基于符号表示的可度量shapelets提取的时序分类研究

王礼勤, 万源, 罗颖

引用本文

王礼勤, 万源, 罗颖. 基于符号表示的可度量shapelets提取的时序分类研究[J]. 计算机科学, 2024, 51(8): 106-116.

WANG Liqin, WAN Yuan, LUO Ying. Measurable Shapelets Extraction Based on Symbolic Rrepresentation for Time Series Classification [J]. Computer Science, 2024, 51(8): 106-116.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230500161

基于符号表示的可度量 shapelets 提取的时序分类研究

王礼勤　万　源　罗　颖
武汉理工大学理学院　武汉 430070
(319578@whut.edu.cn)

摘　要　在时序分类问题中,基于符号表示的 shapelets 提取方法具有良好的分类精度和分类效率,但对符号进行质量度量的过程,如计算 TFIDF 分数,耗时较长且计算量大,导致分类效率较低。此外,提取的 shapelets 候选数量仍然较多,判别力有待提高。针对这些问题,本文提出了一种基于符号表示的可度量 shapelets 提取方法,该方法包含时间序列数据预处理、确定 shapelets 候选集和学习 shapelets 3 个阶段,可以快速得到高质量 shapelets。在数据预处理阶段,将时间序列转化为符号聚合近似(SAX)表示以降低原始时间序列的维度。在确定 shapelets 候选集阶段,利用 Bloom 过滤器过滤重复的 SAX 词,并将过滤后的 SAX 词存储在哈希表中进行质量度量。随后,对 SAX 词的相似性进行判别,基于相似性和覆盖度等概念确定最终的 shapelets 候选集。在学习 shapelets 阶段,采用 logistic 回归模型学得真正的 shapelets 用于时序分类。在 32 个数据集上进行了大量实验,实验结果表明,所提方法的平均分类精度和平均分类效率均排名第二。与现有的基于 shapelets 的时序分类方法相比,该方法可以在保证精度的同时提高分类效率,并且具有良好的可解释性。

关键词:时间序列分类;shapelet;SAX 表示;Bloom 过滤器;logistic 回归
中图分类号　TP391

Measurable Shapelets Extraction Based on Symbolic Rrepresentation for Time Series Classification

WANG Liqin,WAN Yuan and LUO Ying
School of Science,Wuhan University of Technology,Wuhan 430070,China

Abstract　In the time series classification problems,shapelets extraction method based on symbol representation has good classification accuracy and efficiency,but the quality measurement of symbols,such as calculating TFIDF scores,is time-consuming and computatively heavy,leading to low classification efficiency. In addition,there are still a large number of shapelets candidates extracted,and the discriminating power needs to be improved. To solve these problems,this paper proposes a measurable shapelets extraction method based on symbolic representation, which includes three stages: time series data preprocessing, determining shapelets candidate set and learning shapelets,so that high-quality shapelets can be obtained quickly. In the data preprocessing stage,the time series is transformed into a symbolic aggregation approximation(SAX)representation to reduce the dimensions of the original time series. In the stage of determining the candidate set of shapelets,Bloom filters are used to filter repeated SAX words,and the filtered SAX words are stored in the hash table for quality measurement. Then,the similarity of SAX words is discriminated,and the final shapelets candidate set is determined based on the concepts of similarity and coverage. In the learning phase of shapelets,the logistic regression model is used to learn real shapelets for time series classification. In this paper,a large number of experiments are conducted on 32 datasets,and the experimental results show that the average classification accuracy and average classification efficiency of the proposed method rank second on 32 datasets. Compared with the existing time series classification methods based on shapelets,the proposed method can improve the classification efficiency while ensuring the accuracy,and has good interpretability.

Keywords　Time series classification,Shapelet,SAX means,Bloom filters,Logistic regression

1　引言

时间序列分类(Time Series Classification,TSC)在生物[1]、医疗卫生[2]、金融[3]等领域有着广泛应用。传统的时间序列分类方法将整条时间序列数据作为研究对象,使用时间序列的相似性度量作为特征来构建分类器[4],其中相似性度量方法主要有欧氏距离[5]和动态时间弯曲距离(Dynamic Time Warping,DTW)[6]。这类基于全序列的方法需要消耗

到稿日期:2023-05-24　返修日期:2023-10-22
基金项目:中央高校基本科研业务费专项资金(2021III030JC)
This work was supported by the Fundamental Research Funds for the Central Universities of Ministry of Education of China(2021III030JC).
通信作者:万源(wanyuan@whut.edu.cn)

大量的时间和空间来对整个时间序列数据集进行搜索和存储,限制了其在大规模时间序列数据集上的应用。此外,通常对时间序列分类起关键作用的是局部信息,因为局部信息更能反映出不同类别数据之间的差别。

为解决上述问题,2009 年 Ye 等[7]首次提出了"shapelet"的概念。Shapelet 是时间序列中具有判别性的子序列,能够最大程度地代表某一类时间序列。与传统的时间分类方法相比,基于 shapelets 的时间序列分类方法提高了分类精度,其分类结果还具有可解释性。因此,基于 shapelets 的方法成为时间序列分类领域中的研究热点,这类方法的关键在于对具有判别性子序列的提取。

原始的子序列提取方法是 Ye 等[7]提出的暴力搜索算法,该算法直接从时间序列的子序列中搜索所有可能的 shapelets,搜索规模和时间消耗都很大。为了加快搜索过程,他们提出了子序列距离早期放弃策略和可容许熵剪策略[7],但这些策略不适用于大规模数据集。为了进一步提高搜索 shapelets 的效率,许多运用加速技术的算法被提出,如使用在线聚类和剪枝筛选技术的可扩展 shapelets 发现算法(SD)[8]、使用子类分裂技术和局部最远偏差点的快速 shapelets 选择算法(FSS)[9]、改进的快速 shapelets 选择算法[10]等。虽然这些方法在一定程度上加快了 shapelets 的提取过程,但得到的 shapelets 候选数量仍然很大,分类的准确率也有待提高。

Grabocka 等[11]提出的学习时间序列 shapelets 方法(Learning Time-series Shapelets,LTS)为 shapelets 的提取提供了一种新思路。该方法不是从原始时间序列中搜索子序列作为 shapelets 候选,而是通过逻辑回归模型构建 shapelets 提取的目标函数,并使用随机梯度下降法对目标函数进行优化,实现了较好的分类效果,但在大规模数据集中耗时较长。作为 LTS 的改进,Hou 等[12]提出的融合 Lasso 广义特征向量法将广义特征向量和融合 Lasso 结合起来学习 shapelets 的位置,然后将这些位置上的子序列进行提取作为 shapelets,与 LTS 相比提高了分类速度,但分类精度有所降低。Zhang 等[13]基于维数提出了一种 shapelets 提取方法,该方法结合 Fisher 判别分析和两种稀疏来学习 shapelets 的位置,获得了较高的分类精度,但分类效率较低。

除了上述直接在原始时间序列空间中提取 shapelets 的方法外,Rakthanmanon 等[14]提出了一种快速的 shapelets 发现算法,该方法将时间序列转化为符号聚合近似(Symbolic Aggregate approximation,SAX)以降低空间维度,然后在 SAX 空间中提取 shapelets 候选,实现了较快的分类速度,但分类精度较低。与此不同,Fang 等[15]在分段聚合近似(Piecewise Aggregate Appoximation,PAA)空间中寻找 shapelets 候选,并用 TFIDF 分数来度量 PAA 词的质量,该方法提高了分类精度,但复杂的分数计算导致分类速度较慢。为了在提高分类效率的同时不降低分类精度,Li 等[16]基于 SAX 技术提出了一种高效的 shapelets 学习算法,该算法首次在 shapelets 发现中使用 Bloom 过滤器,大大减少了 shapelets 候选的数量,还为 SAX 词构造了位图以度量其质量,从而筛选出高质量的 shapelets 候选。

上述基于符号表示的 shapelets 提取方法能够在一定程度上提高分类速度,同时保持良好的分类准确率,但大规模数据集中,仍然存在符号的质量度量消耗时间大、shapelets 候选数量较多,判别力不足等问题。

