面向时间序列有序分类的 Shapelet 抽取算法

杨 骏[1,2,3]，敬思远[2,4*]，钟 勇[1,3]

(1. 中国科学院成都计算机应用研究所 成都 610041；2. 乐山师范学院电子信息与人工智能学院 四川 乐山 614000；
3. 中国科学院大学计算机科学与技术学院 北京 石景山区 100049；4. 厅市共建智能终端四川省重点实验室 四川 宜宾 644000)

【摘要】当前面向时间序列有序分类的 Shapelet 抽取算法，首先计算 Shapelet 与时间序列之间的欧式距离及其类别标签之间的距离，然后根据两种距离的皮尔逊相关系数或斯皮尔曼相关系数来对 Shapelet 进行评价，效率较低。针对该问题，提出一种基于 SAX 表示时间序列的 Shapelet 评价指标 CD-Cover，该指标同时考虑 Shapelet 对时间序列数据集的覆盖集中度和覆盖优势度。其次，提出一种基于随机采样的 Shapelet 抽取算法，该算法采用布隆过滤器对候选 Shapelet 进行预剪枝，采用移除自相似策略对抽取结果进行后剪枝。在 11 个时间序列公开数据集上的实验结果表明，相比现有方法，该算法抽取的 Shapelet 具有更好的有序分类能力，且算法的计算效率也更高。

关 键 词 特征评价； 有序分类； Shapelet； 时间序列
中图分类号 TP273 **文献标志码** A doi:10.12178/1001-0548.2022278

Shapelet Extraction Algorithm for Time Series Ordinal Classification

YANG Jun[1,2,3], JING Siyuan[2,4*], and ZHONG Yong[1,3]

(1. Chengdu Institute of Computer Application, Chinese Academy of Sciences Chengdu 610041;
2. School of Electronic Information and Artificial Intelligence, Leshan Normal University Leshan Sichuan 614000;
3. School of Computer Science and Technology, University of Chinese Academy of Sciences Shijingshan Beijing 100049;
4. Intelligent Terminal Key Laboratory of Sichuan Province Yibin Sichuan 644000)

Abstract The current Shapelet extraction algorithm for time series ordinal classification, which suffers from low efficiency, needs to figure out the Pearson's correlation coefficient or the Spearman's correlation coefficient between the Euclidean distances and the label distances from time series to Shapelets to evaluate the Shapelets. To handle this problem, this paper first proposes a Shapelet measure CD-Cover (concentration and dominance of coverage) based on the SAX (symbolic aggregate approximation)-represented time series. The measure takes into account both the concentration and the dominance of coverage of a Shapelet on the time series dataset. Secondly, this paper also proposes a Shapelet extraction algorithm based on random sampling. The algorithm uses the Bloom filter to pre-prune Shapelet candidates and employs a strategy of removing self-similar Shapelets to post-prune the extracting results. Experimental results on 11 time series public datasets show that the Shapelet extracted by the proposed algorithm has better ability for ordinal classification than the existing methods, and meanwhile, the computing efficiency of the proposed algorithm is superior to that of the existing methods.

Key words feature evaluation; ordinal classification; Shapelet; time series

时间序列有序分类（time series ordinal classification, TSOC）是时间序列数据挖掘领域的一项重要任务。该任务旨在训练一个分类器，实现对类别标签有序的时间序列数据的自动分类。与传统时间序列分类[1]任务不同，TSOC 采用错分代价来衡量算法有效性。如在医疗辅助诊断系统中，将危重型病症错分成轻型病症的代价远高于将其错分成重型病症的代价。除了医疗辅助诊断外，本任务在金融投资、气象预测等领域都有重要应用。但目前关于该任务的研究非常少，尚处于起步阶段[2]。

收稿日期：2022－08－13；修回日期：2022－10－28
基金项目：四川省科技计划重点研发项目（2021YFS0019）；四川省科技成果转移转化示范项目（2020ZHZY0002）；厅市共建智能终端四川省重点实验室开放基金（SCITLAB-1002）
作者简介：杨骏 (1976－)，男，博士生，副教授，主要从事数据挖掘方面的研究.
*通信作者：敬思远，E-mail：syjing628@126.com

基于 Shapelet 的时间序列分类近年来受到学界广泛关注[3-5]。Shapelet 是指时间序列中具有良好分类能力的子序列，最早由文献 [6] 提出，它采用 Brute-Force 算法搜索子序列，并用信息增益（information gain, IG）对其分类能力进行评价，最后利用计算得到的 Shapelet 构造决策树。由 Shapelet 构造的决策树具有非常好的可解释性。随后，文献 [7] 提出了 ST（shapelet transformation）方法。该方法获得 Top-K 个最优 Shapelet，然后将原始时间序列转换到新的特征空间并采用传统方法训练分类器，使算法的分类能力大幅提升。但是，上述两类算法都采用暴力方法搜索 Shapelet，计算效率低。为解决该问题，文献 [8] 提出了基于三角不等式的剪枝策略，以及提前计算距离的方法；文献 [9] 用符号聚合近似（symbolic aggregate approximation, SAX）表示时间序列，采用随机投射技术，计算最优 Shapelet 集合；文献 [10] 提出了基于随机采样的 Shapelet 抽取算法；文献 [11] 提出了基于随机采样的随机 Shapelet 森林算法 gRSF；文献 [12] 在上述工作基础上进一步改进，提出了压缩随机 Shapelet 森林算法 CRSF。CRSF 采用 SAX 表示时间序列，通过随机采样构建一个高质量的 Shapelet 池，然后从池中随机选择 Shapelet 生成决策树节点，提升了算法性能。近年来，基于 Shapelet 构建深度学习模型也受到学界广泛关注[13-14]，但考虑到深度神经网络可解释性不足，本文不再详细介绍。

文献 [15] 提出了基于 Shapelet 的时间序列有序分类算法。该工作主要提出了两项面向 TSOC 的 Shapelet 评价指标，Spearman 相关系数和 Pearson 相关系数。与 IG 不同，这两项评价指标通过计算 Shapelet 与时间序列的欧式距离和标签距离之间的相关系数来评价 Shapelet。实验结果表明，评价指标与 IG 相比有效地降低了算法的错分代价。但是，该方法中提出的 Shapelet 评价指标都需要计算 Shapelet 到时间序列的距离，计算效率较低。

本文提出一种基于覆盖集中度和覆盖优势度的 Shapelet 评价指标 CD-Cover（concentration and dominance of coverage），以及一种面向时间序列有序分类的 Shapelet 抽取算法。该算法采用 SAX 表示时间序列，通过随机采样 Shapelet，使用 CD-Cover 指标评价 Shapelet，抽取最终的 Shapelet 结果集。然后，通过 ST 方法将时间序列数据集转换到新的特征空间并训练有序分类器。最后，在 UCR 和 UEA 时间序列分类资源库[16] 挑选适合 TSOC 任务的 11 个数据集上，采用 CCR（correct classification rate）和 Weighted-κ 两个指标对所提算法进行了实验验证。

