计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(11):138−146 [doi: 10.15888/j.cnki.csa.007113]

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

基于微局部特征的时序数据二分类算法①

舒伟博

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)
通讯作者: 舒伟博, E-mail: weiboshu@mail.ustc.edu.cn

摘　要: 在诸多时序数据分类算法中, 有一类算法借助时序数据的局部特征对时序数据进行分类, 它们取得了不错的分类结果, 然而其时间复杂度以及分类精度依旧存在可见的提升空间. 本文提出的微局部特征二分类算法, 着眼于局部特征本身的性质, 对局部特征集进行限制, 进而改进现有的基于局部特征的分类算法. 新算法通过理论分析支撑, 将经典算法的局部特征集大幅缩小, 进而显著提升了分类算法的时间性能. 另一方面通过重定义局部特征的评价标准, 新算法选出性质更为优良的局部特征, 提升了分类精度.

关键词: 时序数据分类; 特征选择; 有监督学习; 机器学习; 人工智能

引用格式: 舒伟博.基于微局部特征的时序数据二分类算法.计算机系统应用,2019,28(11):138–146. http://www.c-s-a.org.cn/1003-3254/7113.html

Time Series Binary Classification Based on Mini Local Features

SHU Wei-Bo

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Among all kinds of time series classification algorithm, algorithms based on local features of time series data have achieved reasonable results. However, there is still abundant space for improvements of them in time complexity and accuracy. In this study, we propose an improved algorithm based on local features. It focuses on the property of local features and put restrictions on the set of local features. On the one hand, supported by theoretical analysis, our new algorithm cuts the size of set of local features and consequently reduces the time and space complexity. On the other hand, we redefine the criteria of selecting local features so that we can select more discriminative local features.

Key words: time series classification; feature selection; supervised learning; machine learning; artificial intelligence

1　引言

时间序列是一种重要且特殊的高维数据, 它的各个维度之间存在固定的先后次序, 这些次序中隐藏大量的有利于分类的特征信息. 在现实生活中, 时间序列有着广泛的应用. 例如天气预报中的气象数据、对外贸易中浮动的货币汇率、医疗器械捕获的电波图像, 工程应用中的连续信号等, 这些数据都可以看成是时间序列.

时间序列分类是时序数据分析中的主要任务之一. 当前的时序数据分类算法可大致分为两类, 一类是将整个时间序列看成一个整体, 即距离空间中的一个点, 通过构造合适的距离度量方式, 在距离空间中寻找分类边界.

第二类是采用滑动窗口的方式捕捉时间序列的子序列, 即所谓的“捕捉局部特征”. 在通过某些方式选择具有良好分类性能的局部特征后, 通过这些局部特征来构造各式各样的分类器, 进而完成分类.

通过一些综合的比较, 基于局部特征的分类方法整体性能优于第一类算法, 尤其体现在更好的分类性能上[1]. 因而基于局部特征的分类方法也是现在研究的主要方向.

① 收稿时间: 2019-03-25; 修改时间: 2019-04-18; 采用时间: 2019-04-23; csa 在线出版时间: 2019-11-06

本文针对现阶段在基于局部特征进行时序数据分类的研究领域内存在的一些问题, 设计了一个新的算法, 该算法着重解决现阶段存在于该领域内的如下两个问题:

(1) 基于局部特征的分类算法在分类精度上依旧存在可以提升的空间. 该问题尤其体现在一些二分类问题上.

(2) 基于局部特征的分类算法在现阶段存在太多的冗余局部特征, 使得时间复杂度相对较高.

针对这两个问题, 本文提出的算法分别采用如下的策略进行改进:

(1) 本算法针对二分类问题, 采用一种新的指标来评价局部特征, 使得原数据在转换到特征空间后, 具备更高的线性可分性.

(2) 本算法在选择局部特征前, 首先抛弃大量局部特征, 仅保留长度非常短的局部特征, 使得局部特征数量大幅减小, 因而在评估并选择局部特征时, 时间会被大幅减少.

对于这两处创新, 本文也会给出理论依据以证明其可行性, 同时, 实验结果也证明了其带来的时间优势和分类性能上的优势.

2 相关工作

2.1 几个术语的定义

定义 1. 时间序列及其分类器: 时序数据的一个样本点是一个序对 (x, y), x 是一个 m 维的有序观测值 $(x_1, x_2, \cdots, x_m)$, y 为该样本点的类标, 在不需要讨论其类别信息时, 我们也会在文中将其简写为 x. 整个样本集表示为 $T=(X, Y)=((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n))$, 在不需要讨论其类别信息时, 我们也会在文中将其简写为 X.

定义 2. 局部特征 (shapelet): 局部特征又叫shapelet, 本质上是时间序列的连续子序列, 其具有一定的判别性能[2]. 所以一个局部特征由一段连续子序列以及一个类标组成, 该类标即为该子序列的父序列 (即某一原始时间序列) 的类标. 我们在文中用 $(\boldsymbol{s}, z)$ 来表示一个局部特征, 其中 $\boldsymbol{s}$ 表示其代表的连续子序列, z 为其类标, 在不需要讨论其类别信息时, 我们也会在文中直接用 $\boldsymbol{s}$ 指代. 为了尊重提出 shapelet 的原作者, 从此处开始, 下文中的“局部特征”皆用“shapelet”替代.

