软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(10):3216–3237 [doi: 10.13328/j.cnki.jos.005852]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种时间序列鉴别性特征字典构建算法*

张 伟, 王志海, 原继东, 郝石磊

(北京交通大学 计算机与信息技术学院,北京 100044)

通讯作者: 原继东, E-mail: yuanjd@bjtu.edu.cn

摘 要: 时间序列数据广泛产生于科技和经济的多个领域.基于符号傅里叶近似(symbolic Fourier approximation)和滑动窗口的定长单词抽取算法是目前时间序列特征字典构建过程中最有效的特征生成算法之一,但是该算法在特征生成过程中不能根据不同滑动窗口长度动态地选择保留的最优傅里叶值的个数,而且特征字典构建过程中缺少从生成的海量特征中对鉴别性特征进行有效选择的算法.为此,提出一种鉴别性特征字典构建算法.首先,提出一种针对不同长度滑动窗口学习最优单词长度的基于Fourier近似的可变长度单词抽取方法;其次,构建了一种新的特征鉴别性评价指标,并依据其动态阈值对生成的特征进行选择.实验结果表明,基于构建的特征字典的逻辑回归模型不仅分类精度高,而且可以有效发现预测过程中的鉴别性特征.

关键词: 时间序列分类;特征生成;鉴别性特征选择;特征字典学习

中图法分类号: TP311

中文引用格式: 张伟,王志海,原继东,郝石磊.一种时间序列鉴别性特征字典构建算法.软件学报,2020,31(10):3216–3237. http://www.jos.org.cn/1000-9825/5852.htm

英文引用格式: Zhang W, Wang ZH, Yuan JD, Hao SL. Time series discriminative feature dictionary construction algorithm. Ruan Jian Xue Bao/Journal of Software, 2020,31(10):3216–3237 (in Chinese). http://www.jos.org.cn/1000-9825/5852. htm

Time Series Discriminative Feature Dictionary Construction Algorithm

ZHANG Wei, WANG Zhi-Hai, YUAN Ji-Dong, HAO Shi-Lei

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Time series data are widely generated in many fields of science, technology and economy. Time series feature generation algorithm based on Symbolic Fourier Approximation (SFA) and sliding window transformation mechanism is one of the most effective feature dictionary construction algorithms, but there are some obvious shortcomings in this kind of methods. Firstly, the number of optimal Fourier values cannot be dynamically selected for different sliding window lengths in the process of transformation. Secondly, there is a lack of effective algorithm to select discriminant features from the generated massive features. To this end, a new variable length feature dictionary building algorithm is proposed in this study. First, a variable length word extraction method based on SFA is proposed. The method dynamically selects the optimal number of Fourier values for different sliding window lengths. Second, a new feature discriminant evaluation indicator is designed, and the generated features are selected according to its dynamic threshold. Experimental results show that, based on the proposed time series dictionary, the logistic regression model can achieve high classification accuracy and find the discriminant features in the prediction process.

Key words: time series classification; feature generation; discriminant feature selection; feature dictionary learning

时间序列是一系列按时间进行排序的实值数据组成的集合.在许多研究领域或实际应用领域之中存在着大量的时间序列数据,例如恶意软件检测、风能预测、工业自动化、电压稳定评估、移动设备追踪等领域[1−3].

* 基金项目: 中央高校基本科研业务费专项资金(2018JBM014); 国家自然科学基金(61702030, 61672086); 北京市自然科学基金(4182052); 北京市优秀人才项目资助(2017000020124G056)

Foundation item: Fundamental Research Funds for the Central Universities (2018JBM014); National Natural Science Foundation of China (61702030, 61672086); Beijing Natural Science Foundation of China (4182052); Beijing Excellent Talents (2017000020124G056)

收稿时间: 2018-10-23; 修改时间: 2019-01-01; 采用时间: 2019-04-22

目前,获得的时间序列通常具有如下两个特点:时间序列数据集的规模很大,同时,每条时间序列数据的维度都很高.

时间序列分类(time series classification,简称 TSC)技术的研究涉及到许多方面的技术问题,这些问题可能包括时间序列特征的发现或生成以及如何存储或压缩时间序列等.Esling 等人对时间序列的研究领域给出了详细介绍[4].Bagnall 等人对各种时间序列分类算法进行了详细分析[5].一般 TSC 算法可大致分为两类:基于完整时间序列的方法和基于局部特征的方法[6].前者基于全局相似性进行分类预测,主要研究相似性的不同度量方式和使用方法[7−11];而后者基于时间序列的局部特征进行分类预测,通常利用不同的特征生成和转换技术来发现局部特征,然后基于建立的特征集合直接构建分类模型或对时间序列数据进行转换[12−15].然而,目前多数的 TSC 算法无法以足够的精度和速度充分处理大规模的数据.一些精度较高的分类算法,例如,基于局部鉴别性特征 Shapelet 的转换算法的时间复杂度为 $O(N^2\times n^4)$[16],其中,N 为时间序列数据集中的实例数,n 为时间序列的长度.基于时间序列转换的集成分类算法(the collective of transformation-based ensembles,简称 COTE)[17]的分类精度显著地优于常见的时间序列分类算法,但该算法中包含多个时间复杂度非常高的基分类器.

为了提高 TSC 算法的分类效率,时间序列的快速转换表示方法成为当前的一个研究热点.本文对基于特征包(bag-of-pattern,简称 BOP)的分类模型进行研究,这类模型具有分类精度高和运行速度快的特点.BOP 模型将时间序列分成一系列子结构,并将这些子结构作为离散化特征构建特征字典,最后将基于特征频数的向量作为模型训练和分类的基础.最早的特征包模型(bag-of-patterns using symbolic aggregate approximation,简称 BOP-SAX)[18]使用固定长度的滑动窗口和符号聚合近似技术(SAX)将每个窗口中的子序列转换成离散化特征,然后使用基于频数向量的欧几里德距离作为相似性度量方式,最后通过 1NN 分类器进行分类预测.Senin 等人利用 SAX 技术和向量空间模型构建了一种新的时间序列分类模型(symbolic aggregate approximation and vector space model,简称 SAX-VSM)[19],该模型基于词频-逆向文档频率(term frequency-inverse document frequency,简称 tf-idf)对符号化特征进行加权,同时使用余弦距离代替欧式距离进行相似性度量;此外,它只为每个类构建一个特征向量,而不是每个样本一个向量,这极大地缩短了模型的运行时间.Neuyen 等人[14]对基于 SAX 的转换方法的可变长度的单词进行了研究,并将序列学习算法用于转换后的时间序列分类问题.SAX 技术本质上仍然是在时域空间对时间序列进行处理,然而,实际中的一些问题在时域空间进行处理时会比较困难,而在频域空间更容易取得好的处理效果.例如,通常可在频域空间将代表噪声的频率成分去除来实现降噪.此外,频域中包含一些时域难以发现的鉴别性信息.

离散傅里叶变换(discrete Fourier transform,简称 DFT)[20,21]可将时域空间的时间序列转换为一组频域空间的不同频率的正弦波.因此,基于离散傅里叶变换的时间序列符号化表示方法受到各国研究人员的重视.Schafer 等人[22]提出用符号傅里叶近似(symbolic Fourier approximation,简称 SFA)来代替 SAX 对时间序列进行离散化表示.接着,Schafer 又提出了一种基于 SFA 的特征包算法(bag-of-SFA-symbol,简称 BOSS)[23],但该算法对时间序列离散化过程中只简单挑选前 l 个傅里叶值,未考虑傅里叶值的鉴别性.为此,Schafer 等人进一步提出一种用于时间序列的单词抽取方法(word extraction for time series classification,简称 WEASEL)[24],该方法用鉴别性较强的 top-l 个傅里叶值代替前 l 个傅里叶值对时间序列进行符号化,但是该方法存在一个明显的缺陷,其将所有长度子序列转化得到的单词设为定长,即,所有子序列经离散傅里叶变换后保留的傅里叶值的个数相同.然而,实际中不同周期的时间序列子序列所含有的鉴别性频域信息量可能不同,单一的固定长度会导致损失大量鉴别性信息或包含冗余信息.为此,针对目前基于 SFA 的时间序列离散化表示方法存在的问题,本文首先提出一种用于时间序列分类的基于 SFA 的可变长度单词抽取算法(variable-length word extraction algorithm,简称 VLWEA).该算法为每个窗口长度学习性能最优的单词长度.图 1 展示了本文提出的变长单词抽取算法相对于定长单词抽取算法所具有的优势.图中,w 表示滑动窗口长度,l_w 表示长度为 w 的滑动窗口中提取的单词长度,c_1 和 c_2 表示类别.

图 1(b)中给出了长度为 6 和 8 的滑动窗口获得的 4 个子序列经过 SFA 转换得来的完整字符序列.图 1(a)所示为用定长单词抽取算法得到的特征(假定共同单词长度为 4),其中,“6aabc”表示一个长度为 6 的滑动窗口子

序列抽取出的长为 4 的字符序列,即特征.图 1(c)中给出了变长单词抽取算法获得的两种长度滑动窗口对应的特征,变长单词抽取算法旨在获得具有更强鉴别性的子序列.从图 1 可以看出,定长单词“8abab”无法区分长为 8 的两类子序列,而变长单词“8ababc”和“8ababd”可区分两类子序列.此外,定长单词“6aabc”和“6acbc”虽可区分长度为 6 的两类子序列,但与变长单词“6aa”“6ac”相比,包含了更多的冗余成分.

Fig.1 Comparison of variable length and fixed length word extraction algorithm

图 1 变长单词生成和定长单词抽取算法比较

与此同时,针对基于 VLWEA 建立的特征字典中存在大量冗余特征的问题,本文基于 tf-idf 的基本思想定义了一种新的特征鉴别性强弱度量方式来对鉴别性特征进行选择,并能根据不同周期生成的特征的整体分类性能动态设定各周期生成特征的选择阈值.本文的主要贡献概括如下.

(1) 本文提出了一种基于网格搜索的滑动窗口单词长度学习算法.该算法为不同窗口长度学习分类性能最优的单词长度,不同窗口长度可能生成不同长度的单词.变长单词可有效减少鉴别性信息的损失,在此基础上可以获得更优的分类效果.与此同时,我们分析了 Bigrams 语法模型对基于变长单词特征字典的模型分类性能的影响.

(2) 本文基于 tf-idf 的基本思想定义了一种新的特征鉴别性评价统计量用于特征选择.

(3) 针对采用固定阈值进行特征选择忽略了不同周期转换得来的特征存在性能差异的问题,我们提出一种根据不同滑动窗口生成特征的整体分类性能以动态地设定各窗口生成特征选择阈值的机制,该机制可有效缩小时间序列特征空间的规模并提高模型的分类性能.

本文第 1 节介绍基本概念和理论基础.第 2 节介绍本文使用的相关算法.第 3 节给出实验方式和实验结果.第 4 节总结全文.

1 定义与理论基础

本节介绍时间序列的基本概念和基于 SFA 的时间序列转换表示方法的理论基础.

1.1 时间序列

下面我们首先介绍本文用到的一些基本概念.

定义 1(时间序列). 时间序列 T 是由 n 个有序的实际观测值 $t_0,t_1,\ldots,t_{n-1}$ 组成的一个实值序列,即 $T=\{t_0,t_1,\ldots,t_{n-1}\}$,其中,$t_i\in R$.

用 $D=\{T_0,T_2,\ldots,T_{N-1}\}$表示包含 N 条时间序列的数据集合,其中,每条时间序列 $T_i=\{t_{i1},t_{i2},\ldots,t_{in}\}$.

定义 2(时间序列子序列). 给定一条完整的时间序列 $T=\{t_0,t_1,\ldots,t_{n-1}\}$,该序列 T 中长为ω的窗口 S 之中包含的ω个连续值组成的序列称为时间序列 T 的子序列,若其在时间序列 T 上的起始位置为 a,则 $S(a,\omega)=(t_a,t_{a+1},\ldots,t_{a+\omega-1})$,其中,$0\leqslant a\leqslant n-\omega$.

定义 3(特征字典). 特征字典是用于表示时间序列数据的特征集合.