1 相关理论及问题描述

通常，时间序列是按照固定时间间隔采集得到的数值型数据序列，可以表示为$X=\langle x_1,x_2,\cdots,x_m\rangle$，$x_i\in\mathbb{R}$，$m$ 为时间序列的长度。进一步，将二元组 $T=\langle X,c\rangle$ 称为时间序列样本，c 是时间序列样本的类别标签（简称标签）。$D=\{T_1,T_2,\cdots,T_n\}$为一个时间序列数据集，其中，n 称为时间序列数据集的大小。$Y=\{c_1,c_2,\cdots,c_Q\}$表示时间序列数据集 D 中所有样本标签的集合，Q 是标签数。在 TSOC 问题中，要求$Q\geqslant 3$，且标签之间存在全序关系$c_1\prec c_2\prec\cdots\prec c_Q$。

时间序列数据通常都是高维实数，不仅需要大量存储空间，而且计算代价也很高。文献 [17] 提出一种时间序列的 SAX 表示方法，将数值型时间序列转换为符号型时间序列，可以达到数据降维、降低噪声、节省存储和简化计算等目的。SAX 表示方法的转换过程如下。

给定时间序列$X=\langle x_1,x_2,\cdots,x_m\rangle$、动态窗口长度 ω 和字母表 Σ，将$X$平均分成$t=\lceil m/\omega\rceil$段，得到$\overline{X}=\langle\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_t\rangle$，其中$\bar{x}_i$为：

$$\bar{x}_i=\frac{1}{\omega}\sum_{j=1}^{\omega}x_{(i-1)\cdot\omega+j}\qquad i=1,2,\cdots,t \tag{1}$$

进一步，基于映射函数φ，将$\bar{x}_i$映射到字母表空间，记为$\hat{x}_i=\varphi(\bar{x}_i)$，$\hat{x}_i\in\Sigma$。采用 SAX 表示后的时间序列记为$\widehat{X}=\langle\hat{x}_1,\hat{x}_2,\cdots,\hat{x}_t\rangle$，时间序列样本为$\widehat{T}=\langle\widehat{X},c\rangle$，时间序列数据集为$\widehat{D}=\langle\widehat{T}_1,\widehat{T}_2,\cdots,\widehat{T}_n\rangle$。

传统的 Shapelet 是指时间序列的任意子序列，本文的候选 Shapelet 是基于 SAX 表示的时间序列，下面给出其定义。

定 义 1（候选 Shapelet） 给定 SAX 表示的时间序列$\widehat{X}=\langle\hat{x}_1,\hat{x}_2,\cdots,\hat{x}_t\rangle$，称该时间序列的任意子序列$s_b^l$为一个候选 Shapelet，其中 b 和 l 分别表示候选 Shapelet 在时间序列中的开始位置和长度，且$1\leqslant b\leqslant t+1-l$，$0<l\leqslant t$。

为表述方便，将s_b^l统一表示为 s。如果没有特别说明，本文中时间序列、时间序列样本、时间序列数据集和 Shapelet，均采用 SAX 表示。

基于 Shapelet 的时间序列分类，核心是找出最具代表性的候选 Shapelet 集合，称为 Shapelet 抽取。

问题描述：给定 SAX 表示的时间序列数据集 $\widehat{D}$，Shapelet 抽取就是从$\widehat{D}$中找出最优候选 Shapelet 集合 S，使评价函数$\Psi(\cdot)$取得最大值，为：

$$J(\widehat{D}) = \arg\max_{S} \Psi(\widehat{D}, S) \tag{2}$$

文献 [6] 最早采用 IG 对 Shapelet 的分类能力进行评价。IG 也是当前研究和实践中最常用的 Shapelet 评价指标。文献 [7] 提出了 Kruskal-Wallis、F-statistic 和 Mood's median 这 3 种指标，并通过实验证明了其有效性。但上述指标没有考虑错分代价，在 TSOC 任务中表现不佳。文献 [15] 提出了 Spearman 相关系数和 Pearson 相关系数两项指标，充分考虑了类别有序这一特征。但是，两项指标都需要计算 Shapelet 到时间序列之间的距离，计算成本较高。

2　面向 TSOC 的 Shapelet 抽取算法

2.1　分类算法框架

本文设计的面向时间序列有序分类算法框架如图 1 所示。首先，采用 SAX 方法将数值型时间序列数据集转化为符号型时间序列数据集。然后，采用随机采样+布隆过滤+Shapelet 评价的策略，实现对 Shapelet 的抽取。其中，随机采样+布隆过滤旨在提升算法效率以及进行预剪枝。该策略在文献 [18] 中已被验证有效。本文提出一种新的面向 SAX 表示时间序列有序分类的 Shapelet 评价指标 CD-Cover。图 1 中◆图标表示是否满足约束条件，如果不满足约束条件，继续采样和评价，否则，结束采样评价。接下来，算法对抽取出的 Shapelet 进行自相似移除，降低特征冗余度。最后，选取指定大小的 Shapelet 集合作为数据集的新特征，使用新特征对原始时间序列进行特征空间转换，并训练有序分类器。

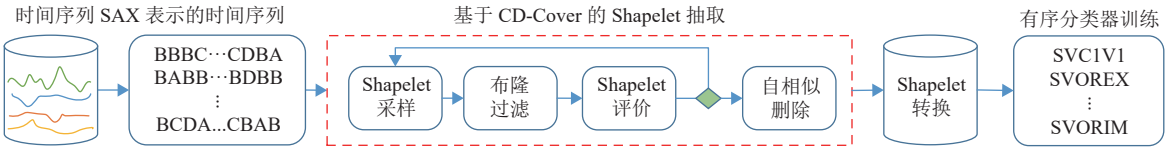


图 1　面向时间序列有序分类算法框架

2.2　Shapelet 评价指标 CD-Cover

为提升 Shapelet 评估效率，本节介绍一种适用于时间序列有序分类的 Shapelet 评价指标 CD-Cover。首先，给出覆盖、覆盖集中度和覆盖优势度的定义。

定义　2（覆盖）　给定一个 SAX 表示的时间序列数据集$\widehat{D} = \{\widehat{T}_1, \widehat{T}_2, \cdots, \widehat{T}_n\}$和一个 Shapelet s，$\widehat{D}$的标签数为Q，称$\Phi(s) = \langle \lambda_1, \lambda_2, \cdots, \lambda_Q \rangle$为$s$在$\widehat{D}$上的覆盖。其中，$\lambda_i$为包含$s$的$c_i$类时间序列样本数。