为此,本文基于符号表示提出了一种可度量的 shapelets 提取方法(Measurable Shapelets Extraction based on Symbolic Representation,SR-MSE)。该方法首先将时间序列转化成 SAX 词,受到 Li 等[16]的启发,利用 Bloom 过滤器对重复的 SAX 词进行过滤,然后将过滤后的 SAX 词储存在哈希表中并对其质量进行度量,再基于相似性、覆盖度、覆盖增益等概念来确定 shapelets 候选。最后,运用 logistic 回归模型对 shapelets 候选进行学习得到真正的 shapelets。这样的 shapelets 可能不是时间序列的子序列,而是能够对类别进行划分的特征,可以提高泛化能力,使分类结果更具可解释性。在 UCR 数据库中的 32 个数据集上对本文提出的 shapelets 提取方法进行了验证,实验结果表明,本文算法在保证分类精度的同时提高了分类效率,并且具有良好的可解释性。

2 相关工作

现有的基于 shapelets 的时间序列分类方法大体可以分为 3 类,分别为基于搜索的方法、基于学习的方法和基于符号的方法。

基于搜索的方法是直接从原始时间序列中搜索所有可能的 shapelets 候选,并使用 Kruskall-Wallis 检验[17]、Mood 中位数检验[18]、信息增益[11]、F 统计量[19]等方法评估候选的判别力。传统的暴力搜索算法[7]利用信息增益对所有子序列候选进行质量度量,时间复杂度为 $O(m^2n^4)$,其中 m 是时间序列的数量,n 是时间序列的长度。2011 年 Mueen 等[20]对该方法做了改进,运用了多个统计量以减少重复的距离计算,并采用信息增益上界作为质量度量降低了时间复杂度,为 $O(m^3n^2)$。Yuan 等[21]在此基础上运用 shapelets 之间的逻辑组合关系提高了分类精度,但在效率方面没有得到提升。为了提高效率,Lines 等[22]提出利用关键点来提取 shapelets 候选,然后基于最优 shapelets 构造决策树以对时间序列进行分类。Zou 等[10]利用改进的 K 均值法对时间序列进行聚类,然后根据时间序列的重要数据点来选择 shapelets 候选。

基于学习的方法是将 shapelets 的提取问题转化成数学方面的优化问题,通过优化目标函数,从训练数据中提取具有高判别力的 shapelets。Grabocka 等[11]提出的 LTS 算法是最先使用基于学习的方法提取 shapelets 的算法,其时间复杂度很高,为 $O(IN_umn^2)$,其中 N_u 是每一次迭代需更新的 shapelets 数量,I 是迭代次数。为了降低时间复杂度,许多改进的 shapelets 学习方法被提出[12-13],但这些方法很难同时实现良好的分类速度和准确率。为了同时提高分类的精度和效率,Zhao 等[23]提出了一种正则化的 shapelets 学习框架(RSLA),该方法使用融合 Lasso 正则化器和不同的损失函数作为目标函数来学习 shapelets,然后计算 shapelets 和时间序列之间的欧氏距离以应用于常规分类器来对时间序列进行分类。

基于符号的时间序列分类方法是将时间序列转换成符号表示,如 PAA[15],SAX[24],SFA[25]。2013 年 Senin 等[26]基于 SAX 技术提出了 SAX-VSM 算法,该算法提出了向量空间

模型并根据时间序列模式对类的重要程度对模式进行了排序，这为分类结果提供了可解释性但计算量大。Nguyen等[27]将各种可变长度的符号单词表示和高效的线性序列学习方法(SAX-VSEQL[28])进行了结合，利用梯度下降法从训练数据中提取具有判别力的子序列，实现了高效的时间序列分类。Liang等[29]在SAX空间中为数据集的每个类别提取了特定于该类的shapelets候选，使得分类结果具有良好的可解释性。基于PAA技术，Zhang等[30]提出了ELIS++算法，为了避免手动设置参数，该算法提出了一种基于贝叶斯优化的方法，还使用了数据增强、shapelets细粒度化、并行计算等技术，实现了良好的分类速度。

除了上述方法之外，还有几种分类器集成的方法，例如COTE[31]，该方法将多种分类器相结合以对时间序列进行分类；此外还有运用深度学习的方法，例如ResNet[32]，该方法运用神经网络对时间序列进行分类。

3 相关定义与符号

定义1(时间序列 T) 时间序列 T 是按时间顺序排列的一组数字序列，表示为 $T=(t_1,t_2,\cdots,t_n)$，其中 n 为时间序列的长度。

定义2(时序数据集 D) 时序数据集 D 是由一组带有类别标签的时间序列构成的集合，表示为 $D=\{T_1,T_2,\cdots,T_m\}$，其中类别标签集 $C=\{0,1,2,\cdots,|C|-1\}$，$|C|$ 是数据集中类别的数量。

定义3(PAA序列 $\bar{T}$) 长度为 n 的时间序列 T 可以转化为长度为 ω 的PAA序 $\bar{T}=(\bar{t}_1,\bar{t}_2,\cdots,\bar{t}_\omega)$，其中 $\bar{t}_i$ 是第 i 个片段内的时间序列均值，按式(1)计算：

$$\bar{t}_i=\frac{\omega}{n}\sum_{j=\frac{n}{\omega}(i-1)+1}^{\frac{n}{\omega}i}t_j \tag{1}$$

定义4(SAX序列 $\hat{T}$) SAX序列 $\hat{T}=(\hat{t}_1,\hat{t}_2,\cdots,\hat{t}_\omega)$ 由PAA序列映射到遵循高斯分布的断点列表[33]值表示，Σ 表示字符集的大小，它决定了SAX序列由多少个字符表示。

图1给出了一条长度为35($n=35$)的时间序列的SAX表示，利用式(1)转化为了长度为5($\omega=5$)的PAA序列 $\bar{T}$。若选定大小为8($\omega=5$)的字符集，则可将PAA序列 $\bar{T}$ 映射为SAX序列 $\hat{T}=FCBGG$。

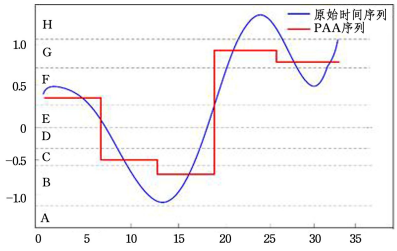

图1 时间序列的SAX表示

Fig. 1 SAX representation of time series

定义5(shapelets的判别力) shapelets的判别指shapelets对时间序列类别的区分能力，高判别力的shapelets能将不同类别的时间序列明显区分开来。

4 本文方法

为了能够快速找到具有高判别力的shapelets，在提高时间序列分类速度的同时保证分类准确率，本文提出了基于符号表示的可度量shapelets提取方法SR-MSE。该方法由3部分构成：1)数据预处理，将时间序列转化为低维的符号聚合近似(SAX)表示，可以进行数据压缩从而节省空间；2)确定shapelets候选集，利用Bloom过滤器对重复的SAX词过滤，基于哈希表对过滤后的SAX词进行质量度量，再根据相似性、覆盖度、分类错误率等概念确定shaplets候选集；3)学习真正的shapelets，运用logistic回归模型学习真正的shapelets，借助已选择的shapelets候选对shapelets进行初始化，并使用梯度下降法对目标函数进行求解。本文方法的整体框架如图2所示。

