ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 31,No. 1,Jan. 2016,pp. 168-177
DOI:10.16337/j.1004-9037.2016.01.018

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:+86-025-84892742

多源时间序列中具有显著时间间隔的 Shapelet 对挖掘

李钟麒[1,2] 段 磊[1,2] 胡 斌[3] 邓 松[4] 秦 攀[2]

(1. 武汉大学软件工程国家重点实验室,武汉,430072;2. 四川大学计算机学院,成都,610065;
3. 国家电网智能电网研究院,南京,210003;4. 南京邮电大学先进技术研究院,南京,210003)

摘 要:Shapelet 作为时间序列特征,具有较好的可解释性。Shapelet 在行为识别、聚类分析及异常检测等方向均得到了广泛应用。但在电力运行监测、医学图像分析以及流媒体监测等领域,时间序列具有多源、同步的特点,仅对单一源上的时间序列提取 Shapelet 可能丢失序列间相关性。在 Shapelet 概念基础上,本文提出 p-Shapelet 作为不同源的 Shapelet 间关于时间间隔的特征表达,从而实现分析不同源 Shapelet 间的相关性。具体地,为找出不同类别样本间时间间隔具有最显著差异的 Shapelet 对,设计并实现了并行化挖掘的算法 p-Shapelet miner。算法采用信息增益对不同源间的 Shapelet 对进行评价,并找出能最大化信息增益的 Shapelet 对(p-Shapelet)。利用 CMU 人体动作捕捉数据集进行实验,验证了算法的有效性与执行效率。

关键词:时间序列;特征提取;信息增益;Shapelet
中图分类号:TP311 **文献标识码**:A

Mining Pair of Shapelets with Significant Time Lags From Multi-Sources Synchronous Time Series

Li Zhongqi[1,2], Duan Lei[1,2], Hu Bin[3], Deng Song[4], Qin Pan[2]

(1. State Key Laboratory of Software Engineering, Wuhan University, Wuhan, 430072, China;
2. School of Computer Science, Sichuan University, Chengdu, 610065, China;
3. State Grid Smart Grid Research Institute, Nanjing, 210003, China;
4. Institute of Advanced Technology, Nanjing University of Posts & Telecommunications, Nanjing, 210003, China)

Abstract: As the feature of a time series, Shapelet holds a good interpretability. Shapelet is widely applied recently in behavior reorganization, clustering and outlier detection, et al. However, time series data are synchronized and multi-sources in domains of electric power operation monitoring, medical image processing and streaming media, The relevance among time series are ignored if only extracting Shapelet from single source independently. Thus, to analyze the relevance of Shapelets from different sources, p-Shapelet is proposed as a new feature expressing time lags among Shapelets based on Shapelet. Specifically, for mining pair of Shapelets with most significant time lags from different classes, a parallel algorithm called the p-Shapelet miner is designed. It evaluates pair of Shapelets from different sources by information gain, and find the one(p-Shapelet) maximums information gain. The effectiveness and efficiency of the algorithm is verified by experiments using CMU motion capture datasets.

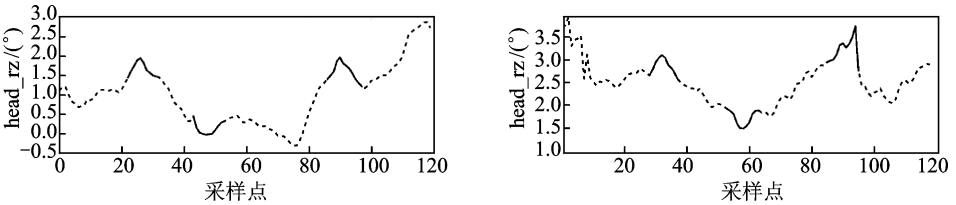
Key words: time series; feature extraction; information gain; Shapelet

---

基金项目:国家自然科学基金(61103042)资助项目;软件工程国家重点实验室开放研究基金(SKLSE2012-09-32)资助项目。
收稿日期:2014-06-05;**修订日期**:2015-06-09

引　言

时间序列作为序列分析中的常见数据类型,引起了广泛关注,如用户行为分析[1]、事件预测[2]以及异常监测[3]等。在时间序列的分类上,最常用的最近邻[4-6]方法虽然效率较高,但可解释性较差。Ye 等首次提出了以 Shapelet 作为时间序列挖掘的特征表达[4],并以此构建决策树来解决时间序列分类问题。Shapelet 本质为一段子序列,将其作为序列特征,有助于提高结果的可解释性。目前,Shapelet 已广泛地应用于时间序列分析,如:通过手部运动的时间序列识别持枪与非持枪状态[7];将手部 x 光医学图像转换为时间序列可提高分类精度[8];利用 Shapelet 对信号序列进行聚类分析[9]。再如,电力网络中根据传感器工作状况进行设备状态估计,利用事件时间间隔变化对故障进行探测。此外,Shapelet 也可用于序列数据质量评估。在康复医学领域,医生对患者的受损关节的康复情况进行检查,通过传感器记录相关关节在纵向(X),横向(Y)和竖直(Z)3 个方向上摆动幅度的时间序列。图 1 给出了在步行状态和奔跑状态下,某患者头部 Z 轴方向摆动幅度的时间序列。