定义　3（覆盖集中度）　给定 Shapelet s在时间序列数据集$\widehat{D}$上的覆盖$\Phi(s) = \langle \lambda_1, \lambda_2, \cdots, \lambda_Q \rangle$，$Q$为$\widehat{D}$的标签数。则$s$在$\widehat{D}$上的覆盖集中度$\mathrm{con}(s)$定义为：

$$\mathrm{con}(s) = 1 - \frac{\mathrm{Var}(\Phi(s))}{\mathrm{Var}_{\max}}$$

$$\mathrm{Var}(\Phi(s)) = \sum_{i=1}^{Q} p_i \left(i - \sum_{i=j}^{Q} p_j j \right)^2$$

$$p_i = \lambda_i \Big/ \sum_{j=1}^{Q} \lambda_j \tag{3}$$

式中，$\mathrm{Var}(\Phi(s))$为$\Phi(s)$的方差；$\mathrm{Var}_{\max}$为方差上界。根据方差性质可知，当覆盖取值平均分布在值域两端时取得最大值，因此覆盖集中度的方差上界$\mathrm{Var}_{\max} = (1-Q)^2 \big/ 4$。从上述定义容易看出，$\mathrm{con}(s)$取值范围为 [0,1]，值越大，表明覆盖的方差越小，覆盖越集中。特殊情况下，如果s仅覆盖一类时间序列，则方差为 0，$\mathrm{con}(s)$值为 1；如果s仅覆盖类别标签最小和类别标签最大的时间序列，且覆盖比率相同，则方差达到上界，$\mathrm{con}(s)$值为 0。

定义　4（覆盖优势度）　给定 Shapelet$s$在时间序列数据集$\widehat{D}$上的覆盖$\Phi(s) = \langle \lambda_1, \lambda_2, \cdots, \lambda_Q \rangle$，且$\widehat{D}$中$Q$类样本数分别为$n_1, n_2, \cdots, n_Q$，称$\Pi(s) = \langle \kappa_1, \kappa_2, \cdots, \kappa_Q \rangle$为 Shapelet$s$在数据集$\widehat{D}$上的覆盖率，其中$\kappa_i = \lambda_i / n_i$为类别$c_i$的覆盖率。令$\Pi'(s) = \langle \kappa'_1, \kappa'_2, \cdots, \kappa'_Q \rangle$为$\Pi(s)$降序排列的结果，则$s$在$\widehat{D}$上的覆盖优势度为$\mathrm{dom}(s) = \kappa'_1 - \kappa'_2$。

换言之，覆盖优势度为类别最高覆盖率与次高覆盖率之差。覆盖优势度$\mathrm{dom}(s)$的取值范围是 [0,1]，值越大，表明覆盖的优势越明显。如果 Shapelet

s 仅覆盖一个类别的时间序列，则覆盖优势度为该类类别的覆盖率。

定 义 5（CD-Cover） 给定 Shapelet s和时间序列数据集$\widehat{D}$，s在$\widehat{D}$上的覆盖集中度和覆盖优势度分别为$\mathrm{con}(s)$和$\mathrm{dom}(s)$，则s的 CD-Cover 评价值为：

$$\sigma(s)=\alpha\mathrm{con}(s)+(1-\alpha)\mathrm{dom}(s)\qquad 0<\alpha<1 \tag{4}$$

式中，α是权重因子，用来调节覆盖集中度和覆盖优势度的权重，默认两者权重相同，即α取 0.5。如果覆盖集中度越高，则$\mathrm{con}(s)$值越大；如果覆盖优势度越大，则$\mathrm{dom}(s)$越大，并且，$\sigma(s)$取值范围为[0,1]。因此，如果 CD-Cover 值越大，表明其分类能力越强。

下面，通过 4 个算例来展示 CD-Cover 指标的计算过程及其有效性。

算例 1 时间序列数据集$\widehat{D}$标签$Y=\langle 1,2,3,4\rangle$，即$Q=4$，且每类时间序列样本数分别为$\langle 5,5,2,2\rangle$。如果给定一个 Shapelet s_1，且s_1在数据集$\widehat{D}$上的覆盖$\Phi(s_1)=\langle 4,1,0,0\rangle$。则计算可得，$\mathrm{Var}(\Phi(s_1))=0.17$，而$\mathrm{Var}_{\max}=(1-4)^2/4=2.25$，所以，$s_1$的覆盖集中度$\mathrm{con}(s)=1-0.17/2.25=0.924$。根据定义 4，覆盖率为$\Pi(s_1)=\langle 4/5,1/5,0/2,0/2\rangle$，因此，类别 1 覆盖率最高为 0.8，类别 2 覆盖率次高为 0.2。所以，覆盖优势度$\mathrm{dom}(s_1)=0.8-0.2=0.6$。根据式（4），计算$s_1$的 CD-Cover 为$\sigma(s_1)=0.5\times 0.924+(1-0.5)\times 0.6=0.762$。

算例 2 算例 1 相同的时间序列数据集$\widehat{D}$中，如果给定 Shapelet s_2，在$\widehat{D}$上的覆盖$\Phi(s_2)=\langle 2,0,0,1\rangle$，则 $\mathrm{Var}(\Phi(s_2))=1.25$，$\mathrm{con}(s_2)=1-1.25/2.25=0.444$；$s_2$在$\widehat{D}$上的覆盖率$\Pi(s_2)=\langle 2/5,0/5,0/2,1/2\rangle$，覆盖优势度为$\mathrm{dom}(s_2)=0.5-0.4=0.1$。因此，$s_2$的 CD-Cover 为$\sigma(s_2)=0.5\times 0.444+(1-0.5)\times 0.1=0.272$。

算例 3 算例 1 相同的时间序列数据集$\widehat{D}$中，如果给定 Shapelet s_3，在$\widehat{D}$上的覆盖$\Phi(s_3)=\langle 1,1,0,0\rangle$，则 $\mathrm{Var}(\Phi(s_3))=0.125$，$\mathrm{con}(s_3)=1-0.125/2.25=0.944$；$s_3$在$\widehat{D}$上的覆盖率$\Pi(s_3)=\langle 1/5,1/5,0/2,0/2\rangle$，覆盖优势度为$\mathrm{dom}(s_3)=0.2-0.2=0$。因此，$s_3$的 CD-Cover 值为$\sigma(s_3)=0.5\times 0.944+(1-0.5)\times 0=0.472$。

算例 4 算例 1 相同的时间序列数据集$\widehat{D}$中，如果给定 Shapelet s_4，在$\widehat{D}$上的覆盖$\Phi(s_4)=\langle 1,0,0,2\rangle$，则 $\mathrm{Var}(\Phi(s_4))=1.25$，$\mathrm{con}(s_4)=1-1.25/2.25=0.444$；$s_4$在$\widehat{D}$上的覆盖率$\Pi(s_4)=\langle 1/5,0/5,0/2,2/2\rangle$，覆盖优势度为$\mathrm{dom}(s_4)=1-0.2=0.8$。因此，$s_4$的 CD-Cover 值为$\sigma(s_4)=0.5\times 0.444+(1-0.5)\times 0.8=0.622$。

