一种时间序列动态聚类的算法*

谢福鼎[1]，赵晓慧[2]，嵇　敏[2]，平　宇[3]

(1. 辽宁师范大学 城市与环境学院，辽宁 大连 116029；2. 辽宁师范大学 计算机与信息技术学院，辽宁 大连 116081；3. 同济大学 电子信息工程学院，上海 201804)

摘　要：针对时间序列传统静态聚类问题，提出了对时间序列进行动态聚类的方法。该方法首先提取时间序列的关键点集合，根据改进的 FCM 算法找到动态特征明显的时间序列，再利用提出的动态聚类算法确定此类时间序列在不同时间段的所属类别，在改进的 FCM 算法中采用兰氏距离可以使其对奇异值不敏感。实验结果反映出动态特征明显的时间序列类别随时间演化的特性，表明了方法的可行性和有效性。与已有算法相比，该方法揭示了时间序列的部分动态特征。该方法还可以运用于研究数据挖掘的其他问题。

关键词：时间序列；关键点；兰氏距离；模糊聚类算法；动态聚类

中图分类号：TP391　　**文献标志码**：A　　**文章编号**：1001-3695(2012)10-3677-04

doi:10.3969/j.issn.1001-3695.2012.10.019

Dynamic clustering algorithm for time series

XIE Fu-ding[1], ZHAO Xiao-hui[2], JI Min[2], PING Yu[3]

(1. *School of Urban & Environmental Science, Liaoning Normal University, Dalian Liaoning* 116029, *China*; 2. *School of Computer & Information Technology, Liaoning Normal University, Dalian Liaoning* 116081, *China*; 3. *School of Electronics & Information Engineering, Tongji University, Shanghai* 201804, *China*)

Abstract: This paper proposed a dynamic clustering algorithm for time series aiming at solving the shortcoming of traditional static clustering. Firstly, the method extracted the key point set of each time series, and then obtained the dynamic time series by using improved FCM algorithm. At last, detected the cluster of dynamic time series which belonged to each time segment based on the dynamic clustering algorithm. The adoption of L-W distance in FCM algorithm could avoid the shortcoming of sensitivity to singular value. The experimental results obtained by the proposal reflect the evolutional property that the clusters of the dynamic time series change over time, and show the validity and the feasibility of the method. Compared with existed algorithms, the proposed algorithm indicates the dynamic characteristic of time series when clustering them. This algorithm can also be applied to other problems in data mining.

Key words: time series; key points; L-W distance; FCM algorithm; dynamic clustering

时间序列是按照时间顺序排列的一组数据，它反映了某一事物或现象的状态或程度随时间变化的规律，其广泛存在于生物、医学、工程和经济等各个领域。目前在对时间序列相似性、关联性以及预测模型等方面的研究中，聚类分析作为模式发现的主要方法，越来越受到人们的重视[1~4]。

聚类方法一般可以分为硬聚类和软聚类两种。硬聚类方法是将每个数据对象明确地分派到单个类中，如经典的 K-means、K-medoids 和谱聚类算法；软聚类方法通常被称做模糊聚类，是当数据对象并不明确属于一个类时，将其合理地放入多个类中，避免了随意分派数据对象到单个类的问题，如 fuzzy C-means 和 fuzzy C-medoids 算法[5]。由于时间序列具有规模大和维数高的特点，对其直接进行聚类效果并不理想，因此需要通过提取时间序列特征的方法达到降维的目的，如 Maharaj 等人提出的自相关系数[6]、谱密度特征[7~9]和小波法[10,11]，Corduas[12]通过 ARMA 模型表示时间序列，Vilar 等人[13]引入了预测密度法。