(a) 步行状态 (a) Walking state　　(b) 奔跑状态 (b) Running state

图 1　不同运动状态下的时间序列(单一源)

Fig. 1　Time series of different motions state (single source)

对比图 1 (a)与图 1 (b)可见序列之间存在整体相似性,仅用 Shapelet(图中实线部分)不能准确地区别步行和奔跑状态。考虑实际情况中,除了头部监测以外,还有患者其他部位(如:胸部)的监测序列。而在多源情况下,同时考虑源 head_rz 与 thorax_rz(胸部 Z 轴角度传感器读数),虽然在单个源上 head_rz(或 thorax_rz)挖掘得到的 Shapelet 并没有较好的区分度,但不同源上 Shapelet 之间的时间间隔却明显不同。

多源同步时间序列 $MS = \{S_1, S_2, \cdots, S_g\}$ 为一组时间序列集合,其中 S_k 表示由源 g_k 生成的序列,所有时间序列在时间上同步并且长度相等,即 $|S_1| = |S_2| = \cdots = |S_g|$。图 2 给出了在多源同步时间序列下,Shapelet 时间间隔变化。在步行状态下(如图 2(a)),head_rz 与 thorax_rz 上 Shapelet(实线部分)之间的时间间隔明显大于奔跑状态下(如图 2(b))head_rz 与 thorax_rz 上 Shapelet 之间的时间间隔。可见,对于多源同步时间序列,通过分析不同源 Shapelet 之间的时间间隔,可以发现区别不同类别的特征。

Shapelet 作为数据样本中的子序列,由于其长度短以及可解释性较强,将其作为时间序列特征得到了广泛认可。Ye 等首先提出 Shapelet 作为时间序列特征的概念,并利用 Shapelet 构建决策树[7]。Jason 等以挖掘得到的 Shapelet 为样本,对数据进行转换后得到了更高的分类精度[8]。虽然文献[7]提出了基于距离的 Early abandon 和最大信息增益的剪枝,但获取 Shapelet 的计算量仍然较大,使得 Shapelet 难以得到广泛的应用。为提升 Shapelet 挖掘效率,Rakthanmanon 等提出了利用 SAX[10]将时间序列转化成符号序列的快速挖掘算法[11]。将时间序列转化为符号序列,会在一定程度上丢失信息,并失去时间序列的连续性特点,导致问题退化到了符号序列模式挖掘,但其结果在一定误差范围内可以接受。He

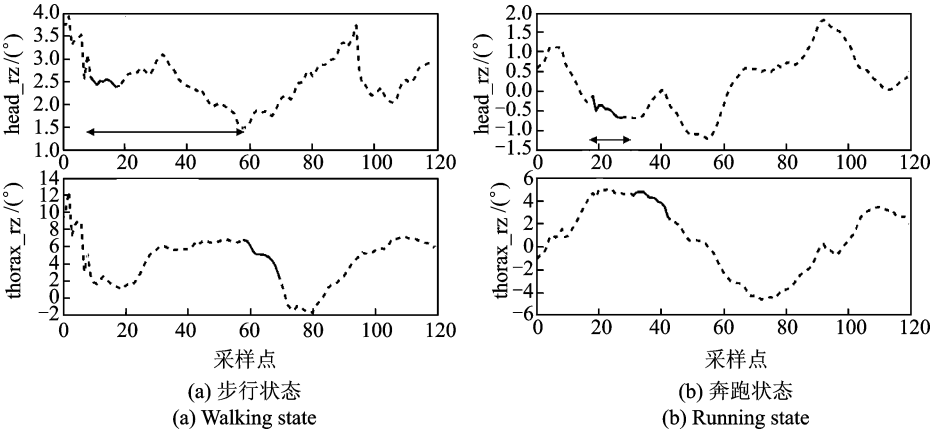


图 2 步行与奔跑状态下 head_rz 与 thorax_rz 子序列间时间间隔变化

Fig. 2 Time lag changes between subsequences on head_rz and thorax_rz of walking and running

等提出了利用非频繁 Shapelet 作为分类特征[12]，然而非频繁 Shapelet 在数据集上的支持度较低，可能丢失一般性，虽然可以作为分类特征，但并不具备良好的可解释性。在 Mustafa 等的研究中，Shapelet 被扩展应用到多维时间序列上，通过近似求解和投票机制来加速候选集的计算[13]。同时，Shapelet 也被应用到无监督学习领域[9]。Jesin 等利用文献[6]中的提出的 u-Shapelet 构造出了新的时间序列聚类算法[14]。Xing 等利用 Shapelet 对时间序列进行早期分类，并采用核密度估计的方法给出了合理估计 Shapelet 到时间序列距离阈值的方法，并在生成 Shapelet 候选时，利用增量计算方式，降低了算法复杂度[15]。Abdullah 等提出了采用多个 Shapelet 的组合作为分类的特征依据，实践表明，多个 Shapelet 的组合的确能够更好地表征数据特征[16]。