通过以上算例可以发现，CD-Cover 的目标是找出最优的 Shapelet。这些 Shapelet 覆盖的时间序列集中在某个类别附近，且对该类时间序列有很高的覆盖比例。如果 Shapelet 覆盖且只覆盖一个类别的所有时间序列，其 CD-Cover 值为 1。

2.3 基于随机采样的 Shapelet 抽取算法

基于随机采样的面向时间序列有序分类的 Shapelet 抽取算法核心步骤算法如下。

输入 D: 原始时间序列训练集; α: 权重因子; ε: CD-Cover 阈值; ω: SAX 窗口大小, Σ: SAX 字符集大小; τ: Shapelet 评价时间限制；N: Shapelet 数目

输出 S: 抽取得到的最佳 Shapelet 集合

1. $\widehat{D}\leftarrow$ convert_*to*_SAX(D,ω,Σ);
2. $S\leftarrow\emptyset$;
3. BF $\leftarrow$ initialize_bloom_filter();
4. while not timeout(τ) do
5. $\quad\widehat{T}\leftarrow$ random_sampling_time_series$\left(\widehat{D}\right)$;
6. $\quad s\leftarrow$ random_sampling_shapelet$\left(\widehat{T}\right)$;
7. $\quad$if exist_in_bloom_filter(BF,s) then
8. $\quad\quad$continue;
9. $\quad$BF $\leftarrow$ update_bloom_filter(BF,s);
10. $\quad$cd_cover $\leftarrow$ CD_Cover$(s,\widehat{D},\alpha)$;
11. $\quad$if cd_cover $>\varepsilon$ then
12. $\quad\quad S\leftarrow S\cup s$;
13. $\quad$if $|S|>2*N$ then
14. $\quad\quad S\leftarrow$ remove_worst(S);
15. $\quad\quad\varepsilon\leftarrow$ update_threshold(S);
16. $S\leftarrow$ remove_similar_and_select_top(S,N);
17. return S;

算法第 1 行：根据参数ω和Σ将原始时间序列训练集 D 转换为 SAX 表示的时间序列训练集$\widehat{D}$。

算法第 2、3 行：初始化 Shapelet 结果集合S和布隆过滤器 BF。本算法采用布隆过滤器检索当前候选 Shapelet 是否已经出现并评价，如果此前已经出现则不再进行评价，提升算法效率[18]。

算法第 4 行：进入循环，当计时器超过限定时间，结束 Shapelet 抽取。

算法第 5、6 行：随机从$\widehat{D}$中选择一条时间序列，然后从该时间序列中随机抽取一个候选 Shapelet。

算法第 7、8 行：通过布隆过滤器判断 Shapelet s是否已经出现；如果已出现，则跳过 CD-Cover 值计算，重新进行 Shapelet 采样。

算法第 9、10 行：更新布隆过滤器，并计算候选 Shapelet s的 CD-Cover 值。

算法第 11、12 行：判断当前候选 Shapelet s的 CD-Cover 值是否大于给定阈值ε，只有评估值大于ε，才将其加入S，保证抽取高质量 Shapelet。

算法第 13、14、15 行：判断抽取得到的 Shapelet 集合大小是否大于预设数目N的两倍，如果满足条件，则从S中移除 CD-Cover 值最小的候选 Shapelet，同时将阈值ε更新为S中 Shapelet 的最小 CD-Cover 值。保留预设数N两倍数量的候选 Shapelet 是后续还要从中移除自相似的 Shapelet。

算法第 16 行：首先从S中移除自相似的 Shapelet。自相似 Shapelet 指的是两个候选 Shapelet取自同一条时间序列，且存在重叠部分。大量自相似的 Shapelet 会造成特征空间冗余，降低算法分类效果。移除自相似 Shapelet 的算法在文献 [12,19] 中均有介绍。最后，返回剩余候选 Shapelet 中 CD-Cover 值最高的N个 Shapelet。

2.4　时间复杂度分析

如果时间序列数据集 D 的大小为 n，所包含的时间序列长度均为 m，SAX 窗口大小为ω。则经过转换，采用 SAX 表示的时间序列数据集$\widehat{D}$中包含 n 条字符形时间序列，每条长度均为$t=\lceil m/\omega \rceil$。

在上述算法中，将原始时间序列转换为 SAX 表示时间序列的时间复杂度为$O(nm)$（第 1 行），移除自相似算法的时间复杂度依赖于候选 Shapelet 的数量，该数量远小于时间序列数据集的大小 n 与长度 m 的乘积nm。因此，第 1 行和第 16 行的时间复杂度合计为$O(nm)$。

本文采用的随机采样框架设置了 Shapelet 评价时间限制，通常应采用单位时间内的吞吐量来分析算法性能。但是，反过来，也可以通过分析抽取单个候选 Shapelet 的时间复杂度来评价算法性能。Shapelet 抽取算法的主要时间开销为 CD-Cover 值计算，需要判断候选 Shapelet 是否存在于某条时间序列中。本文采用 KMP 算法，其最坏时间复杂度为$O(t)$。在 n 条时间序列中进行字符串匹配，其最坏时间复杂度为$O(tn)$。该时间复杂度明显低于文献 [15] 提出的 Spearman 相关系数和 Pearson 相关系数的时间复杂度。换言之，在相同时间约束内，本文提出的算法抽取和评价的 Shapelet 数量要远多于文献 [15] 提出的算法。

3　实验与分析

为了验证所提 Shapelet 抽取算法的有效性，在公开数据集上进行实验，训练生成有序分类器，并采用不同指标来评价这些分类器的分类能力。

3.1　实验准备

3.1.1　实验环境和数据集

实验硬件配置为 Intel Xeon Gold 5215 CPU（2.5 GHz 主频，8 核）和 64 GB 内存。操作系统为 Windows Server 2016。实验工具采用 Python3.6.8、时间序列挖掘工具包 sktime 0.9.0 和有序分类开源框架 ORCA（ordinal regression and classification algorithm）[20]。ORCA 的运行环境为 MATLAB 2018b。

从时间序列分类资源库 UCR 和 UEA[16]，根据以下条件筛选出 11 个实验数据集：1）类别标签之间有明确的全序关系；2）类别数不小于 3；3）时间序列长度相等。数据集的基本信息如表 1 所示。