本文构建的特征字典中的每个元素为一个指定长度滑动窗口生成的字符序列.表 1 给出文中涉及的常用符号.

Table 1 Symbol table

表 1 符号表

符号	释义
D	时间序列集合
N	时间序列集合 D 中的实例数
T	任意一条时间序列
n	时间序列 T 的长度
min	滑动窗口最小长度
max	滑动窗口最大长度
min$F$	最小单词长度
max$F$	最大单词长度
k	交叉验证重数
$S(a,\omega)$	时间序列 T 中起始位置为 a 长为 ω的子序列
x_f	第 f 个傅里叶系数
$real_i$	第 i 个傅里叶系数的实部
$imag_j$	第 j 个傅里叶系数的虚部
$\boldsymbol{F}$	离散傅里叶变换矩阵
$dict$	数据集 D 的特征字典
$TfIdfDict$	基于 tf-idf 统计量建立的特征字典

1.2 时间序列离散傅里叶变换

本节我们介绍时间序列的离散傅里叶变换过程[20,21].

给定一条由 n 个数值组成的离散序列 $T=\{t_0,t_1,\ldots,t_{n-1}\}$,其离散傅里叶变换公式为

$$x_f=\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}t_i\mathrm{e}^{-\frac{2\pi j}{n}fi}=\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}t_iW^{fi},\ f=0,\ 1,\ldots,\ n-1 \tag{1}$$

其中,j 为虚数单位,即 $j=\sqrt{-1}$, W 为

$$W=\mathrm{e}^{-\frac{2\pi j}{n}}.$$

经公式(1)可将离散数值序列 T 转换为序列 $\boldsymbol{x}$,转换过程可表示为

$$\boldsymbol{x}=\begin{bmatrix}x_0\\x_1\\x_2\\\vdots\\x_{n-1}\end{bmatrix}=\frac{1}{\sqrt{n}}\begin{bmatrix}1&1&1&\cdots&1\\1&W^1&W^2&\cdots&W^{n-1}\\1&W^2&W^4&\cdots&W^{2(n-1)}\\\vdots&\vdots&\vdots&\ddots&\vdots\\1&W^{n-1}&W^{2(n-1)}&\cdots&W^{(n-1)(n-1)}\end{bmatrix}\begin{bmatrix}t_0\\t_1\\t_2\\\vdots\\t_{n-1}\end{bmatrix}.$$

上式可简单记作:

$$\boldsymbol{x}=\boldsymbol{F}T \tag{2}$$

其中,$\boldsymbol{F}$ 称为离散傅里叶变换矩阵.

矩阵 $\boldsymbol{F}$ 中第 i 行表示第 i 个正弦波,x_i 表示时间序列在这个正弦波上的投影,即,时间序列 T 中包含的第 i 个频率正弦波成分的多少.它反映了时间序列与该频率正弦波的相关性.

数据 $\boldsymbol{x}=\{x_0,x_1,\ldots,x_{n-1}\}$为经离散傅里叶变换得到的数据,称为频域向量.时间序列 T 可通过逆变换来恢复,即,

$$T=\boldsymbol{F}^H\boldsymbol{x} \tag{3}$$

其中,$\boldsymbol{F}^H$ 表示正交矩阵 $\boldsymbol{F}$ 的共轭转置.

定理 1(Parseval theorem)[25]. 若 $\boldsymbol{x}$ 为序列 T 经离散傅里叶变换得到的序列,则有

$$\sum_{i=0}^{n-1}|t_i|^2=\sum_{i=0}^{n-1}|x_i|^2.$$

上述定理说明序列 T 在时域空间的能量与在频域空间的能量相同.

时间序列的每个元素可以表示为

$$t_i=\frac{1}{\sqrt{n}}\sum_{f=0}^{n-1}x_f\mathrm{e}^{\frac{2\pi j}{n}fi},i=0,1,...,n-1.$$

复数 x_f 通常称作傅里叶系数(Fourier coefficient).

通过欧拉公式可以将公式(1)表示为

$$x_f=\sum_{i=0}^{n-1}t_i\mathrm{e}^{-\frac{2\pi j}{n}fi}=\sum_{i=0}^{n-1}t_i\left(\cos\left(\frac{2\pi f}{n}i\right)-j\sin\left(\frac{2\pi f}{n}i\right)\right),f=0,\ 1,\ ...,\ n-1.$$

傅里叶系数 x_f 可分为 $real_f$ 实数部分和 $imag_f$ 虚数部分:

$$real_f=\sum_{i=0}^{n-1}t_i\cos\left(\frac{2\pi f}{n}i\right),f=0,1,...,n-1,$$

$$imag_f=-\sum_{i=0}^{n-1}t_i\sin\left(\frac{2\pi f}{n}i\right),f=0,1,...,n-1.$$

经过傅里叶变换,我们得到的傅里叶系数值序列可以表示为

$$\boldsymbol{x}=(real_0,imag_0,real_1,imag_1,...,real_{n-1},imag_{n-1}) \tag{4}$$

由于 $real_{n-f}=real_f,imag_{n-f}=-imag_f,imag_0=0$,所以上述 $2n$ 维数组可用 n 维数组表示.

当 n 为偶数时,$imag_{n/2}=0$,$\boldsymbol{x}$ 可以表示为

$$\boldsymbol{x}=(real_0,real_{(n-1)/2},real_1,imag_1,...,real_{n/2-1},imag_{n/2-1}) \tag{5}$$

当 n 为奇数时,$\boldsymbol{x}$ 可以表示为

$$\boldsymbol{x}=(real_0,imag_{(n-1)/2},real_1,imag_1,...,real_{(n-3)/2},imag_{(n-3)/2},real_{(n-1)/2}) \tag{6}$$

考虑到傅里叶值的对称性,为了减少计算量和存储空间,一般用 n 维数组对傅里叶值进行存储.由于 $\boldsymbol{x}$ 中的傅里叶值代表了固定周期内的时间序列的频率成分,因此,符号傅里叶的近似过程就是对这些傅里叶值进行选择和符号化转换.下面我们介绍基于 SFA 的特征生成过程.

1.3 基于SFA的特征生成

基于 SFA 的特征生成过程分为两步:首先,从子序列转换得到的傅里叶值数组中选择鉴别性最强的 top–l 个傅里叶值;其次,利用分箱技术将选择的傅里叶值依序转换为符号组成单词,即,特征.

傅里叶值的鉴别性度量本质上是根据一系列标定类属性的实值来判断这组值对应的频率成分的鉴别性强弱.Schafer 等人[24]提出利用 F 统计量对傅里叶值的鉴别性强弱进行度量,然后选择鉴别性最强的前 l 个傅里叶值.这样处理比简单挑选 top–l 个傅里叶值生成的特征更具有鉴别性.本文我们也使用 F 统计量进行傅里叶值鉴别性度量,与 Schafer 所提方法不同的是,我们为每个窗口长度选择分类性能最优的傅里叶值个数进行特征生成,而 Schafer 等人使用的方法所有滑动窗口都选择鉴别性 top–l 个傅里叶值.计算 $real_i$ 或 $imag_j$ 对应的实数集上的 F 统计量的类内方差 MS_W 和类间方差 MS_B 的计算公式[24]为

$$MS_W=\frac{1}{q-k}\sum_c\sum_{x\in Q_c}|x-\overline{x}_c|^2,$$

$$MS_B=\frac{1}{k-1}\sum_c q_c|\overline{x}_c-\overline{x}|^2,$$

其中,Q 为 $real_i$ 或 $imag_j$ 对应的傅里叶值集合,q 为集合 Q 中实数的个数,k 为集合 Q 中的类属性取值个数,x 为集合 Q 中的任意实数.Q_c 为集合 Q 中类属性值为 c 的傅里叶数值集合,q_c 为集合 Q_c 中实数的个数,$\overline{x}$ 为集合 Q 中所有实数的均值,$\overline{x}_c$ 为 Q_c 中所有实数的均值.

F 值的计算公式为

$$F=\frac{MS_B}{MS_W} \tag{7}$$

当 F 值用于傅里叶值选择时,我们希望找到最大的 F 值,其等价于不同类均值之间的最大差,此时,傅里叶值的鉴别性最强.

例如,给定一个长为 10 的子序列,假设我们选择的前 4 个傅里叶值为“0.11,0.23,0.02,0.63”,利用分箱技术我们可将该实值序列转换成为一个字符序列,假设为“abcd”,由于不同长度的滑动窗口代表不同周期的频率成分,它们生成的特征不同,为此,每个单词都对应固定的滑动窗口长度,上述生成的单词通常表示为“10abcd”.Schafer在文献[23]中对 SFA 的具体符号化过程进行了详细介绍,这里不再赘述.

n_0 元语法模型(n_0-gram)是自然语言处理中很重要的统计语言模型.该模型在实际应用中通常假设某个词出现的概率只与它前面的一个或几个词有关,即,马尔可夫假设.当 $n_0=i$ 时(i 为正整数),称为 i 元语法,也称为 i 阶马尔可夫链,此时第 j 个词出现的概率只与其前 i–1 个词有关.n_0 取 1、2 和 3 时,基于 n_0 元语法表示获得的词组称为一元单词(unigram word)、二元单词(bigram word)和三元单词(trigram word).此外,若特征字典有 m 个特征,则基于 n_0 元语法要考虑的词组合可能有 $O(m^{n_0})$ 个.因此,为了缩小特征字典的规模,通常只考虑一元和二元词组组成的特征.为了弥补所有周期抽取定长单词所导致的鉴别性信息的损失,Schafer 等人[24]将 n_0 元语法用于时间序列的特征构建过程中,他们将连续出现的两个特征组成一个新的特征加入到特征字典.本文将一元和二元语法模型分别记为 Unigram 和 Bigrams,并对 Bigrams 语法模型对变长单词特征字典的性能影响进行了实验分析.

1.4 特征的鉴别性评价

tf-idf 是一种用于寻找文本中关键词的统计方法[26,27].它通常用来评估一个词与一篇文档的主题的相关程度.词对一篇文档的重要性与其在该文档中出现的频率成正比关系,同时,与其在语料库中出现的频率成反比.如果某个词在所有的文档中都出现,则意味着与主题并不相关.本文基于 tf-idf 的基本思想设计了一种新的 tf-idf 统计量对特征的鉴别性进行评价.新定义的鉴别性评价指标主要从两个方面对各类特征的鉴别性进行评价.

(1) 某类特征和该类的相关程度.主要通过 tf 值来度量,我们用某类特征在该类所有实例中的出现频率来度量该特征和所属类别的相关性.

(2) 某类特征对该类的鉴别性强弱.实际中,某类中高频特征也可能是其他类中的常见特征.此时,该特征不能有效区分不同类别.为此,我们用 idf 度量该特征对所属类别的鉴别性强弱.

下面首先对本文定义的 tf-idf 计算公式进行介绍.我们用特征在某类中出现的相对频率代替其在某个实例中出现的频率以衡量它与某类实例的相关程度.我们用 $f(t,c)$表示特征 t 在其类属性 c 对应的实例集合中出现的次数,即,类属性为 c 的特征 t 在类属性为 c 的所有实例中出现的次数总和.特征 t 在其类属性 c 对应的特征字典中的频率 $tf(t,c)$的计算公式为

$$tf(t,c)=\frac{f(t,c)}{\max\limits_{p\in dict(c)} f(p,c)},$$

其中,$dict(c)$表示类属性 c 对应的特征字典,p 表示任意特征.