2.2 基于 shapelet 的时序数据分类算法

一个 m 维的数据, 它的 shapelet 的长度可以是 1 到 m, 所以其一共产生 $m(m+1)/2$ 个 shapelet. 如果数据集里面有 n 个时间序列, 那整个数据集一共拥有 $nm(m+1)/2$ 个 shapelet. Ye LX 和 Keogh E 首次提出用 shapelet 进行时间序列分类. 其使用方法是选择具有判别性的 shapelet 来构造一棵决策树[2]. 该决策树的判定节点为一个 shapelet, 而属性即为时间序列与该判定节点中的 shapelet 的距离, 通过距离所处的区间来将时间序列选择递交给某下一个节点处理[2].

因为 shapelet 是从原始数据上截取下来的子序列, 所以与原始数据并不等长, 这里特别说明一下如何计算 shapelet 与原始数据之间的距离.

设原始数据的一个样本点 $x=(x_1, x_2, \cdots, x_m)$, 某个 shapelet $s=(s_1, s_2, \cdots, s_j)$, 其中 $j \le m$, 那么它们之间的距离用如下函数 $D(.,.)$ 计算:

$$Dist(s,x) = \min\left\{d(s,P)\middle| \begin{matrix} P=(x_i,\cdots,x_{i+j-1}), \\ i=1,\cdots,m-j+1 \end{matrix}\right\} \quad (1)$$

其中, $d(\cdot,\cdot)$ 是欧式距离度量函数.

该算法较好地利用了数据的局部特征进行分类, 具有不错的分类精度和可扩展性, 但其缺点在于时间复杂性高. 算法拥有 $O(n^2m^4)$ 的计算复杂性, n 是数据个数, m 是数据维数, 即时间序列数据的长度[2].

针对该方法时间复杂度高的问题, 领域内的研究者提出了种种解决方案[3–6], 这里介绍其中几种比较有代表性的方案. 该算法时间复杂度高的第一大原因在于候选 shapelet 太多了, 对每一个 shapelet 进行评估的单位时间, 被候选 shapelet 的数量极大规模地放大. 所以有一类叫做“快速 shapelet”的方法针对候选 shapelet 的数量进行改进, 它们牺牲掉一些 shapelet 的覆盖率, 大幅减少 shapelet 的候选数量, 换取运行速率上的提升, 但是精度上也有明显退化[7]. 而另一类叫做“learning shapelet”的算法, 它并不从原始数据直接获取 shapelet, 而是通过学习的方式习得最佳的 shapelet. 该 shapelet 可能与原始数据中的任意一个数据段都不匹配, 因为它是通过 feedback 的学习方式创造出的用于分类的 shapelet. 该方法开辟了另一条道路, 然而其也存在过拟合的问题, 时至今日尚无可观的突破[8,9].

2.3 Shapelet 转换算法与 shapelet 集成算法

Ye LX 和 Keogh E 使用 shapelet 构造决策树, 该算法时间复杂度高的另一个原因在于, shapelet 的评估需要在每个节点的特征选择阶段进行一次. 这是因为在构造完一个节点之后, 数据集会被该节点划分为若干

新子集, 而新子数据集与原来的数据集不相同了, 所以 shapelet 的判别性能需要在新子集上做重新的评估. 同样地, 每一轮评估的时间被评估的轮数——即决策树上的节点数——放大后, 时间复杂度变得相当高.

于是有一类叫做“shapelet 转换”的方法, 它通过解决该问题来降低时间复杂度. 该类方法通过一次性选取若干 shapelet 来构建分类器, 所以只需要对所有的 shapelet 进行一轮评估, 在评估并选取了合适的 shapelet 之后, 则基于它们将原始数据转换到一个特征空间中, 转换的方式是通过计算原始数据与第 i 个 shapelet 的距离来作为原始数据在特征空间中的第 i 个坐标[10]. 当这些原始时序数据被转换到特征空间中之后, 它再使用 kNN 算法来完成分类[10]. 该算法有效地避免了多轮 shapelet 评估, 但是其时间开销依然不容小觑, 主要原因还是上节中提到的候选的 shapelet 数量太多. 如果要减少候选 shapelet 的数量, 势必会引起精度的下降, 但是其相较于原来经典的 shapelet 算法已有了革命性的突破, 使得后续的研究者工作开始围绕如何快速地找到用来构建特征空间的若干 shapelet 来进行 [11].

“shapelet 转换”的方法会有不稳定的特性, 因为数据在被转换到特征空间之后, 很难准确预料它们的分布特性, 如果使用单一分类器, 必须承担错误估计它们分布特性的风险, 所以很多时候特征空间中的单一分类器会带来低分类性能的后果. 为了解决这一问题, 有人提出了“集成 shapelet 转换”的方法, 即在特征空间中集成若干经典分类器来完成分类[12]. 该方法能让分类精度得到大幅提升, 但是训练集成分类器带来的时间开销也是很可观的.

3 算法理论分析

该节对当前已有的基于局部特征的分类算法上存在的缺陷进行分析, 探讨如何能够有效地改进它们.

3.1 时间序列与 shapelet 的距离计算

从式 (1) 可以看到, 计算一个 shapelet 与一个时间序列的距离需要计算该 shapelet 与该时间序列的所有与该 shapelet 等长的连续子序列的距离, 然后选择其中最小的距离. 如果假设一个时间序列的长度为 m, 一个 shapelet 的长度为 k, 那么计算它们之间的距离需要 $O(k(m-k))$ 的时间复杂度, 当 $m>>k$ 时, 该复杂度接近 $O(km)$.

