# 基于二分迭代SAX的时序相似性度量算法

张建辉 王会青 孙宏伟 郭芷榕 白莹莹
(太原理工大学计算机科学与技术学院 太原 030600)

**摘 要** 时序降维是解决时间序列高维问题的关键技术。符号聚集近似表示(SAX表示法)作为一种时序降维技术,具有良好的维度约简能力与性能稳定的下界距离算法,但算法中分段数的选取需根据当前时序数据的特征而人为设定。针对这一问题,引入了滑动窗口算法与统计学方法,提出了基于二分迭代SAX的时序相似性度量算法。实验结果表明,该算法不仅解决了分段数设定困难的问题,而且降低了时序降维表示的复杂度,提高了SAX算法在多种时序数据上的分类准确性。

**关键词** 时序降维,符号聚集近似,滑动窗口
**中图法分类号** TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.046

## Similarity Measure Algorithm of Time Series Based on Binary-dividing SAX

ZHANG Jian-hui WANG Hui-qing SUN Hong-wei GUO Zhi-rong BAI Ying-ying
(College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030600, China)

**Abstract** Time series dimentionality reduction technology is used to resolve high dimensionality time series. Symbolic aggregate appro ximation (SAX representation) is a time series dimensionality reduction technique which benefits from its brief representation in dimensionality reduction and high-performance lower bound distance algorithm, but there is a question that the number of segments, a parameter in SAX, is set artificially based on the characteristic of individual time series. To solve this problem, similarity measure algorithm of time series based on binary-dividing SAX was presented by introducing sliding window and statistical methods. The experimental results show that binary-dividing SAX algorithm not only solves the difficulty to choose the number of segments, but also reduces the complexity of time series representation in dimensionality reduction and improves classification accuracy by using the SAX algorithm in a variety of time series data.

**Keywords** Dimensionality reduction, Symbolic aggregate approximation, Sliding window

## 1 引言

多元时间序列、数据流处理是数据挖掘领域研究的热点之一,其中存在大量有价值的时间序列待被挖掘。面对如此多元复杂亦或海量高维的时间序列处理的挑战,若直接对如此庞杂的时间序列进行挖掘,不但需要消耗大量的时间与空间,而且最终也很难得到科学合理的结果[1],提高时序挖掘技术的效率是必要而有意义的。时序降维技术不仅可以降低时间序列表示的复杂度,进而提高任务的运算速度,而且能够有效地对序列的特征信息进行提取[2],提高相关数据挖掘技术的性能。各领域研究人员相继提出了多种时间序列的降维表示方法,包括离散小波变换(Discrete Wavelet Transform)、离散傅里叶变换(Discrete Fourier Transform)、奇异值分解(Singular Value Decomposition)、分段聚集近似(Piecewise Aggregate Approximation)、符号聚集近似(Symbolic Aggregate approXimation, SAX)[3]等。SAX表示法因其较好的维度约简特性与下界距离算法,在聚类、分类与异常检测等研究领域中得到了广泛的应用[2]。由于时间序列数据种类多样且特征复杂,导致SAX表示法在不同时间序列中进行分段数的选择时不能一概而论,而SAX表示法的有效性与精度在很大程度上依赖于分段数的选择。在处理未知特征的时间序列数据时,如何选择合适的分段数成为了SAX算法的关键。

针对这一问题,本文引入了滑动窗口算法与统计学方法,转变了SAX算法中划分分段的方式与顺序,提出了基于二分迭代的SAX表示法并改进了SAX下界距离算法,有效地解决了SAX表示法的分段数设定与非等长符号序列间下界距离计算困难的问题。

本文首先介绍了SAX表示法的相关研究;第3节通过分析SAX算法的现有问题,提出了基于二分迭代的SAX表示法和基于二分迭代SAX表示的下界距离算法;第4节通过实验验证了算法的性能,并讨论与分析了实验结果。

到稿日期:2016-03-29 返修日期:2016-08-17 本文受国家基金项目(61402318),山西省科技攻关项目(20130313012-2,201603D221037-2),校青年团队项目(2013T0490),博士点基金项目(20131402120009)资助。

张建辉(1991—),男,硕士生,主要研究方向为机器学习与人工智能,E-mail:ZJH_0518@126.com;王会青(1978—),女,博士,副教授,主要研究方向为数据挖掘、机器学习,E-mail:tywanghq@163.com(通信作者);孙宏伟(1989—),男,硕士,主要研究方向为数据挖掘与机器学习;郭芷榕(1991—),男,硕士生,主要研究方向为机器学习与人工智能;白莹莹(1992—),女,硕士生,主要研究方向为机器学习与人工智能。

