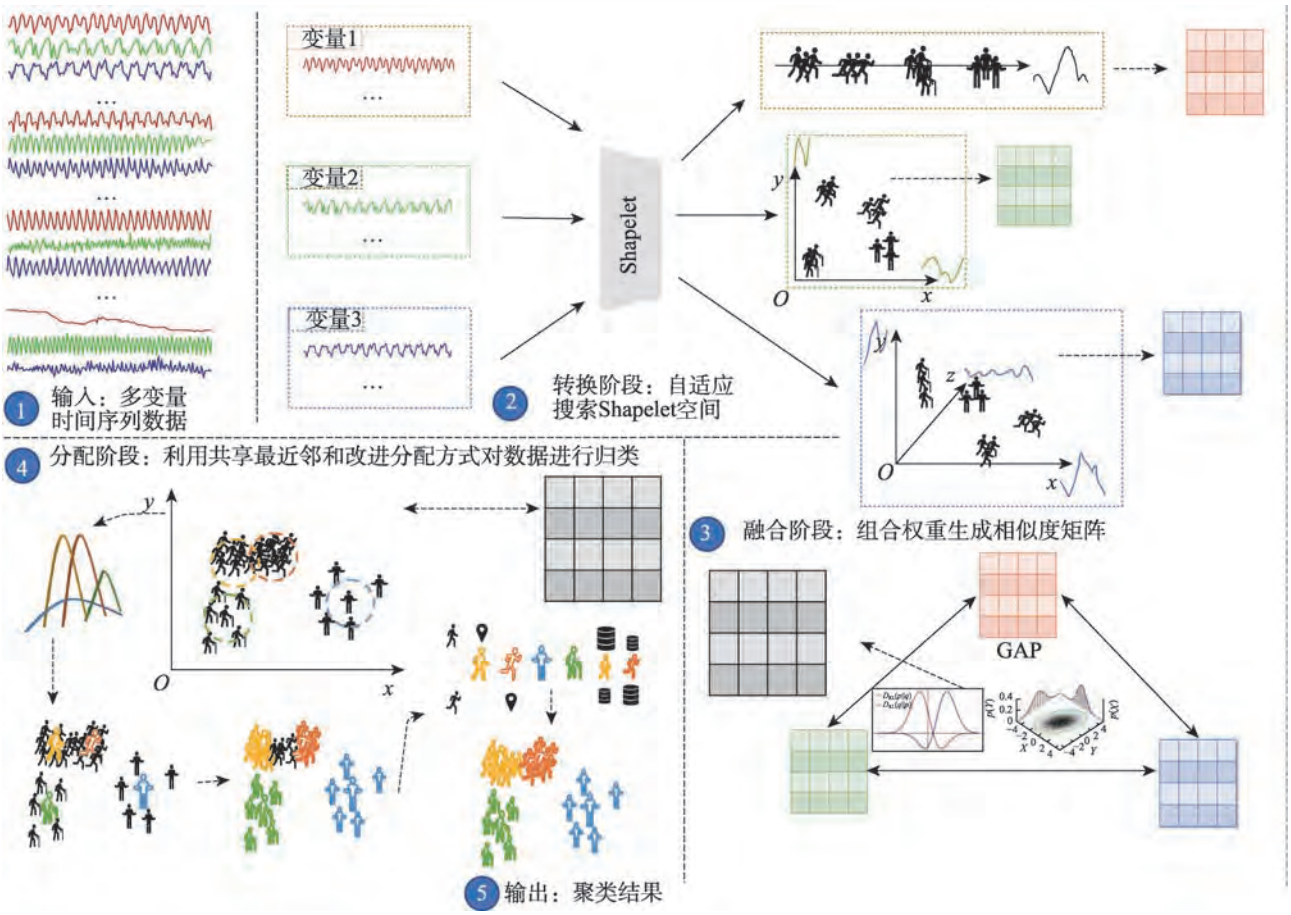


图2 MDCS算法流程

Fig.2 MDCS algorithm process

MTS聚类算法在23个数据集进行对比，MDCS展现了其较为优异的聚类性能。

1 相关概念

表1给出本文需要使用的记号摘要。

1.1 U-Shapelet

U-Shapelet（unsupervised-shapelet）[22]由Zakaria首次提出，其中Shapelet概念是由Ye和Keogh等人[23]在时间序列分类领域首次命名并指代为UTS中最具有判别类别信息的连续子序列。Shapelet思想由于使用了一些局部序列段代替整条时序数据，能够很好地捕捉到同类数据最显著的共同信息，同时忽略大量的无效内容与噪声。另外，该方法与人类认知指标接近，具有天然的可解释性；并且由于利用了局部原理，该方法还能够对不同长度的时序数据集进行实验，具备高度的拓展性。在介绍U-Shapelet相关聚类算法前，还需要给出如下的一些概念与定义。

表1 记号摘要

Table 1 Summary of notations

符号	意义
T_i	一条UTS数据实例，其中长度为 l，故 $T_i=(t_{i1},t_{i2},\cdots,t_{il})$
$T_{i(a,b)}$	一条UTS子序列，其中 $T_{i(a,b)}=(t_{ia},t_{ia+1},\cdots,t_{ib})$，$0\leqslant a\leqslant b\leqslant l$
M_i	一条MTS数据实例，其中变量个数为 m，即 $M_i=(T_i^1,T_i^2,\cdots,T_i^m)$
$M_{i(a,b)}$	一条MTS子序列，其中 $M_{i(a,b)}=(T_{i(a,b)}^1,T_{i(a,b)}^2,\cdots,T_{i(a,b)}^m)$，$0\leqslant a\leqslant b\leqslant l$
D	一个MTS数据集，其中共有数据实例 n 条，即 $D=(M_1,M_2,\cdots,M_n)$
T_{is}	单条U-Shapelet
S	U-Shapelet集合
SsD	Shapelet空间
X_{SsD}	Shapelet空间距离矩阵
c	真实类别数
$\mathbb{C}$	类标签集合
K	最近邻个数

定义1（子序列之间的距离计算）两条子序列 $T_{u(a,b)}=\{t_{ua},t_{ua+1},\cdots,t_{ub}\}$ 和 $T_{v(c,d)}=\{t_{vc},t_{vc+1},\cdots,t_{vd}\}$ 之间的距离 $sdist$ 如式(1)所示。注意：这里假设 $T_{u(a,b)}$ 的长度小

于或等于 $T_{v(c,d)}$ 的长度，即 $(b-a+1)\leqslant(d-c+1)$，$J=d-c-b+a$。

$$sdist(T_{u(a,b)},T_{v(c,d)})=\min_{j=\{1,2,\cdots,J\}}\frac{1}{b-a+1}\sum_{q=a}^{b-a+1}(t_{uq}-t_{v(j+q-1)})^2 \quad (1)$$

定义2（U-Shapelet）U-Shapelet T_{is} 是一条特殊的子序列，它能将数据集中的所有数据划分为两个集合 D_A 和 D_B，其中划分依据满足 $sdist(T_s,D_A)\ll sdist(T_s,D_B)$，$len(D_A)>1$ 即与 D_A 中时序数据之间的距离远小于 D_B 中的距离。

定义3（Gap value）给定一个距离阈值 dt，若将与U-Shapelet T_{is} 之间的距离小于 dt 的数据划分为 D_A，反之划分为 D_B，则当前U-Shapelet的Gap value如式(2)所示：

$$Gap(T_{is},dt)=\mu_{D_B}-\sigma_{D_B}-(\mu_{D_A}+\sigma_{D_A}) \quad (2)$$

其中，μ_{D_A}、μ_{D_B}、σ_{D_A} 和 σ_{D_B} 分别代表 D_A 和 D_B 中时序数据与U-Shapelet之间距离的均值与标准差。

定义4（Shapelet空间）将U-Shapelet集合 S 与UTS数据集进行距离计算，所得的新数值空间称为Shapelet空间 SsD，由于Shapelet空间只保留了原始时序数据中具有类别区分度的显著子序列之间的信息，该数值空间能够更好地展示不同簇之间的分布。需要注意的是，如果 S 的个数为 v，那么对于UTS来说，数值空间的大小会由原先的 $n\times l$ 变为 $n\times v$。而对于MTS来说，由于各个变量的数值空间会独立进行变换，每个变量产生的U-Shapelet数量并不保证相同。图3展示了一个UTS数据集上关于U-Shapelet集合与Shapelet空间的例子，可以看到产生的Shapelet空间具备了良好的区分度和聚类准确性。

暴力U-Shapelet搜索（brute force U-Shapelet search，BFUS）是首个面向U-Shapelet的聚类算法。其实现过程基于Gap value，利用子序列计算方法循环得到多个在当前数据集中具有最优区分度的U-Shapelet，然后将原始的时序数据与 S 进行距离计算，进而产生了一个新的基于Shapelet的数值空间，最终使用 K-means算法对该空间进行聚类。