显而易见, 该距离度量方式的时间复杂度比较高, 而且此种度量方式只关心该 shapelet 代表的局部特征是否明显出现在与其计算距离的时间序列中, 而并不关心其出现的位置, 这是因为最终的距离是取所有子序列与该 shapelet 距离中的最小值. 而对于不同的时间序列, 取得最小距离的子序列的位置并不是固定的. 由于 shapelet 本身来自于时间序列, 所以其本身就携带了位置信息, 所以我们认为这种忽略位置信息的距离计算方式存在一定缺陷.

针对以上问题, 我们设计了固定位置的距离度量方式, 如下式所示:

$$D(s,x)=\left\{d(s,P)\middle|\begin{array}{c}P=(x_i,\cdots,x_{i+j-1}),\\ i=\text{start position of } s\end{array}\right\} \tag{2}$$

其中, x, s, j 和 $d(\cdot,\cdot)$ 的含义同式 (1), 由于 s 本身是某个时间序列的子序列, 所以它拥有自己在原始时间序列数据中的起始位置, 所以式 (2) 所表达的即是用 x 中与 s 位置对齐的子序列与 s 计算欧氏距离来作为 x 与 s 的距离. 我们把这个距离称作“定点距离”.

式 (2) 的计算加入了位置信息, 既关注该时间序列是否具备 shapelet 所代表的局部特征, 还关注了其是否在对应的位置上与该局部特征有很好的近似性, 同时, 其时间开销从 $O(km)$ 降低至 $O(k)$, 其中 k 是 shapelet 的长度, m 是原始时间序列的长度. 所以, 式 (2) 无论从最后的预测精度这一角度还是从时间开销这一角度来说, 都要优于式 (1).

考虑到时间序列数据经常会发生迟滞, 噪音等情况, 我们对式 (2) 加入适当的松弛, 得到如下式 (3) 的距离计算公式:

$$D(s,x)=min\left\{d(s,P)\middle|\begin{array}{c}P=(x_{i_1},x_{i_2}\cdots,x_{i_k}),\\ i=\text{start position of } s,\\ i_1<i_2<\cdots<i_j<i+j+r,\\ -l\le i_1-i\le r\end{array}\right\} \tag{3}$$

上式中的 x, s, m, j 和 $d(\cdot,\cdot)$ 的含义同式 (2), 而 l 是左松弛因子, r 是右松弛因子, 都是超参数. 我们把这个距离称作“定点浮动距离”.

3.2 固定候选 shapelet 集中 shapelet 的长度

基于 shapelet 的算法因候选的 shapelet 数量太大而具有很高的时间复杂度, 而之所以需要如此庞大的候选 shapelet 集, 是为了保证局部特征的全覆盖, 因为你无法判定理想的局部特征所对应的 shapelet 的长度应该是多少, 所以只能选取所有长度的 shapelet 来评估. 这样的话, 我们时序数据的个数为 n, 长度为 m, 则

根据 2.2 节中的分析, 整个数据集产生的 shapelet 个数达到 $O(nm^2)$ 的量级, 非常庞大, 由于评估一个 shapelet 的时间开销也不容小觑, 所以单位评估时间被这个数量放大之后, 时间开销爆炸性增长.

然而, 我们发现, 如果我们使用“shapelet 转换”的方式配合定点浮动距离 (式 (3)) 来构造分类器的话, 我们可以通过固定 shapelet 的长度来大幅缩减 shapelet 候选集的规模, 接下来我们就来说明这件事.

我们看到“shapelet 转换”的第一步是从候选 shapelet 中选择判别性强的若干个 shapelet 出来, 第二步是基于这些选择出来的 shapelet 构造特征空间, 将原始数据转换至特征空间, 第三步是在特征空间中对转换后的数据进行分类.

在这个过程中, 我们能够发现最后对数据分类所倚赖的关键是构建特征空间的 shapelet 的判别性. 而 shapelet 的判别性体现在, 和该 shapelet 同类的时间序列, 与该 shapelet 的距离要足够小, 而反之则要与该 shapelet 的距离足够大. 而根据转换的方式 (参见 2.3 节), 当原始数据被转换到特征空间之后, 这就体现在和 shapelet 同类的时间序列, 转换后在该 shapelet 对应的坐标轴上的范数要比较小, 而反之则对应的范数要比较大.

所以我们看到, 在特征空间中分类的关键依据, 其实质是原始数据被转换到特征空间之后的范数. 如果有两个不同的特征空间, 原始数据被转换到它们之中后拥有相同的空间范数, 那最终两个特征空间中的分类依据就是相同的, 分类效果也会大同小异, 在某种程度上, 我们可以认为这两个特征空间是等价的.

现在假设原来的 shapelet 候选集是 A, 如果我们找到一个 A 的很小的子集 B, 使得: 从 A 里面找出的任意一组 shapelet P, 任意原始数据 $\boldsymbol{x}$ 被转换到 P 构造的特征空间中的范数记录为$\|\boldsymbol{x}\|_P$, B 中都存在对应的一组 shapelet Q, 原始数据 $\boldsymbol{x}$ 被转换到 Q 构造的特征空间中的范数记录为$\|\boldsymbol{x}\|_Q$, 且任意 $\boldsymbol{x}$, 都有$\|\boldsymbol{x}\|_P \approx \|\boldsymbol{x}\|_Q$, 即这两个特征空间是等价的. 那么这样的 B 显然具备构造等价特征空间的能力. 而如此一来, 我们就能够抛弃原来巨大的候选 shapelet 集 A, 而只选用它的很小的子集 B, 这样时间开销会得到大幅降低.

