DOI:10.16451/j.cnki.issn1003-6059.2009.05.014
第22卷 第5期 2009年10月

Vol.22 No.5 Oct 2009

基于EMD与K-means算法的时间序列聚类*

刘慧婷[1] 倪志伟[2]

[1](安徽大学 计算机科学与技术学院 合肥 230039)
[2](合肥工业大学 计算机网络系统研究所 合肥 230009)

摘 要 有效实现时间序列聚类的重要前提是序列的维数得到约简,序列中包含的噪声能够被滤除.文中提出一种能够对时间序列进行有效预处理的方法.该方法先通过经验模态分解实现时间序列趋势的提取,再利用自底向上算法对趋势序列进行分段,最后转换成由{-1,0,1}构成的齐序列.为了证明该方法既能实现降维,也可实现数据序列中噪声的滤除,文中利用K-means算法对经过上述方法预处理后的序列进行聚类.实验结果表明,与直接对原序列进行聚类相比,对预处理后的数据序列进行聚类,空间复杂度较低、准确性较高.

关键词 时间序列,分段序列,降维,经验模态分解,K-means算法
中图法分类号 TP391

Clustering Method of Time Series Based on EMD and K-means Algorithm

LIU Hui-Ting[1], NI Zhi-Wei[2]
[1](School of Computer Science and Technology, Anhui University, Hefei 230039)
[2](Institute of Computer Network System, Hefei University of Technology, Hefei 230009)

ABSTRACT

Dimension reduction of time series and noise in sequences filtering are important prerequisites for effective realization of time series clustering. A method is proposed to preprocess time series effectively. Firstly, the trend of a time sequence is got by using empirical mode decomposition method. Then, the trend series are divided into several segments by bottom-up algorithm. Finally, the piecewise series are translated into uniform sequences, and each of them is composed of -1, 0 and 1. To prove that the proposed method can achieve dimensionality reduction and filter out the noise from the data sequence, K-means algorithm is utilized to finish clustering of pretreated time series. Experimental results show clustering of pretreated data sequences is better than that of the original series.

Key Words Time Series, Piecewise Series, Dimension Reduction, Empirical Mode Decomposition, K-means Algorithm

*国家863计划资助项目(No.2007AA04Z116)、国家自然科学基金项目(No.70871033)和安徽高校省级自然科学研究项目(KJ2007B303ZC)资助
收稿日期:2008-10-10,修回日期:2009-07-15
作者简介 刘慧婷,女,1978年生,博士,副教授,主要研究方向为算法分析与设计、机器学习.E-mail:wangph168@yahoo.com.cn 倪志伟,男,1963年生,教授,博士生导师,主要研究方向为机器学习、数据挖掘和智能决策支持系统.

1 引 言

时间序列是指按时间顺序排列的一种数据.作为数据库中的一种数据形式,它广泛存在于各种大型的商业、医学、工程和社会科学等数据库中.随着数据库知识发现和模式识别等计算机技术的发展,出现基于大规模甚至海量数据库的数据挖掘技术,其研究目的是从大量时间序列数据中发现未知的重要模式和知识,并据此作出具有知识驱动的决策.

聚类问题是时间序列模式发现的一个重要问题[1],聚类的一个重要方面是数据的降维.由于时间序列的数据量很大,在进行聚类之前需要对数据维度进行约简.文献[2]~[4]中提出基于傅立叶变换的约简算法.但傅立叶变换只是一种纯频域的分析方法,反映的是信号在全部时间上的整体频率特征,不能提供任何局部时间上的频率特征.加窗傅立叶变换将一个时间窗口函数和待分析函数点乘,再进行傅立叶变换,结果可以描述某一局部时间段上的信息.但对一个时变的非稳态信号,很难找到一个合适的时间窗口来适合不同的时间段[5].经验模态分解方法(Empirical Mode Decomposition, EMD)可有效处理非线性非稳态信号[6],文中将利用 EMD完成数据维度的约简.

为了提高聚类的准确性,本文对时间序列进行K-means聚类前,先利用提出的基于 EMD的时间序列降维方法进行预处理.即先通过 EMD提取出序列的趋势特征.再利用自底向上算法进行趋势序列的分段.最后把分段序列转换成由{-1,0,1}构成的齐序列,从而完成序列数据维度的约简.实验证明对序列进行预处理后再进行聚类,可以降低 K-means算法的空间复杂度,提高聚类的准确性.在把分段序列转换成齐序列这一过程中,本文也提出相关的转换算法.

