基于主成分分析的时间序列 Shapelet 提取方法①

李祯盛, 何振峰

(福州大学 数学与计算机科学学院, 福州 350100)

摘 要: Shapelet 序列分析为时间序列分类提供了一种快速分类的方法, 但 Shapelet 序列抽取速度很慢, 限制了它的应用范围. 为了加快 Shapelet 序列的提取, 提出了一种基于主成分分析的改进方法. 首先运用主成分分析法(PCA)对时间序列数据集进行降维, 采用降维后的数据表示原数据, 然后对降维后的数据提取出最能代表类特征的 Shapelet 序列. 实验结果表明: 本方法在保证分类准确率的前提下, 提高了运算速度.

关键词: 主成分分析; 时间序列; Shapelet; 降维

Time Series Shapelet Extraction Based on Principal Component Analysis

LI Zhen-Sheng, HE Zhen-Feng

(School of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

Abstract: Shapelet provides a fast classification method in time series classification, but the extraction of time series Shapelet is so slow that it restricts the application of the Shapelet. In order to speed up the extraction of time series Shapelet, an improved method is proposed based on the principal component analysis. Firstly, it uses the principal component analysis (PCA) to reduce the dimension of time series data set and chooses the reduced data to represent the original data. Secondly, it can extract the most discriminatory Shapelet sequence from the reduced data. Lastly, the experimental results show that the improved method improves the speed of the extraction and ensures the accuracy of classification.

Key words: principal component analysis; time series; Shapelet; dimensionality reduction

1 引言

在数据挖掘领域, 时间序列的分类是一个非常广泛的主题, 在很多领域中都得到了应用[1]. 目前已经提出了很多时间序列分类解决方案, 例如: 时间序列的 Shapelet, 时间序列的模式抽取, 基于时间序列合并的变换, 时间序列的相似性研究等[2][3][4][5]. 其中, Shapelet 方法分类速度更快, 分类更准确, 空间更节省, 同时具有更好的解释性, 所以它在解决时间序列分类问题中具有更大的潜力[2].

Shapelet 最先由 Ye, L, Keogh, E 等在 2009 年引入[6], Shapelet 序列是能够代表时间序列类特征的子序列, 通过这个子序列, 能够对未知类的时间序列进行标记. 该篇文章的作者通过计算数据集的每个候选序列的信息增益, 选取信息增益达到最大的候选序列作为 Shapelet 序列, 然后以 Shapelet 序列为基础构造出一个决策树分类器, 但是信息增益的计算量过大, 为此, Lines 等人提出了用 Kruskal-Wallis 统计测试方法来作为候选序列的质量度量措施[7]. 上述方法虽然避免了信息增益方法中对每个可能的分割点的测试, 但运算时间还是很长, 这是因为候选序列的数量非常庞大. 针对提取 Shapelet 序列计算量大的问题, Ye, L, Keogh, E 等提出了一种早起去除的方法, 它可以在计算候选序列到各个时间序列的距离时, 储存一个当前的最短距离, 在后续的计算中, 一旦当前点的距离大于最短距离, 则计算马上终止[2]. 同时, 由于每个候选序列都要计算到每一个时间序列间的距离, 那么可能会有重

① 收稿时间:2014-03-06;收到修改稿时间:2014-04-01

复计算的情况发生. 比如, 当一个以点 a 开始的候选序列与点 b 开始的子序列进行计算时, 可能存在以点(a-1)开始的候选序列与(b-1)开始的子序列的距离已经被计算过了, 那么此时再计算以点 a 开始的候选序列与点 b 开始的子序列的距离时就会进行重复计算, 为此, Abdullah 等在提取 Shapelet 序列时, 预先存储已经计算过的距离, 当产生重复计算时, 则可以直接调用, 从而达到减少计算量的效果, 但这种方法会使得空间的消耗大量的增加[8].

针对上述问题, 本文提出了基于主成分分析的时间序列的 Shapelet 序列提取方法 PShapelet, 它通过对数据集进行降维, 缩短了时间序列, 从而减少了候选序列. 基本步骤为: 首先运用主成分分析法对数据集进行降维, 然后对降维后的数据进行 Shapelet 序列的提取.