我们现在就来证明这件事, 即存在上述那样一个小子集 B, 其具备构造同等特征空间的 shapelet. 此事关键在于证明如下的定理:

定理 1. 将一个 shapelet s 分割成若干段 $C=\{s_1, s_2, \cdots, s_n\}$, 使得它们按序拼接起来构成完整的 s, 我们称这样的 C 为 s 的一个划分集. 对任意时间序列 x, 其通过定点距离 (式 (2)) 转换至 s 构造的特征空间中的欧氏范数记录为$\|x\|_s$, 而其通过定点距离转换至 C 构造的特征空间中的欧氏范数记录为$\|x\|_C$, 则我们有$\|x\|_s=\|x\|_C$.

证明: 不妨设 $s=(s_i, s_{i+1}, \cdots, s_{i+k})$, 其中 i 是 s 在原始时间序列中的起始位置, $k+1$ 为其长度. 令 t_j 为 C 中 s_j 的终止位置, 且规定 $t_0=i-1$, $t_n=i+k$, 那么我们有 $s_j=(s_{t_{j-1}+1}, s_{t_{j-1}+1}, \cdots, s_{t_j})$. 对于任意时间序列 x, 我们设 $x=(x_1, x_2, \cdots, x_m)$, 显然 $m>i+k$.

则根据 shapelet 转换的规则, 我们有:

$$\begin{aligned}\|x\|_C &= \left\| \left(\sqrt{\sum_{p=t_0+1}^{t_1}(x_p-s_p)^2}, \cdots, \sqrt{\sum_{p=t_{n-1}+1}^{t_n}(x_p-s_p)^2} \right) \right\|_2 \\ &= \sqrt{\sum_{p=i}^{t_1}(x_p-s_p)^2 + \sum_{q=2}^{n}\sum_{p=t_{q-1}+1}^{t_q}(x_p-s_p)^2} \\ &= \sqrt{\sum_{p=i}^{t_n}(x_p-s_p)^2} = \sqrt{\sum_{p=i}^{i+k}(x_p-s_p)^2} = \|x\|_s\end{aligned} \tag{4}$$

证毕.

根据上述定理我们发现, 在当前叙述背景下, 一个 shapelet 构造的特征空间, 和它的划分集构造的特征空间可以看成是等价的. 尽管最后我们用作转换的距离度量方式是定点浮动距离 (式 (3)) 而不是定点距离 (式 (2)), 但是定点浮动距离只是定点距离的松弛版本, 所以最后转换后的数据的范数与定理 1 中的范数并不会相差太远, 这样我们依旧有$\|x\|_C \approx \|x\|_s$. 所以在定点浮动距离作为转换坐标的计算公式的前提下, 我们得到 shapelet 构造的特征空间和它们的划分集构造的特征空间是近似的.

而另一方面, 每一个 shapelet 都可以划分为若干短的 shapelet, 所以我们只需要保留 shapelet 候选集里足够短的 shapelet, 就足够我们找到好的 shapelet 来构造好的特征空间了. 如此一来, 我们只需要选取长度为某个固定小数值的所有 shapelet 来作为 shapelet 候选集即可, 这个数值一般取 3 或者 4 即可, 我们把这样的 shapelet 称作为“微局部特征”. 我们将长度设置为 3 或者 4 是一种折衷, 当长度设置比 3 更小的时候, 这些短 shapelet 将失去统计意义, 因为时序数据是连续的数据,

在某个时间点的值并不能构成统计意义上的特征, 它们提供的分类信息也因而不具备高可信度. 而当长度比 4 还大时, 将无法覆盖某些短 shapelet, 比如长度为 4 的 shapele, 失去这些 shapelet 会对分类结果造成影响. 由这些微局部特征构成的集合正是我们需要寻找的原候选集的小子集. 对于 n 个长度为 m 的时间序列构建的数据集, 我们构建的 shapelet 候选集只有 $O(nm)$ 的规模, 而不再是 $O(nm^2)$ 的量级.

3.3 shapelet 判别性的评价指标

在具备 shapelet 候选集后, 需要从中选取判别性强的 shapelet 来作为构建特征空间的一组基底.

容易知道, 选取 shapelet 需要量化 shapelet 的判别性能, 目前普遍采用的做法是用最佳信息增益来量化 shapelet 的判别性能. 该做法如算法 1 所示.

算法 1. 计算 shapelet 的信息增益

输入: shapelet s, data set $T=(X, Y)$
输出: prime information gain of s

1) **For** each time series (x, y) **in** T
2) Calculate $Ds, x)$, namely distance between s and x;
3) Depict $D(s, x)$ with its label y in a real line r;
4) **End for**
5) Find each possible segmentation in real line r to build C;
6) Set *prime_information_gain*=0;
7) **For** each segmentation c **in** C
8) Calculate information gain of c as g;
9) **If** g>*prime_information_gain*
10) *Prime_information_gain*=g;
11) **End if**
12) **End for**
13) **Return** *prime_information_gain*;

从该算法中可以看到, 如果要计算一个 shapelet 的最佳信息增益, 则需要计算所有“分割”的信息增益再挑出里面最大的. 这样做非常耗时, 尤其当数据集里面数据比较多时, 则可能的“分割”的数目指数增长, 评价一个 shapelet 的代价变得相当大.