现有研究均只考虑了单源 Shapelet，并没有涉及广泛存在的多源同步时间序列以及 Shapelet 之间的时间间隔。据作者所知，尚没有 Shapelet 相关研究考虑了多源同步时间序列环境下 Shapelet 之间时间间隔的关系。为保证结果完备性，已有研究均采用遍历的方式产生候选 Shapelet。考虑各个源的序列间是相互独立的，因而可并行化产生候选 Shapelet。另外，与本文工作最为相近的文献[16]考虑了将多个 Shapelet 作为特征，但没有考虑多源情况下 Shapelet 的时间间隔。

1 问题定义

时间序列 $S=\langle S[t_1], S[t_2], \cdots, S[t_n]\rangle$ 为一组有序的实数序列，其中 t_i 为时间序列的时刻。不失一般性，本文假设时间序列时刻单位、采样频率一致。为便于分析，t_i 取值为非负整数。称 S 中的实数个数为 S 的长度，记为 $|S|$。用 g 表示生成时间序列 S 的源。给定时间序列 S，用 Q 表示 S 上长度为 l ($0\leqslant l\leqslant |S|$) 的子序列。即 $Q=\langle S[p], S[p+1], \cdots, S[p+l-1]\rangle$，其中 $1\leqslant p\leqslant |S|-l+1$。

定义 1 时间序列间的距离 给定等长时间序列 S_1 和 S_2，它们之间的距离 $\mathrm{Dist}(S_1, S_2)$ 为

$$\mathrm{Dist}(S_1, S_2)=\sqrt{\sum_{i=1}^{|S_1|}(S_1[i]-S_2[i])^2} \tag{1}$$

参照现有 Shapelet 相关研究[7,14-16]，本文采用欧氏距离作为时间序列间距离度量。在实际应用中，可选择其他距离函数。给定时间序列 S，用户定义的候选 Shapelet 长度 l。使用 S^l 表示 S 上所有长度为 l 的子序列，即

$$S^l=\{\langle S[p], S[p+1], \cdots, S[p+l-1]\rangle \mid 1\leqslant p\leqslant |S|-l+1\} \tag{2}$$

定义 2　子序列与时间序列的距离　给定候选 Shapelet Q,时间序列 S,Q 与 S 之间的距离 $\mathrm{SubseqDist}(Q, S) = \min(\{\mathrm{Dist}(Q, S') \mid S' \in S'\})$。根据定义 2,$\mathrm{SubseqDist}(Q, S)$ 即 Q 与 S 所有长度为 $|Q|$ 的子序列距离的最小值,将 S 上对应的子序列 S' 的起始位置称为 Q 与 S 的最佳匹配位置,记为 $\mathrm{BMP}(Q, S)$,即 $\mathrm{Dist}(Q, \langle S[\mathrm{BMP}(Q, S)], S[\mathrm{BMP}(Q, S)+1], \cdots, S[\mathrm{BMP}(Q, S) - |S'|+1]\rangle) = \mathrm{SubseqDist}(Q, S)$。图 3 给出了子序列 Q 在序列 S 上匹配最小距离位置,可得到 $\mathrm{BMP}(Q, S) = 20$。

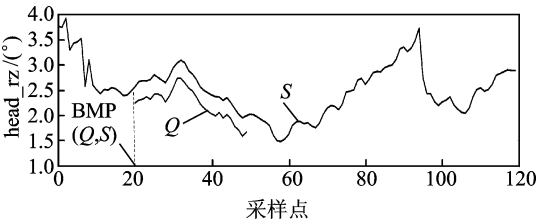


图 3　子序列 Q 与时间序列 S 间的距离

Fig. 3　Distance between subsequence Q and time series S

定义 3　子序列对的时间间隔　给定源 g_1,g_2 的 Shapelet 对$\langle Q_1, Q_2\rangle$,多源同步序列 MS,令 $MS[g_i]$ 表示 MS 中由源 g_i 产生的序列,$\langle Q_1, Q_2\rangle$的时间间隔记为:$\mathrm{Lag}(\langle Q_1, Q_2\rangle, MS) = \mathrm{BMP}(Q_1, MS[g_1]) - \mathrm{BMP}(Q_2, MS[g_2])$,即子序列 Q_1,Q_2 在多源同步时间序列 MS 上源 g_1 序列 S_1,源 g_2 序列 S_2 中最佳匹配位置差值。由于不同源之间序列不可比,故不考虑 MS 上除 g_1,g_2 外其他源的序列。为挖掘出 Shapelet 间存在的时间间隔关系,需要考察 Shapelet 对$\langle Q_1, Q_2\rangle$是否存在显著的时间间隔。