表 1　实验数据集

编号	数据集	训练样本	测试样本	标签数	序列长度
1	AbnormalHeartbeat	302	304	5	3 053
2	Beef	30	30	5	470
3	ChlorineConcentration	467	3 840	3	166
4	Colposcopy	99	101	6	180
5	DistalPhalanxOutlineAgeGroup	400	139	3	80
6	DistalPhalanxTW	400	139	6	80
7	EthanolLevel	504	500	4	1 751
8	MiddlePhalanxOutlineAgeGroup	400	154	3	80
9	MiddlePhalanxTW	399	154	6	80
10	ProximalPhalanxOutlineAgeGroup	400	205	3	80
11	ProximalPhalanxTW	400	205	6	80

3.1.2　有序分类器和评价指标

传统 ST 方法将抽取的 Shapelet 用于训练标准分类器，如支持向量机、随机森林等。但标准分类器没有考虑类别标签有序的问题。实验采用了 3 种有序分类研究中常用的分类器 SVC1V1[21]、SVOREX [22] 和 SVORIM [22] 来验证本文所提 Shapelet 抽取算法。其中，SVC1V1 是标量分类器，SVOREX 和 SVORIM 是有序分类器，这些分类器的实现均来自 ORCA 开源框架[20]。

实验采用 CCR 和 Weighted-κ 作为分类器分类能力的评价指标。前者是最常用的分类评价指标，但没有考虑类别标签有序的特征；后者是目前研究

验证的、更适用于评价有序分类能力的指标[23]。CCR 和 Weighted-κ 分别由式（5）和式（6）计算所得。

$$\mathrm{CCR}=\frac{1}{n}\sum_{i=1}^{n}\delta(c_i,\widehat{c_i}) \tag{5}$$

式中，n 是时间序列测试集大小；c_i和$\widehat{c_i}$分别表示测试集中时间序列样本T_i的实际标签和分类器预测标签；$\delta(c_i,\widehat{c_i})$是 Kronecker 函数，当且仅当$c_i$和$\widehat{c_i}$相同时，函数值为 1，否则函数值为 0。CCR 取值范围为 [0,1]，值越大，表明分类器性能越好。

$$\mathrm{Weighted}-\kappa=1-\frac{\sum_{i=1}^{Q}\sum_{j=1}^{Q}\theta_{i,j}M_{i,j}}{\sum_{i=1}^{Q}\sum_{j=1}^{Q}\theta_{i,j}e_{i,j}} \tag{6}$$

式中，Q 表示时间序列数据集的标签数；$\theta_{i,j}$表示将c_i类时间序列预测为c_j类的权重（错分代价），通常采用线性权重或二次权重，本文采用线性权重，即$\theta_{i,j}=\left|O(c_i)-O(c_j)\right|$，其中$O(c)$表示类别标签 c 的序号，所有序号是连续整数，如果类别标签本身是连续整数，则可简化为$\theta_{i,j}=\left|c_i-c_j\right|$；$M_{i,j}$表示分类器将$c_i$类时间序列预测为$c_j$类的样本数；$e_{i,j}$表示期望的预测结果，计算式为：

$$e_{i,j}=\frac{1}{n}\sum_{k=1}^{Q}M_{i,k}\sum_{k=1}^{Q}M_{k,j} \tag{7}$$

Weighted-κ 取值范围为 [−1,1]，值越大，表明分类器的分类效果越好。

3.1.3 实验参数设置

实验中，本文提出的 Shapelet 抽取算法的参数设置如表 2 所示。其中 CD-Cover 计算公式中的权重因子α设置为 0.5，换言之，设定覆盖集中度和覆盖优势度有同等的重要性；Shapelet 抽取过程中根据候选 Shapelet 的 CD-Cover 值是否大于初始阈值ε决定是否对其保留，ε设置为 0.5。此外，SAX 的参数ω和Σ分别设置为 1 和 10。这样设置有两个理由：1）文献 [12] 通过实验证明该设置在大部分数据集上能取得较好的分类结果；2）本文主要是利用 SAX 的符号表示能力，即将数值型时间序列转换为符号型时间序列，并不特别关注其对数据的维度约减能力。此外，训练有序分类器的时候，3 种支持向量分类器的惩罚参数和核函数参数都设置为相同的搜索范围$\{10^{-3},10^{-2},\cdots,10^{2},10^{3}\}$，并采用 5 折交叉验证，搜索最优参数组合。时间约束τ设定为 300 s。

表 2 实验参数设置

序号	参数	取值
1	α	0.5
2	ε	0.5
3	ω	1
4	Σ	10
5	τ/s	300

3.2 实验结果

本节首先验证所提 CD-Cover 指标的有效性。实验将根据 IG、Spearman 相关系数、Pearson 相关系数和 CD-Cover 4 种评价指标抽取的 Shapelet，分别用来训练 3.1.2 节中的 3 种分类器，然后比较 4 种指标在实验数据集上的有序分类效果。由于算法具有随机性，为保证结果的公平性，CCR 和 Weighted-κ 结果都取 50 次实验的平均值。

1）CCR

采用 3 种分类器 SVC1VC、SVOREX 和 SVORIM 在 11 个数据集上进行实验，得到的 CCR 结果如表 3 所示，表中 IG、ρ、γ和σ分别代表 IG、Spearman 相关系数、Pearson 相关系数和 CD-Cover 指标。对每个数据集而言，同一分类器不同指标的 CCR 最高值用粗体标注。表格最后的“胜出”，表示采用当前分类器的 4 种 Shapelet 评价指标中，各指标在 11 个数据集上取得 CCR 最优分类结果的数据集个数。如采用 SVC1V1 分类器结合 IG 评价指标，分别在第 2、第 6 和第 9 个数据集上取得 3 次最优分类结果。“排名”表示采用当前分类器的 4 种 Shapelet 评价指标中，各指标在 11 个数据集上取得的 CCR 值的平均排名。例如，采用 SVORIM 分类器结合 Pearson 相关系数，在 11 个数据集上的平均排名为 2.55。CD-Cover 指标在 3 种分类器上“胜出”的数据集都是 6 个（超过数据集的半数），且在 3 种分类器上的平均排名都最高，分别为 1.77、1.91 和 1.86。

2）Weighted-κ

表 4 给出了采用 3 种分类器结合 4 种 Shapelet 评价指标在 11 个数据集上获得的 Weighted-κ 结果。由表 4 可知，在 11 个数据集上，CD-Cover 指标在 3 种分类器上分别取得了 6 次、6 次和 7 次最优的表现。同时，CD-Cover 指标在 3 种分类器上都取得了最高的平均排名，分别为 1.77、1.64 和 1.50。