我们在自然对数尺度下对特征频率进行比较分析,对频率公式 $tf(t,c)$进行如下处理:

$$tf(t,c)=\ln(1+tf(t,c)) \tag{8}$$

如果 $tf(t,c)$值越大,则特征 t 在类属性 c 对应的实例中出现的频率越高,意味着它与类属性越相关,但不能说明它对于类属性 c 的鉴别性越强.为了准确度量特征的鉴别性,我们提出一个新的 idf 计算公式.在定义的 idf 公式的分子中用实例总数减去类属性为 c 的实例数,即,只考虑类属性不为 c 的实例中包含特征 t 的情况;idf 公式分母中只计数类属性不为 c 且包含特征 t 的实例数,这样可以直接反映特征 t 在其他各类实例中的出现频率.我们使用的类属性为 c 的特征 t 的逆文档频率 $idf(t,c)$的计算公式为

$$idf(t,c)=\ln\frac{N-N_c+1}{df(t,\overline{c})+1} \tag{9}$$

其中,N 为实例总数,N_c 为类属性为 c 的实例数,$df(t,\overline{c})$ 为数据集中包含特征 t 同时类属性不为 c 的实例数.$idf(t,c)$值越大,说明类属性为 c 的特征 t 在其他类中出现的频率越低.基于 tf-idf 的基本思想,我们定义的类属性为 c 的特征 t 的鉴别性度量值 $d(t,c)$的计算公式为

$$d(t,c)=tf(t,c)\times df(t,c) \tag{10}$$

上式说明:类属性为 c 的特征 t 在类属性为 c 的实例中出现的相对频率越高($tf(t,c)$值越大),而在其他类实例

中出现的次数越少($idf(t,c)$值越大),则该特征对类属性 c 的鉴别性越强.即,类属性为 c 的特征 t 的 $d(t,c)$值越大,其对类属性 c 的鉴别性越强.为了给出我们定义的特征的鉴别性度量公式的直观解释,下面我们通过一个计算实例来与其他研究人员常使用的 tf-idf 公式进行对比.

Table 2 A calculation example of formula tf-idf

表 2 tf-idf 公式计算实例

	t_1	t_2	t_3	N_i
c_1	5	10	90	100
c_2	50	100	2	100

表 2 中 t_i 表示第 i 个特征,其所在列中数字表示包含该特征的各类别实例数,c_i 表示第 i 类,N_i 表示数据集中第 i 类实例的个数.这里,为了计算方便,我们假定每个特征在每个包含该特征的实例中的出现频数为 1.

Senin 等人[19]和 Schafer[28]使用的 tf-idf 计算公式虽有小的不同,但本质上是一致的.这里,我们使用 Schafer 在文献[28]中的公式进行说明.基于文献[28]中的公式,特征 t_1 对类属性 c_1 的 tf 值为该特征在 c_1 类中频数取对数加 1,即,$tf(t_1,c_1)=1+\ln(5)=2.6$;idf 值为数据类别总数和包含该特征的类别数的比取对数加 1,即,$idf(t_1,c_1)=1+\ln(2/2)=1$;特征 t_1 对类属性 c_1 的鉴别性评价值为 $df(t_1,c_1)=2.6\times1=2.6$.

同理可得:

$$tf(t_1,c_2)=1+\ln(50)=4.9;\ idf(t_1,c_2)=1+\ln(2/2)=1;\ df(t_1,c_2)=4.9\times1=4.9,$$
$$tf(t_3,c_1)=1+\ln(90)=5.5;\ idf(t_3,c_1)=1+\ln(2/2)=1;\ df(t_3,c_1)=5.5\times1=5.5,$$
$$tf(t_3,c_2)=1+\ln(2)=1.7;\ idf(t_3,c_2)=1+\ln(2/2)=1;\ df(t_3,c_2)=1.7\times1=1.7.$$

上面的计算公式主要存在两个问题.

(1) 在 tf 值的计算过程中直接对频数取对数,极大地缩小了频数间的差异.例如,c_1 类中特征 t_1 和 c_2 类中特征 t_1 的频数比为 1:10,而计算 tf 值后比例变为 1:1.9.

(2) idf 值不能准确反映特征的鉴别性.例如,特征 t_1 和 t_3 对各类别的 idf 值都为 1,这导致实际的鉴别性评价值仅为 tf 值,不能准确反映这两个特征对不同类别的鉴别性.这是由于 Schafer 在文献[28]中的 idf 计算公式使用类属性取值个数和包含某特征的类的个数的比值来衡量特征对于某一类的鉴别性.然而,实际上,由于类内变异的存在,某类中的少数实例中可能具有其他类的特征,这会导致 idf 度量出现偏差.

下面我们介绍基于本文定义的 tf-idf 计算公式得到的特征鉴别性评价值.

首先,由于我们假定每个包含特征 t 的实例中特征 t 出现的频数都为 1,因此,某类特征的最大频数不超过该类实例数.为了简便起见,在计算过程中,我们假定各类特征的最大频数为该类实例数.同时,根据表 2,我们可知类别为 c_1 的特征 t_1 的频数为 5,则其频率为 5/100=0.05,因此 tf 值为 $tf(t_1,c_1)=\ln(1+0.05)=0.05$.

由于在其他类实例(即,c_2 类,该类实例数为 100)中包含类 c_1 的特征 t_1 的实例数为 50,因此,根据公式(9),idf 值为 $idf(t_1,c_1)=\ln[(200-100+1)/51]=0.7$.进而可得 $df(t_1,c_1)=0.05\times0.7=0.035$.

同理可得:

$$tf(t_1,c_2)=0.4;\ idf(t_1,c_2)=2.8;\ df(t_1,c_2)=0.4\times2.8=1.12,$$
$$tf(t_3,c_1)=0.6;\ idf(t_3,c_1)=3.5;\ df(t_3,c_1)=0.6\times3.5=2.1,$$
$$tf(t_3,c_2)=0.02;\ idf(t_3,c_2)=0.1;\ df(t_3,c_2)=0.02\times0.1=0.002.$$

从上面的计算结果可以看出,基于本文的计算公式,特征 t_1 对类 c_1 和 c_2 的 tf 值比为 1:8,与实际的频数比 1:10 接近,更准确地反映了特征和各类别的相关性.另一方面,特征 t_3 的对类别 c_1 和 c_2 的 idf 值分别为 3.5 和 0.1.从中可以看出,特征 t_3 对类别 c_1 具有更强的鉴别性.这样就有效解决了由简单地通过特征所属的类的个数进行鉴别性评价所导致的误差.因此,从计算得到的 $df(t,c)$值更容易区分同一特征对不同类别的鉴别性.例如,基于 Schafer 等人给出的公式,特征 t_3 对类别 c_1 和 c_2 的 $df(t,c)$值比为 3.2:1,而基于本文定义的公式比值为 1050:1.综上所述,利用本文定义的鉴别性度量公式,更有利于区分各类别的鉴别性特征.

1.5 特征选择阈值设定

通过固定阈值进行特征选择忽略了不同长度滑动窗口生成的特征的分类性能,这会降低生成特征字典的的质量.为此,我们提出了一种动态阈值设定机制,该机制利用单一长度滑动窗口转换得到的特征字典对应的分类性能为其生成的特征设定阈值.首先,我们用交叉验证在单一长度滑动窗口转换得到的训练集上进行分类预测,假设获得的精度为 a;然后,基于该精度和最大精度的差值为其生成特征设定选择阈值.滑动窗口对应精度越高,则其特征选择阈值越小.函数 $f(a)$将某窗口长度精度映射为该窗口生成特征的阈值因子,定义如下:

$$f(a)=\begin{cases}0, & a_{\max}-a\leqslant 0.05\\ 0.1, & 0.05<a_{\max}-a\leqslant 0.15\\ 0.2, & 0.15<a_{\max}-a\leqslant 0.25\\ 0.3, & 0.25<a_{\max}-a\leqslant 0.35\\ 0.4, & 0.35<a_{\max}-a\leqslant 0.45\\ 0.5, & a_{\max}-a>0.45\end{cases} \tag{11}$$

其中,$a_{\max}$为单一滑动窗口获得的最大交叉验证精度.

下面我们给出不同长度窗口对应的阈值计算公式:

$$\delta(l)=\theta\times f(a_l) \tag{12}$$

其中,θ为加权因子(θ>0),a_l为长度为 l 的滑动窗口对应的交叉验证精度.

加权因子θ用于校正阈值因子存在的偏差.滑动窗口对应的交叉验证精度和最优值差距越大,则对应窗口生成特征的选择阈值越大,从而根据整体分类性能对特征进行选择可有效提升所建立的特征字典的有效性.

2 算法设计

本节对我们提出的特征字典构建算法进行详细介绍.

2.1 滑动窗口最优单词长度学习算法

各种特征包方法(BOP)的不同之处在于将实值序列转换为单词的具体过程.本文我们使用的数值序列离散化方法是 SFA[22].本文提出为每个窗口长度在指定范围内动态学习性能最优的单词长度.下面首先介绍本文提出的为每个窗口长度学习最优单词长度的算法.

算法 1. *computeBestWindowsF*(*D*,min,max,min*F*,max*F*,*k*).

输入:训练集:*D*,最小和最大窗口长度:min 和 max,最小和最大傅里叶值个数:min*F*,max*F*,交叉验证重数:*k*;

输出:各窗口最优傅里叶值个数数组:*bestWindowsF*.

```
1: int[] bestWindowsF=new int[max−min+1];
2: int[][][] AF←supervisedSFA(D,min,max,minF,maxF);
3: for i=0 to numWindows−1
4:     int maxCorrect=0;
5:     for j=minF to maxF
6:         transformedData←transformedWithSingleWindow(AF,i,j)
7:         int correct←crossVvalidation(transformedData,k)
8:         if (correct>maxCorrect){
9:             maxCorrect=correct;
10:            bestWindowsF[i]=j;
11:        }
12:        j←j+1;
13: return bestWindowsF
```

算法 1 给出了本文从指定区间为每个窗口长度寻找最优傅里叶值个数(即,最优单词长度)的过程.首先,我们使用监督 SFA 对训练集 D 进行转换得到数组 $\boldsymbol{AF}$(第 2 行),其中,$\boldsymbol{AF}$ 第 1 维对应不同长度滑动窗口的序号,维度为 max−min+1;第 2 维对应训练实例的序号,维度为 N;第 3 维为转换后的实例,维度为对应滑动窗口从该实例上获得的子序列的个数,且实例中的每个单词长度为 max$F$(若滑动窗口长度小于 max$F$,则单词长度为窗口长度).由文献[22]可知,使用单一滑动窗口对 N 个实例进行监督傅里叶符号化转换的时间复杂度为 $O(|W_i|\times w_i\log w_i)$,其中,$w_i$ 为第 i 个滑动窗口长度,W_i 为该滑动窗口在数据集 D 上生成的子序列集合,$|W_i|$表示集合 W_i 中元素个数.由于$|W_i|$包含 $O(N\times n)$个子序列,$w_i\leqslant n$,因此,算法 1 第 2 步的算法复杂度为 $O(N\times n^3\log n)$.然后,我们在转换后的数据 $\boldsymbol{AF}$ 的基础上从区间[min$F$,max$F$]中学习各滑动窗口对应的性能最优的单词长度(第 3 行~第 12 行).

在学习各窗口最优单词长度的过程中,我们依次只使用单一滑动窗口长度对训练实例进行转换(第 6 行),这一转换过程只需要截取每个单词序列($\boldsymbol{AF}[i][x][y]$)的前 j 个字母即生成指定长度的单词,这一过程的时间复杂度可忽略.然后在转换得到的数据集上利用交叉验证进行预测(第 7 行),并将正确预测实例数最大值对应的傅里叶值的个数作为该窗口长度对应的最优单词长度(第 8 行~第 11 行).整个单词长度的学习过程需执行 max×max$F\times k$ 次模型训练和预测过程.本文我们使用的预测算法是一种适用于处理大规模稀疏表示数据的运行较快的逻辑回归算法[29],其算法复杂度取决于使用的梯度下降算法的收敛速度,这不在本文讨论范围之内.我们假定使用的分类模型的时间复杂度为 $T(n)$,则算法 1 的时间复杂度为 $O(\text{maximum}(N\times n^3\log n,\text{max}\times\text{max}F\times k\times T(n)))$.实验过程中,我们将交叉验证重数统一设为 10.

2.2 鉴别性特征生成

本节我们介绍如何对训练集和测试集中的实例进行符号化转换,这一过程是算法 1 中数据符号化转换的主要内容.转换过程主要分为两步.

(1) 利用监督 SFA 技术将时间序列或子序列转换为一组傅里叶值序列;

(2) 利用离散化技术将傅里叶值转换为字母表中的字母.