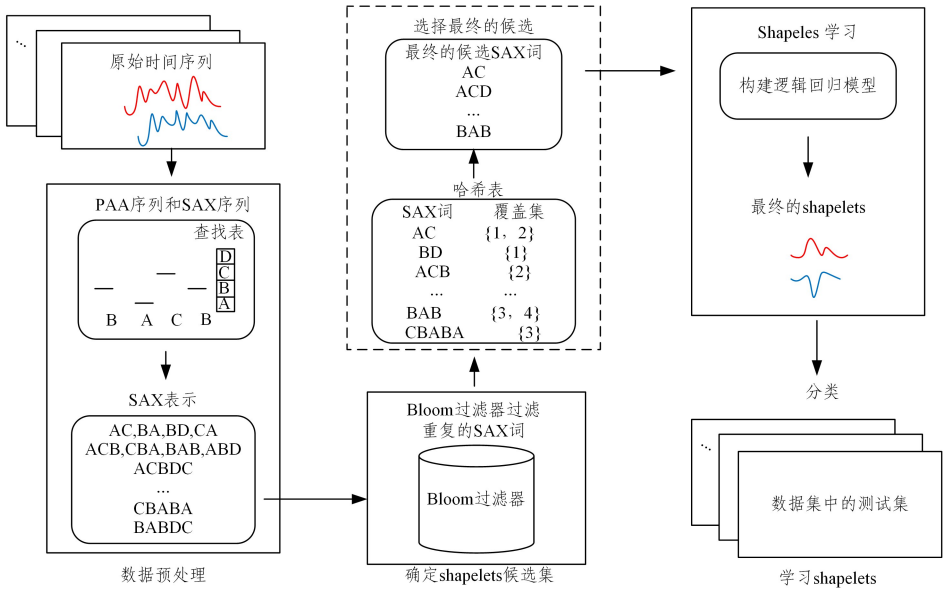

图2 本文方法的整体框架

Fig. 2 Overall framework of the proposed method

4.1 时间序列 SAX 词的生成

在大规模数据集中,直接从原始时间序列空间中搜索 shapelets 候选不仅需要很大的存储空间还非常耗时,因此,将时间序列转换为低维的 SAX 表示,在低维空间寻找 shapelets 候选以节省空间和时间是十分必要的。基于此,本文将数据集中的每一类时间序列都转化成 SAX 表示。

首先将 n 维时间序列 $T=(t_1,t_2,\cdots,t_n)$ 转化为 ω 维 PAA 序 $\bar{T}=(\bar{t}_1,\bar{t}_2,\cdots,\bar{t}_\omega)$,其中 $\bar{t}_i$ 是第 i 个片段内的时间序列均值,按式(1)计算得到。PAA 序列的维数 ω 决定了时间序列信息的丢失程度和所需的存储空间,ω 越大表示维数越多包含的信息特征也越多,但所需的存储空间也会增大。为此,对于不同长度的时间序列,本文设置了不同的 ω 值,如下所示:

$$\omega=\begin{cases}\dfrac{n}{2}, & 1\leqslant n\leqslant 200\\ 200, & 200<n\leqslant 600\\ 300, & 600<n\leqslant 1\,200\\ 400, & 1200<n\leqslant 2\,000\\ 500, & n>2000\end{cases}\tag{2}$$

对于 PAA 序列 $\bar{T}$,选定字符集 Σ 的大小,通过查找高斯分布断点表[33]将其映射为 SAX 序列 $\hat{T}$。

在 SAX 序列 $\hat{T}$ 上利用不同大小的滑动窗口 $\varphi=\left\{\left[\frac{\omega}{4}\right],\left[\frac{\omega}{3}\right],\left[\frac{\omega}{2}\right],\omega\right\}$ 生成多种长度的 SAX 词 e,并将其加入 SAX 词集 $\Omega_c(c\in C)$ 中。

图 3 给出了将原始时间序列转换成 SAX 词的一个例子。假设原始时间序列的长度 $n=8$,PAA 序列的长度 $\omega=4$,字符集的大小 $\Sigma=4$,利用式(1)将时间序列转换为 PAA 序列,然后根据高斯分布断点表将 PAA 序列映射为 SAX 序列 BCCA。在 SAX 序列上,生成滑动窗口 $\varphi=2$ 的 SAX 词集 $\{BC,CC,CA\}$。

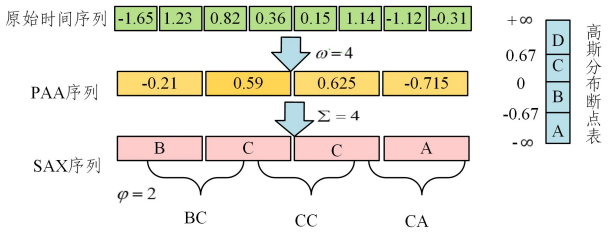

图 3 时间序列 SAX 词的生成示例

Fig. 3 Example of generating SAX words in time series

4.2 确定 shapelets 候选集

4.2.1 构造 Bloom 过滤器过滤重复的 SAX 词

将时间序列转化成 SAX 词后,有部分 SAX 词会在数据集的多个类别中重复出现,说明这类 SAX 词的判别力较弱,对分类所起的作用较小。因此本文使用 Bloom 过滤器来过滤在多个类别中重复出现的 SAX 词以提高效率,具体过程如算法 1 所示。

算法 1 Bloom 过滤器的构造

输入:SAX 词集 Ω_c

输出:过滤后的 SAX 词集 Ω_c^{BF}

1. Initialize BF_c for each c in C;//初始化数据集中每一类别的 Bloom 过滤器
2. 使用 MD5 哈希函数将 Ω_c 中的每个 SAX 词 e 加入过滤器 BF_c 中;
3. Initialize Ω_c^{BF};=ϕ //初始化通过 Bloom 过滤器的 SAX 词集 Ω_c^{BF};
4. 将 Ω_c 中的每一个 SAX 词 e 都作为其他类过滤器的查询,并返回查询结果;
5. 如果查询结果为"一定不存在",则将该 SAX 词 e 添加到 Ω_c^{BF} 中。如果查询结果为"可能存在",则不添加;
6. return Ω_c^{BF}。

在算法 1 中的过滤阶段,每一类 SAX 词集 $\Omega_c(c\in C)$ 中的每个 SAX 词对其他类的 Bloom 过滤器而言都是一个查询。查询的结果有两种,分别为"可能存在"和"一定不存在"。"可能存在"意味着该 SAX 词有很大的概率存在于其他类时间序列中,需要将其在所有时间序列中剔除。"一定不存在"表明该 SAX 词一定不会在其他类的时间序列中出现,具有很高的判别力,可以被保留在候选集 Ω_c^{BF} 中以进行下一步的处理。具体操作如图 4 所示。

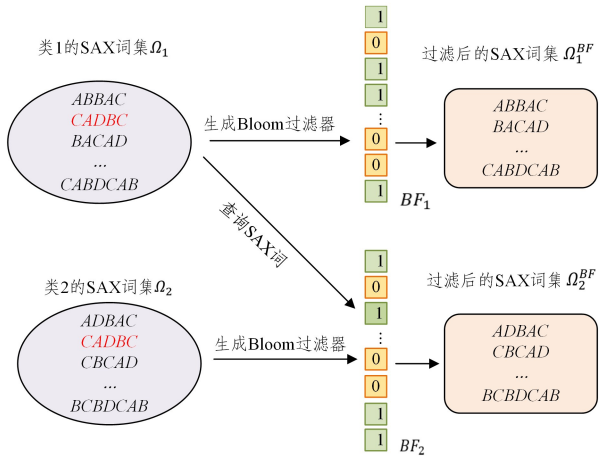

图 4 利用 Bloom 过滤器过滤重复的 SAX 词

Fig. 4 Filter duplicate SAX words using Bloom filter

在图 4 中,类别 1 的 SAX 词集 $\Omega_1=\{ABBAC,CADBC,\cdots\}$,类别 2 的 SAX 词集 $\Omega_2=\{ADBAC,\cdots,BCBDCAB\}$。分别为 Ω_1 和 Ω_2 构造一个 Bloom 过滤器 BF_1 和 BF_2,然后将类别 1 中的每个 SAX 词作为 BF_2 的查询并返回查询结果。如 SAX 词 $CADBC$ 同时存在于类别 1 和类别 2 中,查询结果为"可能存在",则将其从所有时间序列中剔除。其他通过 Bloom 过滤器的 SAX 词具有很高的判别力,继续保留在候选集中。

4.2.2 基于哈希表的 SAX 词的质量度量

为了寻找高判别力的 shapelets 候选,需要对 SAX 词的质量进行度量。现有的符号质量度量方法需要消耗大量时间,这会降低分类的效率。哈希表是一种数据结构(本质上是一个数组),存储的是键值对,键通过哈希函数得到数组的索引,进而存取索引位置的值。其数据存储和查找速度快,需要消耗的时间很少。因此,对通过 Bloom 过滤器的每一类 SAX 词集 $\Omega_c^{BF}(c\in C)$ 的每一个 SAX 词,我们将其存储在一个哈希表