为了避免这个问题, 我们决定采用广义雷利熵来作为 shapelet 的判别性能的评价指标. 对于一个实数轴上的二分类问题, 我们假设两类数据的集合分别为 P 和 Q, 则广义雷利熵的计算公式如下式所示:

$$GRQ(P,Q)=\frac{|\mu(P)-\mu(Q)|}{\sigma^2(P)+\sigma^2(Q)} \tag{5}$$

式中, $\mu(.)$ 是均值函数, $\sigma^2(.)$ 是方差函数.

利用广义雷利熵来作为判别指标后, 我们有效避开了寻找最佳分割的过程, 不再需要计算每种分割的信息增益来寻找最佳信息增益了, 这种耗时的操作因此也被去除掉了. 因为 shapelet 的评估是一个单位操作, 所以在单位操作上带来的时间节省, 被操作次数放大后, 会得到非常可观的优化效果. 如此一来, 评估一个 shapelet 的算法如下所示.

算法 2. 计算 shapelet 的判别性能 (广义雷利熵)

输入: shapelet s, data set $T=(X, Y)$
输出: prime information gain of s

1) **For** each time series (x, y) **in** T
2) Calculate $D(s, x)$ by formula (3);
3) Depict $D(s, x)$ with its label y in a real line r;
4) **End for**
5) Calculate general Rayleigh quotient *grq* of points in r;
6) **Return** *grq*;

3.4 特征空间中的分类器

正如我们 2.3 节中所述, 使用“shapelet 转换”的方式必须承担转换后的数据在特征空间中的分布不定性这一代价, 所以要想保证分类精度, 需要在转换后的特征空间中训练多个类型不同的分类器, 保证不遗漏可能的数据分布. 但是这样做的时间代价相当高.

自然地, 我们想要避免这种操作来降低时间开销. 由于我们无法保证特征空间中的数据具有某种特定的分布特性, 所以我们只能用不同类型的分类器把所有可能的分布特性都考虑进去. 而造成这种现象的根源在于选择 shapelet 的环节. 我们确实选择了最具判别性的 shapelet 来作为特征空间的基底, 但是却忽略了它们的组合效应. 在经典的基于 shapelet 的算法中, 他们选择最佳信息增益作为判别指标, 这使得在“shapelet 转换”后, 特征空间的数据在每个坐标轴上的投影具有非常好的线性可分性, 但是在每个坐标上的分量具有很好的线性可分性, 并不能保证在整个空间上具备很好的线性可分性, 这是问题的关键所在.

所以若我们可以保证原始数据被转换到相应的特征空间之后, 具备线性可分性这一分布特性. 我们就能使用单个的 SVM 去替代集成的分类器而避免训练集成分类器这一耗时操作.

我们现在断言, 如果我们使用广义雷利熵作为 shapelet 的评价指标, 那么高得分的 shapelet 能够保证特征空间中数据的线性可分性, 这倚赖于如下的定理:

定理 2. 对于一个时序数据的二分类问题, shapelet 的广义雷利熵与特征空间中数据的线性可分性存在正

相关.

证明: 我们假设选择了 k 个 shapelet 构造好了一个特征空间, 那我们要证明的是, 将其中某个 shapelet s_1 替换为拥有更大广义雷利熵的 shapelet s_2, 数据的线性可分性会提高. 我们假设原始数据中有两个类的时间序列, 根据中心极限定理, 它们通过这 k 个 shapelet 转换到特征空间之后实际上形成了两个随机向量 A, B, 且 A 和 B 服从球形正态分布.

我们自然地按如下方式来定义这两类数据的线性可分性:

$$LS = \max_{||\vec{w}||=1,\vec{w}\in R^k} P((A-B)\cdot\vec{w}>0) \quad (6)$$

式中, LS 是线性可分性, $P(\cdot)$ 是概率函数. 我们现规定 $\mu(\cdot)$ 是均值函数, 返回随机向量的均值向量; $\sigma^2(\cdot)$ 为随机向量的协方差矩阵的对角线函数, 它返回由协方差矩阵的对角线构成的向量; $[\cdot]^2$ 是平方函数, 对向量中每一维度的数据取平方来得到一新向量. 根据 A, B 服从球形高斯分布, 我们有:

$$(A-B)\cdot\vec{w}\sim N((\mu(A)-\mu(B))\cdot\vec{w},(\sigma^2(A)+\sigma^2(B))\cdot[\vec{w}]^2) \quad (7)$$

式中, $N(\cdot,\cdot)$ 为正态分布的符号. 结合式 (7), 根据线性可分性的定义以及$\vec{w}$的取值范围 (式 (6)), 对于最佳的 $\vec{w}$, 我们显然可以得到如下关系:

$$(\mu(A)-\mu(B))\cdot\vec{w}^*\geq 0 \quad (8)$$

$$(\mu_i(A)-\mu_i(B))\cdot\vec{w}_i^*\geq 0 \quad 1\leq i\leq k \quad (9)$$

式中, $\vec{w}^*$表示其为使式 (6) 取得最大值的最佳$\vec{w}$.