目前已有的大多数时间序列聚类算法，均是将长度为 n 的时间序列看做 n 维空间的一个点，利用处理静态数据的方法聚类时间序列。但是时间序列具有随时间演化的动态性，将它们作为整体聚类，不能反映出时间序列随时间变化的特性。如图 1 所示，类 1~3 中的时间序列整体形态基本相同，可以将其聚类到各自类中；但也存在如虚线所示的时间序列，在不同时间段表现出不同类的特征，将这样的时间序列聚类到一个类中是不能反映出该序列的基本特性的。

针对上述问题，本文提出了一种时间序列动态聚类的算法。该算法通过提取时间序列的关键点达到降维的目的，同时采用兰氏距离克服了算法对奇异值敏感的缺点，有效地解决了时间序列数据集中大量噪音影响聚类结果的问题。对于如图 1 中虚线表示的时间序列，本文算法能够根据不同时间段将其聚类到不同类别，所得到的结果可以帮助人们进一步理解时间序列的动态演化性质，准确有效地把握其结构特征。

1　关键点与 FCM 算法

1.1　关键点的定义

定义 1　关键点。若点(x_i, t_i)是时间序列 X 的关键点，则

收稿日期：2012-03-23；**修回日期**：2012-04-30　　**基金项目**：国家自然科学基金资助项目(10771092)

作者简介：谢福鼎(1965-)，男，教授，博士，主要研究方向为人工智能、数据挖掘(fuding.xie@gmail.com)；赵晓慧(1987-)，女，硕士研究生，主要研究方向为数据挖掘；嵇敏(1971-)，女，讲师，主要研究方向为数据挖掘；平宇(1988-)，男，硕士研究生，主要研究方向为数据挖掘.

其满足以下两个条件：

a) $-1 \leqslant \cos\alpha = \dfrac{m \cdot n}{|m||n|} \leqslant a \quad a \in [-1, 0]$ 或

$$0 \leqslant \cos\alpha = \frac{m \cdot n}{|m||n|} \leqslant b \quad b \in [0, 1] \tag{1}$$

b) 点 (x_j, t_j) 也满足条件 a)，且与 (x_i, t_i) 相邻并满足条件

$$|t_j - t_i| \geqslant \lambda \Delta t \tag{2}$$

其中：$m = (x_i - x_{i-1}, t_i - t_{i-1})$；$n = (x_{i+1} - x_i, t_{i+1} - t_i)$；$\Delta t = t_{i+1} - t_i$；$a$、$b$、$\lambda$ 为阈值。条件 b) 中的相邻是指在所有满足条件 a) 的点集中与 (x_i, t_i) 相邻。若时间序列步长相等，则 Δt 为常数。本文用到的时间序列均为等步长。

由于时间序列的关键点包含了重要信息，它们可能是极大值、极小值或拐点，所以用关键点集合代替原始时间序列，不仅节省了储存空间，降低了计算代价，而且保持了时间序列的主要形态特征，提高聚类算法的准确性和可靠性。

1.2 改进的 FCM 算法

提取关键点得到的关键点序列不等步长且不等长。为了计算两条时间序列的相似度，本文提出了合并关键点下标的方法，使其达到等长的目的。设有两条时间序列 X_h 和 X_i，由关键点定义得到关键点序列 $\tilde{X}_h = \langle x_{hr_1}, x_{hr_2}, \cdots, x_{hr_p} \rangle$ 和 $\tilde{X}_i = \langle x_{it_1}, x_{it_2}, \cdots, x_{it_s} \rangle$。合并 $\tilde{X}_h$ 和 $\tilde{X}_i$ 的下标集合 $\{r_1, r_2, \cdots, r_p\}$ 和 $\{t_1, t_2, \cdots, t_s\}$，并按升序排列，若有重复，只记一次。得到新的下标集合为 $\{kp_1, kp_2, \cdots, kp_l\}$，其中 $l \leqslant p + s$。得到新的等长序列 $X_h^* = \langle x_{hkp_1}, x_{hkp_2}, \cdots, x_{hkp_l} \rangle$ 和 $X_i^* = \langle x_{ikp_1}, x_{ikp_2}, \cdots, x_{ikp_l} \rangle$。