2 基于主成分分析的Shapelet序列提取

在Shapelet序列的提取过程中, 主要的时间消耗是在计算每个候选序列到时间序列数据集之间的距离, 假设在数据集T中含有n个时间序列, 这些时间序列的最大长度为m, 那么在T中的候选子序列的总数为$O(nm^2)$. 对每一个候选子序列, 在计算它到T中的n个时间序列的距离所用的时间为$O(nm^2)$. 所以, 提取Shapelet序列的算法的时间复杂度为$O(n^2m^4)$. 可以看出, 通过缩短时间序列的长度能够很好的提高Shapelet序列的提取速度. 假设运用某种方法使得时间序列的长度变成$\frac{m}{k}$, 那么算法的时间复杂度就会变为$O(n^2\frac{m^4}{k^4})$, 加速的效果是很明显的. 由于Shapelet序列的提取方法中计算时间序列间距离的方法是欧氏距离, 不用像DTW那样考虑时间序列的偏移, 那么, 我们可以选取主成分分析法(PCA)作为缩短时间序列的方法. 同时, 相关研究也证明了主成分分析法是可以应用于时间序列的[9][10],因此, 主成分分析的方法在这里是可行的.

2.1 主成分分析法

主成分分析法[11](PCA)是一种通用的降维方法. 本质上讲, PCA 就是将高维数据通过线性变换投影到低维空间上去, 并且投影后的数据不能失真, 也就是说, 被 PCA 降掉的那些维度只能是那些噪声或是冗余的数据. 假设有一个样本集 X, 里面有 N 个样本, 每个样本的维度为 r,即:

$$X=\{X_1,X_2,....,X_N\}\text{其中，}X_i=(x_{i,1},x_{i,2}...x_{i,r}),i=1,2...,N$$

将这些样本组成矩阵的形式, 每行一个样本, 每列一个维度, 得到样本矩阵 Q. 首先, 先将样本矩阵中心化为 Q_1, 即保证每个维度的均值为 0; 然后对样本矩阵计算协方差矩阵 M, 计算公式如下:

$$M=\frac{Q_1^TQ_1}{N-1},M\in R^{r\times r}$$

然后, 找到正交矩阵 P,满足: $P^TDP=\Lambda$,即对 M 对角化. 其中特征值矩阵为 Λ, 特征向量矩阵为 P. 假如, 选取最大的前 p 个特征值对应的维度, 那么这 p 个特征值组成了新的对角矩阵 $\Lambda_1\in R^{p\times p}$, 对应的 p 个特征向量组成了新的特征向量矩阵 $P_1\in R^{r\times p}$, P_1 即为投影矩阵. 最后, 用矩阵 Q_1 右乘投影矩阵 P_1, 即得到了降维后的新样本矩阵. 主成分分析法在模式识别, 单变量时间序列, 多元时间序列上的应用都取得了一定的成果[9,10,12,13].

2.2 基于主成分分析的 Shapelet 序列的提取

主成分分析法在 Shapelet 序列提取中的应用是通过把数据集投影到低维的空间中, 缩短了时间序列, 使得候选子序列的数量大大减少, 为 Shapelet 序列的提取减少了很多计算量. 从而, 提出了基于主成分分析的Shapelet序列提取的方法(PShapelet), 该方法首先对原始的时间序列数据集进行主成分的提取, 达到去除噪声和冗余数据的目的, 并且减少了原始时间序列的数据量, 然后再对降维后的时间序列进行 Shapelet 提取, 具体的步骤为:

输入: 训练集 T;

输出: Shapelet 序列集合 best 和分割点集 sp 构建的决策树

1) 对 T 运用主成分分析法(PCA)降维, 得到新的训练集 D, 同时初始化一个未处理的训练集的集合 $K=\{\varnothing\}$, 并把 D 加入到 K 中.

2) 若 K 为空, 则终止训练. 否则取 K 中的一个元素 Ki, 如果 Ki 中的序列都在同一类 c 中, 则说明 Ki 不用分割, 将其移出 K, 否则对 Ki 进行 Shapelet 序列的提取, 得到一个 Shapelet 序列和一个分割点组合成的二元组($Shapeleti$, spi), 它把 Ki 划分为 Ki_1, Ki_2, 把 Ki_1, Ki_2 加入集合 K 中, 同时把二元组($Shapeleti$, spi)分别加入到 $best$ 和 sp 中, 并把其当做决策属性.

3) 最后, 用 Shapelet 序列集合 *best* 和分割点集 *sp* 构建出类似图 1 的决策树.