给定多源同步时间序列集合 D 和 Shapelet 对$\langle Q_i, Q_j\rangle$,D 上的时间间隔集合为 $\mathrm{Lag}(\langle Q_i, Q_j\rangle, D) = \{\mathrm{Lag}(\langle Q_i, Q_j\rangle, MS) \mid MS \in D\}$。为评价 Shapelet 对$\langle Q_i, Q_j\rangle$及其时间间隔是否能对数据进行有效划分,利用决策树[17]中的信息熵与信息增益评价其区分度。给定数据集 D,包含两个类 D_+ 和 D_-,满足 $D = D_+ \cup D_-$,$D_+ \cap D_- = \varnothing$,数据集 D 的熵值为

$$I(D) = -\frac{|D_+|}{|D|}\log\left(\frac{|D_+|}{|D|}\right) - \frac{|D_-|}{|D|}\log\left(\frac{|D_-|}{|D|}\right) \tag{3}$$

给定时间间隔阈值 $\mathrm{lag} \in \mathrm{Lag}(\langle Q_i, Q_j\rangle, D)$,可根据 lag 将数据集划分为两类,满足

$$\begin{aligned} D_1 &= \{MS \in D \mid \mathrm{Lag}(\langle Q_i, Q_j\rangle, MS) \leqslant \mathrm{lag}\} \\ D_2 &= \{MS \in D \mid \mathrm{Lag}(\langle Q_i, Q_j\rangle, MS) > \mathrm{lag}\} \end{aligned} \tag{4}$$

利用 Shapelet 对$\langle Q_i, Q_j\rangle$及时间间隔 lag 对数据集 D 进行分类后,可以得到 D_1 与 D_2 两类数据,分割后数据集的熵 $\hat{I}(D_1, D_2)$为

$$\hat{I}(D_1, D_2) = \frac{|D_1|}{|D|}I(D_1) + \frac{|D_2|}{|D|}I(D_2) \tag{5}$$

利用信息熵,可得到信息增益 Gain 评价分割策略

$$\mathrm{Gain}(D_+, D_-, \langle Q_i, Q_j, \mathrm{lag}\rangle) = I(D) - \hat{I}(D_1, D_2) \tag{6}$$

定义 4　最优时间间隔　给定含有两个类别的数据集 D,对于 Shapelet 对$\langle Q_i, Q_j\rangle$,最优时间间隔 $\mathrm{lag}_{\mathrm{opt}}$,对 $\forall\ \mathrm{lag} \in \mathrm{Lag}(\langle Q_i, Q_j\rangle, D)$满足

$$\mathrm{Gain}(D_+, D_-, \langle Q_i, Q_j, \mathrm{lag}_{\mathrm{opt}}\rangle) \geqslant \mathrm{Gain}(D_+, D_-, \langle Q_i, Q_j, \mathrm{lag}\rangle) \tag{7}$$

最优时间间隔即为使信息增益最大化的时间间隔。最后形式化定义本文的挖掘目标 p-Shapelet (pair-Shapelet)。

定义 5　p-Shapelet　给定多源同步时间序列集合 $D(D = D_+ \cup D_-, D_+ \cap D_- = \varnothing$,$C$ 为所有满足长度限制的子序列集合。p-Shapelet 为一对 Shapelet 与其最优时间间隔组成的三元组$\langle Q_i, Q_j, \mathrm{lag}_{\mathrm{opt}}\rangle$使得对任意 Shapelet 对$\langle Q_m, Q_n\rangle$及其最优时间间隔 $\mathrm{lag}_{\mathrm{opt}}$满足

$$\mathrm{Gain}(D_+, D_-, \langle Q_i, Q_j, \mathrm{lag}_{\mathrm{opt}}\rangle \geqslant (\max\{\mathrm{Gain}(D_+, D_-, \langle Q_m, Q_n, \mathrm{lag}_{\mathrm{opt}}\rangle) \mid Q_m, Q_n \in C\} \tag{8}$$

2　p-Shapelet 挖掘

为挖掘 p-Shapelet,本文设计并实现了并行化算法 p-Shapelet miner 来处理多源同步时间序列。p-

Shapelet 来自多源序列上的不同源，因此，首先需要生成每一个源上所有候选 Shapelet，然后对不同源间的候选 Shapelet 两两组合，产生候选 p-Shapelet $\langle Q_i, Q_j \rangle$，通过最优时间间隔和信息增益评价该候选区别 D_+ 和 D_- 的能力。概括地讲，p-Shapelet 的挖掘过程有 3 个步骤：(1)在每个源上挖掘 Shapelet；(2)对 Shapelet 进行两两组合作为候选 p-Shapelet；(3)对步骤(2)得到的候选计算最优时间间隔与信息增益，选取信息增益最大的候选作为 p-Shapelet。