欧氏距离是数据挖掘中使用最广泛的距离度量方法，但是该距离与变量的量纲有关，变差大的变量在距离中的作用（贡献）更大。若某个关键点数值出现偏差，则会影响聚类效果。所以本文选择由 Lance 等人提出的兰氏距离[14]计算相似性。这是一个无量纲的量，它克服了欧氏距离有量纲的缺点，对大的奇异值不敏感。兰氏距离定义为

$$d^{\mathrm{L}}(X_i, X_j) = \frac{1}{m}\sum_{t=1}^{m}\frac{|x_{it} - x_{jt}|}{|x_{it}| + |x_{jt}|} \quad i, j = 1, 2, \cdots, n \tag{3}$$

其中：m 为时间序列 X_i、X_j 的长度。

FCM 算法[15]的隶属度表明每条时间序列属于各个类的程度。若隶属度较为平均，说明此时间序列很大程度上表现出不同类的特征，这样的时间序列的类别具有明显不确定性。为了理解此类序列的性质，把握序列的结构，需要进一步考虑该类序列的类别随时间变化的情形。

基于兰氏距离的 FCM 算法如下：

$$\min J_m = \sum_{i=1}^{n}\sum_{j=1}^{k}\mu_{ij}^m d^{\mathrm{L}}(X_i^*, C_j^*) = \sum_{i=1}^{n}\sum_{j=1}^{k}\left(\mu_{ij}^m \frac{1}{l}\sum_{q=1}^{l}\frac{|x_{ikp_q} - c_{jkp_q}|}{|x_{ikp_q}| + |c_{jkp_q}|}\right) \tag{4}$$

$$\text{a)}\ \mu_{ij} \in [0, 1], \forall i, j; \quad \text{b)}\ \sum_{j=1, k}\mu_{ij} = 1, \forall i \tag{5}$$

其中：$1 \leqslant m \leqslant \infty$，是 FCM 算法中用来控制模糊度的阈值。其值越大，模糊程度越高；越接近 1，FCM 算法与传统的 K-means 算法的行为就越接近。通常情况下，$m \in [1.5, 2.5]$。本文实验取 $m = 2, 2.3$。$C_j^* = \langle c_{jkp_1}, c_{jkp_2}, \cdots, c_{jkp_l} \rangle$ 是第 j 类的质心序列。每次迭代仅合并两条时间序列的关键点下标集合，使其等长的同时最大程度地降维。μ_{ij} 和 c_j 的迭代如下：

$$\mu_{ij} = \frac{(1/d^{\mathrm{L}}(x_i^*, c_j^*))^{\frac{1}{m-1}}}{\sum_{h=1}^{k}(1/d^{\mathrm{L}}(x_i^*, c_h^*))^{\frac{1}{m}}} \tag{6}$$

$$c_j = \frac{\sum_{x_i \in X}\mu_{ij}^m x_i}{\sum_{x_i \in X}\mu_{ij}^m} \tag{7}$$