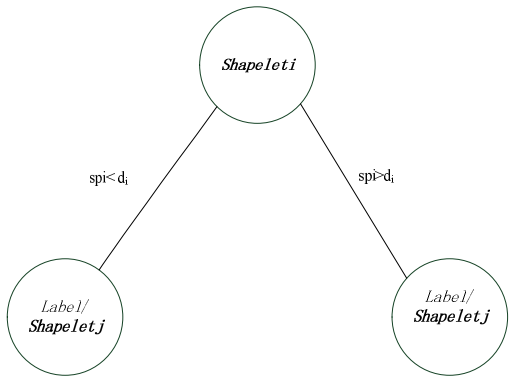

图1 决策树模型

图 1 中的 *Shapeleti* 和 *spi* 是一个 Shapelet 序列和与它对应的分割点, *di* 为需要分类的时间序列与 *Shapeleti* 间的距离. 对决策属性{*Shapeleti, spi*}分割出的节点, 如果不是终止节点(类标号节点 *label*), 那么就是另外一个决策属性节点{*Shapeletj, spj*}.

3 实验结果及分析

3.1 **Shapelet** 序列的提取

Shapelet 序列的提取的主要步骤为: 1)从训练集中生成 Shapelet 序列的候选序列集合, 即候选集; 2)计算候选集中的每个序列和数据集中每个样本之间的距离(Shapelet 距离), 然后对这些距离进行排序; 3)运用质量度量措施对每个候选序列进行评价, 评价最好的就是 Shapelet 序列. 这里, 不同的质量度量措施有不同的评价方法.

3.1.1 候选集的生成

对于一个含有 n 个时间序列的数据集 $T=T_1,T_2,\ldots..T_n$, 其 Shapelet 序列候选集是各个序列候选序列集的并集. 在数据集 T 中, 对于长度为 m 的时间序列 $T_i=t_{1,i},t_{2,i},t_{3,i},\ldots,t_{m,i}$,其候选序列的长度可以为 *1,2…m*, 其中长度为 l 的候选序列有 $(m-l+1)$ 条, 起点为 p, 长度为 l 的候选序列可以记为 $S_{p,i}=t_{p,i},t_{p+1,i}\ldots.,t_{p+l-1,i}$, $1\le p\le m-l+1$, $l\le m$. 故时间序列 T_i 中所有长度为 l 的子序列集合为 $S_{T_i}^{l}=\{S_{p,i},1\le p\le m-l+1\}$, 从而 T_i 的 Shapelet 候选序列的集合为 $w_i=\bigcup_{l=1}^{m}s_{T_i}^{l}$, 数据集 T 的候选集为 $w=\bigcup_{i=1}^{n}w_i$.

3.1.2 Shapelet 距离的计算

一般采用欧氏距离来度量时间序列的子序列间的距离. 两个长度都为 l 的子序列 S 和 R 之间的欧氏距离为:

$$dist(S,R)=\sum_{i=1}^{l}(s_i-r_i)^2$$

一个时间序列 T_i 和一个长度为 l 的子序列 S 之间的距离为子序列 S 和 T_i 中所有长度为 l 的子序列之间的距离的最小值:

$$d_{i,s}=\min_{R\in S_{T_i}^{l}}dist(S,R)$$

其中 $S_{T_i}^{l}$ 为 T_i 中长度为 l 的子序列的集合. 那么子序列 S 到数据集 T 中每一个时间序列的距离的集合可以表示为 $D_S^{'}=\{d_{1,s},d_{2,s}\ldots..,d_{n,s}\}$, n 为 T 中时间序列的个数. $D_S^{'}$ 中的数据是与 T 中的时间序列一一对应的, 即 $D_S^{'}$ 的第 i 个数据与 T 中的第 i 个时间序列属于同一类, 对 $D_S^{'}$ 进行排序可以得到排序后的距离的集合 D_S.

3.1.3 评价 Shapelet 候选序列

对 Shapelet 候选序列进行评价的方法主要有信息增益法(Information Gain)和统计测试的方法. 在统计测试中, Kruskal-Wallis(KW)统计测试对 Shapelet 候选序列的评价更具有代表性.