BFUS在执行的过程中依赖数据的输入特征，具有重要形态区分的时序数据能够得到较好的聚类结果，但由于BFUS需要反复计算子序列之间的距离，

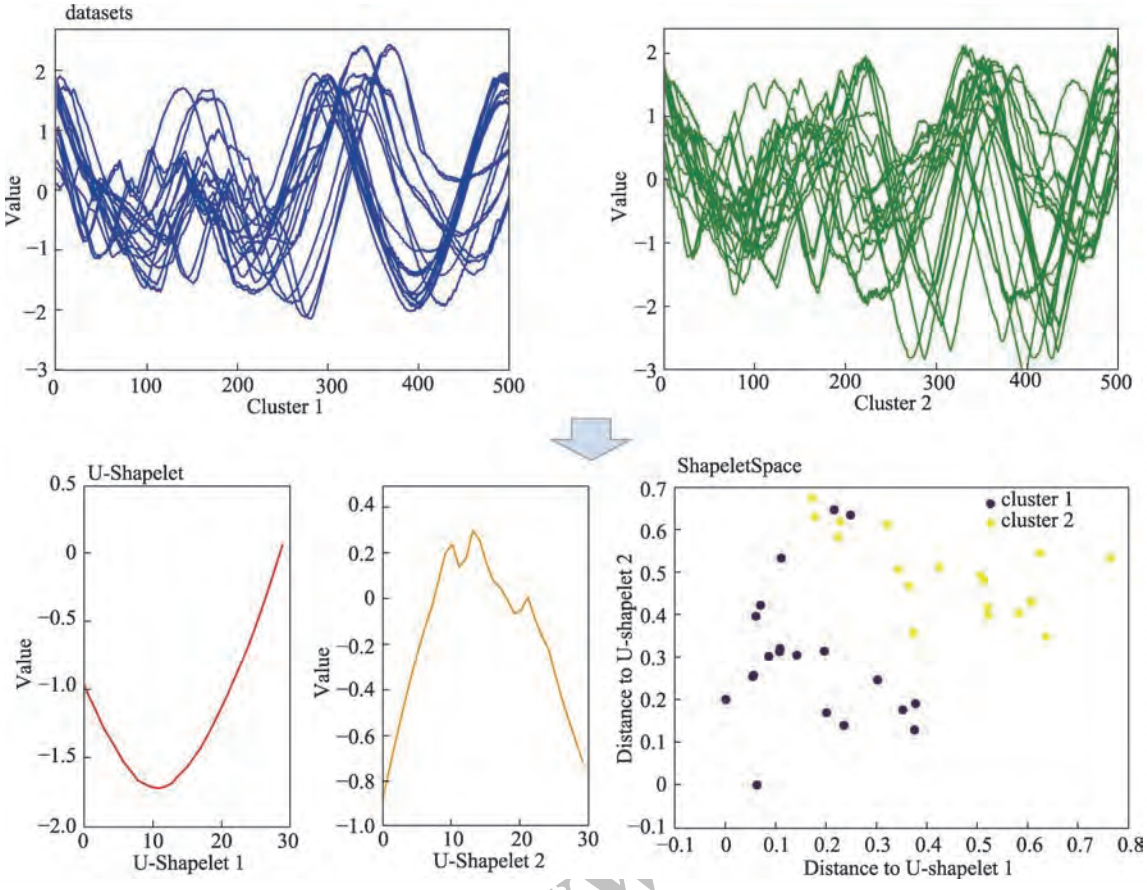


图3 U-Shapelet与Shapelet空间的聚类准确性

Fig.3 U-Shapelet and clustering accuracy in Shapelet space

算法在最坏情况下的时间复杂度为 $O(n^2l^4)$，其中评估单条子序列质量的时间复杂度为 $O(nl^2)$。近年来，诸多学者提出了针对BUFS时间复杂度问题的改进。例如Ulanova等人[24]利用符号化技术对UTS进行降维，同时利用随机映射的方式快速筛选候选子序列；Zhang等人[25]利用期望最大化思路迭代学习U-Shapelet等，但是由于这些算法族只面向UTS，无法在MTS上执行。

1.2 SNNDPC算法

共享最近邻密度峰值聚类算法（shared nearest neighbor density peaks clustering，SNNDPC）[21]是为了解决密度峰值聚类（density peaks clustering，DPC）容易产生的密度分布不均和链式错误而提出的。它能够有效应对复杂的数据分布，适合任意形状任意密度的空间。SNNDPC在DPC的基本思路上额外增加了共享最近邻、局部相似度矩阵和二次分配的概念。

定义5（共享最近邻） $SNN(T_i,T_j)=\Gamma(T_i)\cap\Gamma(T_j)$ 表示为两个数据 T_i 和 T_j 各自 K 近邻集合之间的交集，其中 $\Gamma(T_i)$ 表示距离 T_i 最近的 K 个数据点集合。

定义6（局部相似度矩阵） 局部相似度矩阵是一个方阵，方阵中的值表示对应数据之间的局部相似度，计算过程如式（3）所示：

$$Sim(T_i,T_j)=\begin{cases}\dfrac{|SNN(T_i,T_j)|^2}{\sum\limits_{p\in SNN(T_i,T_j)}(dist(T_i,T_p)+dist(T_j,T_p))}, & T_i,T_j\in SNN(T_i,T_j)\\ 0, & T_i,T_j\notin SNN(T_i,T_j)\end{cases}\tag{3}$$

二次分配是指将数据按照密度特征二分类后，利用分类结果进行的两次分配过程。其中每个数据的划分依据必须先基于寻找到的密度峰值点，如果数据 T_i 与某个密度峰值点之间的共享最近邻个数大于 $K/2$，则认为该数据 T_i 为当前密度峰值点的不可避免从属点，反之为可能从属点。第一次分配过程会将所有的不可避免从属点归类，第二次分配过程则会循环统计可能从属点各自最近邻数据中已分配数据集合所在的最大簇，将拥有最大数量的可能从属点分配到其对应的最大簇中，直到所有可能从属点分配完毕。

总结SNNDPC算法，它利用定义5计算数据的局部密度 ρ_i 和距离最近的较大密度值之间的距离 δ_i 构建决策图，进而根据决策图选择密度峰值点，最后对非密度峰值点进行二次分配。SNNDPC算法最终的时间复杂度为 $O((K+l)n^2)$。

2 聚类算法

2.1 算法动机

从概念可知，MTS中的每个变量均为被测物体在同一时间段上某一特定角度的信息，因此不同变量的子序列之间必然存在关联性。从聚类角度来说，这种关联性也可以细化为两类，其一是互斥的关联性，其二是平凡的关联性。进一步解释这两种关联性，前者更侧重于子序列之间不具备可替代性，即子序列均能够显著代表自己所属的类，而后者与之相反，子序列基本都是趋同相近的，因而提供的信息差异较少。图4给出了在StandWalkJump数据集中单条MTS数据两个变量的数值曲线，这两条曲线的标注部分形象化地介绍了这两种关联性，即红色箭头指向的互斥关联性和黑色直线囊括的平凡关联性。从图中可明显发现，挖掘互斥关联性而忽略平凡关联性能够获得更有意义的信息。

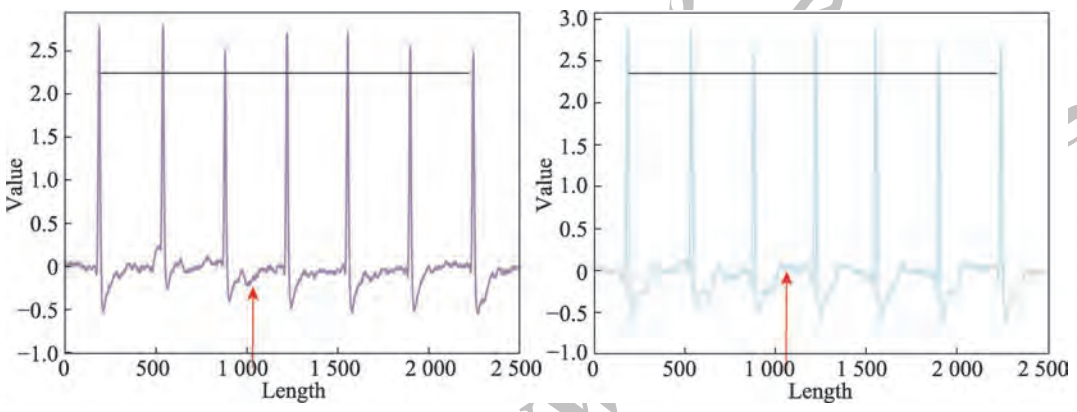


图4 StandWalkJump中单条数据的两个变量

Fig.4 Two variables for one data in StandWalkJump

U-Shapelet思想能够有效搜索UTS的显著局部子序列集合，故利用U-Shapelet思想挖掘互斥关联性具备可行性。当前背景下，最简单和常见的思路是将时间段对齐，记为 $M_{i(a,b)}$，并在整条长度内寻找相应的解，但是这种方式并不能保证每个变量的显著信息同时出现在某个时间段内，因而获得的子序列集合普遍不能达到最优解。如图5在Lp5数据集中两条数据的两个变量通过搜索后找到的子序列集合（用两条橙色竖线包裹）在第一个变量（紫色）上区分能力较强，而第二个变量（绿色）中该子序列不具备明显差别。目前，这种局部最优问题在U-Shapelet学习算法上相当普遍。例如MUSLA（multiview unsupervised multivariate Shapelet learning with adaptive neighbors）[26]利用多视图方法学习多条不同长度的 $M_{i(a,b)}$，虽然该方法聚类效果较好，但其本身增大了U-Shapelet集合的冗余度且参数设置非常依赖人为经验。故本文认为，在多元U-Shapelet选择算法上对各变量单独做出最优U-Shapelet选择能够避免对齐问题，同时有效缓解局部最优问题。

此外，对图5变量之间的形状进一步观察可以发现，MTS中每个变量提供的信息量不完全相同，因此各个变量之间应该赋予一定的权值，并尽可能让提供更多信息的变量（如图5中的第一个变量）得到更多的权重。