2.1 p-Shapelet 候选生成

由于获取 Shapelet 计算开销较大，文献[15]提出了增量计算子序列与时间序列距离的算法，这里采用该算法进行 Shapelet 与时间序列距离计算。由用户给定 Shapelet 长度限制[MINLEN, MAXLEN]以及阈值 $\alpha \in (0,1]$。用长度[MINLEN, MAXLEN]的滑动窗口枚举候选 Shapelet，用 D_{S_i} 表示多源同步时间序列集合 D 上源 g_i 产生的时间序列集合。对每一个候选 Shapelet Q_i，计算 Q_i 到 D_{S_i} 上所有序列 S 的距离以及最佳匹配位置 BMP(Q_i, S)。在所有长度相等的候选 Shapelet 中，根据阈值 α 得到所有样本平均距离最小的 Shapelet 集合。算法 1 给出了挖掘 Shapelet 的伪代码。得到每个源上的 Shapelet 后，将不同源间的 Shapelet 两两组合，作为 p-Shapelet 候选。

算法 1 mineShapelet

输入： 源 g_i 时间序列数据 D_{Si}，Shapelet 长度区间[MINLEN, MAXLEN]，阈值 α。

输出： 源 g_i 上的 Shapelet S。

```
(1)   S ← ∅
(2)   C ← D_Si 上长度[MINLEN,MAXLEN]子序列   //候选子序列
(3)   For each Q ∈ C do
(4)     For S ∈ D_Si do
(5)       dist ← subseqDist(Q, S)   //计算子序列距离
(6)       BMP(Q, S)//最佳匹配位置
(7)     End For
(8)     S ← S ∪ {BMP(Q, S), dist}   //保存距离与匹配位置信息
(9)   End For
(10)  For l ∈ [MINLEN, MAXLEN] do   //根据阈值 α 选择距离最短的子序列
(11)    保留 α * 100%长度 l 的 Shapelet
(12)  End For
(13)  return S
```

2.2 p-Shapelet 挖掘

对每一个 p-Shapelet 候选，可得到的时间间隔集合 Lag 并计算使信息增益最大的 lag，为使 p-Shapelet 具有较强的类别区分能力，保留信息增益最大的候选作为 p-Shapelet，如算法 2 所示。在算法 2 中，利用算法 1 得到单一源上 Shapelet 集合(步骤 4)，将不同源间的 Shapelet 两两组合作为候选的 p-Shapelet(步骤 6)，计算最大信息增益。将信息增益最大的候选作为 p-Shapelet 保留。

算法 2 mine p-Shapelet

输入： 正例数据集 D_+，负例数据集 D_-，子序列长度区间[MINLEN, MAXLEN]，阈值 α。

输出： p-Shapelet。

```
(1)   p-Shapelet ← ∅, gain ← 0, S ← ∅
(2)   D ← D+ ∪ D-
(3)   For each d ∈ D do
(4)     S ← S∪mineShapelet(D_Si, MINLEN, MAXLEN, α)   //算法 1,获取单源上的 Shape-
```

let
(5) End For
(6) For Q_i, $Q_j \in S$ do
(7) If Gain(D_+, D_-, $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$) > gain then //计算最优 lag 及信息增益
(8) p-Shapelet ← $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$
(9) gain ← Gain(D_+, D_-, $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$) //保存当前最优结果
(10) End If
(11) End For
(12) return p-Shapelet

2.3 并行化策略

虽然文献[7,15]都对该算法进行了相应的改进,但其时间复杂度仍旧较高。为提升算法效率,在算法 1 和算法 2 的基础上,将单源上的 Shapelet 挖掘与候选 p-Shapelet 的评价并行化,进一步减少算法运行的时间。

从算法 1 的描述中可见,对于 D_{Si} 上的每一个时间序列 S,其生成候选 Shapelet 以及最佳匹配距离的计算相互独立,满足并行化条件。因此,可同时对多个候选 Shapelet 在 D_{Si} 上进行距离与匹配位置计算。算法 2 中,每个源上的 Shapelet 需要与其他源的 Shapelet 两两组合,并计算信息增益与最优 lag,该计算过程相互独立,因此可根据计算量进行分片,同时进行多个 p-Shapelet 候选评价,对算法 2 并行。结合算法 1 与算法 2 的并行策略,算法 3 给出了 p-Shapelet miner 的伪代码。

算法 3: p-Shapelet miner
输入: 正例数据集 D_+,负例数据集 D_-,子序列长度区间[MINLEN,MAXLEN],阈值 α。
输出: p-Shapelet。
(1) p-Shapelet ← ∅,gain ← 0,S ←∅
(2) $D \leftarrow D_+ \cup D_-$
(3) For $d \in D$ do
(4) 并行计算 d 上的 Shapelet 候选,得到 Shapelet 集合 S //算法 1 并行优化
(5) End For
(6) For Q_i, $Q_j \in S$ do //算法 2 评估部分并行化
(7) If Gain(D_+, D_-, $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$) > gain then //计算最优 lag 及信息增益
(8) p-Shapelet ← $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$
(9) gain ← Gain(D_+, D_-, $\langle Q_i, Q_j, \text{lag}_{\text{opt}}\rangle$) //保存当前最优结果
(10) End If
(11) End For
(12) return p-Shapelet