1)信息增益

信息增益[2](IG)是用以度量两种不同概率分布之间的差异的非对称度量措施, 被广泛应用于机器学习的领域. 一般在处理分类问题时, 信息增益的计算是针对数据的属性而言的, 可以用来衡量各属性含有信息量的大小. 在 Shapelet 应用中, 候选序列 S 的质量是基于排序后的 D_S 和通过计算每一个可能的分割点 sp 处的信息增益来评估的. 其中可能的分割点 *sp* 是指在 D_S 中两个连续的距离之间的平均值. IG 的计算方法为: D_S 中小于 sp 的元素放入 A_S 中, 其它的元素放入 B_S 中. 在 *sp* 点处的信息增益的计算公式如下:

$$IG(D_S,sp)=H(D_S)-\frac{|A_S|}{|D_S|}H(A_S)-\frac{|B_S|}{|D_S|}H(B_S)$$

其中 $|A_S|$ 是集合 A_S 的基数, $H(A_S)$ 是 A_S 的熵. 熵的计算如下:

$$H(D)=-\sum_{c\in class\{D\}}p_c\log_2 p_c$$

其中 p_c 是集合 D 中类标号为 c 的数据的概率. 那么候选序列 S 的信息增益可以用相关系数 *infos* 来表示:

$$infos=\max_{sp\in D_S}IG(D_S,sp)$$

当一个候选序列 S 的 *infos* 达到最大时, 那么该候选序列即为 Shapelet 序列.

2)Kruskal-Wallis 统计测试

Kruskal-Wallis[14](KW)是一种观察数据是否来源于一个单一分布的非参数测试, 通过这种方法计算出来的统计数字代表了一个类秩和全局平均秩之间的平方加权的差异. 在 Shapelet 的应用中, 子序列 S 的 KW 计算公式如下:

$$KW_S = \frac{12}{|D_S|.(|D_S|+1)}\sum_{i=1}^{k}\frac{R_i^2}{n_i} - 3(|D_S|+1)$$

其中 $|D_S|$ 为 D_S 的基数, k 为 D_S 中类的个数, n_i 是 D_S 中属于类 i 的实例的个数, R_i 是属于类 i 的实例的秩的总和. KW 值最大的候选序列即为 Shapelet 序列, 在提取 Shapelet 序列时, 当所使用的候选序列进行评价的方法是 Kruskal-Wallis 统计测试方法时, 不用像信息增益的方法那样对候选序列的每一个可能的分割点进行测试, 只需要计算一次候选序列的 KW, 因此减少了计算量[7].

3.2 实验数据和评价方法

本文选用的时间序列数据集来自 UCI, 分别为 SonyAIBORobotSurfaceII, ECGFiveDays, Coeff, CBF TwoLeadECG 和 SonyAIBORobotSurface. 分别在这些数据集上使用 PShapelet 和 Shapelet 方法对训练集提取 Shapelet 序列后对测试集进行分类, 在运算时间和分类错误率上进行了比较, 实验得到的运算时间为处理一个数据集所得到的的总时间, 包括训练时间和测试时间.

3.3 Shapelet 分类的实现

Pshapelet 方法在实验中使用留一法把原时间序列数据集分割出训练集和测试集, 即依次选取数据集中的一条时间序列作为测试集, 其它为训练集. 运用 Pshapelet 方法提取 Shapelet 序列, 并通过其对数据进行分类的主要步骤为: 1)用主成分分析法对训练集和测试集降维, 其中主成分分析的累计贡献率至少为 90%; 2)对训练集提取出候选集, 候选序列的最小长度分别为 3, 最大长度为时间序列长度的一半, 然后对每个候选序列计算其到训练集中的每个时间序列的距离, 排序后得到 D_S, 运用 KW 对每个 D_S 进行评价, 进而得到 Shapelet 序列. 然后用信息增益法对 Shapelet 序列的 D_S 进行分析, 找到最好的分割点, 进而构造出类似图 1 所示的决策树; 3)用构建好的决策树作为分类器对测试集进行分类, 得到分类结果和运算时间; 4)最后, 将每次实验的分类结果和运算时间进行统计, 即可得到该方法在这个数据集上的分类错误率和运算时间. Shapelet 方法在实验室中的步骤与 Pshapelet 方法是一样的, 唯一的区别就是 Shapelet 方法在选取训练集和测试集后没有对它们进行 PCA 分析.