表 3　不同 Shapelet 评价指标的 CCR 值

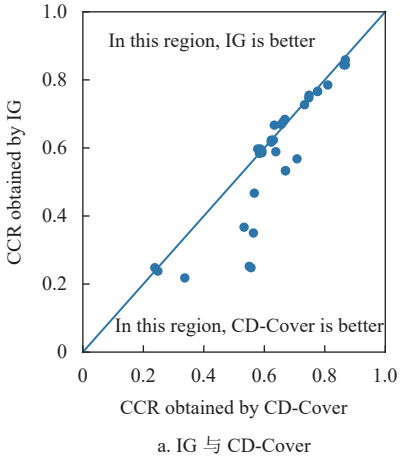
数据集	SVC1V1				SVOREX				SVORIM			
	IG	ρ	γ	σ	IG	ρ	γ	σ	IG	ρ	γ	σ
1	0.589	0.622	0.592	**0.638**	0.586	0.589	0.576	**0.592**	**0.592**	0.586	0.586	**0.592**
2	**0.667**	0.600	0.533	0.633	0.367	**0.533**	0.500	**0.533**	0.467	0.533	0.367	**0.567**
3	0.568	0.611	0.655	**0.708**	0.533	0.570	0.655	**0.670**	0.533	0.572	0.655	**0.670**
4	0.218	0.307	0.327	**0.337**	0.248	0.218	**0.297**	0.238	0.238	0.218	**0.307**	0.248
5	0.727	0.741	**0.755**	0.733	**0.755**	0.748	**0.755**	0.748	0.747	**0.755**	**0.755**	0.747
6	**0.669**	0.633	0.633	0.656	0.684	**0.691**	0.656	0.668	0.676	**0.684**	0.676	0.662
7	0.350	0.258	0.258	**0.564**	0.252	0.258	0.252	**0.550**	0.248	0.248	0.262	**0.556**
8	0.623	**0.636**	0.623	0.630	0.617	**0.630**	0.623	0.623	0.623	**0.630**	0.623	0.623
9	**0.597**	0.591	**0.597**	0.578	0.584	0.565	**0.597**	0.584	**0.597**	0.526	0.552	0.587
10	0.844	0.859	**0.863**	**0.863**	0.844	0.863	0.849	**0.868**	0.859	0.859	0.859	**0.868**
11	0.785	0.756	0.785	**0.810**	0.766	0.771	0.771	**0.776**	0.766	0.771	0.771	**0.776**
胜出	3	1	3	**6**	1	3	**3**	**6**	2	3	2	**6**
排名	2.86	2.82	2.55	**1.77**	3.09	2.41	2.50	**1.91**	2.86	2.64	2.55	**1.86**

表 4　不同 Shapelet 评价指标的 Weighted-κ 值

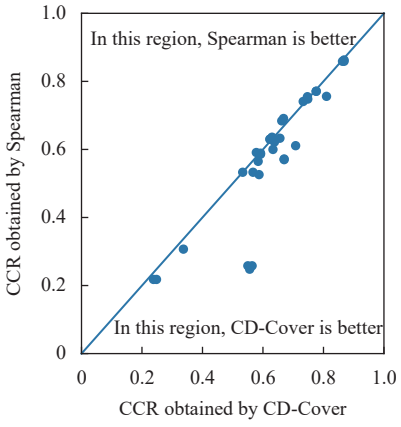
数据集	SVC1V1				SVOREX				SVORIM			
	IG	ρ	γ	σ	*IG*	ρ	γ	σ	*IG*	ρ	γ	σ
1	0.129	**0.177**	0.156	0.132	0.079	0.099	**0.115**	0.109	0.073	0.039	0.044	**0.122**
2	**0.655**	0.534	0.478	0.504	0.429	0.268	0.340	**0.562**	0.396	0.363	0.409	**0.552**
3	0.151	0.398	0.489	**0.575**	0.000	0.378	0.378	**0.528**	0.000	0.378	0.378	**0.548**
4	0.063	0.009	0.156	**0.174**	0.000	0.000	0.000	**0.026**	0.008	0.000	0.000	**0.054**
5	0.587	**0.637**	0.622	0.612	0.615	0.580	0.573	**0.641**	**0.615**	0.591	0.573	0.612
6	0.783	0.775	0.808	**0.810**	0.829	0.780	**0.837**	0.821	0.813	0.777	**0.831**	0.830
7	0.238	0.020	0.042	**0.541**	0.000	0.006	0.149	**0.548**	0.000	0.064	0.096	**0.546**
8	0.254	0.228	**0.260**	0.255	0.244	0.254	**0.264**	0.246	**0.252**	**0.252**	0.213	0.250
9	0.688	**0.735**	0.691	0.722	0.676	0.698	**0.715**	0.708	0.679	0.707	**0.715**	0.710
10	0.751	0.771	0.746	**0.789**	0.742	0.771	**0.780**	0.778	0.742	0.771	0.780	**0.785**
11	0.875	0.867	**0.884**	**0.884**	0.856	0.858	0.869	**0.880**	0.856	0.858	**0.876**	**0.876**
胜出	1	3	2	**6**	0	0	5	**6**	2	1	3	**7**
排名	3.09	2.73	2.41	**1.77**	3.36	3.05	1.95	**1.64**	2.95	3.14	2.41	**1.50**

3.3　实验分析

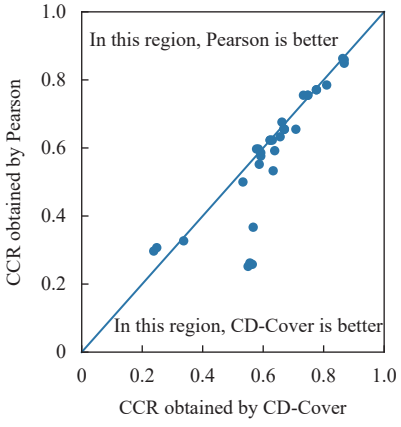
为了更清晰地对实验结果进行分析，将 3.2 节的实验数据绘制为散点图，如图 2 所示。3 个散点图分别代表 CD-Cover 与 IG、Spearman 相关系数和 Pearson 相关系数在 3 种算法、11 个数据集上“一对一”比较的结果，每个散点图中散点数目合计 33 个。图 2 的横坐标和纵坐标分别表示采用 CD-Cover 指标与对比指标得到的 CCR 值。散点如果正好处于对角线上，表明采用两种指标得到的 CCR 值相同；如果散点处于对角线右下方，表明采用 CD-Cover 指标得到的 CCR 值更优；如果散点处于对角线左上方，表明采用对比指标得到的 CCR 值更优。从图 2 可以发现，大多数散点处于对角线右下方，或者在对角线附近；明显处于对角线左上方的散点数目较少。