根据广义雷利熵的计算式 (式 (5)), 我们若证明任意 $0<i<k+1$, $|\mu_i(A)-\mu_i(B)|$与式 (6) 中的线性可分性成正相关以及$\sigma_i^2(A)+\sigma_i^2(B)$与其成负相关, 则可完成定理的证明.

但这件事并不难, 假设我们已经拥有最佳的 LS 值 (式 (6)), 现在将某个特定的$\mu_i(A)-\mu_i(B)$替换为拥有更大绝对值的$\mu_i(A')-\mu_i(B')$, 保持其它数值不变. 我们不妨假设$(\mu_i(A')-\mu_i(B'))\cdot\vec{w}_i^*>0$, 因为若其小于 0, 我们只需将相应的$\vec{w}_i^*$取成相反数即可. 由于$(\mu_i(A')-\mu_i(B'))\cdot\vec{w}_i^*>(\mu_i(A)-\mu_i(B))\cdot\vec{w}_i^*\geq 0$, 其他项不变, 所以$(\mu(A')-\mu(B'))\cdot\vec{w}^*>(\mu(A)-\mu(B))\cdot\vec{w}^*\geq 0$. 再根据正态分布的特性及式 (8), 我们有:

$$LS'\geq P((A'-B')\cdot\vec{w}^*>0)>P((A-B)\cdot\vec{w}^*>0)=LS \quad (10)$$

这样即证明了$|\mu_i(A)-\mu_i(B)|$与式 (6) 中的线性可分性成正相关, 同样地方法可以证明与$\sigma_i^2(A)+\sigma_i^2(B)$式 (6) 中的线性可分性成负相关, 此处不再赘述.

证毕.

定理 2 为特征空间中的转换后的数据的线性可分性这一分布特性提供了理论支撑, 所以在我们提出的新算法中, 当原始时间序列被转换到特征空间之后, 我们可以放心地使用单个的 SVM 来执行分类, 而不再需要训练集成分类器, 这一结果极大优化了时间性能.

4 算法框架与实验结果

4.1 主体算法

我们在此节给出最终的算法框架, 如算法 3 和算法 4 所示.

算法 3. 基于微局部特征的时序数据二分类算法 (分类器构造)

输入: shapelet length L, binary data set $T=(X, Y)$, shapelet number N, relaxation factor r and l
输入: a priority queue of N shapelet, a SVM classifier

初始化: a null priority queue Q, a null set S, a null vector set V
1) **For** each time series (x, y) **in** T
2) Capture all the L-length shapelet in $(\boldsymbol{x}, y)$ and put them in S;
3) **End for**
4) **For** each L-length shapelet (s, z) **in** S
5) Input (s, z) and T to Algorithm 2, and get a return value grq;
6) Add (s, z) to Q, and set its priority as grq;
7) **End for**
8) Q=N-length subqueue of Q;
9) **For** each time series (x, y) **in** T
10) Transform (x, y) to a N-dimensional label vector (v, y) by calculate distance with shapelet in Q by formula (3);
11) Add (v, y) to V;
12) **End for**
13) Train a SVM svm **in** V;
14) **Return** Q, svm.

算法 4. 基于微局部特征的时序数据二分类算法 (数据分类)

输入: shapelet queue Q, SVM classifier svm, data x
输出: class label of x

1) Transform x to the N-dimensional vector v by calculate distance with shapelet in Q by formula (3);
2) Use svm to classify $\boldsymbol{v}$, get a label y;
3) **Return** y.

4.2 实验数据集

为了公平起见, 我们选择 Bagnall 等人在其工作中所使用的数据集[1]. 他们在相关研究工作中精心筛选数据集以及各种算法, 并做出了比较公平公正的对比, 他们所使用的数据集也被选作时序数据分类算法社区的

标准数据集[10]. 先对数据集做如下介绍:

Ham, 火腿光谱图数据, 通过对光谱图进行分类来判断火腿的种类, 训练集 109 个数据, 测试集 105 个数据, 数据长度 431.

MPOC, 全称 Middle Phalanx Outline Correct, 手指中部骨节的 X 光投影轮廓图. 科学家根据该数据来判断人们所处的年龄阶段, 训练集 600 个数据, 测试集 291 个数据, 数据长度 80.

Eq, 全称 Earthquakes, 用传感器捕捉的地震波数据, 用来判断近期内是否会有地震发生, 数据来自于北加利福利亚地震研究中心. 训练集 322 个数据, 测试集 139 个数据, 数据长度 512.

Herring, 鲱鱼的耳石轮廓, 该数据用于生物多样性研究, 通过耳石轮廓对应的时序数据来判定鲱鱼生活的地区. 训练集 64 个数据, 测试集 64 个数据, 数据长度 512.

IPD, 即 Italy Power Demand, 意大利人民不同季度生活用电时序数据, 不同类别的时序数据对应不同季度的用电水平. 训练集 67 个, 测试集 1029 个, 数据长度 24.

Wine, 葡萄酒的光谱图, 光谱图上不同种类的时序数据对应不同种类的葡萄酒. 训练集 57 个数据, 测试集 54 个数据, 数据长度 234.

用于做实验的数据集来自于实际应用的各方各面, 包括天文地理, 衣食住行等多个领域, 也从侧面反映了时序数据有着广泛的应用.