改进的 FCM 算法步骤如下：

输入：时间序列数据集 X，聚类个数 k，终止阈值 ε。

输出：隶属度矩阵 $U = [\mu_{ij}]$。

a) 随机初始化质心 $C_1, C_2, \cdots, C_k$，初始化 $U = [\mu_{ij}]$，$J_m = 0$。

b) 迭代重复以下步骤：

(a) $C^{\text{previous}} \leftarrow C$ 且 $J_m^{\text{previous}} \leftarrow J_m$。

(b) 对于 $X_i \in X$，提取其关键点序列，得到 $\tilde{X}_i = \langle x_{it_1}, x_{it_2}, \cdots, x_{it_s} \rangle$；对于每一类的质心序列 $C_j \in \{C_1, C_2, \cdots, C_k\}$，提取其关键点序列为 $\tilde{C}_j = \langle c_{jr_1}, c_{jr_2}, \cdots, c_{jr_p} \rangle$；合并后的关键点序列下标 $\{t_1, t_2, \cdots, t_s\} \cup \{r_1, r_2, \cdots, r_p\} \rightarrow \{kp_1, kp_2, \cdots, kp_l\}$；$\{kp_1, kp_2, \cdots, kp_l\}$ 映射到 X_i 上，得 $X_i \rightarrow X_i^* = \langle x_{ikp_1}, x_{ikp_2}, \cdots, x_{ikp_m} \rangle$；$\{kp_1, kp_2, \cdots, kp_l\}$ 映射到 C_j 上，得 $C_j \rightarrow C_j^* = \langle c_{jkp_1}, c_{jkp_2}, \cdots, c_{jkp_m} \rangle$；根据式(3)计算 X_i^* 和 C_j^* 的距离 $d^{\mathrm{L}}(X_i^*, C_j^*)$。

(c) 根据式(6)更新权值矩阵 U，根据式(7)更新 $C_j \in C$。

(d) 计算 J_m。

(e) 直到 $|J_m - J_m^{\text{previous}}| \leqslant \varepsilon$，跳出循环。

2 动态聚类算法

在 FCM 算法得到的隶属度矩阵中，每条时间序列的最大隶属度 μ_{ij} 表示其最有可能属于的类，设定隶属度阈值 δ。若 $\mu_{ij} \geqslant \delta$（一般地，$\delta$ 取 0.7），说明时间序列 X_i 的整体结构很大程度上属于第 j 类；若 $\mu_{ij} < \delta$，说明其仅属于一个类的可能性较小，即此类时间序列在不同时间段可能属于不同的类，其形态特征跳跃转换，所以称此类时间序列为跳转序列（switching time series，STS）。形态特征跳跃转换的点包含了更多数据信息，此为关键点。上述算法得到了跳转序列和所有质心序列的关键点，于是用调整后的兰氏距离，如式(8)所示：

$$d^{\mathrm{L}}(\text{sub-STS}_i, \text{sub-}C_j) = \frac{1}{2}\left[\frac{|\text{STS}_{ikp_{q-1}} - c_{jkp_{q-1}}|}{|\text{STS}_{ikp_{q-1}}| + |c_{jkp_{q-1}}|} + \frac{|\text{STS}_{ikp_q} - c_{jkp_q}|}{|\text{STS}_{ikp_q}| + |c_{jkp_q}|}\right] \quad i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, k \tag{8}$$

逐段计算跳转子序列（sub-STS）与等长的质心子序列（sub-C）的距离，将跳转子序列逐段聚类到距离最近的类中。在式(8)中，$c_{jkp_{q-1}}$ 与 c_{jkp_q} 是质心子序列 sub-C_j 端点值，$\text{STS}_{ikp_{q-1}}$ 与 STS_{ikp_q} 是跳转子序列 sub-STS_i 的端点值。动态聚类算法步骤如下：

输入：隶属度矩阵 U，阈值 δ。

输出：聚类结果和跳转序列的逐段聚类结果。

a) 对给定的序列集合 X，首先利用 FCM 算法得到隶属度矩阵，然后选取矩阵中每一个时间序列的 μ_{ij}。若存在 $\mu_{ij} \geqslant \delta$，将此时间序列聚类到相应类中，并更新该类的质心序列；否则标记此时间序列为 STS。

b) 对所标记的每条 STS 和质心序列，提取其关键点集合。

c) 利用式(8)，对每条 STS 逐段计算其与 μ_{ij} 不为 0 的类的质心序列在相应段的兰氏距离，将此段归到最近的类中。

d) 输出聚类结果和跳转序列的逐段聚类结果。

3 实验结果及分析

为说明该算法的可行性和有效性，本文进行了仿真实验和实际数据测试。仿真数据集为 Eamonn Keogh 提供的 SonyAIBORobot Surface 和 CBF。这两个数据集均对其中的每个时间序列进行了类标号。其中 SonyAIBORobot Surface 分成两类，每