接下来将具体实现这部分构思。

2.2 转换阶段

当前UTS聚类算法普遍采用了一种顺序增加 T_{is} 并聚类的方式获取到最佳Shapelet空间。该方法每次添加单条 T_{is} 后就进行聚类，将聚类后得到的兰德系数（Rand index，RI）[27]与前一次聚类得到的RI进行计算，得到CRI（change RI，当前RI与前一次的差值），最终利用肘部法寻找最优的 $\mathbb{C}$ 作为聚类结果。这种手段存在如下问题：（1）需要提前输入聚类个数；（2）聚类多次，算法实际运行效率不高；（3）如果直接拓展到MTS中，那么单个变量对 c 的敏感度可能不同，若该变量只存在平凡关联性子序列，则其提供的信息会变得非常有限。MDCS利用了一种自适应方法寻找最优Shapelet空间，通过对不同Shapelet

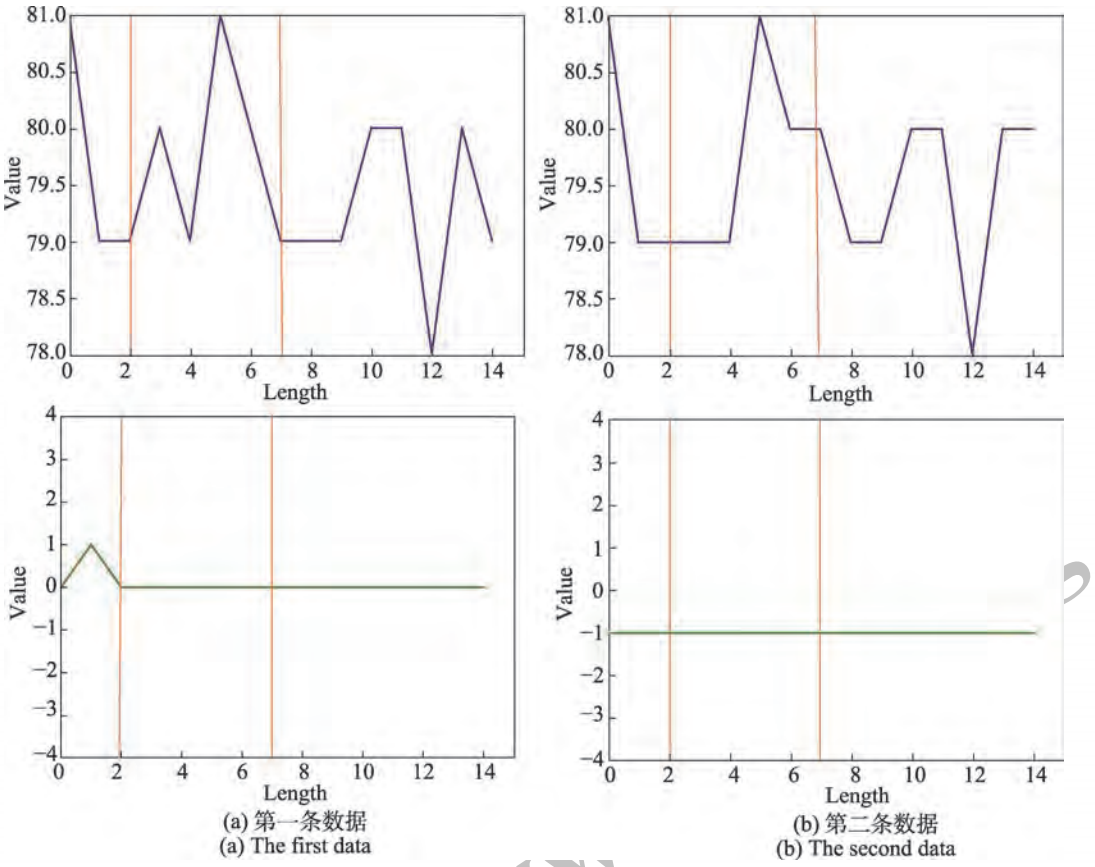


图5 Lp5中两条数据的两个变量

Fig.5 Two variables for two data in Lp5

空间计算衡量其分布能力的指标 MSD，将顺序增加 T_{is} 并聚类的模式改变为寻找所有 Shapelet 空间中 MSD 最好的过程，相较之前的方法更加容易用代码实现，并且不需要执行额外的聚类，详细细节见算法 1。

$$dist(T_i^x S, T_j^x S) = \sum_{y=1}^{len(S_{local})} gap_y \cdot ED(T_i^x S^y, T_j^x S^y) \tag{4}$$

$$MSD = avg \sum_{i=1}^{K} \frac{\sum \Gamma_{\phi \cdot n}^{dis}(T_i^x S)/\phi \cdot n}{\sum \Gamma_{\psi \cdot n}^{dis}(T_i^x S)/\psi \cdot n} \tag{5}$$

算法 1 自适应方式的 Shapelet 空间计算

输入：D，超参数 K、ϕ 和 ψ。

输出：多个不同的 Shapelet 空间 $\{SsD_i\}_{i=1}^m$。

步骤 1 获取 D 中的变量个数 m 并对每个变量执行 U-Shapelet 搜索算法，得到各个变量相应的 S 和 S 中各条 U-Shapelet 的 Gap value。

步骤 2 将各个变量单独执行步骤 3~步骤 6 中的内容，将结果汇总到步骤 7。

步骤 3 按搜索算法获取到 U-Shapelet 的顺序提取单条 U-Shapelet 及其 Gap value 放入临时 U-Shapelet 集合 S_{local} 并按照式(4)计算 Shapelet 空间距离矩阵 $\boldsymbol{X}_{SsD}$。

步骤 4 统计每个数据最近的百分之 ϕ 数据量的距离和，选择最小的前 K 个数据点作为候选点，并按照式(5)量化 Shapelet 空间分布能力 MSD。

步骤 5 判断当前 U-Shapelet 是否为 S 中的最后一条 U-Shapelet，是则跳入步骤 6，否则回到步骤 3。

步骤 6 统计 MSD 中数值较大的前百分之 ψ 个 Shapelet 空间并降序排序，依次计算每个空间与排序前相邻空间之间的差值，寻找差值最小的空间作为该变量的 Shapelet 空间 SsD。

步骤 7 返回各个变量的 Shapelet 空间 $\{SsD_i\}_{i=1}^m$。

式(4)中的 gap 是经过归一化后的权重，$T_i^x S^y$ 代表了 MTS 数据 M_i 在第 x 个变量上与第 y 个 U-Shapelet 的 $sdist$，ED 为欧式距离函数。式(5)中的 $\Gamma_{\phi \cdot n}^{dis}(T_i^x S)$ 与 $\Gamma_{\psi \cdot n}^{dis}(T_i^x S)$ 分别表示候选点最近的前 $\phi \cdot n$ 与 $\psi \cdot n$ 数据之间的距离值。

详细分析 MSD 的作用，它被量化为各个候选点在 $\Gamma_{\phi \cdot n}^{dis}(T_i^x S)$ 和 $\Gamma_{\psi \cdot n}^{dis}(T_i^x S)$ 上距离和平均之比的均值，体现了理想 Shapelet 空间下各个高密度区域的局部特征即数据与近邻数据之间呈现明显的扩散分布。在离群值较多或数据点随机分布的异常情况下，MSD 同样会较大，但这种异常值不会具有连续性，因此为了避免这个问题，设计了步骤 6 中的最优选择方式即比率较大的同时与近邻空间之间的指标差值较小。另外，步骤 4 中候选点的选择主要考虑了最近邻数据集合中距离之和较小的数据，而通常情况下这些数据是密度值较高的。

2.3 融合阶段

由 2.1 节可知，MTS 中每个变量所提供的信息量大小并不完全一致，因而在融合多个 Shapelet 空间时需要考虑权重的影响。影响权值的因素有很多，其中最重要的包括以下三种：变量所处空间本身质量、变量之间存在的互补性信息以及一致性信息。MDCS 从非迭代式的路由协议机制中获得启发，使用了组合权重的方式计算每个变量的权重，如式(6)所示：

$$w^m = \lambda_1 \beta_1^m + \lambda_2 \beta_2^m + \lambda_3 \beta_3^m, 0 \leqslant \lambda_1, \lambda_2, \lambda_3 \leqslant 1, \sum_{i=1}^{3} \lambda_i = 1 \tag{6}$$

其中，β_1^m、β_2^m、β_3^m 分别代表了组合权重中的固定权重、互补性权重和一致性权重。

固定权重方面，MDCS 使用了 Gap value 作为每个变量所在 Shapelet 空间中的平均质量，如式(7)所示，优秀的 Shapelet 集合会拥有更高的均值，因此该固定权值会更大。

$$\beta_1^m = avg\left(\sum_{n=1}^{len(S)} gap_n \right) \tag{7}$$

互补性权重方面，MDCS 使用 KL 散度(Kullback-Leibler divergence)作为主要的计算方法。KL 散度在信息系统中又称为相对熵，主要用于衡量两个分布之间的匹配程度或差异性，计算方式如式(8)所示，在本文中它主要起到了量化当前 Shapelet 空间与其他变量中优质 Shapelet 空间拟合度的作用。在实际计算过程中，KL 散度不具备对称性，因此计算结果依赖输入顺序。基于 KL 散度的权重计算模块共分为两部分，分别为成对计算和结果修正。成对计算部分统计每两个不同变量 Shapelet 空间距离分布之间的 KL 散度并将结果保存于一个矩阵 $\boldsymbol{X}_{KL}$ 中。结果修正部分源于如下事实，即单个变量与其他变量计算得到的 KL 散度值会出现两种距离分布结果：其一，当该变量为存在大量互斥关联性的子序列时，互补性结果会出现个别 KL 值较低，大部分 KL 值较高的情况；其二，当该变量为平凡性变量时，KL 值会出现同步较大或较小的情况。为了统一修正这两部分因素，采用式(9)的计算方式($ascsort$ 函数为升序排列)，保证在出现第一种情况下其结果会相对较大，而第二种情况下，覆盖值由于利用 ϕ 进行了缩减，因而最终结果并不会很大。

$$D_{\mathrm{KL}}(p||q)=-\int p(X_{SsD})\ln\frac{q(X_{SsD})}{p(X_{SsD})} \tag{8}$$

$$\beta_2^m=\frac{1}{\sum_{i=1}^{\phi\cdot n} ascsort(X_{\mathrm{KL}}(m,:))[i]} \tag{9}$$