## 2 SAX 表示法的相关工作

符号表示法是通过符号表征实数序列特性的一种时序表示法。由于符号化表示的特征更易被获取，因此基于符号化表示的时序挖掘技术的效率与性能得以提升。SAX 符号表示法结合时序数值分布的特点把整个值域划分为数个分布概率相等的区域，并以符号表示每个划分值域的数值，能够均匀地把出现频率相似、连续的数值归类。Keogh 等人[3-7]提出了多种时间序列的符号表示法：SAX 表示法、Experiencing SAX 表示法、iSAX 符号表示法、iSAX2.0 符号表示法、RA-SAX 等。近几年相继又有许多学者针对 SAX 表示法的局限性进行了相应改进，提出了 Extended SAX 表示法[8]、NSAX 表示法[9]、SAX-TD 表示法[10]、FastSAX 表示法[11]。

### 2.1 SAX 表示法

Keogh 等人[3]在 PAA(分段聚集近似)理论的基础上，结合时序的正态分布的特性，提出了对时间序列符号化转换的 SAX 表示法，它不仅能够有效地降低时序的维度，而且能够支持实时数据的流式转换，在聚类、分类与异常检测等算法中都有着良好的表现。

时间序列的 SAX 表示法的过程如下：

(1)采用正态标准化公式对时间序列进行处理，变换时间序列数据使其服从标准正态分布。

(2)给定长度为 $n$ 的参考时间序列 $C=c_1,c_2,\cdots,c_n$，其经过 PAA 降维处理，用 $w$ 维向量 $\overline{C}=\bar{c}_1,\bar{c}_2,\bar{c}_3,\cdots,\bar{c}_w$ 表示，其中 $\bar{c}_i$ 由式(1)得出：

$$\bar{c}_i=\frac{w}{n}\sum_{j=\frac{n(i-1)}{w}+1}^{\frac{n_i}{w}}c_i \tag{1}$$

其中，$w$ 为分段数，即 PAA 降维后的序列维度。

(3)PAA 降维表示的序列，根据序列的分布特性，利用标准正态分布概率密度函数的性质，设定数个分割点把值域划分为时序数值等概率分布的多个区域，将序列离散化并以符号表示。

符号化序列，首先选定表示符号集与划分数值区域的符号数量(即分割点的数量)，确定离散化的程度。对已标准化的时间序列上的每一个数值，判断其所属的划分区域，并用此数值区域的对应符号表示，最终得出一条完整的符号序列。分割点数值如图 1 所示，图中 $\alpha$ 为符号数，$\beta$ 为分割点取值。

| $\beta_i$ \ $\alpha$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|
| $\beta_1$ | −0.43 | −0.67 | −0.84 | −0.97 | −1.07 | −1.15 | −1.22 | −1.28 |
| $\beta_2$ | 0.43 | 0 | −0.25 | −0.43 | −0.57 | −0.67 | −0.76 | −0.84 |
| $\beta_3$ | | 0.67 | 0.25 | 0 | −0.18 | −0.32 | −0.43 | −0.52 |
| $\beta_4$ | | | 0.84 | 0.43 | 0.18 | 0 | −0.14 | −0.25 |
| $\beta_5$ | | | | 0.97 | 0.57 | 0.32 | 0.14 | 0 |
| $\beta_6$ | | | | | 1.07 | 0.67 | 0.43 | 0.25 |
| $\beta_7$ | | | | | | 1.15 | 0.76 | 0.84 |
| $\beta_8$ | | | | | | | 1.22 | 0.84 |
| $\beta_9$ | | | | | | | | 1.28 |

图 1 分割点取值图

设 $\hat{Q}$、$\hat{C}$ 分别是序列 $Q$ 和 $C$ 经过 SAX 表示后的符号序列，序列 $\hat{Q}$ 与 $\hat{C}$ 中对应的每对符号间的距离为：

$$dist(\hat{q},\hat{c})=\begin{cases}0, & \text{if } |\hat{q}-\hat{c}|\leqslant 1\\ \beta_{\max(\hat{q},\hat{c})-1}-\beta_{\min(\hat{q},\hat{c})}, & \text{otherwise}\end{cases} \tag{2}$$

则序列 $\hat{Q}$ 与 $\hat{C}$ 的下界距离为：

$$MINDIST(\hat{Q},\hat{C})=\sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}(dist(\hat{q}_i,\hat{c}_i))^2} \tag{3}$$

其中，$\hat{Q}$ 与 $\hat{C}$ 分别是序列 $Q$ 与 $C$ 的 SAX 表示序列。

### 2.2 SAX-TD 表示

由于 SAX 表示法中每个符号都是由各分段的均值进行对应变换而来，忽略了每个分段内值的变化趋势，尤其是对于均值相同但变化趋势差异很大的分段间的距离度量，往往会导致与其真实距离存在较大的偏差。Sun 等人[10]针对此问题，用分段起始点与终结点构造了分段间的趋势距离，并整合 SAX 下界距离算法，提出了 SAX-TD 下界距离算法。

已知序列 $Q=q_1,q_2,\cdots,q_n$ 与 $C=c_1,c_2,\cdots,c_n$，SAX-TD 下界距离算法用分段均值、分段起始点取值与终止点取值定义两个序列分段间的趋势距离，如式(4)所示。

$$td(q,c)=\sqrt{(\Delta q(t_s)-\Delta c(t_s))^2+(\Delta q(t_e)-\Delta c(t_e))^2} \tag{4}$$