条时间序列长度为 70,共有 20 条时间序列;CBF 分为三类,数据长度为 128,共有 30 个序列。实际数据集为我国某省 17 个城市大气中 SO_2 含量的日值记录,周期为 180 d。

当已知数据对象的类标号时,可以通过预测类标号与实际类标号的对应程度来衡量聚类结果的好坏。本实验通过计算聚类结果的准确率和熵来评估聚类结果[5]。

定义 2　准确率。数据集中被正确聚类的对象个数与数据集对象总数的比值。

定义 3　熵。熵 $e=\sum_{i=1}^{k}\frac{m_i}{m}e_i$。其中,$k$ 是预测类的个数,m 为样本总数;$e_i=-\sum_{j=1}^{n}p_{ij}\log_2 p_{ij}$,$p_{ij}=m_{ij}/m_i$,其中 m_i 是预测类 i 中对象的个数,m_{ij} 是预测类 i 中实际类 j 的对象个数。

熵表示每个预测类由单个实际类的对象组成的程度,即预测类在多大程度上包含单个实际类的对象,是纯度的一种测量方法。预测类的总熵等于每个预测类的熵的加权和。熵越小说明预测类的纯度越高,聚类效果越好。由于本文采用了关键点技术,所以对数据进行了有效的压缩,压缩率计算公式如下。

定义 4　压缩率。压缩率 =(时间序列的时间点个数 - 压缩后关键点个数)/ 时间序列的时间点个数 ×100%

对数据集 SonyAIBORobot Surface 和 CBF 应用本文提出的算法进行聚类,将时间序列聚类到隶属度最大的类中,并与传统的 FCM 算法[5]、K-means 算法[5] 和谱聚类算法[16] 所得到的结果进行比较,发现本文提出的改进的 FCM 算法所得到的结果,其准确率都在 90% 以上,结果较为理想。表 1 和 2 分别给出了不同聚类算法对数据集 SonyAIBORobot Surface 和 CBF 的聚类结果。

表 1　不同聚类算法对数据集 SonyAIBORobot Surface 的性能比

聚类算法	隶属度指数	关键点余弦阈值	压缩率/%	准确率	熵
改进的 FCM	$m=2$	$a=-0.5$, $b=0.766$	96.4	0.9	0.449 9
		$a=-0.35$, $b=0.644$	91.55	0.95	0.207 1
	$m=2.5$	$a=-0.35$, $b=0.6$	91.55	0.95	0.207 1
传统 FCM	$m=2$	—	—	0.6	0.848 9
K-means	—	—	—	0.85	0.571 5
谱聚类算法	—	—	—	0.55	0.878 2

表 2　不同聚类算法对数据集 CBF 的性能比

聚类算法	隶属度指数	关键点余弦阈值	压缩率/%	准确率	熵
改进的 FCM	$m=2$	$a=-0.8$, $b=0.5$	96.9	1	0
		$a=-0.55$, $b=0.6$	94.57	0.9	0.396 3
		$a=-0.5$, $b=0.5$	95.35	0.9	0.429 6
传统 FCM	$m=2$	—	—	0.7	0.924 7
K-means	—	—	—	0.6	1.089 7
谱聚类算法	—	—	—	0.7	1.088

从表 1 中可以看出,利用本文所提出的算法对该数据集进行聚类的效果好于其他聚类算法,其原因是关键点的提取对时间序列进行了强压缩,除去噪声干扰的同时高效地提取了时间序列的有效信息,提高了聚类的精度,并使每个类的纯度也有了很大提高。

由表 2 可知,选取适当的参数,用改进的 FCM 算法对 CBF 数据集聚类,可以得到与实际情况完全一致的结果。在准确率相同的情况下,压缩率和熵可能不同。选取不同的阈值得到不同的关键点集,则压缩率不同。即使通过提取关键点被正确聚类的数据对象个数相同,但预测类标号有可能不同,所以类的纯度,即熵不同。