如果说互补性权重主要衡量两个 X_{SsD} 之间的匹配程度，那么一致性权重则更侧重衡量空间本身的分布优异程度及与其他分布的同步趋势能力，其也有助于丢弃离散和平凡的空间信息。MDCS使用了相关系数和方差修正进行描述，如式(10)，其中相关系数只统计正相关值。在高质量的Shapelet空间中，经过一致性权重得到的方差会较大且会拥有较高的相关系数值，因此其一致性权重也会变大。

$$\beta_3^m=\left(\sum_{z!=m,r(p,q)>0} r(X_{SsD}^m,X_{SsD}^z)\right)+Var(X_{SsD}^m) \tag{10}$$

当一个理想的Shapelet空间出现时，β_1^m 会被赋予较高的权重。同时该变量与其他变量计算得到的KL散度值会处于结果修正部分的第一种情况即与同步空间的值较小、非同步空间值较大，进而该部分互补权值的结果会通过式(9)进行修正，最终若同步空间越多，β_2^m 取值就越大。β_3^m 方面，更好的Shapelet空间会得到更高的方差值，并且若相似空间越多，则正相关系数的个数会越多，最终权重自然也会更大。相反，平凡或离散的Shapelet空间所计算到的固定权重会较低，β_2^m 和 β_3^m 带来的信息也会被其数值空间呈现的异样分布而稀释，无法与其他空间产生关联，因此最终的权值会更小。

总结融合阶段如算法2所述。需要补充说明的是，为了避免量纲问题带来的影响，最终的输出矩阵会进行归一化，形成相似度矩阵。

算法2 融合Shapelet空间生成相似度矩阵

输入：$\{SsD_i\}_{i=1}^m$、λ_1、λ_2、λ_3。

输出：相似度矩阵 $\boldsymbol{X}_{\mathrm{Sim}}$。

步骤1 对每个 SsD_i 执行式(6)，计算得到 X_{SsD}^m 和权重 w^m。

步骤2 利用 w^m 融合各个 X_{SsD}^m 得到 $\boldsymbol{X}_{\mathrm{Sim}}$。

2.4 分配阶段

由于 $\boldsymbol{X}_{\mathrm{Sim}}$ 还是源于距离的线性叠加，该空间不可避免会出现密度分布不均问题。SNNDPC虽然能够处理该问题，但是在面临更加复杂分布的情况下，原始高斯截断核密度评估方法的快速衰减特性还是容易忽视较远数据。本文选用更长函数尾部特征的柯西核为评估核[28-29]，具体如式(11)所示，其中 T_K 为距离 T_i 第 K 近的数据，最终需要计算的 ρ_i 和 δ_i 如式(12)和式(13)。

$$f^c(T_i,T_j)=\left(\frac{dist(T_i,T_j)^2}{dist(T_i,T_K)^2}+1\right)^{-1} \tag{11}$$

$$\rho_i=\sum_{T_j\in\Gamma(T_i)}\begin{cases}\dfrac{|SNN(T_i,T_j)|^2}{\sum_{p\in SNN(T_i,T_j)}(f^c(T_i,T_p)+f^c(T_j,T_p))}, T_i,T_j\in SNN(T_i,T_j)\\ 0, T_i,T_j\notin SNN(T_i,T_j)\end{cases} \tag{12}$$

$$\delta_i=\min_{i\neq\rho_i>\rho_i}\left\{dist(T_i,T_l)\left[\sum_{T_a\in\Gamma(T_l)}f^c(T_i,T_a)+\sum_{T_b\in\Gamma(T_l)}f^c(T_l,T_b)\right]\right\} \tag{13}$$

在二次分配过程中，为了加快分配速度及提高分配效果，选择将可能从属点依照式(12)计算出的密度值降序排序，并按序将可能从属点 T_i 分配到 $\Gamma(T_i)$ 中距离最近的数据点所在的簇，当最近邻数据都处于未分配情况时，搜索空间会扩大一倍直到最终得到分配。这种改进分配的方式使得每个可能从属点只需计算一次就能够被分配，有效解决了重复统计问题，同时由于数据点一旦出现便立即被分配，算法也不会出现重复统计已分配数据点的问题。

总结上述内容，算法3给出了MDCS最终流程。

算法3 MDCS

输入：D、超参数 K、ϕ 和 ψ、λ_1、λ_2、λ_3。

输出：聚类结果 $\mathbb{C}$。

步骤1 执行算法1得到各自变量的Shapelet空间 $\{SsD_i\}_{i=1}^m$。

步骤2 执行算法2得到融合各个变量Shapelet空间的相似度矩阵 $\boldsymbol{X}_{\mathrm{Sim}}$。

步骤3 利用 $\boldsymbol{X}_{\mathrm{Sim}}$ 与式(12)、式(13)计算每条数据的密度值 ρ_i 和 δ_i。

步骤4 利用决策图选取密度峰值点，并对非密度峰值点进行不可避免从属点和可能从属点的划分，同时开展第一轮分配。

步骤5 进行第二轮分配，寻找当前密度值最高的可能从属点，按照改进的第二轮分配方式对该数据进行分配。

步骤6 记录最终的聚类结果 $\mathbb{C}$。

2.5 复杂度分析

算法1时间复杂度主要来源于Shapelet搜索和自适应距离计算，前者在目前大量优化算法改进的过程中平均时间复杂度不会超过立方级，即 $O(n^2l^3)$，后者主要取决于每个变量各自Shapelet空间 S 的个数

t，故总计为 $O(mtn^2)$，但实际情况下 t 是远小于 n 的常数。算法2的时间复杂度基于各个变量之间的矩阵计算，主要包括互补性权重和一致性权重，故为 $O(m^2n^2)$ 量级。而在分配阶段由于采取了加速策略和直接利用相似度矩阵进行运算，其时间复杂度为 $O(n^2)$。综上所述，MDCS算法总的时间复杂度为 $O(n^2(m^2+mt+l^3))$。

3 实验部分

3.1 实验设置

为了验证MDCS算法的聚类效果，选用了23组UCR公开真实数据集[7,10]进行实验，具体如表2所示，其中部分数据集存在非等长情况，为了便于与其他算法进行对比，本文采用补零手段补齐，但事实上MDCS对非等长数据同样适用。

表2 对比实验数据集

Table 2 Comparative experiment datasets

数据集	数量	变量个数	长度	类别数
ArticularyWordRecognition	575	9	144	25
AtrialFibrillation	30	2	640	3
BasicMotions	80	6	100	4
Cricket	180	6	1 197	12
ECG	200	2	152	2
Epilepsy	275	3	206	4
ERing	300	4	65	6
FingerMovements	416	28	50	2
HandMovementDirection	234	10	400	4
Handwriting	1 000	3	152	26
JapaneseVowels	640	12	29	9
Libras	360	2	45	15
Lp1	88	6	15	4
Lp2	47	6	15	5
Lp3	47	6	15	4
Lp4	117	6	15	3
Lp5	164	6	15	5
Natops	360	24	51	6
RacketSports	303	6	30	4
SelfRegulationSCP1	561	6	896	2
SelfRegulationSCP2	380	7	1 152	2
StandWalkJump	27	4	2 500	3
UWaveGestureLibrary	440	3	315	8

评价指标使用了聚类分析中常用的标准化互信息（normalized mutual information，NMI）和兰德系数（Rand index，RI）[27,29]，其中NMI和RI的取值范围均为[0,1]，指标的数值越大，聚类的效果越好。本文使用的编程环境为Python3.9。

对比算法方面，本文选择了当前流行的11个多变量时间序列聚类算法，分别为SWMDFC[7]、MC2PCA[6]、BCNC[8]、cnNMF[9]、cACM[10]、MK-Shape[11]、MK-SC[11]、FCFW[12]、RP-VWKM[13]、DeTSEC[19]和MUSLA[26]。在实验中，以RI指标为准，选择所有算法在各自参数范围内的最优聚类结果。

MDCS参数共有6个超参数，分别是 K、ϕ 和 ψ、λ_1、λ_2、λ_3。对于最近邻个数 K，本文遵循大部分文献的建议设置范围，将其规定为5~25之间。而 ϕ 和 ψ 的大小主要决定了Shapelet空间的质量和变量之间的互补覆盖量，为了具备普适性并尽可能覆盖到有效的数据量上，本文选择25和50，即覆盖25%和50%的数据。而对于 λ_1、λ_2 和 λ_3 的取值，3.4节的参数实验部分会对其进行详细说明。MDCS算法的最优结果同样以RI最大为准。