3.4 实验结果及分析

本文所使用的实验平台为: MATLAB2010, 硬件配置为: 奔腾双核 CPU(2.5GHz), 2G 内存. 各个数据集使用 Shapelet 方法和 PShapelet 方法所得到的分类错误率和 PShapelet 相对于 Shapelet 方法所能达到的加速比如表 1 所示, 表中 Sony1 指的是 SonyAIBORobot SurfaceII 数据集, ECG1 指的是 ECGFiveDays 数据集, Sony2 指的是 SonyAIBORobotSurface 数据集, ECG2 指的是 TwoLeadECG 数据集.

表 1　2 种方法在数据集上错误率和时间的比较

数据集	错误率		运算时间(s)		加速比
	Shapelet	PShapelet	Shapelet	PShapelet	
Sony1	0.148	0.185	3492	10	349
Coeff	0.170	0.038	118702	14	8478
ECG1	0.130	0.260	19738	7	2820
Sony2	0.300	0.350	1787	32	56
ECG2	0.130	0.217	3713	24	154
CBF	0.100	0.200	25941	138	188
	0.163	0.208			2007

表 1 的最后一行表示这两个算法对这 6 个数据集分类的平均错误率和 PShapelet 方法相对于 Shapelet 序列的提取方法在这 6 个数据集上达到的平均加速比, 表中每个数据集的运算时间指的是使用留一法分割该数据集后, 分别使用原始的 Shapelet 提取方法和 PShapelet 方法在训练集上提取 Shapelet 序列, 并且对测试集分类后所消耗的总时间. 在表 1 中, 除了 Coeff 数据集, 其它数据集在使用 PShapelet 方法时, 错误率

会有小幅上升，但大大降低了运算时间，从实验结果可以看出，使用 Pshapelet 方法处理数据集，减少了候选序列的数量，从而加快了 Shapelet 序列的提取速度，提高了分类的性能. 原来的 Shapelet 序列的提取方法虽然具有很多优点，但由于训练的时间比较长，从而限制了它的应用范围，而 PShapelet 方法极大的提高了 Shapelet 序列的提取速度，虽然错误率会有一定的增加，但是在错误率不是非常敏感的场合，PShapelet 方法是很实用的.

4 结语

Shapelet 方法存在的主要问题就是它在进行 shaplelet 序列提取时的运算时间过长. 本文针对上述问题，先对数据集进行主成分分析，缩小了时间序列，使得 Shapelet 候选序列的数量大大减少，从而达到减少运算时间的目的. 实验结果表明，相对于 Shapelet 方法，PShapelet 方法虽然分类的准确率会小幅度下降，但运算时间却有了显著的提高.

参考文献

1 杨一鸣,潘嵘,潘嘉林,杨强,李磊.时间序列分类问题的算法比较.计算机学报,2007,30(8):1259–1265.

2 Ye L, Keogh E. Time series Shapelets: A novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery, 2011, 22(1): 149–182.

3 Bagnall A, Davis L, Hills J, Lines J. Transformation based ensembles for time series classification. The 12th SIAM International Conference on Data Mining. USA. 2012. 307–318.

4 Pierre G. Pattern extraction for time series classification. The European Conference on Principles and Practice of Knowledge Discovery in Databases. Freipurg, Germany. 2001. 115–127.

5 汤胤.时间序列相似性分析方法研究.计算机工程与应用,2006,42(1):68–71.

6 Ye L, Keogh E. Time series Shapelets: A new primitive for data mining. The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris. 2009. 947–956.

7 Jason L, Anthony B. Alternative quality measures for time series Shapelets. Intelligent Data Engineering and Automated Learning, 2012, 74(35): 475–483.

8 Abdulah M, Eamonn K, Neal Y. Logical-shapelets: An expressive primitive for time series classification. The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA. 2011. 1154–1162.

9 苏木亚,郭崇慧.基于主成分分析的单变量时间序列聚类方法.运筹与管理,2011,20(6):66–72.

10 Kiyoung Y, Cyrus S. A PCA-based similarity measure for multivariate time series. The 2nd ACM International Workshop on Multimedia Databases. Washington, DC, USA. 2004. 65–74.

11 Herve A, Lynne JW. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4): 433-459.

12 磨少清,刘正光,张军.基于图像质量和 PCA 子空间车标识别方法.计算机应用,2010,30(8):2244–2246.

13 李正欣,郭建胜,惠晓滨,宋飞飞.基于共同主成分的多元时间序列降维方法.控制与决策,2013,28(4):531–535.

14 Kruskal W. A nonparametric test for the several sample problem. The Annals of Mathematical Statistics,1952, 23(4): 525–540.