对数据集 SonyAIBORobot Surface 和 CBF,利用上述实验中各自准确率最高的结果进行动态聚类实验。以数据集 SonyAIBORobot Surface 为例,表 3 为其隶属度。

表 3　对 SonyAIBORobot Surface 采用改进的 FCM 算法得到的隶属度

数据对象	1	2	3	4	5	6	7	8	9	10
类 1 隶属度	0.967	0.496	0.961	0.959	0.283	0.779	0.762	0.579	0.853	0.393
类 2 隶属度	0.033	0.504	0.039	0.041	0.717	0.221	0.238	0.421	0.147	0.607
数据对象	11	12	13	14	15	16	17	18	19	20
类 1 隶属度	0.849	0.309	0.675	0.989	0.974	0.968	0.406	0.969	0.066	0.429
类 2 隶属度	0.151	0.691	0.325	0.011	0.026	0.032	0.594	0.031	0.934	0.571

表 3 中的隶属度表明每条时间序列属于各个类的可能性,设定最大隶属度阈值 $\delta=0.7$,最大隶属度大于 0.7 的时间序列聚类到相应的类中,则数据对象为 1、3、4、6、7、9、11、14、15、16、18 的时间序列聚类到类 1 中,数据对象为 5,19 的时间序列聚类到类 2 中,再计算每类的质心。最大隶属度小于 0.7 的数据对象为 2、8、10、12、13、17、20,这七条时间序列为跳转序列,它们的类标号会随着时间的演化而改变。以第 8 和 17 条时间序列为例进行动态聚类,结果如图 2 和 3 所示。

从图 2 和 3 中可以看到,第 8 和 17 条时间序列在不同时间段分别属于不同的类,充分体现了时间序列随时间演化的动态性质。在一定程度上也说明了,传统硬聚类方法在选取不同度量关系时会导致错误分类的部分原因。

对数据集 CBF 进行同样的动态聚类,得到一条跳转序列,其动态聚类结果如图 4 所示。

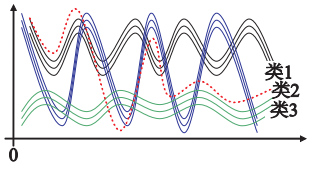

图1　动态特征明显的时间序列(虚线)形态

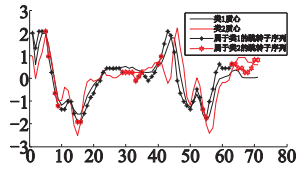
图2　第8条时间序列的动态聚类结果

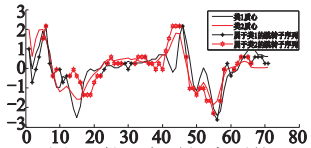
图3　第17条时间序列的动态聚类结果

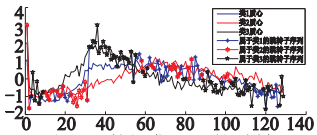
图4　数据集CBF的跳转序列动态聚类结果

实际数据集为我国某省 17 个城市大气中 SO_2 含量 180d 天的日值记录。从地域上分析,所选取的 17 个城市可分为三类。利用本文的算法对此数据集进行聚类,结果也是三类,与实际情况完全吻合,并发现两条跳转序列,它们的演化情况如图 5 和 6 所示。综合其他气象资料和实际的产业结构分析,城市 A 几乎没有造成污染的企业,所以整体上城市空气中的 SO_2 含量几乎为 0,但其相邻城市有一些化工企业,由于风向原因,在部分时间城市 A 受到了很少的污染。城市 B 有一些造成轻度污染的企业,在大部分时间里,空气中有微量的 SO_2,在多风的时间段内,该城市的空气质量好于无风或者小风的时间段。