H_c 中，并基于哈希表对 SAX 词的质量进行度量。

定义 6(SAX 词覆盖) 如果一个 SAX 词能被时间序列 T 的子序列转换得到，则认为该 SAX 词覆盖了时间序列 T。SAX 词覆盖的时间序列索引集称为覆盖集。

在哈希表中，我们将 SAX 词作为键，将 SAX 词覆盖的时间索引集即覆盖集作为值。SAX 词的覆盖集越大，表明其覆盖的时间序列数量越多，更能代表该类时间序列。因此，本文将 SAX 词的覆盖集作为 SAX 词的质量度量，覆盖集越大其质量就越好。

在对哈希表 $H_c(c\in C)$ 中的 SAX 词进行质量度量后，将其按覆盖集的大小进行降序排列，并储存在列表 L_c 中以进行下一步处理。

4.2.3 选择最终的候选

为了减少 shapelets 候选的数量，提高分类速度，在对 SAX 词的质量进行度量后，要对列表 L_c 中的 SAX 词进行筛选，剔除判别力较弱和判别力相似的 SAX 词，只保留少量具有高判别力的 SAX 词并将其作为 shapelets 候选。

现有的符号相似性判别方法在大规模数据集中计算量大且计算过程复杂，因此本文定义了一种计算简单且快速的 SAX 词相似性判别方法，以剔除判别力相似的 SAX 词。

给定两个长度相同的 SAX 词 $e_1=(x_1,x_2,\cdots,x_l)$ 和 $e_2=(y_1,y_2,\cdots,y_l)$，它们之间的 L_1 距离如下：

$$L_1D(e_1,e_2)=\sum_{i=1}^{l}|x_i-y_i| \tag{3}$$

定义 7(SAX 词的相似性) 如果两个长度相同的 SAX 词 e_1 和 e_2 满足以下两个条件，则认为它们相似。

$$|x_i-y_i|\leqslant 1,i=1,2,\cdots,l \tag{4}$$

$$L_1D\leqslant\frac{l}{2} \tag{5}$$

这样做的合理性在于满足以上条件的两个 SAX 词之间的差异很小。

定义 8(覆盖度[15]) 覆盖度 θ 是一个参数，它决定了一个时间序列的特征是否被足够多的候选所覆盖。如果类别 c 的每个时间序列至少被从 L_c 中挑选的候选 SAX 词集覆盖 θ 次，则认为候选 SAX 词集利用覆盖度 θ 已经覆盖了类别 c 中所有时间序列的特征。

定义 9(覆盖增益[15]) SAX 词的覆盖增益等于被其所覆盖的覆盖率低于 θ 的时间序列数量。

对于不同的覆盖度 θ，本文选择 shapelets 候选的具体算法过程如算法 2 所示。

算法 2 选择最终的 shapelets 候选

输入：列表 $L_c(c\in C)$

输出：最终的 shapelets 候选集 Ω_c^{fin}

1. Initialize $\Omega_c^{fin}=\phi$；//初始化数据集中每一类别的最终 shapelets 候选集
2. $\theta=\{1,2,3,4,5\}$；
3. 去除 L_c 中没有覆盖增益的 SAX 词；
4. 检查 SAX 词之间的相似性，若相似，则只选择覆盖集大的将其加入候选集 Ω_c^{θ} 中；
5. 将 SAX 词转换回时间序列；
6. 选择分类错误率最小的 Ω_c^{θ} 作为最终候选集 Ω_c^{fin}
7. return Ω_c^{fin}

在算法 2 中，对不同的覆盖度 θ，首先按顺序遍历 L_c 中的每个 SAX 词，筛除没有覆盖增益的 SAX 词，然后检查 L_c 中 SAX 词之间的相似性。如果两个 SAX 词相似，则只将覆盖集大的 SAX 词加入候选集中，之后将候选集中的 SAX 词转换回时间序列空间作为 shapelets 候选。得到不同覆盖度 θ 的候选后，利用分类错误率来衡量候选的质量，并选择错误率最小的作为最终的候选。

本文 θ 的取值范围为 1～5，在 5.2 节的实验中验证了该取值范围的合理性。

4.3 学习真正的 shapelets

在 4.2 节中，我们为数据集中的每个类选择了 shapelets 候选，这些候选全部是时间序列的子序列。受 LTS[11] 的启发，具有类别判别力的 shapelets 可以是时间序列的子序列，也可以不是。因此，本文采用 logistic 回归模型来调整 shapelets 候选。

与 LTS[11] 中学习所有类共享的 shapelets 不同，本文为数据集中的每个类都构建了一个二元分类器，并通过随机梯度下降法进行求解，以得到每个类的 shapelets，这样有助于提高分类结果的可解释性。

4.3.1 Logistic 回归

使用线性学习模型来预测近似目标值 $\hat{Y}^{(i)}$，如式(6)所示：

$$\hat{Y}^{(i)}=W_0+\sum_{k=1}^{K_c}W_k dist(T_i,S_k) \tag{6}$$

其中，W_0 是偏差项，W_i 是线性权重，K_c 是类别 c 的 shapelet 数量。$dist(T_i,S_k)$ 是时间序列 $T_i=(t_{i,1},t_{i,2},\cdots,t_{i,n})$ 和 shapelet$S_k=(s_{k,1},s_{k,2},\cdots,s_{k,L})$ 之间的距离，计算式如下：

$$dist(T_i,S_k)=\min_{j=1,\cdots,J}\frac{1}{L}\sum_{l=1}^{L}(t_{i,j+l-1}-s_{k,l})^2 \tag{7}$$

其中，$J=n-L+1$。$\hat{Y}^{(i)}$ 是利用 logistic simoid 函数进行分类的，如下所示：

$$\sigma(\hat{Y}^{(i)})=\frac{1}{1+e^{-\hat{Y}^{(i)}}} \tag{8}$$

4.3.2 目标函数

对于类别 c，对损失函数添加正则化项的目标函数表示为：

$$\arg\min_{S,W}\mathcal{F}(S,W)=\arg\min_{S,W}\sum_{i=1}^{m}\mathcal{L}(\hat{Y}^{(i)},Y^{(i)})+\lambda\|W\|^2 \tag{9}$$

其中，S 是类别 c 的 shapelets 集合；$\boldsymbol{W}=(\boldsymbol{W}_0,\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{k_c})$ 是权重向量；λ 是 ℓ_2 正则项系数，在实验中由分类准确率确定。$L(\hat{Y}^{(i)},Y^{(i)})$ 是一个实例的真实目标值和预测目标值之间的分类损失，计算式如下：

$$L(\hat{Y},Y)=-Y\ln\sigma(\hat{Y})-(1-Y)\ln(1-\sigma(\hat{Y})) \tag{10}$$

本文使用随机梯度下降法来优化目标函数。然而，式(7)

中的最小函数是不可微的，因此，使用最小函数的可微近似（即 soft-minimum 函数）来代替最小函数，定义为：

$$\hat{dist}(T_i,S_k)=\frac{\sum_{j=1}^{J}D_{k,L,j}\,e^{\alpha D_{i,k,j}}}{\sum_{j=1}^{J}e^{\alpha D_{i,k,j}}} \tag{11}$$

其中，$D_{i,k,j}=\frac{1}{L}\sum_{l=1}^{L}(t_{i,j+l-1}-s_{k,l})^2$。参数 α 控制函数的精度，由文献[18]可知，$\alpha=-25$ 时足够使 soft-minimum 距离趋近于最小距离，故本文将 α 的值固定为 -25。

随机梯度下降法一次纠正由一个实例引起的预测误差。于是，将式(9)中的目标函数分解为每个实例 T_i 的目标函数，每个分解的目标函数代表每个时间序列实例所造成的分类损失，表示为：

$$\mathscr{F}_i=L(\hat{Y}^{(i)},Y^{(i)})+\frac{\lambda}{m}\sum_{k=1}^{K_s}W_k^2 \tag{12}$$