其中，$\Delta q(t)=q(t)-\bar{q}$，$q(t_s)$ 与 $q(t_e)$ 分别代表序列 $Q$ 中第 $t$ 个分段的起始点变化值与终止点变化值。同理可得 $c(t_s)$、$c(t_e)$ 与 $\Delta c(t)$。

结合分段的趋势变量，序列 $Q$ 与 $C$ 的 SAX-TD 下界距离为：

$$TDIST(Q,C)=\sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}(((dist(q_i,c_i))^2+\frac{w}{n}(td(q_i,c_i))^2)} \tag{5}$$

### 2.3 限定符号误差的 SAX 表示法

SAX 表示法能够降低时序数据中细微噪声对特征提取的影响，但并未量化噪声数据在不同分段数与符号数下导致的 SAX 表示法的误差。Stylios 与 Kreinovich[12]通过分析待处理序列值的分布特征，优化了分割点数量(即符号数)，使 SAX 表示序列的误差达到最小；同时研究了 SAX 表示法在内存中的二进制转换，证实了在不影响精度的情况下能够调整分割点的近似表示，使符号化表示序列所占的空间最少，从而加快运算的速度。

限定符号误差的 SAX 表示法通过将符号误差优化问题转化为最小二乘的优化问题，并构造关于符号数函数关系的拉格朗日乘子式，再结合原误差函数计算其极值，从而得出结论：满足式(6)的分割点划分能够使 SAX 表示法的误差最小：

$$\rho_t(x)=\frac{(\rho(x))^{1/3}}{\int(\rho(y))^{1/3}\mathrm{d}y} \tag{6}$$

其中，$x$ 是序列值域内任意值，$\rho_t(x)$ 是单位值域内分割点数，$\rho(x)$ 是序列 $x$ 的概率密度函数。

### 2.4 其他 SAX 相关方法

研究人员通过结合不同领域技术，提出了多种基于 SAX 符号表示的方法。Lin 等人[4]提出了一种新的降维符号表示法 Experiencing SAX 与其表示法的下界距离算法。Shieh 等人[5]提出了支持百万级时序数据索引的多辨析 *i*SAX 符号表示法。Tayebi 等人[7]针对移动设备上的实时 ECG 数据分析，提出了一种基于移动设备端的时间序列分析技术 RA-SAX。Camerra 等人[6]通过对 *i*SAX 进行改进，提出了针对十亿级的时间序列集检索的数据结构 *i*SAX 2.0。Payam 等人[13]将 SAX 表示法应用于支持模式分析与识别的移动传感数据处理框架。Qinkun 等人[14]基于自组建网络对人体运动捕捉数据，

构建了 SAX 的动态索引结构。Arthit 等人[15]通过将 SAX 表示法与人工神经网络结合，提出了可预测风暴强度的神经网络模型。

以上算法针对经济、医学与大数据等不同领域中的实际应用问题，结合了多种数据挖掘技术，提高了 SAX 算法在各领域的适用性。

由于 SAX 相关算法并未考虑个别数据集的特征，忽略了同类数据不同时间序列间的差异而对其设定相同的参数进行维度约减处理，显然会导致 SAX 算法在个别数据集的表现不稳定，甚至差强人意；SAX 算法中的分段数直接决定了其表示的准确性与精度，人为拟定分段数的不确定性影响了基于 SAX 表示的数据挖掘算法的性能，因此往往需对处理的结果进行必要的选择与优化。

## 3　基于二分迭代的 SAX 相似性算法

分段数与符号数是 SAX 表示法的两个关键参数。符号数决定了数据取值离散化的程度，SAX 表示法的表示符号数越少，其数据离散化的程度就越明显，反之则反。Eamonn 等人[3]指出当符号数在 2～10 之间时，可以明显地反映其时间序列值的特征。Sun 等人[10]通过实验进一步确定符号数的范围在 5～8 时可达到更优的时序分类效果。符号数的选择已经趋近于一个可行的范围，但直接决定降维表示序列长度的分段数仍是现今 SAX 相关方法研究中面临的难题。

为了解决 SAX 算法中分段数选择的问题，提出了基于二分迭代的 SAX 表示法及其下界距离算法，本算法结合了滑动窗口方法中窗口划分的依据，引入了统计学中的方差并转变了窗口划分的方式与顺序，能够自适应地根据时间序列的数值分布特性对序列进行分段划分，解决了分段数选择的不确定性并提高了 SAX 降维表示的准确性。实验验证了基于二分迭代的 SAX 算法在多种时序数据上的分类结果较现有其他 SAX 算法有明显的改善。

### 3.1　二分迭代窗口划分法

由于时间序列数据的来源庞杂、特征各异且长度差异很大，导致 SAX 表示法中分段数的选择受时间序列自身特征分布的影响，往往不能够准确地限定其范围，这严重限制了 SAX 表示法的使用。滑动窗口算法通过设定一个窗口阈值，限制每个时间序列分段中最大值与最小值的差异，有效地按值波动的大小把时间序列分割为波幅相似的分段。二分迭代窗口划分法通过引入滑动窗口算法的思想，把时间序列划分为均匀的分段，为 SAX 表示法中时序降维与符号化处理提供了基础。