在算法 3 的步骤 4,对算法 1 进行优化。生成候选后,同时对多个候选 p-Shapelet 利用信息增益进行评价(步骤 6)。

3 实验评估

3.1 实验环境

本文所有算法均采用 Python 2.7 实现,实验均在配置为 Intel Core i7-3770 3.90 GHz 4 核 CPU,16GB 内存,Windows 8.1 操作系统的 PC 上完成。采用多进程进行并行计算(避免 Python GIL 锁机制带来的伪多线程并行)。

实验分别选取记录人体各个关节活动的 Locomotion, Sports 两组多源时间序列真实数据验证算法的有效性与执行效率,其中 Locomotion 为步行和奔跑状态下利用传感器记录人体各部位在直角坐标系 X,Y,Z 轴上角度随时间变化序列。Sports 为不同运动状态下利用传感器记录人体各部位在直角坐标系 X,Y,Z 轴上角度随时间变化序列。两组数据集均来自 CMU Motion Capture Database [18]。数据集相关描述如表 1 所示。Locomotion 数据集中包含 run 和 walk 两组数据集,run 为人体在奔跑状态的数据,walk 为人体在自然步行状态下的数据。Sports 中包含 soccer 与 basketball 两类数据,其中 soccer 为足球运动数据,basketball 为篮球运动数据。

表 1 数据集特征

Tab. 1 Characteristics of data set

数据集	类别	样本数目	最短长度	最大长度	平均长度	源数目
Locomotion	walk	10	128	1918	477	56
	run	10	128	181	143	56
Sports	basketball	7	385	1222	787	56
	soccer	7	362	801	558	56

3.2 有效性实验

为验证 p-Shapelet miner 的有效性,将算法 2 作为 baseline,利用 baseline 与 p-Shapelet miner 在 Locomotion 数据集上进行有效性测试,并行进程数 $p = 4$,在实验中,baseline 与 p-Shapelet miner 得到的结果完全一致。表 2,3 分别给出了在不同子序列长度限制、不同阈值 α 下 p-Shapelet 及其信息增益变化情况。可以看出,子序列长度会影响到 p-Shapelet miner 的挖掘结果,通常用户可根据应用调整子序列长度限制与阈值 α,以得到满足应用需求的 p-Shapelet。

表 2 不同 α 下 p-Shapelet 挖掘结果

Tab. 2 p-Shapelet with respect to α

α	[MINLEN, MAXLEN]	Shapelet 对	lag	信息增益
0.005	[5,20]		5	0.693
0.01	[5,20]		5	0.693
0.015	[5,20]		−1	0.693
0.02	[5,20]		−1	0.693

表 3 不同长度限制下 p-Shapelet 挖掘结果

Tab. 3 List of p-Shapelets with respect to length limitation

α	[MINLEN, MAXLEN]	Shapelet 对	lag	信息增益
0.01	[5,15]		5	0.693
0.01	[5,20]		5	0.693
0.01	[20,30]		−20	0.525

由表 2 可见,随着阈值 α 的变化,p-Shapelet 会产生变化。因为 α 越大,候选 p-Shapelet 越多。虽然某些候选子序列到原序列的距离较大,但与其他子序列组合后,其信息增益反而较大。当精度要求较高时,可减小 α。若是为了得到信息增益较大的 p-Shapelet,则可设定较大的 α 值。在信息增益相同的情况下,p-Shapelet miner 倾向于选择更长的子序列对作为 p-Shapelet,因为较长子序列更具有代表性。

由表 3 可见,当 Shapelet 长度变化不大时,得到的 p-Shapelet 并不会变化。当大幅调整 Shapelet 长度限制时,p-Shapelet 也会随之产生变化。可见,p-Shapelet 主要取决于 Shapelet 的选择,当 Shapelet 变化时,p-Shapelet 也会随之产生变化。而 Shapelet 主要由其长度限制决定,用户可根据实际需求,选取合适的子序列长度限制。

3.3 执行效率实验

为验证 p-Shapelet miner 的执行效率,本文采用 baseline 与之对比。为保证可比性,数据长度均取采样周期长度(120)。若无特殊说明,执行参数默认值如下:$\alpha=0.01$,MINLEN=5,MAXLEN=$|S|/8$($|S|$为时间序列长度),$p=4$,源数目=6。

图 4 给出了在两组数据集上,不同的 Shapelet 长度限制对算法效率的影响。可以看出,随着 Shapelet 长度限制区间的扩大,运行时间进一步上升。由于 Shapelet 长度限制区间扩大,在进行候选 Shapelet 与时间序列距离计算时,需要考虑的候选以及计算量会呈指数趋势增长。p-Shapelet miner 对数据集进行分片并行,效率明显高于 baseline。