傅里叶值的选择关系到时间序列及其子序列转换得来的特征的鉴别性强弱.不同位置的傅里叶值的鉴别性强弱不同.我们利用 F 统计量度量每个位置的傅里叶值的鉴别性,然后选择整体分类性能最优时对应的傅里叶值个数作为该长度窗口生成单词的长度,最后利用装箱技术将时间序列子序列对应的各最优傅里叶值转换为字母表中的字母,这些字母共同组成一个单词,即,特征.下面介绍本文使用的傅里叶值的挑选算法.

算法 2. *SelectFourierCoefficient*(w_i,W,*bestF*).

输入:滑动窗口长度 w_i,长为 w_i 的子序列集合:W,最优傅里叶值个数:*bestF*;

输出:最优傅里叶值序号集合:*indexBestF*.

```
1: indexBestF←∅
2: double[][] A←supervisedSFA(W)
3: double s←0
4: for int i←0 to w_i−1
5:     s←calculateStatistic(A[i])
6:     tuple←buildTuple(i,s)
7:     indexBestF.add(tuple)
8: sorted(indexBestF) according to the statistic in descending order
9: indexBestF←selectTopK(indexBestF,bestF)
10: return indexBestF
```

算法 2 基于指定长度的子序列集合计算各个位置傅里叶值的鉴别性强弱.首先,对原始子序列进行离散傅里叶变换.由于单个长度为 n 的时间序列进行离散傅里叶变换的时间复杂度为 $O(n\log n)$,因此这一过程的时间复杂度为 $O(N\times n^2\log n)$(第 2 行).矩阵 $\boldsymbol{A}$ 中每行对应一个子序列转换得到的傅里叶值数组,$\boldsymbol{A}[i]$表示所有子序列的第 i 个位置的傅里叶值组成的数组(即,矩阵 $\boldsymbol{A}$ 的第 i 列).算法 2 中依次计算第 i 个位置傅里叶值的鉴别性强弱

统计量 F 值,然后将对应的傅里叶值序号和 F 值组成一对元组添加到集合 *indexBestF* 中(第 5 行~第 7 行).最后,根据 F 值对集合 *indexBestF* 中的元素进行降序排列(第 8 行),并返回其中前 *bestF* 个元组组成的集合(第 9 行~第 10 行).算法 2 的时间复杂度为 $O(N\times n^2\log n)$.

获得子序列集合对应的最优傅里叶值序号数组后,我们对子序列进行符号化转换,傅里叶值符号化的过程是将 $real_i$ 或 $imag_j$ 映射到符号空间的过程.下面给出将时间序列子序列转换成特征的算法.

算法 3(**VLWEA**). *createFeature*(*indexBestF*,*S*,*alphabet*).

输入:指定长度鉴别性最强的傅里叶值序号数组:*indexBestF*,子序列转换得到的傅里叶值数组:*S*,字母表:*alphabet*;

输出:特征:*word*.

```
1: word←S.length
2: for int i=0 to |indexBestF|−1
3:     letter←f(SF[indexBestF[i]],alphabet)
4:     word.combined(letter)
5: return word
```

算法 3 给出了将时间序列子序列转换成单词的算法过程,首先,根据最优傅里叶值序号数组将转换得到的子序列傅里叶值数组 S 中对应位置的傅里叶值依次利用装箱技术映射为字母表中的字母(第 3 行),并将获得的字母依次组合起来构成一个单词(第 4 行),最后得到的单词序列即为一个特征.这一符号化转换过程需要在离散化区间中执行 $O(|S|\times\log|alphabet|)$次运算[22].

2.3 特征选择和鉴别性特征字典构建

本节我们给出基于变长单词生成算法建立特征字典的过程,以及基于 tf-idf 的特征选择算法.首先,给出利用可变长度的单词建立特征字典的算法过程.

算法 4. *createFeatureDictionary*(min,max,*windowBestF*[],*D*,*alphabet*).

输入:滑动窗口最小长度:min,滑动窗口最大长度:max,各滑动窗口对应的最优傅里叶值个数:*windowBestF*[],时间训练数据集:*D*,字母表:*alphabet*;

输出:特征字典:*dict*.

```
1: dict←null
2: for each instance T in D
3:     for int l=min to minimum(|T|,max)
4:         subSequenceSet←generatingSubSequenceSet(T,l)
5:             for each subsequence S in subSequenceSet
6:                 word←createFeature(windowBestF[l−min],S,alphabet)
7:                 dict.add(word,T.classValue)
8: return dict
```

算法 4 给出了基于训练集建立特征字典的算法过程.首先,遍历训练集中的每个实例(第 2 行);然后,利用长度从 min 到 max 的滑动窗口在时间序列上进行滑动获得一系列子序列,并将这些子序列逐个转换为单词(第 3 行~第 6 行);最后,将不重复的单词作为特征加入到特征字典中(第 7 行).算法 4 中特征构建过程的主要部分为训练集实例的 SFA 转换过程,因此,特征库构建过程的时间复杂度也为 $O(N\times n^3\log n)$.

基于特征包的方法转换过程中会生成规模巨大的特征字典.为提高分类模型的效率,通常需要对输入模型的特征进行选择.本文提出了一种根据不同窗口生成特征的整体分类性能来动态设定特征选择阈值的特征选算法.下面介绍本文提出的鉴别性特征字典构建算法.

算法 5(**TfIDfDynamicVLWEA**). *createTfIdfFeatureDict*(*dict*,*corrects*[],θ).

输入:特征字典:*dict*,各滑动窗口对应的交叉验证分类精度:*corrects*[],阈值加权因子:θ;

输出:鉴别性特征字典:*TfIdfDict*.

1: *TfIdfDict*←∅

2: *threhlodFactors*[]←*f*(*corrects*)

3: *thresholds*[]←*generateThresholdsOfAllWindows*(*threhlodFactors*,θ)

4: e_i is the *i*th element in *dict*

5: for *i*=0 to |*dict*|−1

6: double *t*←*computeTfIdf*(e_i)

7: *int index*←*getWindowIndex*(e_i)

8: *double threshold*←*threhlods*[*index*]

9: if (*t*≥*threshold*)

10: *TfIdfDict.add*(e_i)

11: return *TfIdfDict*

算法 5 介绍了本文使用的特征字典建立算法.首先计算各滑动窗口生成特征的选择阈值(第 2 行~第 3 行).然后,遍历字典 *dict* 中的每个特征(第 4 行),计算特征的 $d(t,c)$值(第 6 行),然后将 $d(t,c)$值大于等于其所对应的窗口阈值的特征加入到鉴别性特征字典 *TfIdfDict* 中(第 9 行~第 10 行).

综上所述,算法 1 最优参数学习和算法 4 特征库构建是本文模型的主要构成部分.因此,本文通过动态阈值构建变长特征字典模型的时间复杂度为 $O(\text{maximum}(N\times n^3\log n,\max\times\max F\times k\times T(n)))$,其中,$T(n)$为参数学习过程中使用的分类模型算法复杂度.

2.4 基于特征字典的时间序列转换表示

这节我们给出基于 SFA 的 BOP 模型中用于训练集和测试集中实例转换的算法[24].这一过程主要分为两步:首先,在训练集上利用某长度得到的所有不重叠子序列为每个窗口长度训练用于时间序列转换的监督符号傅里叶近似转换对象,训练转换对象主要是为了计算滑动窗口子序列每个位置的 *F* 值(算法 2)和用于将傅里叶值数组进行离散化的分箱区间(文献[23]中有详细介绍);然后,用训练得到的转换对象对时间序列进行转换[24].下面我们给出时间序列的转换算法.

算法 6. *SupervisedSFATransform*(*TfIdfDict*,*T*,min,max,*transferObjects*[]).

输入:鉴别性特征字典:*TfIdfDict*;要转换的实例:*T*;最小和最大窗口长度:min,max;各滑动窗口训练得到的 SFA 转换对象:*transferObjects*[];

输出:转换后实例:*transformed*.

1: *transformedT*←∅

2: for *l*=min to max

3: for *i*=0 to |*T*|−*l*

4: *word*←*transferObjects*[*l*−min].*sequenceTransform*(*T*(*i*,*l*))

5: if (*TfIdfDict.contain*(*word*))

6: *transformedT.add*(*word*)

7: return *transformedT*

算法 6 给出了将一条时间序列转换为符号序列频数数组的算法.该算法用指定长度滑动窗口从给定时间序列的初始位置进行滑动依次获得一系列可重叠的子序列(第 2 行~第 3 行),然后通过对应长度的监督符号傅里叶近似转换对象将每个子时间序列依次转换成一个单词,即,时间序列包含的特征(第 4 行).若转换得到的特征在给定的鉴别性特征字典中,则将该特征加入到转换后的实例中,作为转换后实例的一个特征(第 5 行~第 6 行).此时,若转换后的实例中没有该特征,则为转换实例加入该特征,并将该特征的取值设为 1;若转换后的实例中已有该特征,则该特征对应的值加 1.最后返回获得的特征频数序列,即为转换得到的时间序列.

逻辑回归(logistic regression)是统计学中的经典分类方法.它是一个对数线性模型.本文我们使用基于 L2

正则化的逻辑回归模型对转换后的实例进行分类[29].此外,本文还利用逻辑回归学习到的权重对特征字典进行分析.

3 实验分析

本节对我们所提模型的相关实验内容进行详细介绍.主要包括 4 个方面:模型参数分析、模型设计的方法分析、分类精度比较和模型的可解释性分析.本文在 UEA & UCR 时间序列知识库提供的 65 个长度小于 750 的时间序列数据集上对模型进行分析[30].表 3 给出了各数据集的信息,其中,Train/Test 表示训练集和测试集的实例数,n 为时间序列长度,$|C|$为类属性取值个数.我们的实验环境 CPU 是 3.40GHz,内存为 16G.

Tabel 3 Introduction to 65 time series datasets

表 3 65 个时间序列数据集介绍

数据集名称	Train/Test	n/$\|C\|$	数据集名称	Train/Test	n/$\|C\|$	数据集名称	Train/Test	n/$\|C\|$
Adiac	390/391	176/37	FaceFour	24/88	350/4	ProximalPOC	600/291	80/2
ArrowHead	36/175	251/3	FacesUCR	200/2050	131/14	ProximalPTW	400/205	80/6
Beef	30/30	470/5	Fish	175/175	463/7	RefrigerationD	375/375	720/3
BeetleFly	20/20	512/2	GunPoint	50/150	150/2	ScreenType	375/375	720/3
BirdChicken	20/20	512/2	Ham	109/105	431/2	ShapeletSim	20/180	500/2
Car	60/60	577/4	Herring	64/64	512/2	SmallKitchenA	375/375	720/3
CBF	30/900	128/3	InsectWS	220/1980	256/11	SonyAIBORobotS	20/601	70/2
ChlorineC	467/3840	166/3	ItalyPD	67/1029	24/2	SonyAIBORobotS	27/953	65/2
Coffee	28/28	286/2	LargeKA	375/375	720/3	Strawberry	613/370	235/2
Computers	250/250	720/2	Lightning2	60/61	637/2	SwedishLeaf	500/625	128/15
CricketX	390/390	300/12	Lightning7	70/73	319/7	Symbols	25/995	398/6
CricketY	390/390	300/12	Meat	60/60	448/3	SyntheticControl	300/300	60/6
CricketZ	390/390	300/12	MedicalImages	381/760	99/10	ToeSegmentation1	40/228	277/2
DiatomSR	16/306	345/4	MiddlePOAG	400/154	80/3	ToeSegmentation2	36/130	343/2
DistalPOAG	400/139	80/3	MiddlePOC	600/291	80/2	Trace	100/100	275/4
DistalPOC	600/276	80/2	MiddlePTW	399/154	80/6	TwoLeadECG	23/1139	82/2
DistalPTW	400/139	80/6	MoteStrain	20/1252	84/2	TwoPatterns	1000/4000	128/4
Earthquakes	322/139	512/2	OliveOil	30/30	570/4	Wafer	1000/6174	152/2
ECG200	100/100	96/2	OSULeaf	200/242	427/6	Wine	57/54	234/2
ECG5000	500/4500	140/5	PhalangesOC	1800/858	80/2	WordsSynonyms	267/638	270/25
ECGFiveDays	23/861	136/2	Plane	105/105	144/7	Yoga	300/3000	426/2
FaceAll	560/1690	131/14	ProximalPOAG	400/205	80/3	–	–	–

3.1 模型参数实验分析

本节我们对模型 TfIDfDynamicVLWEA 的几个重要参数进行实验分析.