a. IG 与 CD-Cover

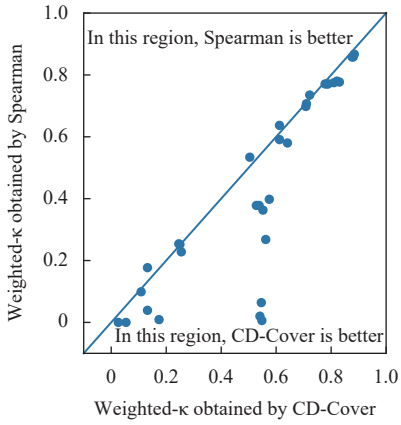


b. Spearman 与 CD-Cover

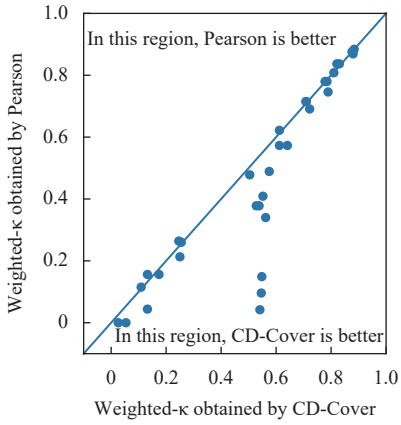


c. Pearson 与 CD-Cover

图 2　CD-Cover 与 3 个指标的 CCR 值比较



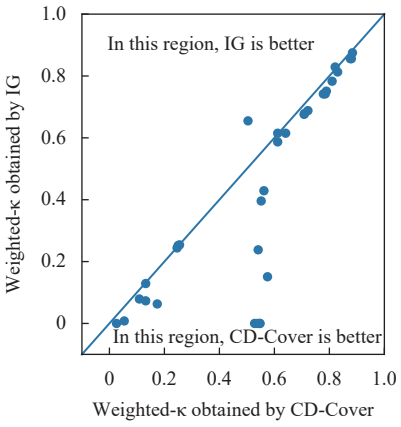
b. Spearman 与 CD-Cover



c. Pearson 与 CD-Cover

图 3　CD-Cover 与 3 个指标的 Weighted-κ 值比较

图 3 为采用 CD-Cover 指标与采用 3 个对比指标得到的 Weighted-κ 值对比结果。图 3 的横坐标和纵坐标分别表示采用 CD-Cover 指标与对比指标得到的 Weighted-κ 值。可以发现，图 3 的大多数散点都处于对角线右下方区域，且相对图 2 而言，偏离对角线更明显。图 2 和图 3 表明，与采用的 3 种对比指标相比，CD-Cover 指标无论是 CCR 还是 Weighted-κ，都能得到更好的结果。



a. IG 与 CD-Cover

3.4　计算效率分析

参与比较的仍然是 3.3 节的 4 种评价指标。区别在于，IG、Spearman 和 Pearson 相关系数不需要将原始时间序列进行 SAX 表示。有序分类器只选择 SVORIM 分类器，分类器的评价指标选择 Weighted-κ。基于不同 Shapelet 评价指标的 Shapelet 抽取算法均持续运行 20 min，并记录下每分钟的 Weighted-κ 值。为保证稳定性，实验结果为 50 次实验的均值，结果如图 4 所示。

图 4 的每个子图 4a～图 4k 分别对应单个数据集的实验结果。所有子图的横坐标均表示 Shapelet 抽取算法的执行时间（min），纵坐标表示基于不同评价指标抽取的 Shapelet 训练的 SVORIM 分类器，在测试数据集上获得的 Weighted-κ 值。分析图 4 可以发现：1）在不同执行时间内，基于 CD-Cover 指标抽取 Shapelet 训练分类器的结果，整体上优于其他 3 种评价指标；2）基于 CD-Cover 指标抽取 Shapelet，在较短时间内就能获得高质量的抽取结果（Colposcopy 数据集为 8 min，其余数据

集均为 2～5 min）。根据上述实验结果可知，本文所提的 CD-Cover 评价指标在计算效率上明显优于其他 3 个评价指标，因为给定相同时间限制，基于 CD-Cover 指标的方法能够评估更多的 Shapelet。