4.3.3 目标函数优化求解

固定 $\boldsymbol{W}_0$ 和 $\boldsymbol{W}_k$，对 S_k 求偏导，可得目标函数(12)相对于 shapeletS_k 在点 l 处的梯度为：

$$\frac{\partial\mathscr{F}_i}{\partial s_{k,l}}=\frac{\partial\mathscr{L}(\hat{Y}^{(i)},Y^{(i)})}{\partial\hat{Y}^{(i)}}\frac{\partial\hat{Y}^{(i)}}{\partial\hat{dist}(T_i,S_k)}\frac{\partial\hat{dist}(T_i,S_k)}{\partial s_{k,l}} \tag{13}$$

其中，损失函数(10)相对于预测目标的梯度、预测目标相对于最小距离的梯度、最小距离相对于 shapeletS_k 的梯度分别为：

$$\frac{\partial\mathscr{L}(\hat{Y}^{(i)},Y^{(i)})}{\partial\hat{Y}^{(i)}}=-(Y^{(i)}-\sigma(\hat{Y}^{(i)})) \tag{14}$$

$$\frac{\partial\hat{Y}^{(i)}}{\partial\hat{dist}(T_i,S_k)}=\boldsymbol{W}_k \tag{15}$$

$$\frac{\partial\hat{dist}(T_i,S_k)}{\partial s_{k,l}}=\sum_{j=1}^{J}\frac{\partial\hat{dist}(T_i,S_k)}{\partial D_{i,k,j}}\frac{\partial D_{i,k,j}}{\partial s_{k,l}} \tag{16}$$

在式(16)中，最小距离相对于分段距离 $D_{i,k,j}$ 的梯度和分段距离相对于 shapeletS_k 在点 l 处的梯度分别为：

$$\frac{\partial\hat{dist}(T_i,S_k)}{\partial D_{i,k,j}}=\frac{e^{\alpha D_{k,L,j}}(1+\alpha(D_{i,k,j}-\hat{dist}(T_i,S_k)))}{\sum_{j=1}^{J}e^{\alpha D_{i,k,j}}} \tag{17}$$

$$\frac{\partial D_{i,k,j}}{\partial s_{k,l}}=\frac{2}{L}(S_k-t_{i,j+l-1}) \tag{18}$$

然后，固定 S_k，分别对 $\boldsymbol{W}_0$ 和 $\boldsymbol{W}_k$ 求导，可得目标函数(12)相对于线性权重 $\boldsymbol{W}_k$ 和 $\boldsymbol{W}_0$ 的梯度：

$$\frac{\partial\mathscr{F}_i}{\partial\boldsymbol{W}_k}=-(Y_i-\sigma(\hat{Y}^{(i)}))\hat{dist}(T_i,S_k)+\frac{2\lambda}{m}\boldsymbol{W}_k \tag{19}$$

$$\frac{\partial\mathscr{F}_i}{\partial\boldsymbol{W}_0}=-(Y_i-\sigma(\hat{Y}^{(i)})) \tag{20}$$

在导出了目标函数相对于 shapelets 和权重的梯度之后，学习算法根据每个训练实例的分类目标在导数的负方向上更新 shapelets 和权重的值。

需要说明的是，式(9)是 S 和 W 的非凸函数，如果初始化开始围绕全局最优所处的区域进行学习，那么梯度可以将参数更新到最优位置。因此，本文使用 4.2 节中发现的 shapelets 候选来对 shapelets 进行初始化，以实现较高的分类精度，$\boldsymbol{W}$ 在 0 附近随机初始化。

5 实验与分析

本章通过对时间序列数据进行分类来验证本文算法 SRMSE 的有效性。首先介绍相关的实验设置，如实验所用的数据集、选择的实验对照组；然后对覆盖度 θ 的选择进行验证；最后给出实验结果，并对其进行分析和讨论。

本文的实验环境为 Java，64 位操作系统的 Windows 10 中文版，Intel i7 CPU 和 16 GB 的内存。

5.1 数据集

本文选取了 UCR 时间序列数据库中的 32 个一元时间序列数据集作为研究对象，这些数据集在基于 shapelets 的时间序列分类研究中被广泛使用，覆盖了人体心电图、电力、图像轮廓信息等多个领域，具有不同长度，含有多个类别标签。数据集的具体信息如表 1 所列。其中，正则项系数 λ 在{0.01，0.1，1}范围内搜索，迭代次数 I 在区间[500，5000]内按步长 500 进行搜索，λ 和 I 的值由分类准确率确定。

表 1 实验使用的数据集

Tabe 1 Datasets used in experiments

数据集	训练数	测试数	类别	长度	λ	I
ArrowHead	36	175	3	251	0.10	2500
Beef	30	30	5	470	0.01	2000
Beetle/Fly	20	20	2	512	0.01	500
CBF	30	900	3	128	0.01	500
ChlorineConcentration	467	3840	3	166	0.01	3000
Coffee	28	28	2	286	0.01	500
Computers	250	250	2	72	0.10	1000
DiatomSizeReduction	16	306	4	345	0.01	1000
DistalPhalanxOutlineCorrect	276	600	2	80	0.01	2000
Earthquakes	139	322	2	512	1.00	500
ECG200	100	100	2	96	0.10	2500
ECG5000	500	4500	5	140	0.10	1000
ECGFiveDays	23	861	2	136	0.01	500
FaceAll	560	1690	14	131	0.01	1000
FaceFour	24	88	4	350	1.00	500
FacesUCR	200	205	14	131	1.00	1500
Gunpoint	50	150	2	150	0.10	2500
Ham	109	105	2	431	0.1	500
HandOutlines	370	1000	2	2709	1.00	1000
InsectWingbeatSound	220	1980	11	256	0.01	2000
Mallat	55	2345	8	1024	1.00	1000
Meat	60	60	3	448	0.10	500
ShapeletSim	20	180	2	500	0.10	2000
SonyAIBORobotSurface1	20	601	2	70	0.01	3500
SonyAIBORobotSurface2	27	953	2	65	0.01	1500
Strawberry	370	613	2	235	0.10	4000
Symbols	25	995	6	398	0.10	500
SyntheticControl	300	300	6	60	0.10	500
ToeSegmentation1	40	228	2	227	0.10	500
TwoLeadECG	23	1139	2	82	0.10	5000
Wafer	1000	6164	2	152	0.10	1000
WormsTwoClass	77	181	2	900	1.00	500

5.2 覆盖度 θ 的验证

覆盖度 θ 值决定了 3.2 节中数据集的每一类得到的 shapelets 候选数量，其值越大，得到的 shapelets 候选的数量越多，涵盖的时间序列特征信息就越多，但进行分类所需要的时间也越长。覆盖度 θ 的取值范围影响着分类的准确率和效率，因此，本节设计了一组实验来验证我们所取 θ 值范围的合理性。

5.2.1 覆盖度对精度的影响

为了研究覆盖度 θ 值的合适取值范围，本文在所有数据集上进行了实验，并从中选取了两个规模较大的数据集（SonyAIBORobotSurface2，ECG5000）和两个规模较小的数据集（SyntheticCotrol，DiatomSizeReduction），来说明覆盖度与精度之间的关系，实验结果如图 5 所示。

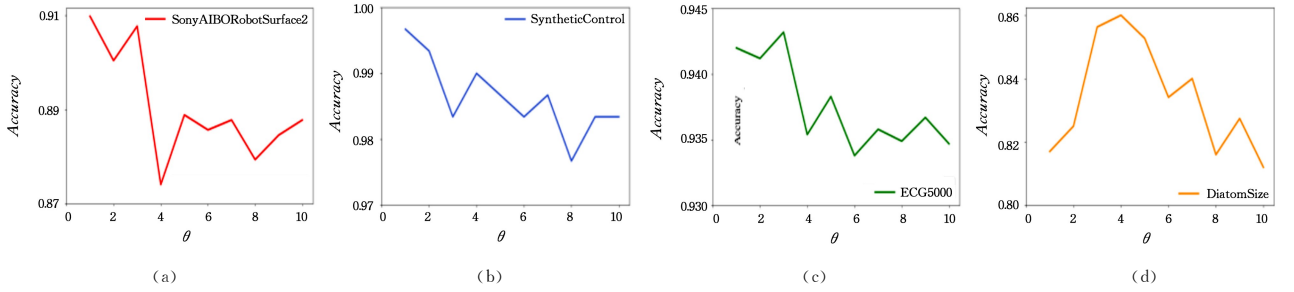

图 5 覆盖度 θ 对精度的影响

Fig. 5 Effect of coverage θ on accuracy

在图 5(a)中，数据集 SonyAIBORobotSurface2 的分类精度在 θ 取值为 1 时达到最大。同样地，在图 5(b)－5(d)中，数据集的分类精度分别在 θ 取值为 1，3 和 4 时达到最大。从整体来看，覆盖度 θ 的值在 1～5 时，数据集的分类精度已经达到了最优。随着覆盖度 θ 值的增大，分类精度并没有呈现持续提高趋势。因此，本文将覆盖度 θ 值的范围设置为 1～5。