图 5 给出了在两组数据集上阈值 α 变化时,程序执行效率的变化情况。由于 Shapelet 挖掘过程中保留的 Shapelet 个数,会随着 α 的增大而增加,因此生成的候选 p-Shapelet 数量也会随之增加。p-Shapelet miner 采用的并行策略,使得其性能高于 baseline。

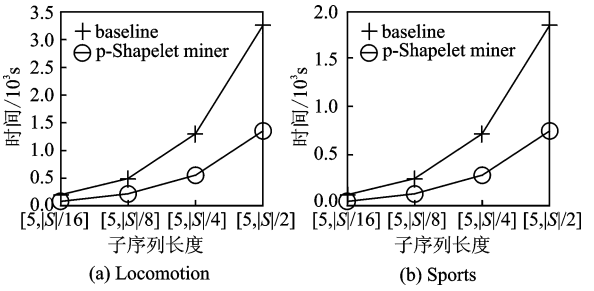


图 4 Shaplet 长度区间对运行时间的影响

Fig. 4 Runtime with respect to length limitation

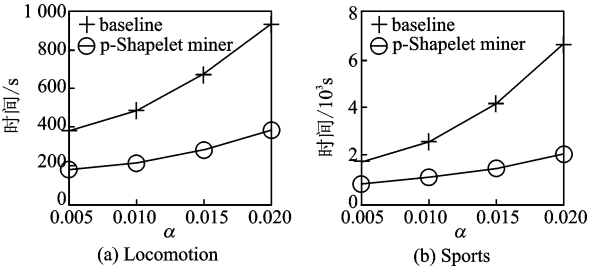


图 5 α 对运行时间的影响

Fig. 5 Runtime with respect to α

图 6 给出了在不同并行进程数下,p-Shapelet miner 与 baseline 的运行时间对比。随着并行进程数的提高,p-Shapelet miner 执行效率不断提高并趋于平稳,受实验环境限制,boseline 在并行度达到 4 以后,再增加并行度,效率提升已不明显。

图 7 给出了在不同源数目下,baseline 与 p-Shapelet miner 的执行效率对比,各个源按照出现顺序依

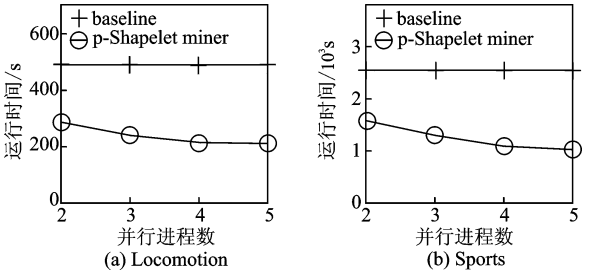


图 6 并行进程数对运行时算的影响

Fig. 6 Runtime with respect to process number

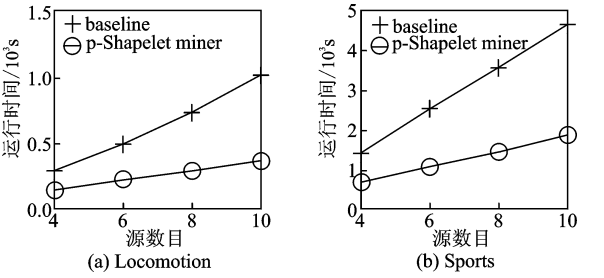


图 7 源数目对运行时间的影响

Fig. 7 Runtime with respect to the source number

次添加。可以看出,随着源数目的不断增加,处理的时间序列数量不断上升,执行时间会随着源数目的增加呈线性增长。

4 结束语

本文就目前 Shapelet 相关研究工作,分析了其在处理多源同步时间序列的不足之处,提出 p-Shapelet 作为多源同步时间序列的特征,设计并实现了高效并行处理多源同步时间序列的 p-Shapelet miner 算法,该算法对不同源上的候选 Shapelet 并行地进行时间间隔关系的挖掘,采用最大信息增益得到最具有区分度的时间间隔阈值,并利用 CMU Motion Capture Database 数据集,验证了算法的有效性与执行效率。本文工作仅仅考虑了两个 Shapelet 之间的时间间隔,如何快速合适地表达出多个 Shapelet 之间的时序关系是一个有趣的问题。另外,除了时间间隔以外,将 Shapelet 时间的时序关系利用多种时序关系进行扩展,会进一步增加特征表达的精确性与代表性。除并行化外,如何快速地对 Shapelet 进行数值计算、剪枝也是可研究的方向。对时间序列进行离散化快速挖掘 Shapelet 也是解决该问题的一种新思路,但如何克服离散化所带来的精确性丢失问题也值得研究。在下一步工作中,计划利用 p-Shapelet 构建多源同步时间序列分类器,并在 Logical Shapelet[16] 的基础上,进一步考虑 Shapelet 之间的时间间隔与时序关系。同时,将现有算法应用到实际的电力监测网络中,挖掘区域用电规律以及对故障传感器进行探测。可见随着子序列长度与阈值指数型增长,无法在一定硬件环境下应对大规模数据处理,因此当前算法仍需改进。故应该设计更为高效的 Shapelet 学习算法,避免目前大量枚举与遍历带来的计算开销。