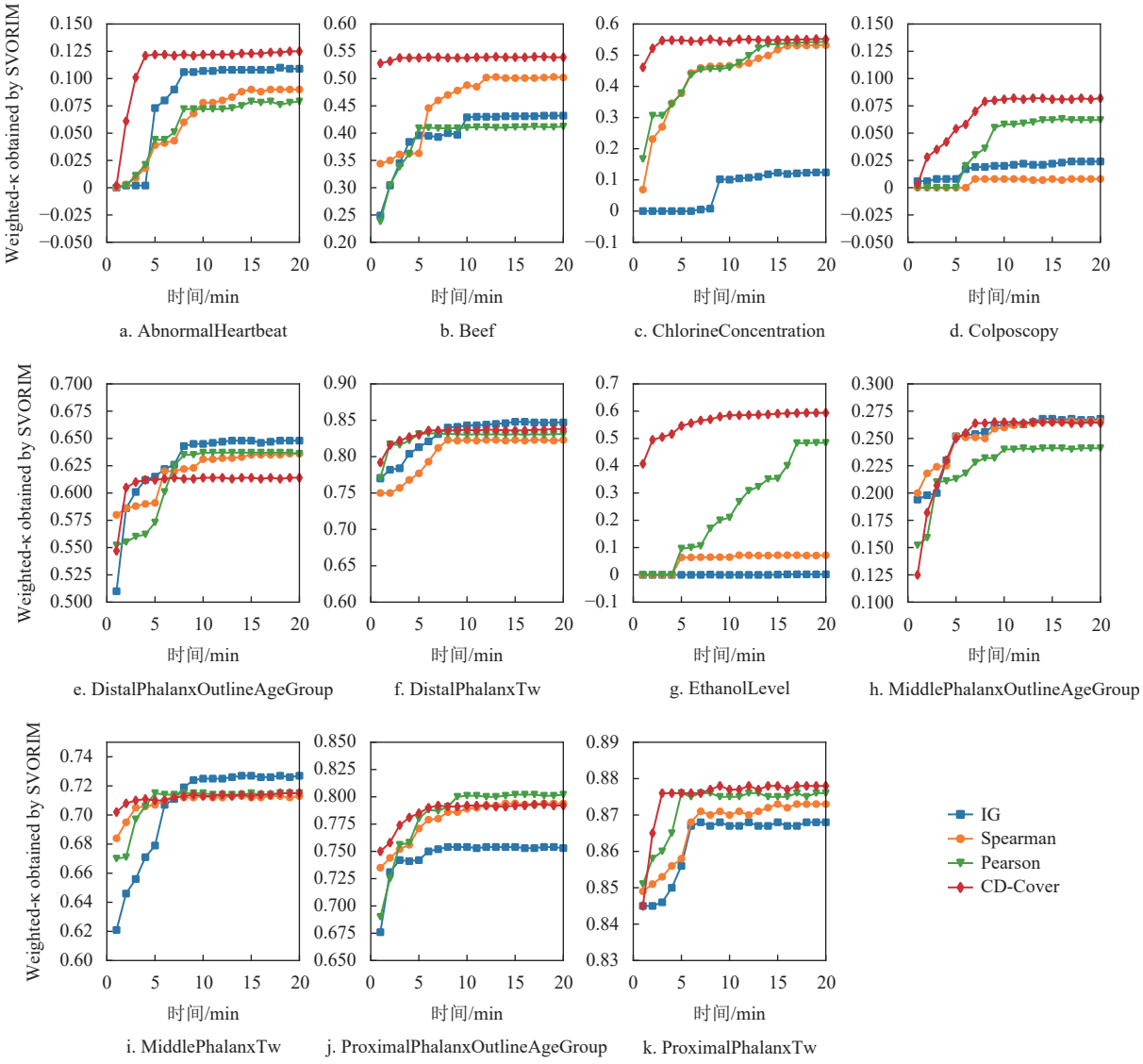


图 4　不同评价指标计算效率对比

4　结 束 语

针对当前基于 Shapelet 的时间序列有序分类研究中，采用 Spearman 相关系数和 Pearson 相关系数进行 Shapelet 抽取时计算效率较低的问题，本文提出了一种基于 SAX 表示时间序列的 Shapelet 评价指标 CD-Cover，以及一种基于随机采样的 Shapelet 抽取算法。在 11 个时间序列数据集上进行了实验。实验结果证明了 CD-Cover 评价指标以及所提 Shapelet 抽取算法的有效性，且展示了其在计算效率上的优越性。下一步将继续研究时间序列长度不相等的情况，并将算法扩展至多元时间序列，并结合实际应用对时间序列有序分类问题进行深入研究。

参　考　文　献

[1] 李海林, 贾瑞颖, 谭观音. 基于 K-Shape 的时间序列模糊分类方法[J]. 电子科技大学学报, 2021, 50(6): 899-906.
LI H L, JIA R Y, TAN G Y. Fuzzy classification for time series data based on K-shape[J]. Chinese Journal of University of Electronic Science and Technology of China, 2021, 50(6): 899-906.

[2] GUIJO-RUBIO D, GUTIÉRREZ P A, BAGNALL A J, et al. Time series ordinal classification via shapelets[C]// Proceedings of the 2020 International Joint Conference on

Neural Networks. Glasgow: IEEE, 2020: 1-8.
[3] 王子一, 商琳. 基于子段距离计算的时间序列分类方法[J]. 小型微型计算机系统, 2018, 39(7): 1386-1389.
WANG Z Y, SHANG L. New method of time series classification based on sub-sequence distance computation[J]. Chinese Journal of Chinese Computer Systems, 2018, 39(7): 1386-1389.
[4] 闫汶和, 李桂玲. 基于 shapelet 的时间序列分类研究[J]. 计算机科学, 2019, 46(1): 29-35.
YAN W H, LI G L. Research on time series classification based on shapelet[J]. Chinese Journal of Computer Science, 2019, 46(1): 29-35.
[5] 赵超, 王腾江, 刘士军, 等. 融合选择提取与子类聚类的快速 Shapelet 发现算法[J]. 软件学报, 2020, 31(3): 763-777.
ZHAO C, WANG T J, LIU S J, et al. Fast shapelet discovery algorithm combining selective extraction and subclass clustering[J]. Chinese Journal of Software, 2020, 31(3): 763-777.
[6] YE L, KEOGH E. Time series shapelets: A new primitive for data mining[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: ACM, 2009: 947-956.
[7] HILLS J, LINES J, BARANAUSKAS E, et al. Classification of time series by shapelet transformation[J]. Data Mining and Knowledge Discovery, 2014, 28(4): 851-881.
[8] MUEEN A, KEOGH E, YOUNG N. Logical-Shapelets: An expressive primitive for time series classification[C]//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego: ACM, 2011: 1154-1162.
[9] KEOGH E, RAKTHANMANON T. Fast Shapelets: A scalable algorithm for discovering time series shapelets[C]//Proceedings of the 13th SIAM International Conference on Data Mining. Austin: SIAM, 2013: 668-676.
[10] WISTUBA M, GRABOCKA J, SCHMIDT-THIEME L. Ultra-Fast shapelets for time series classification[EB/OL]. [2022-07-21]. https://arxiv.org/pdf/1503.05018v1.pdf.
[11] KARLSSON I, PAPAPETROU P, BOSTRÖM H. Generalized random shapelet forests[J]. Data Mining and Knowledge Discovery, 2016, 30(5): 1053-1085.
[12] YANG J, JING S Y, HUANG G Y. Accurate and fast time series classification based on compressed random Shapelet Forest[J]. Applied Intelligence, 2022, DOI: 10.1007/s10489-022-03852-2.
[13] LI G Z, CHOI B, XU J L, et al. ShapeNet: A shape-let-neural network approach for multivariate time series classification[C]//Proceedings of the 35th AAAI Conference on Artificial Intelligence. Virtual Event: AAAI, 2021: 8375-8383.
[14] MA Q L, ZHUANG W Q, LI S, et al. Adversarial dynamic shapelet networks[C]//Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020: 5069-5076.
[15] GUIJO-RUBIO D, GUTIÉRREZ P A, BAGNALL A J, et al. Ordinal versus nominal time Series classification[C]//Proceedings of the Advanced Analytics and Learning on Temporal Data-5th ECML PKDD. Ghent: Springer, 2020: 19-29.
[16] BAGNALL A, LINES J, VICKERS W, et al. The UEA & UCR time series classification repository[EB/OL]. [2022-07-31]. http://www.timeseriesclassification.com.
[17] LIN J, KEOGH E, LONARDI S, et al. A symbolic representation of time series, with implications for streaming algorithms[C]//Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. San Diego: ACM, 2003: 2-11.
[18] LI G Z, CHOI B, XU J L, et al. Efficient shapelet discovery for time series classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(3): 1149-1163.
[19] YAN W H, Li G L, WU Z D, et al. Extracting diverse-shapelets for early classification on time series[J]. World Wide Web, 2020, 23(6): 3055-3081.
[20] GUTIÉRREZ P A, PÉREZ-ORTIZ M, SÁNCHEZ-MONEDERO J, et al. Ordinal regression methods: Survey and experimental study[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(1): 127-146.
[21] HSU C W, LIN C J. A comparison of methods for multiclass support vector machines[J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425.
[22] CHU W, KEERTHI S S. Support vector ordinal regression[J]. Neural Computation, 2007, 19(3): 792-815.
[23] SAKAI T. Evaluating evaluation measures for ordinal classification and ordinal quantification[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. Virtual Event: Association for Computational Linguistics, 2021: 2759-2769.

编 辑 税 红