4.3 对比算法与算法超参

本文针对基于 shapelet 的时序分类算法进行分析与改进, 旨在提升算法的分类精度和降低算法的时间开销, 所选对比算法为基于 shapelet 的时序分类算法中的优秀算法, 介绍如下:

FS, 该算法专注于时间开销, 是现有的基于 shapelet 的算法中平均时间开销比较低的, 但是其牺牲了部分精度, 采用近似的方法选取 shapelet[7].

LS, 是用学习的方式获取 shapelet 的代表算法, 通过把获取判别性 shapelet 这一难题转换为优化问题, 并用梯度下降来习得最优 shapelet, 具有非常不错的时间开销和分类精度[8].

ST, 是当今主流的基于 shapelet 的时序数据分类算法, 它通过 shapelet 将原始数据转换至特征空间, 在特征空间里训练集成分类器进行分类, 拥有较高的时间复杂度, 但是分类精度属于领域内的顶尖[12].

COTE, 是基于 shapelet 的集成算法中的集大成者, 除了集成基于 shapelet 的时序数据分类算法, 也集成了其它类型的时序数据分类算法, 因此是四个对比算法中时间复杂度最高的, 同时也是分类精度最好的[13].

值得一提的是, 在 Bagnall 等人的工作中, ST 和 COTE 不仅是基于 shapelet (局部特征) 的时序数据分类算法中的最好的, 也是所有时序数据分类算法中分类精度最好的[1]. 详情请见图 1[1].

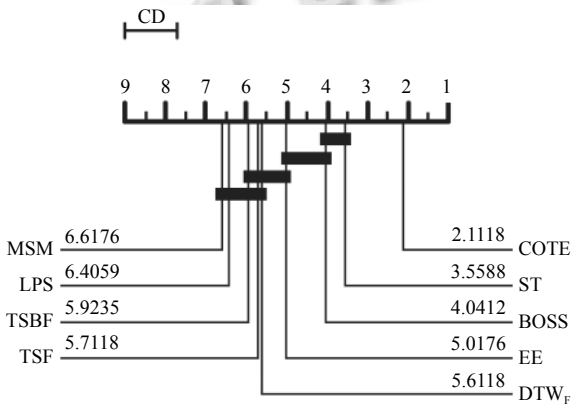

图 1 最佳算法在 UCI 数据集上的分类精度平均排名[1]

另外, 我们提出的基于微局部特征的时序数据二分类算法, 它的英文名称为“mini-shapelet based algorithm”, 简写为“MS”. MS 的超参选择集如表 1 所示.

表 1 MS 的超参取值

MS 的超参	取值范围
左右松弛因子 l, r	{3, 4}
Shaplets 的长度 L	{3, 4}
选择的 shapelet 个数 N	{10, 50, 100, 250, 500, 1000, 1500}

4.4 分类性能对比

关于分类性能, 我们延用时序分类算法社区里面的硬指标, 即分类精度对比以及分类精度的排名对比, 具体的对比数据如表 2, 表 3 以及图 2 所示.

表 2 分类精度对比

数据集	FS	LS	ST	COTE	MS
Ham	0.677	0.832	0.808	0.805	**0.838**
MPOC	0.716	0.822	0.815	0.801	**0.897**
Eq	0.747	0.742	0.737	0.747	**0.871**
Herring	0.558	0.628	0.653	0.632	**0.875**
IPD	0.909	0.952	0.953	0.970	**0.978**
Wine	0.794	0.524	0.926	0.904	**1.000**

表 3 分类精度排名对比

数据集	FS	LS	ST	COTE	MS
Ham	5	2	3	4	1
MPOC	5	2	3	4	1
Eq	2	4	5	2	1
Herring	5	4	2	3	1
IPD	5	4	3	2	1
Wine	4	5	2	3	1
Average	**4.3**	**3.5**	**3.0**	**3.0**	**1.0**

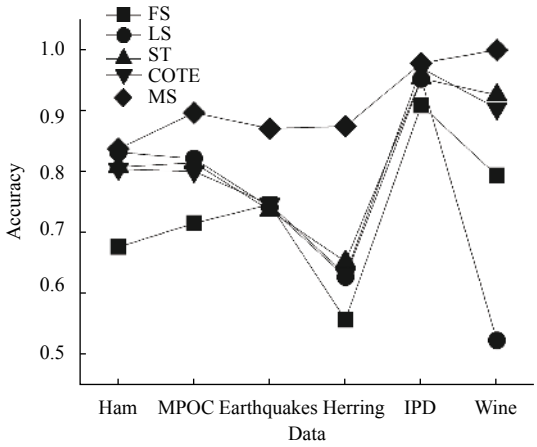

图 2 算法分类精度点线对比图

从表 2 以及表 3 中可以看出, 在所有测试数据集上面, 基于微局部特征的分类算法 (MS) 都取得了最优的分类精度, 并且在某些数据集上还具有非常明显的分类精度优势, 比如 Herring 和 Eq 这两个数据集. 综合来看, 相对于其它 4 种经典的基于 shapelet 的分类算法, 基于微局部特征的分类算法显而易见地具有最佳的分类能力. 而在 Wine 这个数据集上, 它甚至取得了无任何错误的分类结果, 体现了其极强的分类能力. 因而综上所述, 基于微局部特征的时序数据分类算法相较于 4 种对比算法有明显的分类精度优势.