3.2 对比实验与分析

表3与表4分别展示了MDCS与各个对比算法获得的NMI和RI结果，可以看到，MDCS展现出了较好的聚类性能。在NMI上，MDCS完全领先其他算法的数据集共有9组，优于半数算法的数据集有20组，比率分别为39.1%和87%；在RI上，MDCS完全领先的数据集有11组，领先半数的共有21组，比率分别为48%和91.3%。若将这两个指标进行统一整理，那么MDCS共在8组数据集上同时取得了最好的聚类结果，比率达34.8%。由此可见，MDCS通过Shapelet空间和密度聚类思维的方式对数据的合理分配起到了关键作用。将这些数据结果进一步量化处理，得到了表中各个算法的平均聚类结果Avg和平均排名Rank。图6(a)和图6(b)分别对结果进行了可视化，从图中可以清晰地看到，MDCS的综合能力非常突出，平均指标和平均排名都取得了最好的结果，其中NMI平均提高34.4%、领先1.4个位次，RI平均提高9%、领先1.6个位次。

从数据集变量个数、长度等角度观察MDCS算法的聚类结果可以发现如下事实：(1)当时间序列长度偏小且数据量较少时，例如Lp1~5，MDCS执行过程中搜索到的U-Shapelet长度都偏小，因而产生的Shapelet空间并不容易达到最优，但相比K-means类型的算法（例如，MK-Shape和MK-SC）划分算法仍旧具备优势。(2)当时间序列长度和数据量均适中时，MDCS基本优于大部分降维与距离改进的算法，如Epilepsy。(3)当时间序列长度较长时，MDCS算法的聚类性能取决于数据集变量中是否存在明显的局部特征，例如Cricket数据集就存在明显的手势特征因而聚类效果更好；SelfRegulationSCP2数据集则是一种慢速皮质电位特征集，局部特征不明显，故聚类结果相对较低。

表3 NMI结果

Table 3 NMI results

Datasets	SWMDFC	MC2PCA	BCNC	cnNMF	cACM	MK-Shape	MK-SC	FCFW	RP-VWKM	DeTSEC	MUSLA	MDCS
ArticularyWordRecognition	0.492 8	0.527 2	0.506 2	0.889 6	0.850 1	0.858 8	0.824 3	0.431 8	0.737 8	0.429 0	0.838 2	**0.917 0**
AtrialFibrillation	0.153 0	0.204 2	0.219 1	0.075 7	**0.289 0**	0.115 2	0.064 9	0.210 9	0.215 4	0.154 7	0.208 5	0.215 0
BasicMotions	0.199 5	0.535 1	0.349 3	**1.000 0**	0.334 3	0.754 7	0.458 4	**1.000 0**	0.562 0	0.612 0	0.881 2	**1.000 0**
Cricket	0.499 6	0.471 3	0.402 6	**0.975 4**	0.661 8	0.885 3	0.578 1	0.871 4	0.767 2	0.760 9	0.760 9	0.935 3
ECG	0.072 6	0.150 8	0.053 3	**0.296 3**	0.004 3	0.016 4	0.243 5	0.179 1	0.264 8	0.001 3	0.229 2	0.259 9
Epilepsy	0.178 7	0.587 6	0.288 1	**0.627 6**	0.498 8	0.429 0	0.305 1	0.361 7	0.331 4	0.234 5	0.600 8	**0.627 6**
ERing	0.707 7	0.342 3	0.321 1	0.935 6	0.593 6	0.847 1	0.504 8	0.681 7	0.591 4	0.540 5	0.722 3	**0.936 0**
FingerMovements	0.004 8	0.014 2	**0.203 2**	0.003 6	0.016 3	0.011 4	0.001 5	0.012 4	0.000 4	0.005 3	0.000 8	0.008 8
HandMovementDirection	0.034 2	0.022 9	**0.363 2**	0.030 4	0.114 9	0.016 9	0.019 7	0.029 3	0.028 2	0.011 8	0.185 0	0.335 2
Handwriting	0.207 2	0.233 0	0.414 4	0.476 4	0.219 5	0.490 4	0.243 8	0.223 8	0.298 4	0.283 2	0.425 7	**0.512 0**
JapaneseVowels	0.323 2	0.173 0	0.482 7	0.573 8	0.582 6	0.470 8	0.383 2	0.520 0	0.399 6	0.402 1	0.462 6	**0.741 4**
Libras	0.398 5	0.231 6	0.522 5	0.611 6	0.305 7	0.582 3	0.515 7	0.505 9	0.581 0	0.437 0	0.660 9	**0.662 5**
Lp1	0.120 2	0.337 3	0.365 8	0.648 1	0.181 7	0.241 7	0.304 7	**0.685 5**	0.404 7	0.213 2	0.238 2	0.649 9
Lp2	0.310 0	0.408 9	**0.486 6**	0.463 1	0.178 5	0.365 5	0.302 0	0.108 4	0.377 4	0.399 8	0.359 9	0.300 5
Lp3	0.283 1	0.346 7	0.378 9	0.341 1	0.067 0	0.268 6	0.391 2	0.198 5	0.056 2	**0.413 7**	0.176 8	0.396 5
Lp4	0.031 5	0.320 3	0.218 2	**0.469 4**	0.266 3	0.152 0	0.277 9	0.398 9	0.051 3	0.195 2	0.244 6	0.362 2
Lp5	0.165 1	0.160 1	0.296 7	**0.450 8**	0.150 1	0.152 3	0.368 2	0.125 0	0.449 6	0.269 7	0.315 1	0.337 9
Natops	0.564 8	0.192 1	0.367 1	0.692 1	0.368 7	0.648 2	0.529 4	0.558 5	0.243 4	0.582 7	0.483 8	**0.760 4**
RacketSports	0.099 9	0.117 5	0.126 5	0.526 5	0.450 6	**0.547 2**	0.097 7	0.226 7	0.272 3	0.175 7	0.180 3	0.474 4
SelfRegulationSCP1	0.194 4	0.050 6	0.071 3	**0.294 5**	0.149 7	0.021 3	0.024 5	0.373 2	0.012 2	0.256 9	0.256 9	0.203 9
SelfRegulationSCP2	0.002 1	**0.019 4**	0.061 5	0.000 5	0.015 6	0.000 7	0.000 1	0.004 5	0.012 7	0.009 3	0.009 3	0.008 6
StandWalkJump	0.214 6	0.273 1	0.268 2	0.137 1	0.059 8	0.212 7	0.029 4	0.217 7	0.106 0	0.208 6	**0.347 5**	0.268 2
UWaveGestureLibrary	0.234 2	0.369 4	0.276 6	0.744 5	0.474 1	0.757 9	0.487 5	0.492 5	0.533 8	0.391 9	0.727 7	**0.797 3**
Avg	0.238 8	0.264 7	0.306 2	0.489 7	0.297 1	0.384 6	0.302 4	0.366 0	0.317 3	0.303 9	0.405 1	**0.509 2**
Rank	9.04	7.52	6.48	3.83	7.35	6.26	8.30	6.17	7.00	7.74	5.48	**2.83**

表4 RI结果

Table 4 RI results

Datasets	SWMDFC	MC2PCA	BCNC	cnNMF	cACM	MK-Shape	MK-SC	FCFW	RP-VWKM	DeTSEC	MUSLA	MDCS
ArticularyWordRecognition	0.927 3	0.932 3	0.780 3	0.982 9	0.975 6	0.971 2	0.971 8	0.858 6	0.947 4	0.922 8	0.976 8	**0.983 4**
AtrialFibrillation	0.574 7	0.613 8	0.593 1	0.432 2	0.632 2	0.579 3	0.565 5	0.570 1	0.620 7	0.441 4	**0.723 4**	0.574 7
BasicMotions	0.687 0	0.802 2	0.649 7	**1.000 0**	0.620 6	0.894 6	0.764 2	**1.000 0**	0.795 3	0.773 7	0.949 7	**1.000 0**
Cricket	0.826 3	0.880 1	0.763 4	**0.994 8**	0.907 4	0.958 3	0.898 4	0.961 3	0.907 2	0.913 6	0.913 6	0.983 6
ECG	0.586 1	0.628 2	0.543 0	**0.696 9**	0.485 2	0.555 6	0.660 8	0.649 5	**0.696 9**	0.539 7	0.508 8	0.682 0
Epilepsy	0.664 1	0.835 7	0.649 9	0.712 7	0.780 9	0.771 2	0.705 6	0.734 8	0.675 9	0.660 4	0.744 6	**0.931 8**
ERing	0.865 3	0.791 4	0.749 9	0.980 9	0.859 3	0.957 9	0.838 6	0.890 6	0.808 7	0.845 3	0.782 7	**0.982 9**
FingerMovements	0.501 8	0.503 9	0.501 3	0.501 1	0.502 5	0.503 9	0.499 7	0.502 1	0.499 0	0.500 8	0.499 2	**0.507 2**
HandMovementDirection	0.596 3	0.627 3	0.732 3	0.633 2	0.735 5	0.625 8	0.627 2	0.631 1	0.622 5	0.610 6	0.712 4	**0.744 0**
Handwriting	0.919 0	0.923 9	0.781 9	0.939 1	0.708 3	0.937 0	0.927 3	0.744 9	0.891 5	0.925 9	0.937 8	**0.946 9**
JapaneseVowels	0.738 8	0.872 0	0.881 9	0.894 4	0.890 1	0.853 6	0.840 0	0.877 1	0.841 3	0.846 2	0.867 9	**0.919 2**
Libras	0.860 4	0.853 5	0.918 4	0.918 0	0.880 1	0.913 9	0.904 4	0.900 6	0.898 3	0.890 5	0.918 5	**0.919 3**
Lp1	0.614 9	0.664 1	0.699 1	0.826 0	0.591 4	0.676 6	0.674 0	**0.835 9**	0.729 4	0.497 6	0.707 7	0.756 0
Lp2	0.698 4	**0.782 6**	0.735 4	0.770 6	0.625 3	0.754 9	0.715 1	0.552 3	0.743 8	0.762 3	0.722 5	0.752 8
Lp3	0.654 9	0.670 7	0.676 2	0.706 8	0.596 7	0.680 9	0.745 6	0.642 9	0.698 4	**0.729 0**	0.506 0	0.715 1
Lp4	0.516 9	0.699 4	0.592 2	0.695 4	0.637 0	0.584 1	0.598 7	0.619 5	0.621 3	0.541 7	0.589 3	**0.700 0**
Lp5	0.656 7	0.693 1	0.759 1	**0.768 1**	0.739 0	0.688 2	0.749 6	0.696 4	0.723 9	0.665 6	0.728 0	0.692 5
Natops	0.795 2	0.750 8	0.824 0	**0.887 4**	0.725 0	0.810 1	0.804 7	0.838 4	0.656 7	0.754 3	0.769 3	0.878 8
RacketSports	0.643 4	0.655 7	0.702 0	0.728 5	0.768 5	**0.800 3**	0.650 5	0.678 0	0.594 0	0.667 1	0.502 4	0.689 3
SelfRegulationSCP1	0.624 8	0.516 0	0.503 2	0.686 4	0.511 1	0.512 9	0.516 0	**0.725 3**	0.507 6	0.665 2	0.665 2	0.633 9
SelfRegulationSCP2	0.500 1	**0.512 0**	0.504 8	0.499 0	0.501 3	0.499 2	0.498 7	0.501 8	0.505 4	0.499 4	0.499 4	0.504 2
StandWalkJump	0.592 6	**0.683 8**	0.421 7	0.604 0	0.564 1	0.621 1	0.547 0	0.621 1	0.558 4	0.421 7	0.601 4	0.669 5
UWaveGestureLibrary	0.758 4	0.815 7	0.782 2	0.932 2	0.878 7	0.929 3	0.853 6	0.863 8	0.846 5	0.838 4	0.912 9	**0.941 0**
Avg	0.687 1	0.726 4	0.684 6	0.773 5	0.700 7	0.742 6	0.719 9	0.734 6	0.712 6	0.691 9	0.727 8	**0.787 3**
Rank	9.17	6.39	7.65	3.74	6.96	5.78	7.48	5.83	7.48	8.48	6.39	**2.65**