时间序列的波动幅度是其非常重要的特征，而方差是反应数据波动特性的一个重要指标。根据统计学中的方差理论，二分迭代窗口划分法以时间序列的方差作为滑动窗口阈值选定的重要参考。

已知时间序列 $Q=q_1,q_2,\cdots,q_m$，划分窗口的阈值为 $\varepsilon$。序列 $Q$ 划分后每个分段长度占序列总长的比例按原序列中的顺序组成长度比序列 $W=w_1,w_2,w_3,\cdots,w_t$（$w_i$ 分别代表第 $i$ 个分段的长度比），首次划分时只有一个分段（即原序列），且长度比为 1，即 $W$ 为{1}。

(1)序列 Q 转化为服从期望为 0 且方差为 1 的标准正态分布，式(7)为序列标准化公式。

$$q_i=\frac{q_i-\mu_q}{\sigma_q} \tag{7}$$

其中，$\mu_q$ 为序列 $Q$ 的均值，$\sigma_q$ 为序列 $Q$ 的方差。

(2)对已标准化的序列 $Q$ 进行二分迭代窗口划分。在序列 $Q$ 中，若 $\max(q_1,q_2,q_3,\cdots,q_m)-\min(q_1,q_2,q_3,\cdots,q_m)\leqslant\varepsilon$，则此段不具备二分迭代窗口划分的条件，结束此段序列的划分，并返回此段的长度比，记为 $w_t$（$t$ 为此分段的序号）。

若 $\max(q_1,q_2,q_3,\cdots,q_m)-\min(q_1,q_2,q_3,\cdots,q_m)>\varepsilon$，则序列满足二分迭代窗口划分的条件，把原序列均分为两个长度相等的子序列：$\{q_1,q_2,\cdots,q_{\frac{m+1}{2}}\}$ 与 $\{q_{\frac{m+1}{2}+1},q_{\frac{m+1}{2}+2},\cdots,q_m\}$。并由当前序列的长度占比 $w_t$ 得出两个子序列的长度占比$\{\frac{w_t}{2},\frac{w_t}{2}\}$。

然后，再分别对这两个子序列重复上述的划分过程，直至所有分段均不满足划分条件，即认为完成了时序的二分迭代窗口划分。

二分迭代表示法是以方差作为窗口划分的阈值，采用递归的思想，自顶向下逐层地二等分序列，直至所有分段均不符合分段划分的条件。

图 2 为伯克利大学实验数据集 ItalyPowerDemand 中的某一条时间序列数据，图 2 中(a)－(d)分别给出了二分迭代窗口的第一轮到第四轮的划分结果。

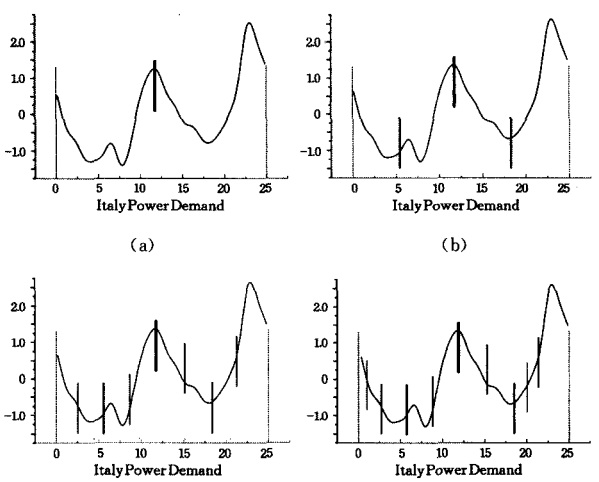



图 2　数据集 ItalyPowerDemand 上的二分迭代窗口划分

### 3.2　基于二分迭代的 SAX 表示法

原序列经过二分迭代窗口划分之后，为实现序列间的相似性度量或下界距离计算，需对序列的每个分段进行 SAX 符号表示。

(1)对于已二分迭代窗口划分的序列，求得每个分段的均值$\bar{q}_i$，得到均值表示序列$\bar{Q}=\bar{q}_1,\bar{q}_2,\bar{q}_3,\cdots,\bar{q}_t$（$t$ 为分段序号）以及分段的长度比序列 $W=w_1,w_2,w_3,\cdots,w_t$。

(2)对序列$\bar{Q}$中的值域离散化处理，可根据图 1 中的分割点，把时间序列的值域分割为数个等概率分布的数值区域，落入每个分割区域的值都对应表示为自定义符号集内的一个符号。即把原序列转化为符号表示序列$\hat{Q}=\hat{q}_1,\hat{q}_2,\hat{q}_3,\cdots,\hat{q}_t$。