(1) 训练集规模和时间序列长度对模型运行时间的影响分析;

(2) 模型精度和压缩比(即,TfIdf 特征库中特征数和初始特征库中特征数的比值)对阈值加权因子θ的敏感性;

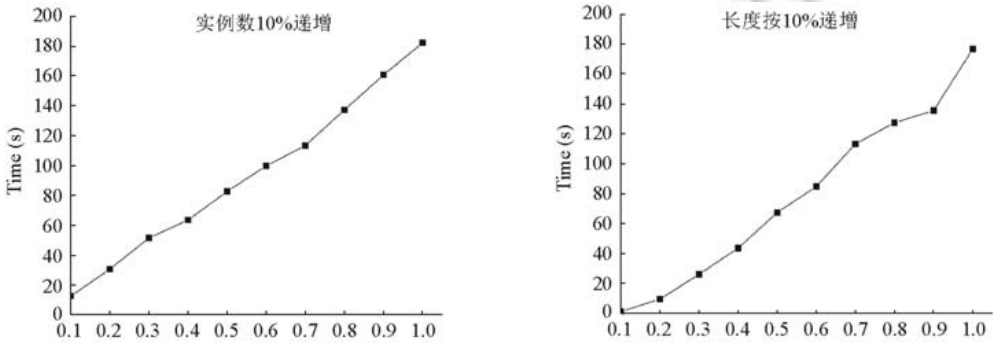
(3) 模型精度对最大滑动窗口长度和最大单词长度的敏感性.

首先,分析模型的运行时间分别与训练集规模、时间序列长度的关系.我们使用一个生成的二分类数据集进行实验.训练集和测试集中各包含 100 个长度为 200 的时间序列,且每个数据集中两类实例个数相同.在实例数递增分析实验中,我们设置初始训练集和测试集各由原数据集中 10%的实例构成,然后训练集和测试集中实例个数以原训练集实例个数的 10%递增,并始终保持数据集中类属性分布和时间序列长度不变.我们分别统计模型在上述 10 组数据上的运行时间.在时间序列长度递增分析实验中,我们将时间序列长度从原长度的 10%开始按 10%递增,同时保持训练集和测试集规模不变,这样进行 10 组实验并统计模型运行时间.图 2 给出了模型 TfIDfDynamicVLWEA 的运行时间的实验结果.

从图 2 可以看出,模型 TfIDfDynamicVLWEA 的运行时间与训练集实例个数呈线性关系,而与时间序列长度呈多项式关系.这与第 2 节分析获得的模型算法复杂度相符.

下面我们在 10 个数据集上对模型各参数的敏感性进行分析.图 3 和图 4 中加粗曲线表示 10 个数据集上模型 TfIDfDynamicVLWEA 的平均精度变化曲线,我们标出了每个点对应的平均精度.

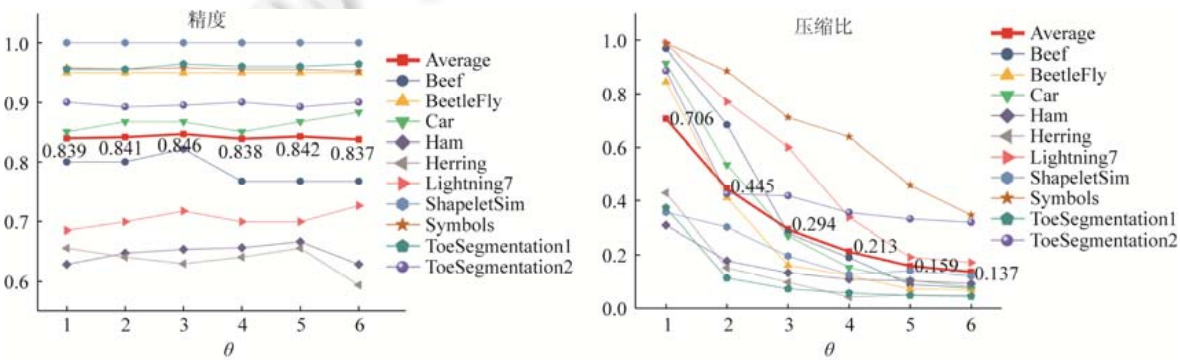
首先,我们分析加权因子对模型 TfIDfDynamicVLWEA 的精度和特征库压缩比的影响.从图 3(b)中可以看出,随着θ的增大,压缩比显著减小,当θ大于 5 后,减速变缓.同时,从图 3(a)中可以看出,随着θ的增大,数据集精度呈现出不同的变化趋势.例如,数据集 Beef 和 Ham 上的模型精度随θ增大,先增大后减小;数据集 BeetleFly、ShapeletSim 等的精度随θ增大,变化不大;数据集 Herring 的精度随θ增大,呈现先减小后增大的趋势.由于当θ为 3 时,模型的平均精度最大,压缩比较优,因此综合考虑,我们选择将模型的参数θ设为 3.



(a) 实例数 (b) 时间序列长度

Fig.2 The running time of model TfIDfDynamicVLWEA

图 2 模型 TfIDfDynamicVLWEA 运行时间分析



(a) θ对精度的影响 (b) θ对压缩比的影响

Fig.3 Analysis of the influence of the parameter θ on model TfIDfDynamicVLWEA

图 3 模型 TfIDfDynamicVLWEA 参数θ的影响分析

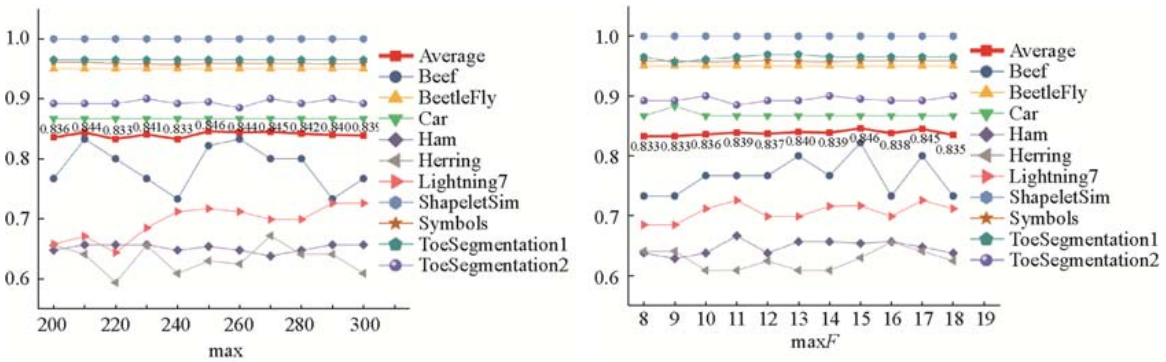
图 4 给出了 10 个数据集上模型 TfIDfDynamicVLWEA 精度及平均精度随最大滑动窗口长度和最大单词长度的变化趋势.

从图 4(a)和图 4(b)可以看出,不同数据集对不同参数的敏感性不同.从图 4(a)中可以看出,Lightning7 数据集上的精度随最大滑动窗口长度的递增总体呈增长趋势,数据集 ShapeletSim、BeetleFly、Symbols、ToeSegmentation1 和 Ham 对最大窗口长度不敏感,数据集 Beef 和 Herring 对最大滑动窗口长度较为敏感,随窗口长度呈递增趋势,精度变化明显.上述实验结果表明,很难设定最优滑动窗口长度.由于最大滑动窗口长度取 250 时,图 4(a)中平均精度值最大,同时也为了保证对比实验的公平性,我们将模型 TfIDfDynamicVLWEA 的最大窗口长度设为 250.

从图 4(b)可以看出,数据集 Beef 和 Herring 对最大单词长度较为敏感,数据集 ShapeletSim、BeetleFly、Car、Symbols 和 ToeSegmentation1 对最大窗口长度不敏感,数据集 Ham、Lightning7 和 ToeSegmentation2 随最大单词长度发生变化,精度有波动.在区间[8,18]中 10 个数据集上的平均精度在 15 时最大,因此,我们将模型中 TfIDf

DynamicVLWEA 的最大单词长度设为 15.

此外,由于目前基于 SFA 对时间序列进行符号转换的研究[19,23,24]表明字母表大小设置为 4 时,BOP 模型具有更强的鲁棒性.因此,本文将字母表大小固定为 4.



(a) 最大滑动窗口长度 (b) 最大单词长度

Fig.4 Sensitivity analysis of model TfIDfDynamicVLWEA accuracy to parameters max and max$F$

图 4 模型 TfIDfDynamicVLWEA 精度对参数 max 和 max$F$ 的敏感性分析

3.2 设计方法分析

我们在表 3 中的 65 个数据集上对本文所提模型中设计的新方法进行分析.主要包括如下 3 个方面.

(1) 可变单词长度和固定单词长度比较;

(2) Bigrams 特征对可变长度特征字典性能的影响;

(3) 基于 tf-idf 统计量的特征选择算法和基于卡方统计量特征选择算法的比较.

我们使用符号 VLWEA 表示采用可变长度单词生成算法的特征字典建立模型,FLWEA 表示采用固定长度单词生成算法的特征字典建立模型.VLWEA_U、FLWEA_U 分别表示对应模型在特征字典建立过程中只将单个单词作为特征,且不对特征进行选择.FLWEA_B、VLWEA_B 分别表示模型在特征字典建立过程中使用 Bigrams 语法模型进行特征生成,即,将连续的两个单词组成一个新的单词作为特征.符号 Chiδ和 TfIdfδ分别表示模型中使用单一阈值的特征选择算法,其中,δ表示使用的阈值.例如,Chi3VLWEA_U 表示模型 VLWEA_U 中使用阈值为 3 的卡方统计量进行特征选择,即,保留卡方统计值大于等于 3 的特征.TfIDfDynamicVLWEA 表示本文提出的使用动态阈值设定的 VLWEA 模型,其中加权因子θ统一取为 3.我们使用的对比模型包括:不利用 Bigrams 语法进行特征生成,只利用可变长度单词或固定长度单词组成的特征字典建立模型 VLWEA_U 和 FLWEA_U,结合不同特征选择算法和不同阈值的模型 Chi3FLWEA_U、Chi3VLWEA_U、TfIdf0.3VLWEA_U、TfIdf0.3VLWEA_B 和 TfIDfDynamicVLWEA_U.实验中,上述所有 VLWEA 和 FLWEA 模型的最小和最大滑动窗口长度都分别设为 4 和 250,字母表的大小统一设定为 4,其他处理方式与定长单词生成模型 WEASEL 相同.同时,使用相同参数设置的分类模型对转换后的测试集进行分类预测.

接下来,我们利用 Demsar 提出的模型分类性能显著性和平均排名比较方法在 65 个数据集上对本文新提出的方法进行分析[31].表 4 中给出了多种条件下建立的特征字典对应的逻辑回归模型在 65 个数据集上的分类精度、平均精度和模型在 65 个数据集上精度最高的数据集个数.表 4 中给出的模型实验结果都是通过在每个数据集上进行 5 次实验取均值获得的.在对特征选择算法的性能对比过程中,为了对比实验的公平性,我们通过选取适当的阈值,使得不同特征选择算法在 65 个数据集上的平均压缩比相近,即,65 个数据集上选择后的特征字典大小和原特征字典大小的比的平均值相近.模型 TfIdf0.3VLWEA_U、TfIdf0.3VLWEA_B、Chi3VLWEA_U 和 Chi3FLWEA_U 的平均压缩比分别为 27.3%、24.6%、32.6%和 36.3%.