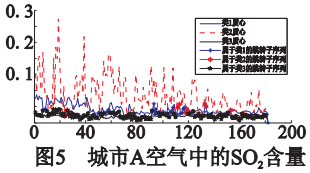
图5　城市A空气中的SO_2含量

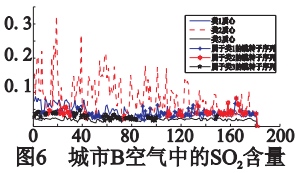
图6　城市B空气中的SO_2含量

4　结束语

时间序列是随时间演化的动态数据,为了充分掌握数据内

部隐含的信息,需要将数据的演化性质反映出来。因为时间序列具有规模大、维数高且大量噪声干扰的特点,所以对其进行降维尤为必要。本文提出的关键点提取方法,既有效达到了降维的目的,同时保持了时间序列的主要特征。此外,合并关键点下标的方法为解决不等长时间序列的聚类问题提供了新思路。本文提出的对时间序列进行动态聚类的算法,是对模糊聚类方法的延伸,揭示了时间序列类别随时间演化的性质,为进一步研究时间序列提供了有力的工具。实验结果表明此方法对时间序列的描述更加精确、具体,更有利于把握时间序列的动态特征。该方法还可以用于时间序列的趋势预测。

参考文献:

[1] FU T. A review on time series data mining[J]. Engineering Applications of Artificial Intelligence,2011,24(1):164-181.

[2] 杨一鸣,潘嵘,潘嘉林,等. 时间序列分类问题的算法比较[J]. 计算机学报,2007,30(8):1259-1266.

[3] 孙吉贵,刘杰,赵连宇. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.

[4] 潘定,沈钧毅. 时态数据挖掘的相似性发现技术[J]. 软件学报,2007,18(2):246-258.

[5] TAN P, STEINBACH M, KUNMAR V. Introduction to data mining [M]. 范明,范宏建,等译. 北京:人民邮电出版社,2006:43-344.

[6] D'URSO P, MAHARAJ E A. Autocorrelation-based fuzzy clustering of time series[J]. Fuzzy Sets and Systems, 2009, 160(24): 3565-3589.

[7] MAHARAJ E A, D'URSO P. Fuzzy clustering of time series in the frequency domain[J]. Information Sciences,2011,181(7):1187-1211.

[8] MAHARAJ E A, D'URSO P. A coherence-based approach for the pattern recognition of time series[J]. Physica A,2010,389(17): 3516-3537.

[9] IOANNOU A, FOKIANOS K J, PROMPONAS V. Spectral density ratio based on clustering methods for the binary segmentation of protein sequences: a comparative study[J]. Biosystems, 2011, 100(2): 132-143.

[10] MAHARAJ E A, D'URSO P, GALGEDERA D U A. Wavelets-based fuzzy clustering of time series[J]. Journal of Classification, 2010,27(2): 231-275.

[11] HSU K, LI Sheng-tun. Clustering spatial-temporal precipitation data using wavelet transform and self-organizing map neural network[J]. Advances in Water Resources,2010,33(2):190-200.

[12] CORDUAS M. Clustering streamflow time series for regional classification[J]. Journal of Hydrology,2011,407(1-4): 73-80.

[13] VILAR J A, ALONSO A M, VILAR J M. Non-linear time series clustering based on non-parametric forecast densities[J]. Computational Statistics and Data Analysis,2010,54(11):2850-2865.

[14] 高惠璇. 应用多元统计分析[M]. 北京:北京大学出版社,2005: 221-222.

[15] DOVZAN D, SKRJANC I. Recursive fuzzy C-means clustering for recursive fuzzy identification of time-varying processes[J]. ISA Trans, 2011,50(2): 159-169.

[16] 赵凤,焦李成,刘汉强,等. 半监督谱聚类特征向量选择算法[J]. 模式识别与人工智能,2011,24(1):48-56.