图 3 为已标准化的某一序列的 SAX 表示过程，序列内每一分段的均值都以符号表示，其最终的 SAX 表示符号序列为：

$\{a,a,c,e,e,c\}$，其中分割的值域由低到高分别由符号表中 $a$，$b$，$c$，$d$，$e$ 来表示。

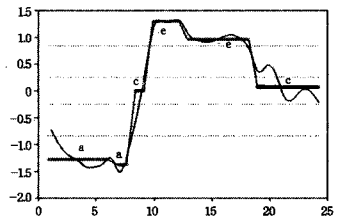


图 3 序列的 SAX 符号表示

### 3.3 基于二分迭代的 SAX 下界距离算法

SAX 下界距离算法只支持长度相同的符号序列间的相似性度量，但基于二分迭代的 SAX 表示符号序列的长度并不确定，导致 SAX 下界算法不能直接计算基于二分迭代的 SAX 表示序列间的下界距离。二分迭代窗口划分法通过不断地分割振幅不均匀的序列，由迭代分割的次数构造序列中每个分段的长度比(分段长度占序列总长的比例)与二分迭代窗口划分法的序列分割点，为不同序列内非等长分段间的匹配与下界距离的计算提供了基础。下文给出基于二分迭代 SAX 表示的下界距离算法。

已知两个时间序列 $Q$ 和 $C$，经过二分迭代 SAX 表示后的基于二分迭代 SAX 表示的两个序列 $\hat{Q}=\hat{q}_1,\hat{q}_2,\hat{q}_3,\cdots,\hat{q}_t$，$\hat{C}=\hat{c}_1,\hat{c}_2,\hat{c}_3,\cdots,\hat{c}_t$；以及序列 $\hat{Q}$ 与序列 $\hat{C}$ 对应的长度比序列 $W_q=w_{q1},w_{q1},w_{q2},w_{q3},\cdots,w_{qs}$ 与 $W_c=w_{c1},w_{c2},w_{c3},\cdots,w_{cs}$。

序列 $\hat{Q}$ 与 $\hat{C}$ 对应匹配点间的下界距离按式(8)计算。

$$dist(q_i,c_j)=\begin{cases} w_{qi}(\text{或 } w_{cj})\times SAX\_dist(q_i,c_j), & \text{当 } w_{qi}=w_{cj}\neq 0 \\ w_{cj}\times SAX\_dist(q_i,c_j)\text{且 } w_{qi}=w_{qi}-w_{cj}, & \text{当 } w_{qi}>w_{cj} \\ w_{qi}\times SAX\_dist(q_i,c_j)\text{且 } w_{cj}=w_{cj}-w_{qi}, & \text{当 } w_{qi}<w_{cj} \end{cases} \tag{8}$$

其中，$SAX\_dist(q_i,c_j)$ 为 $q_i$ 与 $c_j$ 的 SAX 下界距离。当 $w_{qi}=0$ 时，终止 $q_i$ 的计算，开始计算 $q_{i+1}$ 与序列 $\hat{C}$ 的下界距离；$w_{cj}=0$ 时同理。

如式(9)所示，对所有对应匹配点的下界距离求和，即可得到序列 $\hat{Q}$ 与序列 $\hat{C}$ 的基于二分迭代 SAX 下界距离。

$$Dist(\hat{Q},\hat{C})=\sum dist(q_i,c_j) \tag{9}$$

由上述可知，基于二分迭代的 SAX 下界距离算法以分段的长度比为权值引入 SAX 下界距离算法，并通过不断地修改分段权重，遍历完成序列内所有分段间的距离计算，求得分段距离的累加和，即为两序列基于二分迭代的 SAX 下界距离。

## 4 实验与分析

### 4.1 实验环境与数据

实验环境为 Matlab R2012b，Windows 7 Basic，8GB 内存，Intel(R) Core(TM)i7-4790 CPU@ 3.60GHz。

实验数据来源于加利福尼亚州大学河滨分校 Eamonn 教授实验室的开放数据发，即 UCR 时间序列分类数据(UCR Time Series Classification Archive)，是时间序列研究领域非常具有代表性与可信度的实验数据。每个实验数据集划分为训练集与测试集，数据的类别由 2 类至 58 类不等，单一时间序列的长度由 24 至 720 不等，训练集与测试集序列的数量由 28 条至 300 条不等。根据数据不同的特性，选取了其中 9 组差异明显、来源不同的时间序列数据进行实验，用于验证基于二分迭代的 SAX 度量算法的有效性与健壮性，具体数据信息如表 1 所列。

表 1 实验数据

| 数据名称 | 数据长度 | 训练集数目 | 测试集数目 | 分类数 |
|---|---|---|---|---|
| CBF | 128 | 30 | 900 | 3 |
| ECG200 | 96 | 100 | 100 | 2 |
| FaceFour | 350 | 24 | 88 | 4 |
| ItalyPowerDemand | 24 | 67 | 1029 | 2 |
| MedicalImage | 99 | 381 | 760 | 10 |
| MoteStrain | 84 | 20 | 1052 | 2 |
| SonyAIBORobotSurfaceII | 65 | 27 | 953 | 2 |
| TwoLeadECG | 83 | 23 | 1139 | 2 |
| Trace | 276 | 100 | 100 | 4 |