从图 2 中我们可以观察出四种对比算法分类性能不太稳定, 都有大幅度的波动. 而相比之下, 基于微局部特征的分类算法 (MS) 具有较为稳定的分类性能, 其折线波动较小, 相对来说比较平稳. 所以基于微局部特征的时序数据分类算法在分类表现上稳定性更好, 且具有稳定的分类性能优势.

4.5 时间开销对比

关于时间开销, 因为 5 个算法都是使用 eager learning 的方式进行分类, 而且其主要的时间都用在构造分类器上, 所以我们主要比较它们的构建分类器的时间. 我们在同样的硬件条件 (内存: 8 GB; CPU: 2.5 GHz) 及软件条件下 (OS: Win 10; platform: JAVA) 进行实验, 具体数据如表 4 及表 5 所示.

表 4 时间消耗对比 (单位: 秒)

数据集	FS	LS	ST	COTE	MS
Ham	182.8	883.5	51.1	2.8E+05	**8.0**
MPOC	**14.8**	133.5	1.3E+05	2.5E+05	36.9
Eq	1.3E+03	1.9E+03	1.5E+03	9.1E+05	**110.2**
Herring	118.4	678.8	15.3	1.8E+05	**7.0**
IPD	**0.1**	99.1	16.7	88.3	0.5
Wine	5.1	111.1	11.2	4.3E+04	**1.5**

表 5 时间消耗排名对比 (升序)

数据集	FS	LS	ST	COTE	MS
Ham	3	4	2	5	1
MPOC	1	3	4	5	2
Eq	2	4	3	5	1
Herring	3	4	2	5	1
IPD	1	5	3	4	2
Wine	2	4	3	5	1
Average	**2.0**	**4.0**	**2.8**	**4.8**	**1.3**

从表 4 和表 5 中可以观察到, 除了 MPOC 和 IPD 这两个数据集外, 基于微局部特征的分类算法 (MS) 在剩余数据集上都具有最小的时间开销, 而在 MPOC 和 IPD 这两个数据集上, 也仅次于 FS 这一算法, 但是 FS 在 MPOC 和 IPD 上的精度远不如基于微局部特征的时序数据分类算法. 再结合表 5 中的平均排名, 我们可以认为相对于其它 4 个对比算法, 在保持最高分类精度的同时, 基于微局部特征的时序数据分类算法具有最佳的时间性能.

5 结语

本文针对当前基于局部特征的时序数据分类算法中存在的问题与挑战, 在充分的理论依据的支撑下, 使用缩减候选集, 调整判别性评定指标, 修改距离度量以及替换集成分类器 4 项技术设计了高效实用的新型算法. 该基于微局部特征的时序数据分类算法在实验数据集上表现出良好的分类性能和时间性能. 通过实验对比, 也证明了其对当前研究领域内存在的分类精度不足以及时间开销过高等问题有不错的改进.

参考文献

1 Bagnall A, Lines J, Bostrom A, *et al*. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 2017, 31(3): 606–660. [doi: 10.1007/s10618-016-

0483-9]
2 Ye LX, Keogh E. Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery, 2011, 22(1–2): 149–182. [doi: 10.1007/s10618-010-0179-5]
3 He Q, Dong Z, Zhuang FZ, *et al*. Fast time series classification based on infrequent shapelets. Proceedings of the 11th ICMLA International Conference on Machine Learning and Applications. Boca Raton, FL, USA. 2012. 215–219.
4 Deng HT, Runger G, Tuv E, *et al*. A time series forest for classification and feature extraction. Information Sciences, 2013, 239: 142–153. [doi: 10.1016/j.ins.2013.02.030]
5 Zakaria J, Mueen A, Keogh E, *et al*. Accelerating the discovery of unsupervised-shapelets. Data Mining and Knowledge Discovery, 2016, 30(1): 243–281. [doi: 10.1007/s10618-015-0411-4]
6 Baydogan MG, Runger G. Time series representation and similarity based on local autopatterns. Data Mining and Knowledge Discovery, 2016, 30(2): 476–509. [doi: 10.1007/s10618-015-0425-y]
7 Rakthanmanon T, Keogh EJ. Fast shapelets: A scalable algorithm for discovering time series shapelets. Proceedings of the 2013 SIAM International Conference on Data Mining. Austin, TX, USA. 2013. 668–676.
8 Grabocka J, Schilling N, Wistuba M, *et al*. Learning time-series shapelets. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA. 2014. 392–401.
9 Hou L, Kwok JT, Zurada JM. Efficient learning of timeseries shapelets. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, AZ, USA. 2016. 1209–1215.
10 Hills J, Lines J, Baranauskas E, *et al*. Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery, 2014, 28(4): 851–881. [doi: 10.1007/s10618-013-0322-1]
11 Guido RC. Fusing time, frequency and shape-related information: Introduction to the discrete shapelet transform's second generation (DST-II). Information Fusion, 2018, 41: 9–15. [doi: 10.1016/j.inffus.2017.07.004]
12 Bostrom A, Bagnall A. Binary shapelet transform for multiclass time series classification. Madria S, Hara T. Big Data Analytics and Knowledge Discovery. Cham: Springer, 2015. 257–269.
13 Bagnall A, Lines J, Hills J, *et al*. Time-series classification with COTE: The collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(9): 2522–2535. [doi: 10.1109/TKDE.2015.2416723]