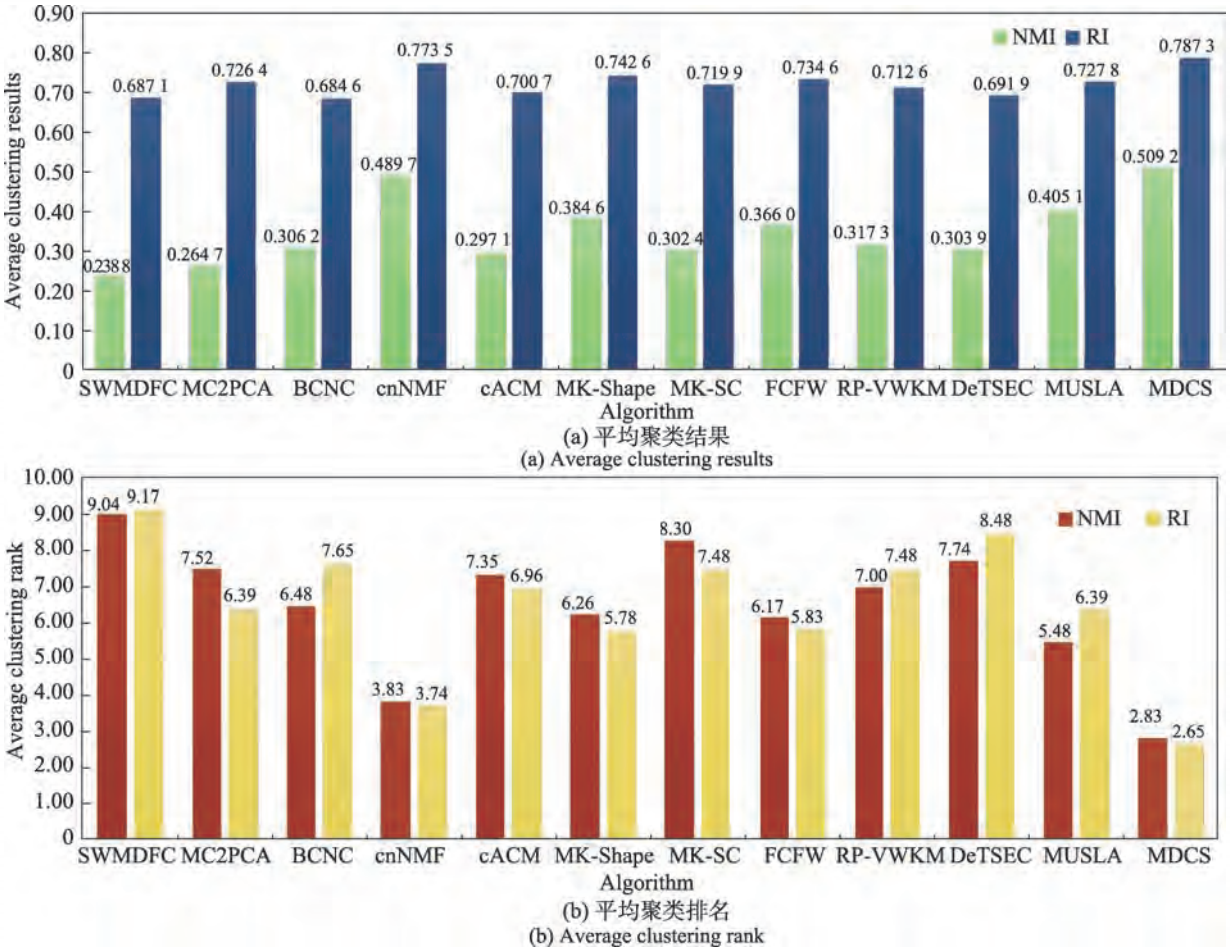


图6 算法聚类结果的统计指标

Fig.6 Statistical indicators of algorithm clustering results

图7展示了由各个算法在所有数据集上的RI值修正后组成的临界差分图，可以再次清晰看到MDCS算法明显优于其他算法。相较于SWMDFC，MC2PCA和BCNC等基于降维的聚类算法，MDCS利用局部思维保留了更多用于聚类的判别子序列Shapelet，提高了其精度与可解释性。而与MK-Shape和MK-SC这些基于距离的方法以及DeTSEC等深度方法相比，MDCS利用组合权重的方式获取变量之间的互补性与一致性，产生了具有更高区分度的相似度矩阵，并使用面向复杂分布的密度聚类方法，最终使得算法的有效性更好。与MUSLA这类基于U-Shapelet思想的学习算法相比，MDCS对每个变量单独进行了Shapelet空间搜索，有效降低了MTS子序列存在的局部最优问题，聚类性能更好，同时可解释性更强。

最后，简要分析MDCS算法与几个代表性聚类算法的计算复杂度，包括划分式聚类算法（MK-Shape、cnNMF等）和U-Shapelet聚类算法（MUSLA）。MK-Shape和cnNMF等划分式的MTS聚类算法执行一次

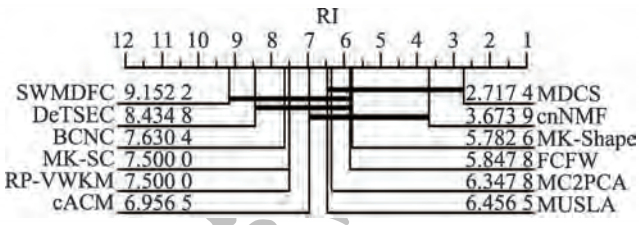


图7 RI值修正后的临界差分图

Fig.7 Critical difference diagram about RI value correction

的运算复杂度为$O(Inml)$，其中I为迭代次数，但由于划分算法都不具备稳定性，故需要执行多次以寻找最优结果，当数据量较为庞大时，其产生的额外代价可能会难以接受。MDCS在整个聚类过程中不存在随机因素，因此不需要多次运行，聚类结果也更加稳定。MUSLA与MDCS都是U-Shapelet算法在多变量上的拓展，MUSLA的时间复杂度依赖输入不同长度子序列的数量，故其运行时间会随着输入的增多而线性上升。MDCS在运行过程中产生的Shapelet空间是自适应得到的，故其U-Shapelet集合是确定的，

因而不存在这类问题。

3.3 消融实验与分析

为了更好说明各个阶段对聚类结果产生的作用，本节对MDCS三个主要的创新点进行了消融实验。其中改动算法的内容分别为MDCS-CRI（不使用自适应方式而采用CRI）、MDCS-NW（不适用加权而直接进行线性叠加）和MDCS-KM（不利用密度聚类而采用*K*-means），同时各个改动算法的参数选择与MDCS保持一致。实验过程方面，选择了表2中前九个数据集进行实验，将这些数据集每三个为一组分为三大组，命名为Group1~3，表5给出了最终外部指标的聚类结果，其中前三大格详细给出了三个数据集的聚类结果，后三大格给出了每组的平均结果。