从图 5 中我们可以看出:使用阈值为 3 的卡方统计量进行特征选择的定长单词生成模型 Chi3FLWEA_U 的

性能排名最低,这说明可变长度单词生成算法有助于提升模型分类性能.使用本文定义的鉴别性特征评价统计量进行特征选择的变长单词生成模型 TfIdf0.3VLWEA_U 在 65 个数据集上特征字典的平均压缩比优于基于卡方统计量的特征选择算法模型(Chi3VLWEA_U 和 Chi3FLWEA_U)压缩比的条件下,分类性能排名更优.这说明,本文提出的基于 tf-idf 的特征选择算法与基于卡方统计量的特征选择算法相比,能够更有效地进行特征选择.与此同时,从图 5 中我们还可以看出:在相同阈值条件下,结合 Bigrams 语法的特征生成模型 TfIdf0.3VLWEA_B 的排名低于模型 TfIdf0.3VLWEA_U,这说明,Bigrams 语法模型不能有效地提升变长特征生成算法的分类性能.因此,综合考虑生成特征字典规模和分类模型的运行效率,在接下来的对比实验中,我们的模型中不再采用 Bigrams 语法进行特征生成.此外,基于动态阈值的特征选择模型 TfIDfDynamicVLWEA_U 在 65 个数据集上的性能平均排名第 1,这显示了本文提出的动态阈值设定算法的有效性.从表 4 中我们还可以看出,TfIDfDynamic VLWEA_U 在 65 个数据集上的平均精度最高和精度最高的数据集个数最多.因此,在接下来的实验分析中,我们使用 TfIDfDynamicVLWEA_U 模型与其他各类模型进行对比,并将其简记为 VLWEA.

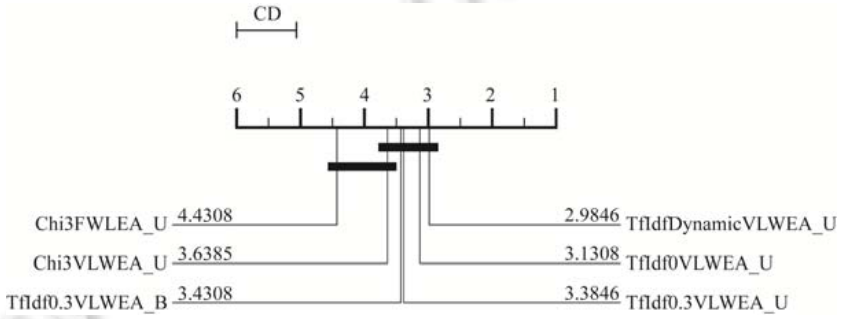


Fig.5 The classification performance significance analysis and the average rank of the dictionaries built under 6 conditions on 65 datasets

图 5 6 种条件下构建的字典在 65 个数据集上的分类性能显著性分析和模型平均排名

3.3 分类精度比较

本节我们将本文所提鉴别性特征字典建立模型 VLWEA 分别与基于特征包的模型、1NN 模型、基于 shapelet 的模型和集成分类模型的分类精度进行对比分析.两个常用的基准 1NN 分类模型为:基于欧式距离的 1NN(ED1NN)和基于动态时间规整的 1NN(DTW1NN);两个基于 Shapelet 的非集成分类模型包括:快速 Shapelet 分类(fast Shapelet,简称 FS)[32]算法和 Grabocka 等人提出的基于启发式 Shapelet 搜索算法的分类(learning Shapelets,简称 LS)模型[33];6 种特征包模型为:SAXVSM、BOSS、TSBF、使用 Bigrams 语法进行特征生成的模型 WEASEL_B、使用 Unigram 语法进行特征生成的 WEASEL 模型 WEASEL_U 以及 Kate 等人提出的基于 DTW 距离的特征生成算法(DTW features,简称 DTW_F)[34];3 种集成分类模型包括:Deng 等人提出的基于 11 种近邻分类模型的集成分类算法(elastic ensemble,简称 EE)[35]、基于 Bostrom 等人提出的 Shapelet 转换表示方法建立的集成分类模型(ST)[16]和 COTE.模型 WEASEL_B 和 WEASEL_U 的参数设置采用 Schafer 等人提供的代码中的默认设置,模型预测精度我们取 5 次实验的平均值(见表 4,其中,C2FB 代表 WEASEL_B,C2FU 代表 WEASEL_U),其他模型采用 Bagnall 等人[5]提供的实验结果.下面我们将本文模型和各类模型在 65 个数据集上分别进行对比,并给出了 VLWEA 和对比模型相比在 65 个数据集中的精度高、平、低的数据集个数,例如,“30/20/15”表示和对比模型相比,VLWEA 有 30 个更准确,20 个相同,15 个更差.

首先,我们对 VLWEA 和 6 个 BOP 模型的比较结果进行分析.从图 6 和具体的“高/平/低”对比结果统计可以看出,VLWEA 模型与模型 WEASEL_B(35/9/21)、WEASEL_U(41/6/18)、TSBF(48/5/12)、BOSS(35/2/28)、SAXVSM(55/2/8)及 DTW_F(51/2/12)相比,分类精度高的数据集个数都是最多的.VLWEA 在不使用 Bigrams 语法进行特征生成的条件下,比 WEASEL_B 在更多的数据集上取得更好的分类效果.这再次说明本文可变长度单

词生成算法在特征生成上的有效性.

从图 7 中和具体的“高/平/低”对比结果统计可以看出,VLWEA 与模型 ED1NN(60/1/4)和 DTW1NN (53/3/9)相比显著更好,分类效果具有绝对优势.

图 8 和对比结果统计说明,VLWEA 和 2 个不使用集成分类算法的基于 Shapelet 的分类模型 LS(49/6/10)、FS(60/3/2)对比同样具有显著优势.

图 9 和具体的精度“高/平/低”对比结果统计说明,VLWEA 和 3 个集成分类模型 COTE(27/7/31)、EE(45/3/17)、ST(40/6/19)相比,比 EE 和 ST 更好,但比 COTE 略差.

Tabel 4 Accuracies of the feature dictionaries built under eight different conditions (%)

表 4 8 种不同条件下建立的特征字典对应的模型分类精度(%)

特征评价指标	tf-idf	tf-idf	tf-idf	Chi	tf-idf	Chi	Chi	Chi
阈值	Dynamic	0.3	0.3	3	0	3	2	2
单词长度	Variable	Variable	Variable	Variable	Variable	Fixed	Fixed	Fixed
n_0 语法模型	Unigram	Unigram	Bigrams	Unigram	Unigram	Unigram	Unigram	Bigrams
Dataset	TDVU	T0.3VU	T0.3VB	C3VU	T0VU	C3FU	C2FU	C2FB
Adiac	80.6	80.3	80.9	81.7	81.2	83.3	82.5	**83.9**
ArrowHead	86.7	**86.9**	86.8	85.1	87.7	74.3	86.3	85.7
Beef	**82.2**	75.3	83.3	76.7	80	73.3	73.3	81.3
BeetleFly	**95**	90	**95**	**95**	**95**	**95**	**95**	**95**
BirdChicken	**90**	**90**	**90**	**90**	**90**	85	**90**	85
Car	86.7	86.7	83.3	85	88.3	85	**88.3**	85
CBF	**99.8**	99.7	99.6	99.6	99.6	96.8	98.5	99.1
ChlorineC	75	**76.1**	75.4	74.6	75	73.5	74.2	75.3
Coffee	**100**	**100**	**100**	**100**	**100**	96.4	**100**	**100**
Computers	70.4	70.5	70.3	69.1	**71.8**	64	62.7	66.4
CricketX	76	72.6	**78.8**	71.9	73.5	75.6	76.4	77.1
CricketY	78.2	79	77.2	79.2	**81.3**	77	76.7	79
CricketZ	77.3	79.2	78.3	77.1	**79.5**	78	78.1	79.2
DiatomSR	90.6	91.4	89.7	91	90.8	**94.4**	93.5	94.1
DistalPOAG	75.5	76	76	77.3	75.2	**79.1**	78.4	77
DistalPOC	**78.7**	76.1	75.4	77.6	78.5	76.1	75.2	78.3
DistalPTW	67.6	68.3	67.2	67	67.6	62.6	67.6	**69.8**
Earthquakes	**75**	74.8	74.8	74.8	74	74.1	74.8	74.1
ECG200	**86.7**	85	85.8	85.6	84.4	85	85	84
ECG5000	94	94.2	94.2	94.2	93.9	94.5	94.4	**95**
ECGFiveDays	**100**	99.9	99.9	**100**	99.9	99.9	99.9	**100**
FaceAll	78.1	78.8	79.1	78.9	79.6	**79.2**	77.4	77.2
FaceFour	98.9	**100**	98.9	100	99.3	**100**	**100**	**100**
FacesUCR	94.4	94.1	93.3	93.9	93.9	94.7	**94.9**	94.7
Fish	**97.7**	96.6	96.6	96.8	97.5	94.9	96.4	96.6
GunPoint	**100**	**100**	**100**	**100**	99.3	**100**	98.7	**100**
Ham	65.4	64.8	62.9	65	**67**	62.9	63.8	64.8
Herring	63	64.7	61.9	59.4	63.8	67.2	**68.7**	60.9
InsectWS	62.7	63.1	63.9	63.6	65	61.7	61.9	**64.1**
ItalyPD	**95.9**	95.1	94.9	94.1	95.6	94.8	94.4	94.9
LargeKA	67.8	67.8	68.3	**70.6**	68.5	60.4	58.5	62.9
Lighting2	67.2	62	63.9	54.1	**69.2**	52.5	63.9	55.7
Lighting7	71.7	70.2	65.8	69	75.1	**76.7**	75.3	74
Meat	**91.7**	91	90	**91.7**	88.3	90	90	90
MedicalImages	75.5	75.1	74.3	74.7	74.9	70	71.1	**75.8**
MiddlePOAG	58.8	**59.6**	58.9	57.9	60.5	54.5	53.2	56.5
MiddlePOC	**82.4**	82.3	83.3	79.9	81.9	79.7	81.4	79.4
MiddlePTW	52.6	51.2	53	53.1	51.4	49.4	49.4	**56.5**
MoteStrain	93.4	**95.6**	93.1	92	95	88.1	74.8	88.2
OliveOil	91.1	90	**93.3**	86.7	92.6	**93.3**	**93.3**	**93.3**
OSULeaf	**98.9**	90.1	90.9	90.6	95.9	88.6	88.4	89.5
PhalangesOC	81.3	79	79.7	81.5	**82.5**	78.7	76.6	80.7
Plane	**100**	**100**	**100**	**100**	**100**	99	99	**100**

Tabel 4 Accuracies of the feature dictionaries built under eight different conditions (%) (Continued)
表 4 8 种不同条件下建立的特征字典对应的模型分类精度(%)(续)

特征评价指标	tf-idf	tf-idf	tf-idf	Chi	tf-idf	Chi	Chi	Chi
阈值	Dynamic	0.3	0.3	3	0	3	2	2
单词长度	Variable	Variable	Variable	Variable	Variable	Fixed	Fixed	Fixed
n_0 语法模型	Unigram	Unigram	Bigrams	Unigram	Unigram	Unigram	Unigram	Bigrams
Dataset	TDVU	T0.3VU	T0.3VB	C3VU	T0VU	C3FU	C2FU	C2FB
ProximalPOAG	84.6	85.6	85.8	**86**	84.4	84.4	83.9	83.9
ProximalPOC	89.9	89.3	88.7	89.6	**90.1**	86.9	88.3	88
ProximalPTW	80.7	78.2	**82.1**	80.2	79.4	81	81.5	78
RefrigerationD	**57.8**	51.4	53.2	52.5	51.6	51.2	51.6	52.8
ScreenType	52.6	55.1	53.9	53.8	55.8	56.8	**58.5**	53.3
ShapeletSim	**100**	**100**	**100**	98.8	76.5	**100**	96.1	**100**
SmallKitchenA	76.3	77	76.7	76.5	**78.7**	76.3	75.2	75.7
SonyAIBORobotS1	84.1	82.7	78.9	82.8	82.7	79.5	**85**	81.9
SonyAIBORobotS2	**95.6**	94.6	94.4	91.3	**95.6**	89.2	93.7	94.8
Strawberry	97.3	97.8	97.6	97.5	97.3	**98.1**	97.8	97.6
SwedishLeaf	96.4	96.3	96.2	96.3	**96.5**	95.8	96.3	96.4
Symbols	95.9	95.7	**96.7**	95.8	96.6	95.5	95.5	96.1
SyntheticControl	98.8	99	98.7	98.8	98.6	**99.3**	**99.3**	99
ToeSegmentation1	**96.5**	94.3	94.8	94.3	94	91.2	89.5	95.2
ToeSegmentation2	89.5	85.9	89.4	83.2	**90.2**	78.5	81.5	82.3
Trace	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
TwoLeadECG	**99.9**	**99.9**	**99.9**	**99.9**	99.8	92.2	95.6	99.4
TwoPatterns	**99.3**	99.2	**99.3**	**99.3**	**99.3**	99.1	99.1	98.9
Wafer	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Wine	86.4	**89.6**	89.3	85.9	87	70.4	75.9	85.6
WordsSynonyms	70.7	71	70.8	71.5	70.3	71.3	70.8	**71.6**
Yoga	91.6	90.7	**92.2**	87.2	90.9	84	85.8	90.4
平均值	**84.6**	83.9	84.1	83.5	84.3	82.2	82.8	83.7
最大值个数	**22**	14	15	13	18	12	13	17