实验一与实验三采用 1NN 分类器，分别用 SAX 下界算法、SAX-TD 下界算法与基于二分迭代的 SAX 下界距离算法对 9 个不同领域的数据集进行分类测试，并以实验测试集的分类错误率来评价 3 种度量算法的优劣。实验二采用限定符号误差的 SAX 表示法得到数据集的优化 SAX 符号数，与实验一得到的结果一致，为实验三的参数选择提供了依据。

### 4.2 实验结果与分析

本节意在通过对上述数据进行分类实验，评价基于二分迭代的 SAX 下界距离算法的准确性与有效性。此实验通过引入 SAX 下界距离算法、SAX-TD 下界距离算法与基于二分迭代 SAX 下界距离算法进行比较。依据 Eammon 等人[16]制定的时序数据实验原则，实验通过限制 3 种算法降维表示序列的长度，分别在 9 组不同的 UCR 数据上进行分类实验，进而分析实验结果并合理地评价基于二分迭代 SAX 下界距离算法。

参考相关的文献可知，SAX 距离度量算法使序列中每个分段一对一地转化为符号，再通过相应符号查表求和得出两个序列间的距离；而 SAX-TD 算法不仅涵括了上述计算过程，同时在参与计算的序列中任意相邻的符号间都插入了趋势距离的计算，在符号序列维度相同的前提下，其实际参与距离计算的序列维度是 SAX 算法的两倍。由此可知，在设计控制变量的对比实验时，算法中分段数的设置需要做一些调整，即 SAX-TD 的分段数应为原 SAX 表示方法的一半，从而达到相同维度以便算法间的实验比较。基于二分迭代的 SAX 下界算法虽改变了 SAX 表示法的降维过程，但参与计算的序列维度与分段数一致，所以其分段数的设置应与 SAX 表示法一致。基于此参数设置的前提，即可开展实验。

实验一 此实验是基于不同窗口分割阈值在两个数据集(MedicalImage，CBF)上的 5 组分类对比实验：在不同符号数与分段数的实验参数设置下，基于二分迭代 SAX、SAX 与 SAX-TD 3 种不同的下界距离算法在 1NN 分类器下进行分类错误率比较。图 4 是在 MedicalImage 上 3 种不同的下界距离算法的对比实验。图 4(a)中二分迭代阈值参数设置为 0.5倍方差，分段数为 27；图 4(b)中二分迭代阈值参数设置为 0.75倍方差，分段数为 26；图 4(c)中二分迭代阈值参数设置为 1 倍方差，分段数为 16；图 4(d)中二分迭代阈值参数设置