2 相关知识概述

2.1 经验模态分解方法理论

EMD方法是 Huang等人[7]提出的一种信号处理方法.从本质上讲,该方法是对一个信号进行平稳化处理,产生一系列本征模函数(Intrinsic Mode Function, IMF),最低频率的 IMF分量通常情况下作为原始信号的趋势.如果原序列表示为 $x(t)$,IMF分量用 $c_i(t)$表示,趋势序列用 $r_n(t)$表示,分解结果表示为

$$x(t)=\sum_{i=1}^{n}c_i(t)+r_n(t).$$

EMD方法对原始序列进行分解,提取出的趋势和原始序列的关系如图 1所示.图 1可以看出,EMD方法对原始序列中的噪声进行过滤,产生的趋势序列准确反映原序列的趋势走向,序列变得更加清晰,而信息量丢失相对较少.

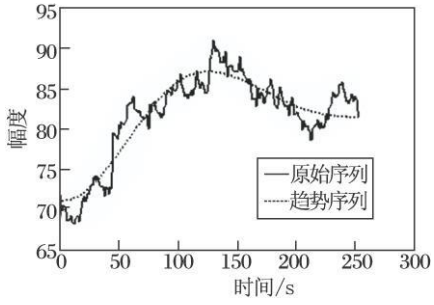


图 1 原始序列及其趋势

Fig. 1 Original sequence and its trend series

2.2 自底向上和 K-means算法

自底向上算法的基本思想是,首先将 N个待分段的时间序列数据点两两连接,划分成不重合的 $N/2$个初始分段并计算合并相邻段的拟合代价,即拟合误差.然后,循环地从中选择拟合代价最小的,如果该最小值小于用户设定的分段阈值,则合并对应的两个相邻段,并计算新合并的分段与它前后的分段的拟合代价.重复该过程,直到所有的拟合代价均不小于分段阈值,分段结束[8-9].

K-means聚类是目前应用最为广泛的聚类算法之一,它具有算法简单且收敛速度快的特点[10].K-means算法的基本思想是,对于给定的聚类数目 k首先随机创建一个初始划分,即随机选择某些数据代表点作为初始聚类中心,根据其余数据点到各聚类中心的距离将其分到各个类中.然后重新确定新的聚类中心,以此类推采用迭代方法将聚类中心不断移动来尝试进一步改进划分,直到聚类中心不再发生变化[11].不同于层次聚类方法,K-means算法并不建立一个树型结构来描述数据集的分组,而是创建一个单一层次的集群.另一个不同是,K-means聚类用数据集合元素的实际值,而不仅仅是它们的近似值来进行聚类.这些差异意味着 K-means更适合于大规模数据集合的聚类.

3　基于 EMD和 K-means算法的时间序列聚类方法

3.1　时间序列维度的约简

本节首先对时间序列进行 EMD分解, 然后再进行分段表示, 达到维度约简的目的. 趋势序列利用自底向上算法分段后要转换成齐序列, 齐序列的定义如下.

定义 1　假设用 [(lx, ly), (rx, ry)] 表示分得的每一段的最左端数据点的横、纵坐标, 最右端数据点的横、纵坐标, 分段序列 S_1 和 S_2 是齐序列当且仅当分段数相等, 且满足 $lx_{1i} = lx_{2i}$, $rx_{1i} = rx_{2i}$, $1 \leqslant i \leqslant N$, 其中 N是序列分段后分得的段数.

$$S_1 = \{[(lx_{11}, ly_{11}), (rx_{11}, ry_{11})], \cdots, [(lx_{1i}, ly_{1i}), (rx_{1i}, ry_{1i})], \cdots, [(lx_{1N}, ly_{1N}), (rx_{1N}, ry_{1N})]\},$$

$$S_2 = \{[(lx_{21}, ly_{21}), (rx_{21}, ry_{21})], \cdots, [(lx_{2i}, ly_{2i}), (rx_{2i}, ry_{2i})], \cdots, [(lx_{2N}, ly_{2N}), (rx_{2N}, ry_{2N})]\}.$$