5.2.2 覆盖度 θ 对效率的影响

图 6 显示了覆盖度 θ 对上述数据集运行时间的影响。从图中可以看出，4 个数据集的运行时间均在覆盖度 θ 取值为 1 时最短，因为 θ 为 1 时所得到的 shapelets 候选数量最少。在图 6(a)中，数据集 SonyAIBORobotSurface2 的运行时间在 θ 取 2～6 时较长，在 6 时达到最长，但其最大值和最小值之间差异很小。图 6(b)中，数据集 SyntheticCotrol 的运行时间在 θ 取 1～5 时逐渐增加，随后趋于稳定。在图 6(c)中，数据集的运行时间在 θ 取值 4 和 7 时较长，但整体时间较为稳定。图 6(d)中，数据集的运行时间在 θ 为 4 时达到了最大值，但同时其分类精度也达到了最优，说明以牺牲较小的时间得到更高的精度是值得的。综合来看，覆盖度 θ 取值为 1～5 时，运行时间在可接受范围内。故本文 θ 的取值范围是合理的。

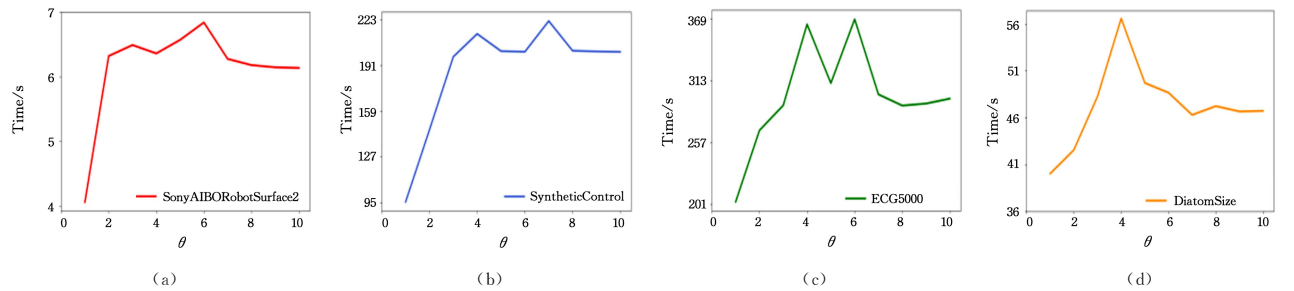

图 6 覆盖度 θ 对效率的影响

Fig. 6 Effect of coverage θ on efficiency

5.3 实验对照组

本文从基于 shapelets 的时间序列分类方法中选取了 7 种方法作为对照组，其中包括基于学习的方法 LTS[11]、基于搜索的方法 FSH[14] 和 SD[8]、基于符号的方法 ELIS＋＋[30] 和 BSPCOVER[16]、一种集成方法 COTE[31] 和一个神经网络方法 ResNet[32]。介绍如下：

(1)FSH[14]：将时间序列转换成 SAX 词，并对其使用随机投影技术来选择 shapelets。

(2)LTS[11]：使用逻辑回归和梯度下降学习最优 shapelets 的算法。

(3)COTE[31]：将多种分类器进行结合对时间序列进行分类。

(4)SD[8]：利用在线聚类和剪枝筛选技术得到最优 shapelets 的方法。

(5)ResNet[32]：运用深度神经网络以对时间序列进行分类。

(6)ELIS＋＋[30]：将时间序列转化为 PAA 词，并使用 TFIDF 分数来度量 shapelets 的质量。

(7)BSPCOVER[16]：通过 Bloom 过滤器和相似性修剪以提高效率的 shapelets 发现方法。

5.4 实验结果分析

5.4.1 分类准确率对比

表 2 列出了本文方法 SR-MSE 与对照组的 7 种方法在 32 个数据集上的分类准确率，每个数据集的最优准确率加粗表示。ELIS＋＋在 HandOutlines 数据集上没有取得结果，

因为运行时间过长。从各算法取得的最优分类准确率次数来看,SR-MSE 排名第三,仅次于 COTE 和 ResNet,在 8 个数据集上实现了最高的分类准确率,在 19 个数据集上分类准确率排名前三。但从一比一的获胜数来看,SR-MSE 略优于 ResNet,在 16 个数据集上获得了更高的分类准确率。LTS 和 BSPCOVER 在 32 个数据集上取得最优分类准确率的次数都为 4,ELIS++在两个数据集上实现了最优的分类准确率,其平均秩为 4.74,优于 FSH 和 SD。

表 2 准确率对比

Table 2 Accuracy comparison

(%)

数据集	FSH	SD	LTS	COTE	ResNet	ELIS++	BSPCOVER	SR-MSE
ArrowHead	59.73	65.70	85.14	81.78	**85.27**	81.92	80.64	82.29
Beef	56.67	50.34	78.50	**87.46**	75.30	66.70	73.33	70.00
Beetle/Fly	70.00	76.25	85.00	80.00	86.38	**100.00**	90.00	90.00
CBF	92.80	97.00	99.11	99.65	99.50	93.14	99.65	**99.67**
Chlorine	55.60	54.36	73.02	72.40	**84.40**	61.95	62.13	64.17
Coffee	93.00	96.10	**100.00**	**100.00**	**100.00**	96.40	**100.00**	**100.00**
Computers	51.74	55.69	58.40	74.00	**80.50**	55.69	66.48	68.00
Diatom	85.56	83.71	**95.10**	92.81	35.97	90.27	87.25	86.00
DistalPhalanx	72.30	71.00	78.35	78.49	79.63	77.20	82.74	**84.00**
Earthquakes	65.93	63.60	74.10	76.75	73.62	76.81	81.68	**83.17**
ECG200	79.57	78.31	88.00	88.00	89.76	90.64	**92.00**	91.00
ECG5000	92.25	90.65	93.41	**94.60**	93.40	78.93	94.44	94.20
ECGFiveDays	99.30	96.48	**100.00**	99.94	96.70	99.86	**100.00**	**100.00**
FaceAll	60.58	70.67	72.92	**90.64**	81.92	74.50	76.00	74.81
FaceFour	89.72	80.30	96.31	92.10	95.50	95.46	96.32	**97.80**
FacesUCR	67.56	83.25	93.90	**94.23**	93.37	70.96	78.68	83.27
Gunpoint	95.82	90.89	**100.00**	**100.00**	97.43	99.30	**100.00**	**100.00**
Ham	64.74	58.47	66.67	68.95	74.92	62.75	75.84	**77.20**
HandOutlines	81.14	76.48	50.06	**92.00**	90.80	—	86.56	85.60
InsectWing.	47.43	44.79	61.10	**65.58**	51.42	53.29	57.26	54.90
Mallat	95.92	89.85	95.01	96.40	**97.20**	78.64	76.58	75.82
Meat	80.36	92.75	73.92	91.67	**96.80**	63.28	75.00	78.34
ShapeletSim	**99.74**	69.34	95.00	96.93	78.53	98.17	83.69	80.56
SonyAIBORobot1	67.16	85.00	81.08	86.27	**94.21**	90.26	88.40	93.60
SonyAIBORobot2	77.30	77.62	87.90	95.17	**95.45**	89.40	92.70	90.56
Strawberry	88.36	86.27	91.08	**95.14**	91.30	81.36	93.30	94.46
Symbols	93.74	92.41	93.17	**96.38**	90.60	93.00	92.82	94.00
SyntheticControl	92.56	95.47	99.67	**100.00**	99.71	96.63	99.58	99.67
ToeSegmentation1	92.82	86.95	93.42	96.61	94.30	**97.40**	96.49	95.18
TwoLeadECG	90.40	88.71	99.60	99.33	**100.00**	99.50	99.65	99.65
Wafer	95.86	98.23	99.61	**100.00**	99.84	99.43	99.33	99.79
WormsTwoClass	70.57	65.27	72.13	**76.91**	73.31	70.50	74.59	72.38
Total best acc	1	0	4	13	9	2	4	8
Ours 1-to-1Wins	29	30	23	11	16	26	16	—
Ours 1-to-1 Draws	0	0	2	2	1	0	6	—
Ours 1-to-1 Losses	3	2	7	19	15	5	10	—
Rank Mean	6.09	6.56	3.91	2.47	3.22	4.74	3.34	3