为 1.5 倍方差，分段数为 11；图 4(e)中二分迭代阈值参数设置为 2 倍方差，分段数为 9。

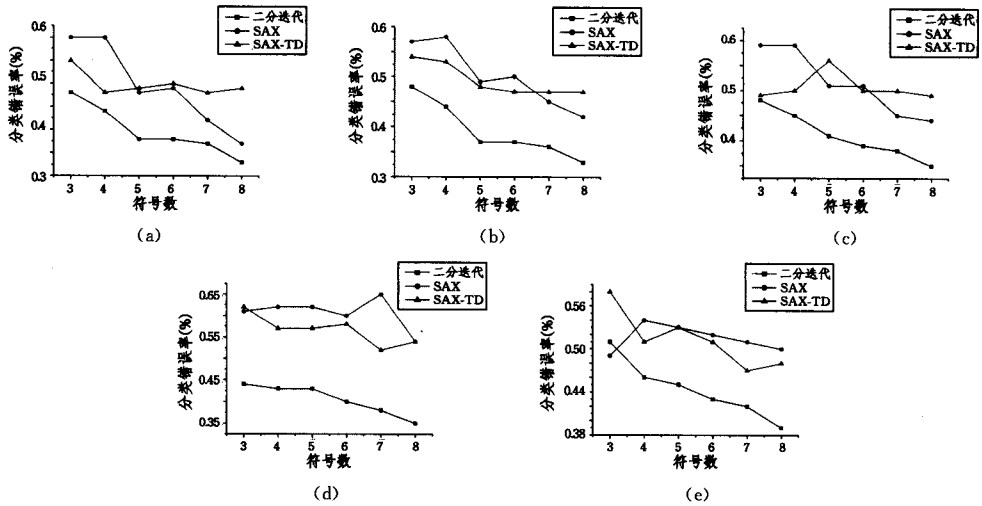



图 4　二分迭代与 SAX、SAX-TD 在 MedicalImage 上的分类错误率

图 5 是 3 种不同的距离度量算法在 CBF 上的对比实验。

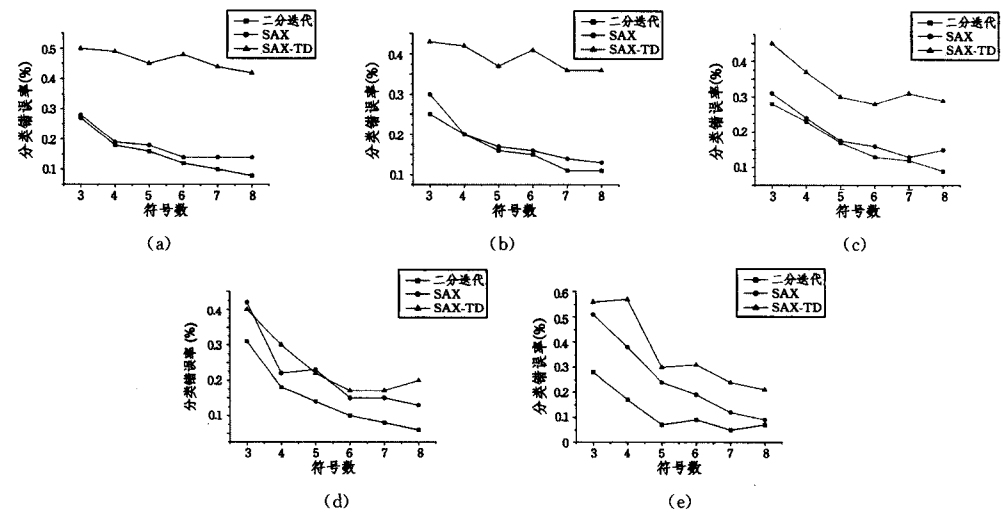



图 5　二分迭代与 SAX、SAX-TD 在 CBF 上的分类错误率

图 5(a)中二分迭代阈值参数设置为 0.5 倍方差，分段数为 100；图 5(b)中二分迭代阈值参数设置为 0.75 倍方差，分段数为 80；图 5(c)中二分迭代阈值参数设置为 1 倍方差，分段数为 60；图 5(d)中二分迭代阈值参数设置为 1.5 倍方差，分段数为 30；图 5(e)中二分迭代阈值参数设置为 2 倍方差，分段数为 16。

由上述实验结果可知：1)基于二分迭代的 SAX 下界距离算法在 1NN 分类下两个数据集上的分类准确率均不同程度地高于另外两种距离度量算法；2)SAX 表示的符号数越高，其分类的准确性越高，3 种算法的分类准确性从符号数 5 开始就逐渐收敛，趋于稳定；3)窗口分割的阈值过大，会导致丢失过多的数据特征，影响分类的准确性；若其阈值过小，则会导致降维表示的分段数过多，极大地影响分类的效率。上述实验证明，窗口分割阈值为单倍方差时，能较好地分割时间序列段，使分类的准确性提高至理想的水平。

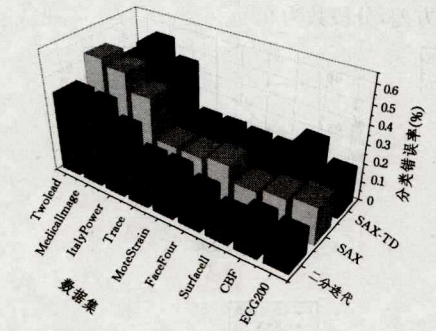
实验二　采用限定符号误差 SAX 表示法对 SAX 表示法的符号数进行进一步优化。SAX 表示法、SAX-TD 表示法与基于二分迭代的 SAX 表示法均未改变原有序列取值分布，因此采用限定符号误差的 SAX 表示法对 3 种算法的符号数优化结果是相同的。以符号误差平方和为优化目标，验证限定符号误差的 SAX 表示法是否与实验一的结果一致，实验数据包含了实验一的数据集(MedicalImage 与 CBF)，便于实验结果间的对照，如表 2 所列。

表 2　限定符号误差 SAX 表示法在 5 个数据集上的符号数优化结果

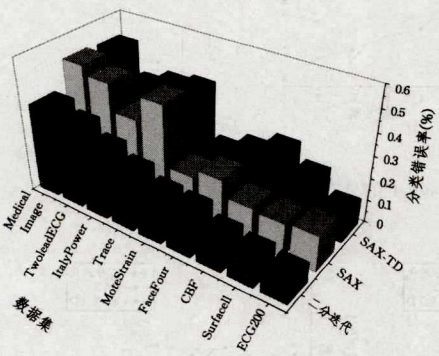
| 数据集名称 | Medical Image | CBF | ECG200 | Trace | Face Four |
|---|---|---|---|---|---|
| 最优符号数 | 5 | 5 | 5 | 5 | 6 |

由表 2 可知，限定符号误差的 SAX 表示法在 5 个数据集上的最优符号数基本选定 5～6 个，通过与实验一的实验结果进行比较可知，实验一与限定符号误差 SAX 的表示法选取的符号数基本一致，此处将最小符号误差的 5～6 个符号数作为实验参数。