表5 消融实验结果

Table 5 Ablation experiment results

数据集或组	算法	NMI	RI
ArticularyWordRecognition	MDCS	0.917 0	0.983 4
	MDCS-CRI	0.907 9	0.981 5
	MDCS-NW	0.500 7	0.916 3
	MDCS-KM	0.522 3	0.936 6
BasicMotions	MDCS	0.215 0	0.574 7
	MDCS-CRI	0.235 0	0.532 2
	MDCS-NW	0.213 7	0.572 4
	MDCS-KM	0.410 9	0.580 5
AtrialFibrillation	MDCS	1.000 0	1.000 0
	MDCS-CRI	1.000 0	1.000 0
	MDCS-NW	0.563 1	0.803 5
	MDCS-KM	0.637 9	0.894 0
Group1	MDCS	0.710 6	0.852 7
	MDCS-CRI	0.714 3	0.837 9
	MDCS-NW	0.425 8	0.764 1
	MDCS-KM	0.523 7	0.803 7
Group2	MDCS	0.674 3	0.865 8
	MDCS-CRI	0.674 3	0.865 8
	MDCS-NW	0.375 3	0.717 3
	MDCS-KM	0.376 2	0.718 5
Group3	MDCS	0.426 7	0.744 7
	MDCS-CRI	0.426 0	0.742 1
	MDCS-NW	0.222 1	0.671 1
	MDCS-KM	0.225 7	0.690 6

从表5中可以看到，各个改动的算法总体性能不及MDCS，其中MDCS-CRI下降程度最低，其次是MDCS-NW和MDCS-KM。在MDCS-CRI方面，原始利用CRI选择最佳Shapelet空间与MDCS的自适应搜索的差别不大，符合算法改进的基本预期，也侧面说明了自适应方式在找到理想的Shapelet空间的同时减少聚类次数是可行且有效的。MDCS-NW由于并未对变量进行加权，因而整体性能下降较多，这也说明了降低平凡变量比重，提高显著变量比重的方式有利于提高聚类性能。而MDCS-KM使用了传统的划分思路解决数值分布问题，当面向的分布较为复杂时，其性能下降最为明显。总体而言，消融实验验证了本文算法的可行性与有效性。

3.4 参数实验与分析

本节对MDCS λ_1、λ_2、λ_3三个超参数的选择进行具体的实验及分析。

从上文可知，λ_1、λ_2和λ_3之间的不同比例影响了$\{SsD_i\}_{i=1}^{m}$的数值分布，其中λ_1更关注S的质量，λ_2更关注变量之间的互补程度，λ_3更偏向变量与其他变量的一致性能力。为了测试各比重对结果可能产生的不同影响，选择在AtrialFibrillation、HandMovementDirection、Lp1和Natops数据集上进行实验，通过固定一个影响值，改变其余两个权值之间的大小进而观察聚类结果RI的不同。图8给出了实验结果，其中(a)~(d)固定λ_1为0.3，调整λ_2和λ_3从0.20~0.50之间以间隔0.05的比率进行变化，(e)~(h)和(i)~(l)则分别固定λ_2与λ_3并进行相同的变化。从图8中可以看到，AtrialFibrillation数据集中每个变量具有较长的长度，因而衡量S质量的λ_1对结果的提高具有较大的影响，λ_1越大，RI值越大；Natops数据集具有较多的变量个数，在λ_2或λ_3更大的情况下拥有更好的聚类结果，特别是λ_3的一致性权重的比例；HandMovementDirection和Lp1数据集则是在数据量、变量个数和长度都较为均衡的代表，前者整体的数据量、变量个数与长度都较大，因而可以看到三个超参数对其结果影响较大，不同组合对RI的波动性影响较为明显；后者的整体数量都偏少，因而其RI值对参数的敏感度较小；但是从图中可以观察到两者均能够在λ_1、λ_2和λ_3三者较为相近的取值下得到优异的聚类结果。

最终对参数设置有如下两个结论：其一，当数据量、变量个数和长度中某一项特别突出的情况下，推荐设置提高突出项的比例，譬如长度较大则提高λ_1的值；其二，当三者都较为均衡时，推荐均衡调整λ_1、λ_2和λ_3的取值范围即设置在0.30~0.35左右。

3.5 可解释性分析

本节简要分析MDCS算法的可解释性能力，可解释性是基于U-Shapelet算法的一种优势，它可以在没有先验知识的情况下对数据集的普遍模式进行关系来源解释，符合人类对于认知结果的定义。

案例1 ERing数据集是一个手指姿势识别数据

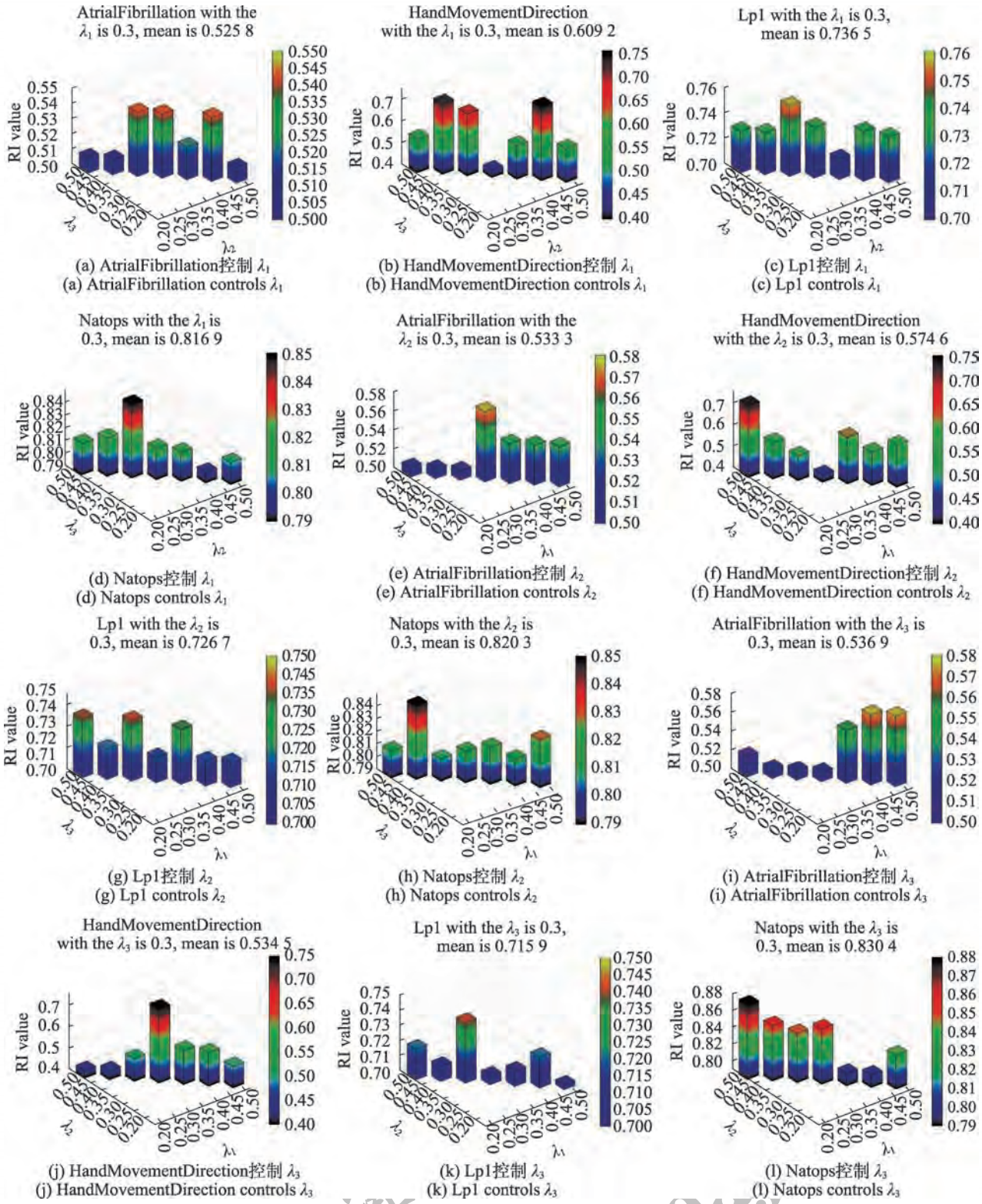


图8 参数设置实验结果

Fig.8 Experimental results of parameter setting

集，由4个变量、长度均为65的序列电极转化而来，每个变量分别代表了手势与手指距离之间的变化。如图9所示，各个变量经过Shapelet转化后产生的 S 可以代表该变量对最终结果的影响。从另一个角度说，最终的6类手势可以由这些 S 按比例所代表与解释，也就是之所以这些数据能够被归类的原因。