上面表 4 中我们将各特征字典构建模型符号表示中的 TfIdf 简记为 T,Chi 简记为 C,Dynamic 简记为 D, VLWEA_X 和 FLWEA_X 分别简记为 VX 和 FX,X 表示使用的 n_0 元特征生成模型,U 表示 Unigram,B 表示 Bigrams.例如,模型 TfIDfDynamicVLWEA_U 记作 TDVU.表 4 中每行加粗数值表示对应行的最优值.

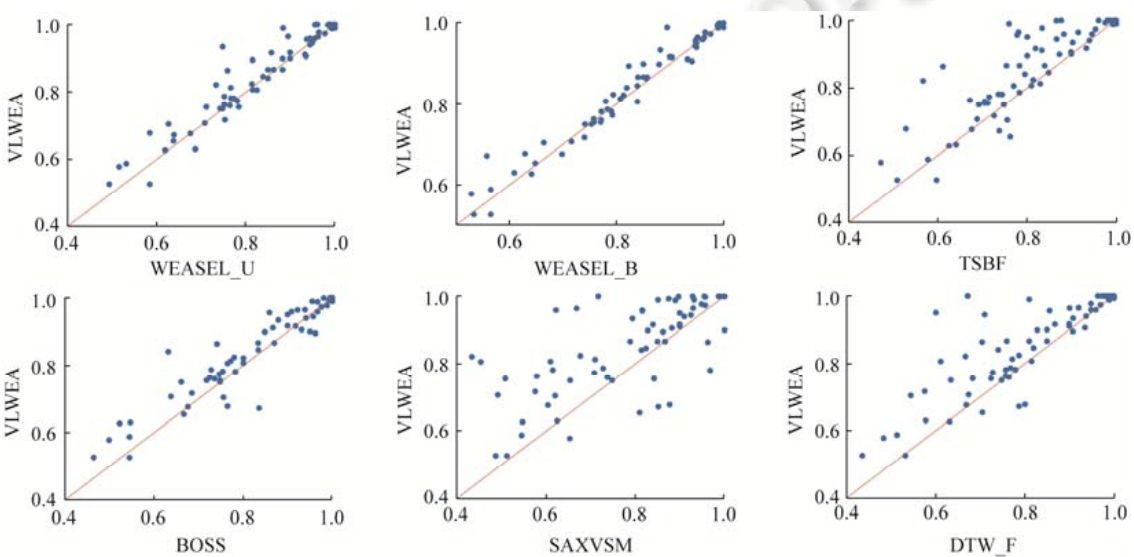


Fig.6 Comparison of accuracy between VLWEA and 6 BOP models on 65 datasets
图 6 VLWEA 和 6 个 BOP 模型在 65 个数据集上的分类精度比较

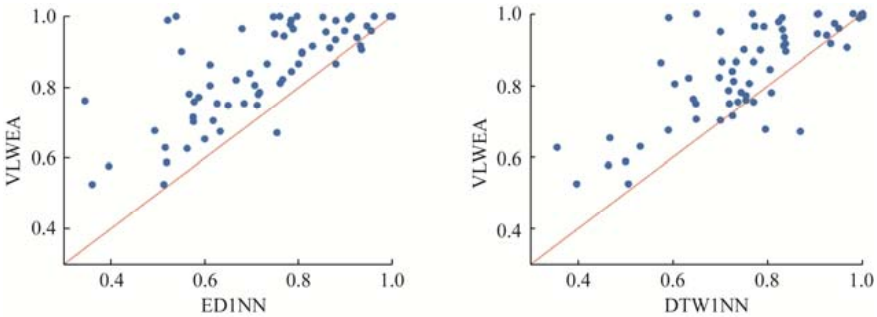


Fig.7 Comparison of accuracy between VLWEA and 2 1NN models on 65 datasets

图 7 VLWEA 和两个 1NN 模型在 65 个数据集上的分类精度比较

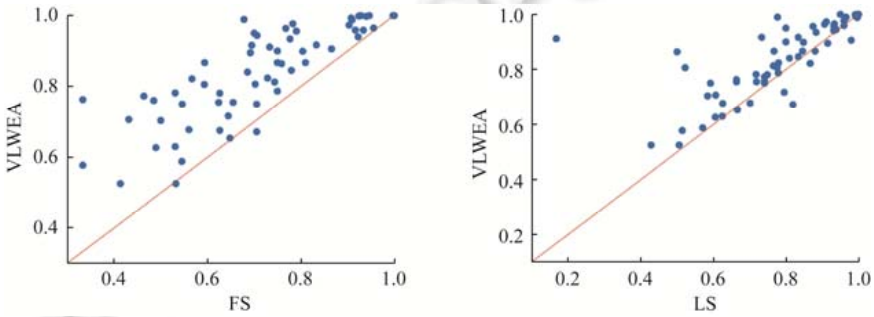


Fig.8 Comparison of accuracy between VLWEA and 2 shapelet models on 65 datasets

图 8 VLWEA 和两个 Shapelet 分类模型在 65 个数据集上的分类精度比较

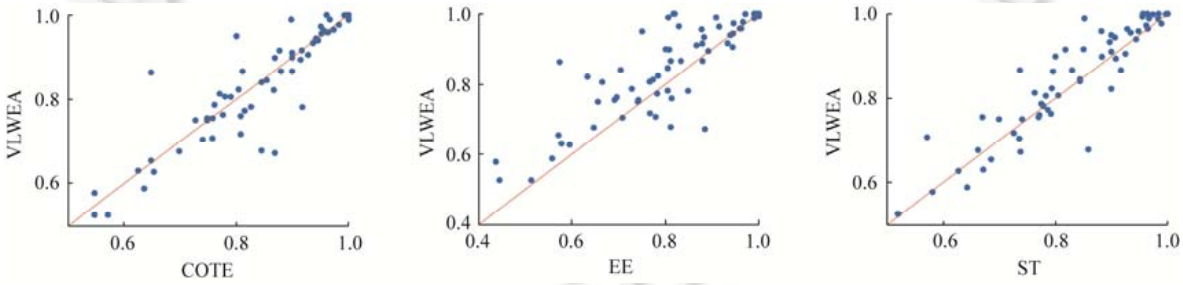


Fig.9 Comparison of accuracy between VLWEA and 3 ensemble models on 65 datasets

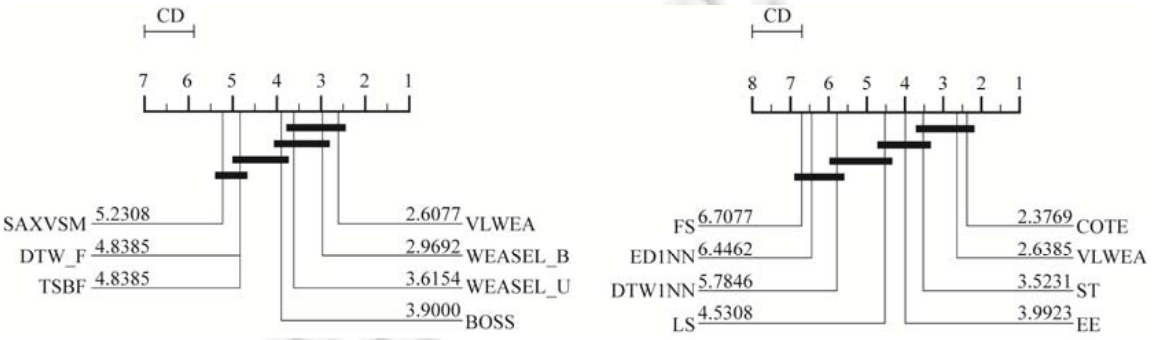
图 9 VLWEA 和 3 个集成分类模型在 65 个数据集上的分类精度比较

最后,我们对本文提出的模型 VLWEA 分别与其同类型和异类型模型的性能显著性和平均排名进行对比分析.从后文所示图 10 可以看出,VLWEA 与其他 6 个 BOP 模型在 65 个数据集上的性能相比没有显著差异,但是 VLWEA 的分类精度排名最高.与非 BOP 模型相比,VLWEA 的排名只比 COTE 差,比 ST、EE、LS、FS、ED1NN 和 DTW1NN 的排名都更好.与当前最先进的模型的对比结果说明了本文所提出的特征字典建立方法的有效性.

3.4 实例分析

本节对 VLWEA 模型的可解释性进行分析.我们选择多分类数据集 CBF 对模型学习到的最优单词长度和生成特征的鉴别性进行分析.该数据集的训练集实例共有 3 类.我们用每类实例各特征的频数平均值组成一个均值序列代表该类实例.图 11 给出了 CBF 各窗口长度的最优单词长度的箱型图和 9 个由原始序列生成的鉴别性子序列图示.

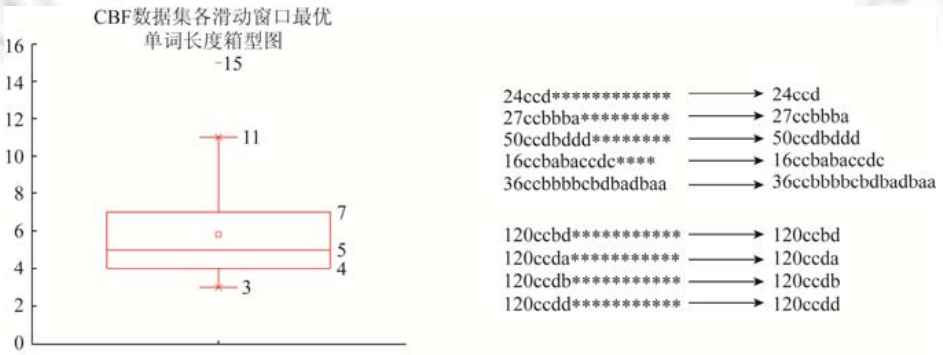
从图 11(a)可以看出:在忽略极少数异常值的情况下(如图 11(a)中的长度 15),25%的最优单词长度落在[3,4]之间,50%的最优单词长度落在区间[4,7]中,25%的最优单词长度位于[7,11]之间,因此,固定长度单词生成算法在单词生成过程中会不可避免地损失鉴别性信息或携带大量冗余信息.从图 11(b)可以看出,本文提出的可变长度单词生成算法可以有效地学习最优单词长度.例如,图中不同长度:24、27、50、16 和 120 对应的特征分别有针对性地将原序列尾部的冗余信息“**…*”去除(符号“*”代表字母表中的任意字母),只保留具有鉴别性的部分.此外,对于图 11(b)所示长度为 120 的 4 个原始序列,若单词长度为 2,则只生成一个特征“cc”;单词长度为 3 时,生成特征“ccb”和“ccd”;单词长度大于 4 时,可以生成 4 个特征,但会包含一定冗余信息.而本文算法可以有效学习单词的最优长度,且不包含冗余信息.这也再次验证了图 11 给出的示例.