1)趋势序列的提取. 在本文方法中, 趋势序列是通过 EMD方法提取的. 假如一个时间序列的极值点数目与零点数目相差 2个 (或 2个以上), 或者上、下包络线的均值不是处处为零, 则该时间序列就需要利用 "筛 " 过程进行平稳化处理[12]. 直到分解得到的剩余部分的极值点个数小于预定值, "筛 " 过程才停止. 为了准确提取出序列的趋势, 预定值是需要根据具体的时间序列确定的. 对于一般的时间序列, 如果剩余部分的极值点个数小于或等于两个, "筛 " 过程停止. 对于波形变化较大的序列, 则如果剩余部分的极值点个数小于或等于原序列极值点个数的十分之一, "筛 " 过程停止. 在实际应用中, 如果提取出的趋势序列不能反映出原序列的走势, 则需要适当调节预定值的大小. 若时间序列表示为 $x(t)$, IMF分量用 $c_i(t)$ 表示, 趋势序列用 $r_n(t)$ 表示, 时间序列的趋势可以表示为

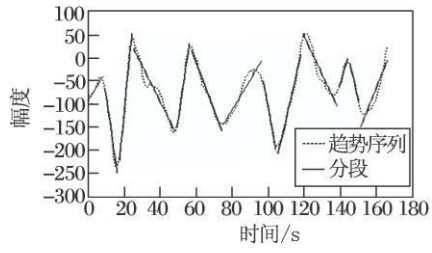
$$r_n(t) = x(t) - \sum_{i=1}^{n} c_i(t).$$

从图 1可以看出, EMD方法对原始序列中的噪声进行过滤, 产生的趋势序列准确反映原序列的趋势走向.

2)利用 Keogh提出的自底向上分段算法[8] 对趋势序列进行分段.

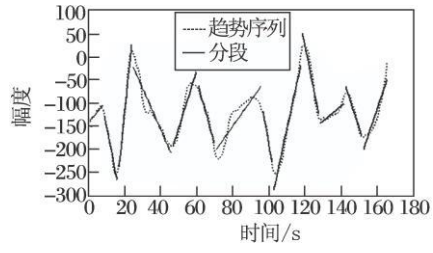
3)实际的趋势序列分段后, 基本不存在齐序列, 如图 2所示. 因此要把分段后的序列转换成齐序列.

从图 2可以看出, 虽然这两个趋势序列的分段序列都是 13段, 但并不是对应的每一段的起始点和终止点都相同, 所以它们两个不是齐序列.



(a)趋势序列 1及其分段序列

(a) Trend sequence$_1$ and its piecewise series



(b)趋势序列 2及其分段序列

(b) Trend sequence$_2$ and its piecewise series

图 2　对 2个趋势序列进行分段所得结果

Fig. 2　Piecewise series for 2 trend sequences

为了把分段序列转换成齐序列, 进行以下步骤.

(1)找出所有分段序列的每一段的最右端数据点的横坐标, 并进行排序. 图 2(a)中的分段序列的每一段的最右端数据点的横坐标分别是

(8, 16, 24, 48, 56, 74, 96, 104, 118, 138, 144, 150, 166).

(b)中的分别是

(8, 16, 24, 46, 60, 70, 96, 102, 118, 128, 142, 152, 166).

把它们进行排序并删除重复数据点后, 表示为

$$P = \{8, 16, 24, 46, 48, 56, 60, 70, 74, 96, 102, 104, 118, 128, 138, 142, 144, 150, 152, 166\}.$$

(2)对于分段序列的每一段, 如果 P中小于或等于该段的最右端数据点的横坐标 rx, 大于该段的最左端数据点的横坐标 lx的元素的个数为 m, 则该段被分为 m个子段.

对于图 2(a)中的分段序列, P中小于或等于它的第一段的 rx并且大于该段的 lx的元素只有一个 8, 所以第一段不需要被划分成子段. P中小于或等于它的第四段的 rx= 48, 并且大于该段的 lx= 25的元素有两个 46, 48. 所以第四段被分成两个子段, 它们的横坐标区间是 [25, 46], [47, 48].

按照上述转换算法, 图 2中的两个分段序列分成 20段, 分段数增加, 每一段的横坐标区间相同, 成

为齐序列.可以对齐序列方便地进行各种操作.

4)把齐序列转换成{-1,0,1}构成的序列.假设齐序列包含 m段,转换后的序列为 STR=(h1,h2,…,hm),其中当 ryi-lyi<0时,hi=-1;ryi-lyi=0,hi=0;当 ryi-lyi>0时,hi=1,i=1,2,…,m,ryi表示线段 i最右端采样点的纵坐标值,lyi表示线段 i最左端采样点的纵坐标值.