与 BSPCOVER 相比,SR-MSE 在 16 个数据集上实现了更高的分类准确率,在 6 个数据集上的分类准确率相同,在 10 个数据集上的分类准确率略低。ELIS++旨在提高分类效率,在 26 个数据集上的分类准确率都低于 SR-MSE,在 5 个数据集上的分类准确率较优。根据平均秩的排名,在本文选取的 32 个数据集中,分类精度由高到低依次为 COTE,SR-MSE,ResNet,BSPCOVER,LTS,ELIS++,FSH,SD。虽然 FSH 和 SD 的分类精度较低,但它们实现了较高的分类效率。

为了更清楚地表明本文算法 SR-MSE 相比其他 7 种算法的差异显著性,我们将其平均排名用临界差图展示,如图 7 所示。从图中可以看出,SR-MSE 的准确率虽然低于最优的 COTE 算法,但两者之间的差异并不明显,与 BSPCOVER 算法和 ResNet 算法的差异也较小,但与其他 4 种算法的差异较为显著。

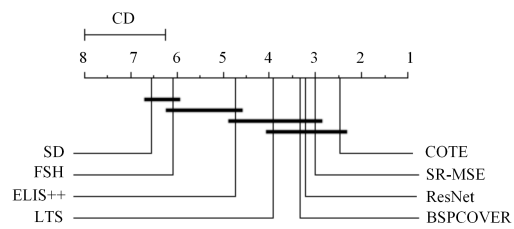

图 7 本文算法和对比算法的临界差图

Fig. 7 Critical difference diagram ofour algorithm and comparison algorithms

综合上述分析,可以看出本文方法 SR-MSE 实现了较好

的分类准确率，优于所对比的绝大多数算法。

5.4.2 Shapelets 的可解释性分析

Shapelets 的优势在于其具有良好的可解释性，本小节选取了两个数据集 TwoLeadECG 和 ECGFiveDays 进行可视化分析，从而验证所提方法 SR-MSE 的可解释性。

(1)TwoLeadECG Shapelet 的可解释性

图 8 中绿色线段表示第 1 类时间序列的 shapelet。图 8(a)和图 8(b)中的蓝色时间序列属于第 1 类时间序列；图 8(c)和图 8(d)中的红色时间序列属于第 2 类时间序列。从肉眼很难分辨这两类时间序列之间的差异，但类别 1 的 shapelet 可以将这两类时间序列很好的区别开来。对比 shapelet 和这两类时间序列可以看出，属于第 2 类时间序列的波谷值略低于第 1 类时间序列，而且第 1 类时间序列的波谷底部要比第 2 类时间序列的波谷底部更尖。

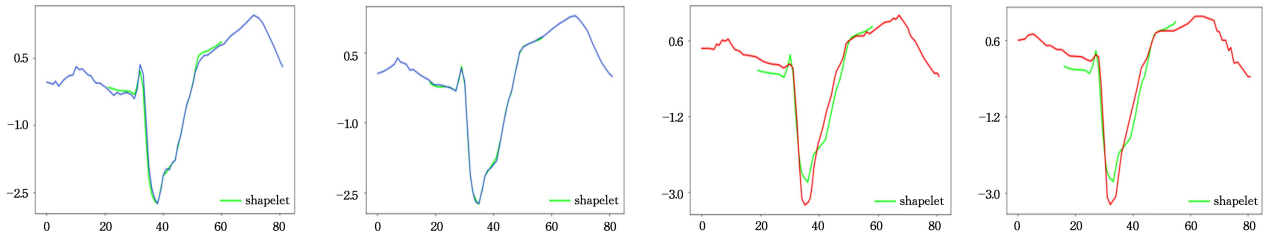

图 8 TwoLeadECG 的可视化图(电子版为彩图)

Fig. 8 Visualization of TwoLeadECG

(2)ECGFiveDays Shapelet 的可解释性

图 9 中绿色线段表示第 2 类时间序列的 shapelet。图 9(a)和图 9(b)中的红色时间序列表示两条异常心电图，属于第 1 类时间序列；图 9(c)和图 9(d)中的蓝色时间序列表示两条正常心电图，属于第 2 类时间序列。从图中可以看出，类别 2 的 shapelet 可以明显地将这两类时间序列区分开来，属于第 2 类时间序列的波谷值明显低于第 1 类时间序列的波谷值，也就是正常心电图的 T 波波谷值要低于异常心电图。

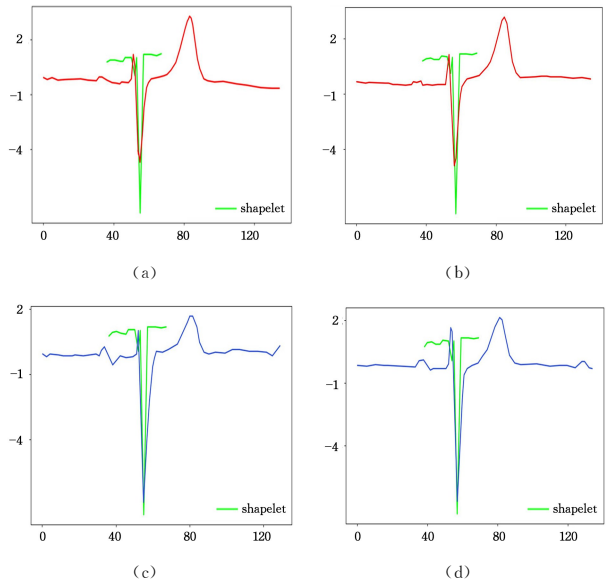

图 9 ECGFiveDays 的可视化图

Fig. 9 Visualization of ECGFiveDays

5.4.3 运行时间对比

本文方法 SR-MSE 旨在在保证分类精度的同时提高分类效率。根据前人的研究，FSH 在 SAX 上采用了随机投影技术，是一种高效的时间序列分类方法，这种方法比对照组中的 COTE 和 LTS 快很多。因此，本小节将 SR-MSE 与 FSH，ELIS++和 BSPCOVER 这几种方法进行了运行时间对比，以验证其效率。运行效率结果如表 3 所列。

表 3 运行时间对比

Table 3 Running time comparison

(s)

数据集	FSH	ELIS++	BSPCOVER	SR-MSE
ArrowHead	7.68	315.89	55.57	60.84
Beef	57.23	204.25	137.27	100.04
Beetle/Fly	12.61	32.41	41.09	6.89
CBF	2.15	20.83	16.43	6.71
Chlorine.	139.60	10 400	194.62	204.17
Coffee	4.39	10.95	12.30	7.78
Computers	405.26	16 920	1067.41	1319.08
Diatom	4.33	130.67	35.76	21.85
DistalPhalanx.	12.84	362.83	58.13	54.80
Earthquakes	1172.62	1145.37	2957.36	357.91
ECG200	2.96	339.25	45.60	82.44
ECG5000	37.53	2880.74	540.29	210.07
ECGFiveDays	0.94	6.13	1.38	1.12
FaceAll	119.86	4345.97	1477.36	577.69
FaceFour	15.47	110.35	35.75	33.81
FacesUCR	49.13	4 986.30	1462.67	2532.37
Gunpoint	1.69	48.68	8.40	20.38
Ham	200.94	264.47	146.71	134.44
HandOutlines	40 680	—	4 320	12 240
InsectWing.	135.62	8756.38	651.38	457.88
Mallat	445.73	3946.62	2764.34	1882.38
Meat	33.52	280.74	40.73	6.92
ShapeletSim	20.64	8.73	405.28	53.44
SonyAIBORobot1	0.85	7.74	5.82	6.69
SonyAIBORobot2	0.91	42.96	5.26	4.77
Strawberry	124.83	3006.24	260.94	229.48
Symbols	16.17	135.74	87.56	41.72
SyntheticControl	10.21	883.50	251.74	95.43
ToeSegmentation1	7.75	175.48	23.06	18.09
TwoLeadECG	0.90	9.27	20.32	8.06
Wafer	78.31	1013.62	807.37	340.76
WormsTwoClass	2216.35	2118.45	1135.64	781.77
RankMean	1.34	3.81	2.94	2.19