(a) 6 个 BOP 模型 (b) 7 个非 BOP 模型

Fig.10 The classification performance analysis and the average ranks of VLWEA and 13 models on 65 datasets

图 10 VLWEA 和 13 个分类模型在 65 个数据集上的分类显著性分析及平均排名



(a) (b)

Fig.11 Optimal word lengths and 9 generation features obtained by VLWEA

图 11 VLWEA 学习得到的最优单词长度和 9 个生成特征

由于建立的特征字典规模巨大,我们根据学习到的权重选择 top-10 个特征对数据集进行表示.图 12 中给出了 3 类实例均值序列的直方图.从图 12 中我们可以看出,这些特征具有明显的鉴别性,例如,特征“74ccadd”只有类属性为 c_0 的实例具有,特征“29ccbbb”和“29acbbb”特征对于类属性为 c_1 的实例具有较强的鉴别性.具有特征“54ccbcd”和“74ccbdd”的实例类属性为 c_1 的可能性很小.

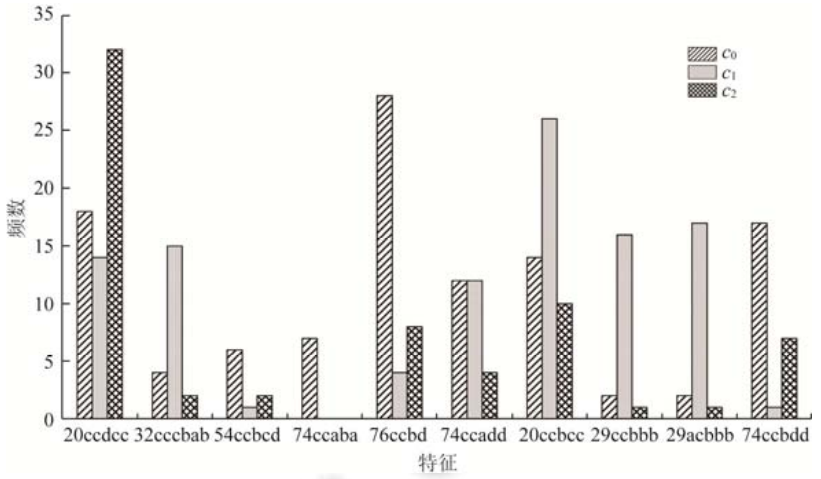


Fig.12 Top-10 discriminant features generated by VLWEA on dataset CBF

图 12 数据集 CBF 上 VLWEA 生成的 top-10 鉴别性特征

4 结 论

在进行时间序列数据挖掘之前,对时间序列数据重新表示是一个重要的研究课题,其目的是,一方面通过减少算法实际处理数据的量来提高算法的运行速度,另一方面,充分表达原始时间序列数据的本质内容以提高分类精度.针对目前基于 SFA 的时间序列进行离散化表示方法存在的问题,本文提出了一种可变长度单词抽取算法,该算法可以有效学习不同滑动窗口对应的最优单词长度.与此同时,针对特征字典规模巨大的问题,本文定义了一种新的鉴别性特征选择统计量,并设计了一种动态阈值设定机制来对生成的特征进行选择,该方法在有效缩小特征字典规模的同时,可以获得较高的分类精度.

References:

[1] Gulisano V, Jerzak Z, Voulgaris S, Ziekow H. The DEBS 2016 grand challenge. In: Proc. of the 10th ACM Int'l Conf. on Distributed and Event-Based Systems (DEBS 2016). New York: ACM Press, 2016. 289–292. [doi: 10.1145/2933267.2933519]

[2] Patri O, Wojnowicz M, Wolff M. Discovering malware with time series Shapelets. In: Proc. of the 50th Hawaii Int'l Conf. on System Sciences. AIS Electronic Library, 2017. 6079–6088. [doi: 10.24251/HICSS.2017.734]

[3] Zhu L, Lu C, Sun Y. Time series shapelet classification based online short-term voltage stability assessment. IEEE Trans. on Power Systems, 2016,31(2):1430–1439. [doi: 10.1109/tpwrs.2015.2413895]

[4] Esling P, Agon C. Time-series data mining. ACM Computing Surveys, 2012,45(1):1–34. [doi: 10.1145/2379776.2379788]

[5] Bagnall A, Lines J, Bostrom A, Large G, Keogh E. 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]

[6] Lin J, Khade R, Li Y. Rotation-invariant similarity in time series using bag-of-patterns representation. Journal of Intelligent Information Systems, 2012,39(2):287–315. [doi: 10.1007/s10844-012-0196-5]

[7] Ding H, Trajcevski G, Scheuermann P, Wang XY, Keogh E. Querying and mining of time series data: Experimental comparison of representations and distance measures. Proc. of the VLDB Endowment, 2008,1(2):1542–1552. [doi: 10.14778/1454159.1454226]

[8] Yuan JD, Wang ZH, Sun YG, Zhang W. *K*-nearest neighbor classifier for complex time series. Ruan Jian Xue Bao/Journal of Software, 2017,28(11):3002–3017 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5331.htm [doi: 10.13328/j.cnki.jos.005331]

[9] Gorecki T. Classification of time series using combination of DTW and LCSS dissimilarity measures. Communications in Statistics-simulation and Computation, 2018,47(1):263–276. [doi: 10.1080/03610918.2017.1280829]

[10] Hoppner F. Improving time series similarity measures by integrating preprocessing steps. Data Mining and Knowledge Discovery, 2017,31(3):851–878. [doi: 10.1007/s10618-016-0490-x]

[11] Yuan JD, Douzal-Chouakria A, Yazdi SV, Wang ZH. A large margin time series nearest neighbour classification under locally weighted time warps. Knowledge and Information Systems, 2018,59(1):117–135. [doi: 10.1007/s10115-018-1184-z]

[12] Yuan JD, Wang ZH, Han M. Shapelet pruning and shapelet coverage for time series classification. Ruan Jian Xue Bao/Journal of Software, 2015,26(9):2311–2325 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4702.htm [doi: 10.13328/j.cnki.jos.004702]

[13] Yuan JD, Wang ZH, Han M, You Y. A logical shapelets transformation for time series classification. Chinese Journal of Computers, 2015,38(7):1448–1459 (in Chinese with English abstract). [doi: 0.11897/SP.J.1016.2015.01448]

[14] Neuyen TL, Gsponer S, Ifrim G. Time series classification by sequence learning in all-subsequence space. In: Proc. of the 33th Int'l Conf. on Data Engineering (ICDE 2017). San Diego: IEEE, 2017. 947–958. [doi: 10.1109/ICDE.2017.142]

[15] Shi M, Wang Z, Yuan J, Liu H. Random pairwise shapelets forest. In: Proc. of the 22th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2018). Cham: Springer-Verlag, 2018. 68–80. [doi: 10.1007/978-3-319-93034-3_6]

[16] Bostrom A, Bagnall A. Binary shapelet transform for multiclass time series classification. In: Hameurlain A, ed. Trans. on Large-scale Data- and Knowledge-centered Systems XXXII. LNCS 10420, Berlin: Springer-Verlag, 2017. 24–46. [doi: 10.1007/978-3-662-55608-5_2]

[17] Bagnall A, Lines J, Hills J, Bostrom A. Time-series classification with COTE: The collective of transformation-based ensembles. IEEE Trans. on Knowledge and Data Engineering, 2015,27(9):2522–2535. [doi: 10.1109/TKDE.2015.2416723]

[18] Lin J, Keogh E, Li W, Lonardi S. Experiencing SAX: A novel symbolic representation of time series. Data Mining and Knowledge Discovery, 2007,15(2):107–144. [doi: 10.1007/s10618-007-0064-z]

[19] Senin P, Malinchik S. SAX-VSM: Interpretable time series classification using SAX and vector space model. In: Proc. of the 13th IEEE Int'l Conf. on Data Mining (ICDM2013). Dallas: IEEE, 2013. 1175–1180. [doi: 10.1109/ICDM.2013.52]

[20] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence databases. In: Proc. of the 4th Int'l Conf. on Foundations of Data Organization and Algorithms. Berlin, Heidelberg: Springer-Verlag, 1993. 69–84. [doi: 10.1007/3-540-57301-1_5]

[21] Rafiei D, Mendelzon A. Efficient retrieval of similar time sequences using DFT. In: Proc. of the 5th Int'l Conf. of Foundations of Data Organization (FODO 1998). Kobe: IEEE, 1998. 249–257. [doi: 10.1109/icde.1998.655778]

[22] Schafer P, Högqvist M. SFA: A symbolic Fourier approximation and index for similarity search in high dimensional datasets. In: Proc. of the 15th Int'l Conf. on Extending Database Technology. Berlin: ACM, 2012. 516–527. [doi: 10.1145/2247596.2247656]

[23] Schafer P. The BOSS is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery, 2015,29(6):1505–1530. [doi: 10.1007/s10618-014-0377-7]

[24] Schafer P, Leser U. Fast and accurate time series classification with WEASEL. In: Proc. of the 26th ACM Int'l Conf. on Information and Knowledge Management (CIKM 2017). ACM, 2017. 637–646. [doi: 10.1145/3132847.3132980]

[25] Oppenheim AV, Schafer RW. Digital Signal Processing. Englewood Cliffs: Prentice Hall, 1975.

[26] Erra U, Senatore S, Minnella F, Caggianese G. Approximate TF-IDF based on topic extraction from massive message stream using the GPU. Information Sciences, 2015,292:143–161. [doi: 10.1016/j.ins.2014.08.062]

[27] Chen K, Zhang Z, Long J, Zhang H. Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Systems with Applications, 2016,66:245–260. [doi: 10.1016/j.eswa.2016.09.009]

[28] Schafer P. Scalable time series classification. Data Mining and Knowledge Discovery, 2016,30(5):1273–1298. [doi: 10.1007/s10618-015-0441-y]

[29] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 2008,9:1871–1874.

[30] Bagnall A, Lines J, Vickers W, Keogh E. The UEA & UCR time series classification repository. http://www.timeseriesclassification.com

[31] Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 2006,7(1):1–30.

[32] Rakthanmanon T, Keogh E. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: Proc. of the 13th SIAM Int'l Conf. on Data Mining. Austin: SIAM Press, 2013. 668–676. [doi: 10.1137/1.9781611972832.74]

[33] Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L. Learning time-series shapelets. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 392–401. [doi: 10.1145/2623330.2623613]

[34] Kate RJ. Using dynamic time warping distances as features for improved time series classification. Data Mining and Knowledge Discovery, 2016,30(2):283–312. [doi: 10.1007/s10618-015-0418-x]

[35] Deng H, Runger G, Tuv E, Vladimir M. A time series forest for classification and feature extraction. Information Sciences, 2013,239:142–153. [doi: 10.1016/j.ins.2013.02.030]

附中文参考文献:

[8] 原继东,王志海,孙艳歌,张伟.面向复杂时间序列的 k 近邻分类器.软件学报,2017,28(1):3002–3017. http://www.jos.org.cn/1000-9825/5331.htm [doi: 10.13328/j.cnki.jos.005331]

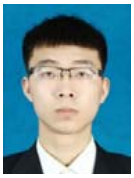
[12] 原继东,王志海,韩萌.基于 Shapelet 剪枝和覆盖的时间序列分类算法.软件学报,2015,26(9):2311–2325. http://www.jos.org.cn/1000-9825/4702.htm [doi: 10.13328/j.cnki.jos.004702]

[13] 原继东,王志海,韩萌,等.基于逻辑 shapelets 转换的时间序列分类算法.计算机学报,2015,38(7):1448–1459. [doi: 0.11897/SP.J.1016.2015.01448]

张伟(1987－),男,博士生,主要研究领域为数据挖掘,机器学习,时间序列分类.

原继东(1989－),男,博士,副教授,CCF 专业会员,主要研究领域为数据挖掘,模式识别.

王志海(1963－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为数据挖掘,机器学习.



郝石磊(1995－),男,博士生,主要研究领域为数据挖掘.