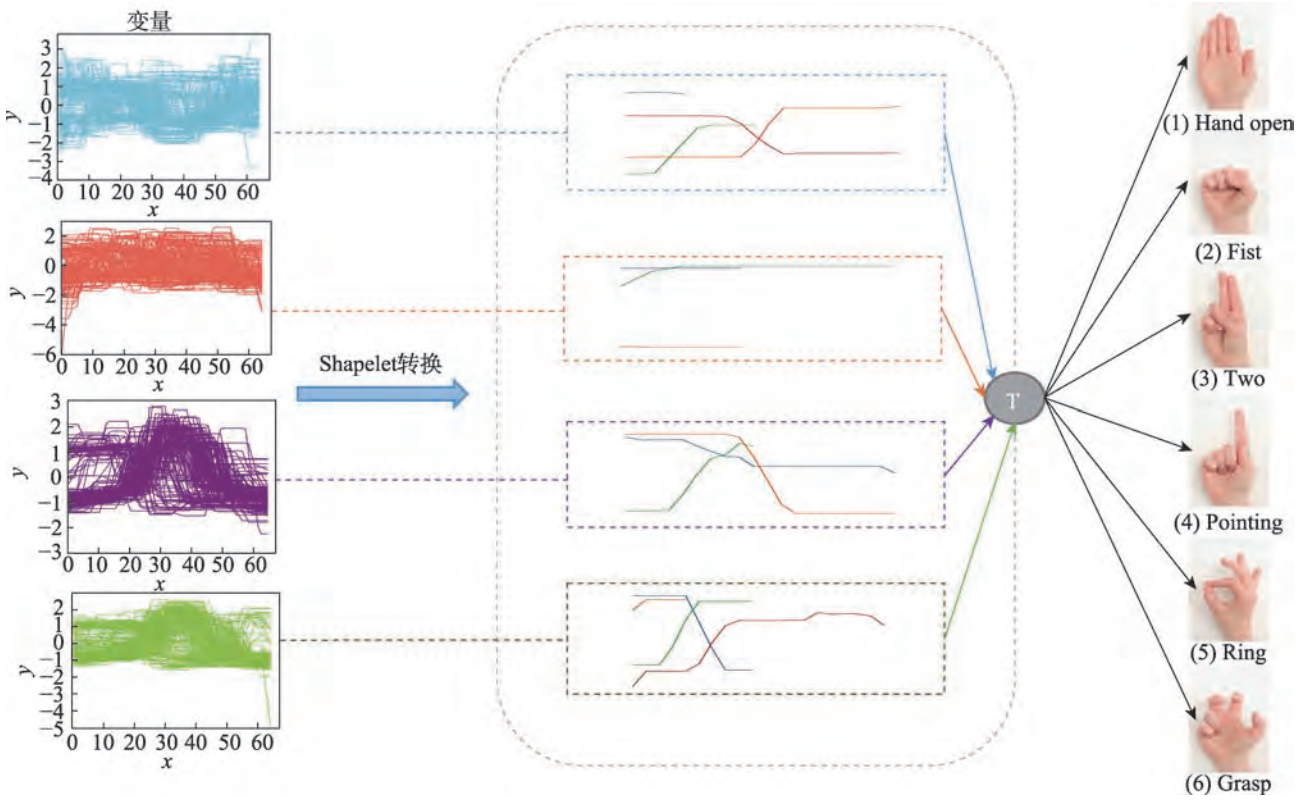


图9 可解释性实验

Fig.9 Experiment about interpretability

4 结束语

本文提出了面向Shapelet空间的多变量时间序列自适应权重密度聚类算法，简称MDCS。算法首先将MTS数据进行自适应的U-Shapelet搜索与空间转换，找到了富含更多类别信息的子序列组；接着利用组合权重的方式提高相似度矩阵的质量，包括获取各个变量之间互补性与一致性；最后利用面向复杂分布的共享最近邻密度峰值聚类算法对MTS数据进行最终分配。实验结果证明，本文算法能够有效处理高维MTS数据集的聚类问题，在提高精度的同时具备了可解释性能力。但是，本文算法的诸多参数需要人为设定，并且由于需要对每个变量进行U-Shapelet搜索，时间消耗较大，探索提高Shapelet空间搜索速度与自适应参数学习等方式将是下一步研究的重点方向。

参考文献：

[1] BAELDE M, BIERNACKI C, GREFF R. Real-time monophonic and polyphonic audio classification from power spectra[J]. Pattern Recognition, 2019, 92: 82-92.

[2] XU R, WUNSCH D. Survey of clustering algorithms[J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645-678.

[3] ASAD M, JIANG H, YANG J, et al. Multi-stream 3D latent feature clustering for abnormality detection in videos[J]. Applied Intelligence, 2022, 52(1): 1126-1143.

[4] SHARIF A, LI J P, SALEEM M A, et al. A dynamic clustering technique based on deep reinforcement learning for Internet of vehicles[J]. Journal of Intelligent Manufacturing, 2021, 32(3): 757-768.

[5] AGHABOZORGI S, SHIRKHORSHIDI A S, WAH T Y. Time-series clustering—a decade review[J]. Information Systems, 2015, 53: 16-38.

[6] LI H. Multivariate time series clustering based on common principal component analysis[J]. Neurocomputing, 2019, 349: 239-247.

[7] HE H, TAN Y. Unsupervised classification of multivariate time series using VPCA and fuzzy clustering with spatial weighted matrix distance[J]. IEEE Transactions on Cybernetics, 2018, 50(3): 1096-1105.

[8] LI H, LIU Z. Multivariate time series clustering based on complex network[J]. Pattern Recognition, 2021, 115: 107919.

[9] LI H, DU T. Multivariate time-series clustering based on component relationship networks[J]. Expert Systems with Applications, 2021, 173: 114649.

[10] 李海林, 王成, 邓晓懿. 基于分量属性近邻传播的多元时间序列数据聚类方法[J]. 控制与决策, 2018, 33(4): 649-656.
LI H L, WANG C, DENG X Y. Multivariate time series clustering based on affinity propagation of component attributes[J]. Control and Decision, 2018, 33(4): 649-656.

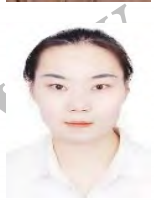
[11] OZER M, SAPIENZA A, ABELIUK A, et al. Discovering

patterns of online popularity from time series[J]. Expert Systems with Applications, 2020, 151: 113337.
[12] LI H, WEI M. Fuzzy clustering based on feature weights for multivariate time series[J]. Knowledge-Based Systems, 2020, 197: 105907.
[13] HE G, JIANG W, PENG R, et al. Soft subspace based ensemble clustering for multivariate time series data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(10): 7761-7774.
[14] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]//Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, Dec 27, 1965-Jan 7, 1966. California: University of California Press, 1967: 281-297.
[15] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.
[16] 吕伟杰, 方一帆, 程泽. 基于模糊C均值聚类和样本加权卷积神经网络的日前光伏出力预测研究[J]. 电网技术, 2022, 46(1): 8-14.
LV W J, FANG Y F, CHENG Z. Prediction of day-ahead photovoltaic output based on FCM-WS-CNN[J]. Power System Technology, 2022, 46(1): 8-14.
[17] TONG W, WANG Y, ZHONG J, et al. A new weight based density peaks clustering algorithm for numerical and categorical data[C]//Proceedings of the 13th International Conference on Computational Intelligence and Security, Hong Kong, China, Dec 15-18, 2017: 169-172.
[18] FRANCESCHI J Y, DIEULEVEUT A, JAGGI M. Unsupervised scalable representation learning for multivariate time series[C]//Proceedings of the 2019 Annual Conference on Neural Information Processing Systems, Vancouver, Dec 8-14, 2019. Cambridge: MIT Press, 2019: 4652-4663.
[19] IENCO D, INTERDONATO R. Deep multivariate time series embedding clustering via attentive-gated autoencoder [C]//Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore, May 11-14, 2020. Cham: Springer, 2020: 318-329.
[20] ZAKARIA J, MUEEN A, KEOGH E. Clustering time series using unsupervised-shapelets[C]//Proceedings of the 12th International Conference on Data Mining, Brussels, Dec 10-13, 2012: 785-794.
[21] LIU R, WANG H, YU X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks[J]. Information Sciences, 2018, 450: 200-226.
[22] 余思琴, 闫秋艳, 闫欣鸣. 基于最佳U-shapelets的时间序列聚类算法[J]. 计算机应用, 2017, 37(8): 2349-2356.
YU S Q, YAN Q Y, YAN X M. Clustering algorithm of time series with optimal U-shapelets[J]. Journal of Computer Applications, 2017, 37(8): 2349-2356.
[23] YE L, KEOGH E. Time series shapelets: a new primitive for data mining[C]//Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, Paris, Jun 28-Jul 1, 2009. New York: ACM, 2009: 947-956.
[24] ULANOVA L, BEGUM N, KEOGH E. Scalable clustering of time series with u-shapelets[C]//Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, Apr 30-May 2, 2015. Philadelphia: SIAM, 2015: 900-908.
[25] ZHANG Q, WU J, ZHANG P, et al. Salient subsequence learning for time series clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(9): 2193-2207.
[26] ZHANG N, SUN S. Multiview unsupervised shapelet learning for multivariate time series clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 4981-4996.
[27] 陈磊, 吴润秀, 李沛武, 等. 加权K近邻和多簇合并的密度峰值聚类算法[J]. 计算机科学与探索, 2022, 16(9): 2163-2176.
CHEN L, WU R X, LI P W, et al. Weighted K-nearest neighbors and multi-cluster merge density peaks clustering algorithm[J]. Journal of Frontiers of Computer Science and Technology, 2022, 16(9): 2163-2176.
[28] DU M, WANG R, JI R, et al. ROBP a robust border-peeling clustering using Cauchy kernel[J]. Information Sciences, 2021, 571: 375-400.
[29] 曹俊茸, 张德生, 肖燕婷. 结合密度比和系统演化的密度峰值聚类算法[J]. 计算机工程与应用, 2022, 58(21): 75-82.
CAO J R, ZHANG D S, XIAO Y T. Density peak clustering algorithm combining density-ratio and system evolution[J]. Computer Engineering and Applications, 2022, 58(21): 75-82.

盛锦超(1997—),男,江苏苏州人,硕士研究生,主要研究方向为聚类分析和时间序列分析。
SHENG Jinchao, born in 1997, M.S. candidate. His research interests include clustering analysis and time series analysis.

杜明晶(1989—),男,江苏徐州人,博士,副教授,硕士生导师,CCF会员,主要研究方向为聚类分析。
DU Mingjing, born in 1989, Ph.D., associate professor, M.S. supervisor, CCF member. His research interest is clustering analysis.

孙嘉睿(1997—),男,江苏南通人,硕士研究生,主要研究方向为聚类分析。
SUN Jiarui, born in 1997, M.S. candidate. His research interest is clustering analysis.



李宇蕊(1998—),女,河北石家庄人,硕士研究生,主要研究方向为聚类分析。
LI Yurui, born in 1998, M.S. candidate. Her research interest is clustering analysis.