实验三　SAX、SAX-TD 与基于二分迭代的 SAX 3 种不同的下界距离算法分别在 9 种不同数据集上进行分类实验比较。图 6(a)中表示在符号数为 5，分段数的设置以阈值单倍方差的二分迭代 SAX 表示法的平均分段数为准，3 种下界距离算法在 9 组不同的数据集上的分类错误率。容易得知，柱形的高度越低，其分类准确性越高。图 6(b)表示在符号数为 6 且其他参数与图 6(a)一致的情况下，3 种距离算法在相同的 9 组数据集上的分类准确性比较。




(a)




(b)

图 6 基于二分迭代的 SAX 算法与 SAX、SAX-TD 在 9 个数据集上的分类错误率

在此实验中，基于二分迭代的 SAX 下界距离算法的分类表现依然较好，但在不同的数据集上其准确性提高的程度有明显的变化，其总体的分类表现优于 SAX 与 SAX-TD 下界距离算法。综上，在一般的时间序列分类过程中，设置单倍测试集方差为窗口划分的阈值且限定的符号集数目为 5～6，基于二分迭代的 SAX 下界算法能够达到最优的分类效果。

**结束语** 本文提出的基于二分迭代窗口划分的 SAX 下界距离算法能够根据时间序列的振幅特征均匀地划分序列，有效地保留时序的特征并提高降维表示的精度，比原 SAX 相关方法具有更简单的参数设置及更低的降维表示的复杂性。

基于二分迭代的 SAX 下界距离算法在某些领域的时序分类结果较 SAX 算法表现得更突出，而在一般时序上分类处理的准确率也有提升。随着序列长度的增加，基于二分迭代的 SAX 下界距离算法的效率相应地下降，但仍与 SAX-TD 下界距离算法相近，不影响其在时序挖掘领域的推广。

## 参 考 文 献

[1] LI Hai-lin, Yang Li-bin. Method of dimensionality reduction and feature representation for time series [J]. Contorl and Decision, 2013 (11): 1718-1722. (in Chinese)
李海林，杨丽彬. 时间序列数据降维和特征表示方法[J]. 控制与决策，2013 (11)：1718-1722.

[2] DAW C S, FINNEY C E A, TRACY E R. A review of symbolic analysis of experimental data[J]. Review of Scientific Instruments, 2003, 74(2): 915-930.

[3] LIN J, KEOGH E, LONARDI S, et al. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms [C]// Proceedings of Acm Sigmod Workshop on Research Issues in Data Mining & Knowledge Discovery. 2003: 2-11.

[4] LIN J, KEOGH E, WEI L, et al. Experiencing SAX: a novel symbolic representation of time series [J]. Data Mining & Knowledge Discovery, 2007, 15(2): 107-144.

[5] SHIEH J, KEOGH E. i SAX: indexing and mining Terabyte sized time series[C]// Proceedings of the14th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining. ACM, 2008: 623-631.

[6] CAMERRA A, PALPANAS T, SHIEH J, et al. iSAX 2. 0: Indexing and Mining One Billion Time Series[C]// 2013 IEEE 13th International Conference on Data Mining. IEEE, 2010: 58-67.

[7] TAYEBI H, KRISHNASWAMY S, WALUYO A B, et al. RA-SAX: Resource-Aware Symbolic Aggregate Approximation for Mobile ECG Analysis[C]// 2011 12th IEEE International Conference on Mobile Data Management. IEEE Computer Society, 2011: 289-290.

[8] LKHAGVA B, SUZUKI Y, KAWAGOE K. Extended SAX: Extension of symbolic aggregate approximation for financialtime series data representation[C]// DEWS 2006. 2006.

[9] HE X, SHAO C, XIONG Y. A non-parametric symbolic approximate representation for long time series[J]. Formal Pattern Analysis & Applications, 2016, 19(1): 111-127.

[10] SUN Y, LI J, LIU J, et al. An improvement of symbolic aggregate approximation distance measure for time series[J]. Neurocomputing, 2014, 138(11): 189-198.

[11] FUAD M M M, MARTEAU P F. Towards a faster symbolic aggregate approximation method[C]// ICSOFT 2010. 2010.

[12] STYLIOS C D, KREINOVICH V. Symbolic Aggregate approXimation (SAX) under interval uncertainty[C]// Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), 2015 Annual Conference of the North American. IEEE, 2015.

[13] BARNAGHI P, GANZ F, HENSON C, et al. Computing perception from sensor data[C]// Sensors, 2012 IEEE. IEEE, 2012: 1-4.

[14] XIAO Q, LUO Y, GAO S. Human Motion Retrieval with Symbolic Aggregate approXimation[C]// Control and Decision Conference. 2012: 3632-3636.

[15] BURANASING A, PRAYOTE A. Storm intensity estimation using symbolic aggregate approximation and artificial neural network[C]// International Computer Science and Engineering Conference. 2014.

[16] KEOGH E, ZHU Q, HU B, et al. Ratanamahatana, The UCR time seriesclassification/clustering homepage [OL]. http://www.cs.ucr.edu/~eamonn/timeseries_data/.