从表 3 的整体数据及平均秩来看，SR-MSE 的运行时间排名第二。运行效率最高的算法是 FSH，ELIS++算法的运行效率最低。虽然 FSH 的运行效率高于 SR-MSE，但从表 2

的结果可知,这是以牺牲精度为代价取得的。与 ELIS++相比,SR-MSE 在 32 个数据集中的 25 个上运行效率提高了 1 倍以上,只有在 ShapeletSim 数据集上的运行效率较低。因为 ELIS++运用的符号质量度量方法(TFIDF 分数)在大型数据集中计算耗时,而本文算法 SR-MSE 是基于哈希表来度量符号的质量,大大缩短了时间。与分类精度处于同一水平的 BSPCOVER 相比,SR-MSE 在 25 个数据集上提高了分类效率。

综合分类准确度和运行时间分析可知,本文方法 SR-MSE 能在保证分类精度的同时提高分类效率。由此表明,本文提出的方法 SR-MSE 是有效的。

结束语 本文针对基于符号表示的 shapelets 提取方法中存在的问题进行研究,提出了基于符号表示的可度量 shapelets 提取方法 SR-MSE。该方法在 SAX 空间中采用 Bloom 过滤器减少了候选的数量,并对 SAX 词的质量提出了基于哈希表的度量方法。此外,还判别了 SAX 词的相似性,基于相似性、覆盖度等概念确定了最终的 shapelets 候选集。最后,运用逻辑回归模型学习得到真正的 shapelets。在 UCR 数据库中的 32 个数据集上进行的实验和分析结果表明,相较于 7 种对比方法,SR-MSE 能在保证分类精度的同时提高分类效率,同时具有良好的可解释性。未来的工作考虑对学习模型的目标函数求解方法进行优化,以实现更好的分类效果。

参 考 文 献

[1] AHMED T,SINGH D. Probability density functions based classification of MODIS NDVI time series data and monitoring of vegetation growth cycle[J]. Advances in Space Research,2020,66(4):873-886.

[2] AL-HADEETH I,ABDULLA S,DIYKH M,et al. Adaptive boost LS-SVM classification approach for time-series signal classification in epileptic seizure diagnosis applications[J]. Expert Systems with Applications,2020,161:113676.

[3] YEUNG J,WEI Z,CHAN K Y,et al. Jump detection in financial time series using machine learning algorithms[J]. Soft Computing,2020,24(3):1789-1801.

[4] BAGNALL A,LINES J,BOSTROMA,et al. The great time series classification bake off:a review and experimental evaluation of recent algorithmic advances[J]. Data Mining and Knowledge Discovery,2017,31(3):606-660.

[5] XI X P,KEOGH E,SHELTONC,et al. Fast time series classification using numerosity reduction[C]// Proceedings of the 23rd International Conference on Machine learning. New York:ACM,2006:1033-1040.

[6] SAKOE H,CHIBA S. Dynamic programming algorithm optimization for spoken word recognition[J]. IEEE Transactions on Acoustics Speech and Signal Processing,1978,26(1):43-49.

[7] YE L X,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. New York:ACM,2009:947-956.

[8] GRABOCKA,WISTUBA M. Schmidt-Thieme. Fast classification of univariate and multivariate time series through shapelet discovery[J]. Knowledge and Information Systems,2016,49(2):429-454.

[9] JI C,ZHAO C,LIUS J,et al. A fast shapelet selection algorithm for time series classification[J]. Computer Networks,2019,148:231-240.

[10] ZOU X N,ZHENG X W,JI C,et al. An improved fast shapelet selection algorithm and its application to pervasive EEG[J]. Personal Ubiquitous Comput,2022,26(4):941-953.

[11] GRABOCKA J,SCHILLING N,WISTUBA M,et al. Learning time-series shapelets[C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM,2014:392-401.

[12] HOU L,KWOK J T,ZURADA J M. Efficient learning of time series shapelets[C]// Proceedings of the 13th AAAI Conference on Artificial Intelligence. Phoenix: AAAI Press, 2016: 1209-1215.

[13] ZHANG Z,ZHANG H,WEN Y,et al. Discriminative extraction of features from time series[J]. Neurocomputing, 2018, 275:2317-2328.

[14] RAKTHANMANON T,KEOGH E. Fast shapelets:A scalable algorithm for discovering time series shapelets[C]// Proceedings of the 2013 SIAM International Conference on Data Mining. Philadelphia:SIAM,2013:668-676.

[15] FANG Z,WANG P,WANG W. Efficient learning interpretable shapelets for accurate time series classification[C]// 2018 IEEE 34th International Conference on Data Engineering. Paris: IEEE,2018:497-508.

[16] LI G Z,BYRON C,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.

[17] KRUSKALW H. A nonparametric test for the several sample problem[J]. Annals of Mathematical Statistics, 1952, 23(4):525-540.

[18] 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.

[19] FAWAZ H I,FORESTIER G,WEBER J,et al. Deep learning for time series classification: A review[J]. Data Mining and Knowledge Discovery,2019,33(4):917-963.

[20] 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. New York:ACM,2011:1154-1162.

[21] YUAN J D,WANG Z H,HAN M. et al. A logical shapelets transformation for time series classification[J]. Chinese Journal of Computers,2015,38:1448-1459.

[22] LINES J,DAVIS L M,HILLS J,et al. A shapelet transform for time series classification[C]// Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Beijing:ACM,2012:289-297.

[23] ZHAO H Y,PAN Z S,TAO W. Regularized shapelet learning for scalable time series classification[J]. Compute Networks, 2020,173:107171.

[24] LIN J, KEOGH E, WEI L, et al. Experiencing sax: A novel symbolic representation of time series[J]. Data Mining and Knowledge Discovery, 2007, 15(2): 107-144.
[25] SCHAFER P, LESER U. Fast and accurate time series classification with weasel[C]// Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. Singapore: ACM, 2017: 637-646.
[26] SENIN P, MALINCHIK S. Sax-VSM: Interpretable time series classification using sax and vector space model[C]// IEEE 13th International Conference on Data Engineering. Dallas: IEEE, 2013: 1175-1180.
[27] NGUYEN T L, GSPONER S, IFRIM G. Time series classification by sequence learning in all-subsequence space[C]// 2017 IEEE 33rd International Conference on Data Engineering. San Diego: IEEE, 2017: 947-958.
[28] IFRIM G, WIUF C. Bounded coordinate-descent for biological sequence classification in high dimensional predictor space[C]// Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego: ACM, 2011: 708-716.
[29] LIANG Z Y, WANG H Z. Efficient Class-Specific shapelets learning for interpretable time series classification[J]. Information Sciences, 2021, 570: 428-450.
[30] ZHANG H, WANG P, FANGZ, et al. ELIS++: a shapelet learning approach for accurate and efficient time series classification[J]. World Wide Web, 24(2): 511-539.
[31] BAGNALL A, LINES J, HILLS J, et al. Time-series classification with COTE: The collective of transformation-based ensembles[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(2): 2522-2535.
[32] FAWAZ H I, FORESTIER G, WEBERJ, et al. Deep learning for time series classification: A review[J]. Data Mining and Knowledge Discovery, 2019, 33(4): 917-963.
[33] LARSEN R J, MARX M L. An Introduction to Mathematical Statistics and its Applications[M]. London: Prentice Hall, 2011.

WANG Liqin, born in 1998, postgraduate. Her main research interests include data mining, machine learning and pattern recognition.

WAN Yuan, born in 1976, Ph.D, professor. Her main research interests include data mining, pattern recognition, manifold learning, machine learning and feature selection.

（责任编辑：何杨）