参考文献:

[1] 常慧君,单洪,满毅,等. 基于时间序列分解的用户行为分析[J]. 数据采集与处理, 2015, 30(2): 441-451
Chang Huiun, Shan Hon, Man Yi, et al. User behavior analysis based on decomposition of time-stamp sequence[J]. Journal of Data Acquisition and Processing, 2015, 30(2): 441-451.

[2] 夏利,王建东,张霞,等. 聚类再回归方法在机场噪声时间序列预测中的应用[J]. 数据采集与处理, 2014, 29(1): 152-156
Xia Li, Wang Jiandong, Zhan Xia, et al. Application of cluster regression in time series prediction of airport noise[J]. Journal of Data Acquisition and Processing, 2014, 29(1): 152-156.

[3] 霍铖宇,倪黄晶,宁新宝. 心率变异时间序列的预处理算法[J]. 数据采集与处理, 2013, 28(5): 591-596
Huo Chengyu, Ni Huangjing, Ning Xinbao. Processing method for heart rate variability time series[J], Journal of Data Acquisition and Processing, 2013, 28(5):591-596.

[4] Ding H, Trajcevski G, Scheuermann P, et al. Querying and mining of time series data: Experimental comparison of representations and distance measures [C]//Proceedings of the 34th VLDB. Berlin, Germany: Springer, 2008, 1(2): 1542-1552.

[5] Salzberg, S. L. On comparing classifiers: Pitfalls to avoid and a recommended approach [J]. Data Mining and Knowledge Discovery, 1997, 1(3): 317-328.

[6] Xi X, Keogh E, Shelton C, et al. Fast time series classification using numerosity reduction [C]//Proceedings of the 23rd International Conference on Machine Learning. New York, USA: ACM, 2006: 1033-1040.

[7] 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. New York, USA: ACM, 2009: 947-956.

[8] 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. New York, USA: ACM, 2012: 289-297.

[9] Zakaria J, Mueen A, Keogh E. Clustering time series using unsupervised-Shapelets [C]//Proceedings of 12th IEEE International Conference on Data Mining. New Jersey, USA: IEEE, 2012: 785-794.

[10] 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.

[11] Rakthanmanon T, Keogh, E. Fast Shapelets: A scalable algorithm for discovering time series Shapelets [C]//Proceedings of the 13th SIAM Conference on Data Mining. Philadelphia, USA: SIAM, 2013: 668-676.

[12] He Q, Dong Z, Zhuang F, et al. Fast time series classification based on infrequent Shapelets [C]//Proceedings of 11th International Conference on Machine Learning and Applications. New Jersey, USA: IEEE, 2012: 215-219.

[13] Cetin M S, Mueen A, Calhoun V D. Shapelet ensemble for multi-dimensional time series [C]//Proceedings of the 15th SIAM International Conference on Data Mining. Philadelphia, USA: SIAM, 2015.
[14] Ulanova L, Begum N, Keogh E. Scalable clustering of time series with U-shapelets [C]// Proceedings of the 15th SIAM International Conference on Data Mining. Philadelphia, USA: SIAM, 2015.
[15] Xing Z, Pei J, Philip, S. Y., et al. Extracting interpretable features for early classification on time series [C]//Proceedings of the 11th SIAM International Conference on Data Mining. Philadelphia, USA: SIAM, 2011: 247-258.
[16] 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, USA: ACM, 2011: 1154-1162.
[17] Breiman L, Friedman J, Stone C J, et al. Classification and regression trees[M]. Boca, Raton, Florida: CRC Press, 1984.
[18] CMU graphics lab motion capture database? [EB/OL]. http://mocap.cs.cmu.edu/,2015-06-10.

作者简介:

李钟麒(1991-),男,博士研究生,研究方向:数据挖掘,E-mail:zqli.scu@foxmail.com。

段磊(1981-),男,副教授,硕士生导师,研究方向:数据挖掘,健康信息学,进化计算,E-mail:leiduan@scu.edu.cn。

胡斌(1980-),男,高级工程师,研究方向:电力信息化,大数据及云计算技术在电力行业的应用,E-mail:123067658@qq.com。

邓松(1980-),男,博士,高级工程师,研究方向:分布式数据挖掘、智能电网信息安全、电力信息物理融合系统,E-mail:ds16090311@163.com。

秦攀(1991-),男,硕士研究生,研究方向:数据挖掘,E-mail:panqin@stu.scu.edu.cn。