图 2中的两个分段序列转换成齐序列后,再通过本步骤的转换,分别表示为

(1,-1,1,-1,-1,1,-1,-1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1),

(1,-1,1,-1,1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,-1,-1,1).

两个趋势序列的长度是 166,通过(2)的分段后均被分成 13段,再通过齐序列的转换成为 20段,最后变成长度为 20的{-1,0,1}构成的序列,实现数据维度的约简.

3.2 降维序列的聚类

度量两个时间序列的相似性有很多方法,文中把两个时间序列经过 3.1节转换后得到的序列的内积,作为判定这两个序列相似的标准.相似性定义如下.

定义 2 对于两个长度均为 n的时间序列 X,Y,它们经过降维转换后分别得到由{-1,0,1}构成的齐序列 $\overline{X},\overline{Y}$,给定相似度阈值 d,如果 $\overline{X},\overline{Y}$的内积满足 $DIS(\overline{X},\overline{Y})=\langle\overline{X},\overline{Y}\rangle>d$,则称序列 X,Y趋势相似.

下面给出 K-means算法的聚类过程.

给定数据集合 $D=\{d_1,d_2,\cdots,d_n\}$.

step1 随机从集合 D中选择 K个对象做为初始簇中心.

step2 将 D中所有的点分配到最近的簇.

step3 重新计算每个簇的中心.

step4 重复 step2,step3直至簇中心不再发生变化或迭代次数超过设定的最大迭代次数.

本文用平均查准率和平均查全率来衡量聚类结果的准确性.查准率 P是指聚类判定的属于类别 C的所有序列中,确实属于类别 C的序列所占的比例.查全率 P是指原本属于类别 C的所有序列中,聚类做出同样判定的序列所占的比例.例如对表 1中所显示的聚类结果来说,其中,类别 C1,C2是广义上的两个类别,用同样的思想可以推广到多类情况,可得

$$P_r=\frac{a}{a+b},\quad P_e=\frac{a}{a+c}.$$

考虑到测试结果的比较问题,所以我们用平均查准率和平均查全率作为本文的评价标准.具体计算公式如下:

$$MacroPr=\frac{1}{n}\sum_{j=1}^{n}Pr_j,\quad MacroPe=\frac{1}{n}\sum_{j=1}^{n}Pe_j$$

其中,n是类别总数,Pr_j为第 j类的查准率,Pe_j为第 j类的查全率.

表 1 聚类结果定义

Table 1 Definition of clustering results

	原本 C1类中的序列个数	原本 C2类中的序列个数
聚类结果中 C1类的数据序列个数	a	b
聚类结果中 C2类的数据序列个数	c	d

4 实 验

实验 1 实验数据来自 http://www.ics.uci.edu/~mlearn/MLRepository.html,本实验用了 Clean1和 Isole5数据库.Clean1数据库包含 476个数据序列,每个数据序列有 166个属性,所有的数据序列被分成两类,第一类包含 207个序列,第二类包含 269个序列.Isole5数据库包含 1 559个数据序列,每个数据序列有 617个属性,所有的数据被分成 26类,本实验选择第 13类、14类中的数据,13类包含 59个序列,14类包含 60个序列.

为了验证本文方法不仅能实现时间序列维度的约简,而且可提高聚类的准确性.实验对数据进行以下处理:1)方法 1,对数据直接进行 K-means聚类.2)方法 2,对数据序列利用自底向上分段算法进行分段,转换成由{-1,0,1}构成的齐序列后,进行 K-means聚类.3)方法 3,利用 EMD方法提取数据序列的趋势,利用自底向上分段算法对趋势序列进行分段,转换成由{-1,0,1}构成的齐序列后,进行 K-means聚类.然后比较这 3种方法的准确性和空间复杂度.比较结果如表 2所示.

从表 2看出,由于方法 1直接对序列进行 K-means聚类,所以参加聚类的每个序列的长度最长,其中 Clean1数据库为 166,Isole5数据库为 617.方法 2先进行降维再聚类,参加聚类的每个序列的长度居中,Clean1数据库为 83,Isole5数据库为 287.方法 3先利用本文方法降维再聚类,参加聚类的每个序列的长度最短,其中 Clean1数据库为 76,Isole5数据库为 279.由于 3种方法中的聚类都是基于 K-means算法的,K-means算法的空间复杂度为 O((n+k)d),其中,n是样本数,k是聚类数,d是样

本特征数. 方法 2和方法 3经过降维, 其特征较少, 即 d值较小, 所以空间复杂度较低, 方法 3的空间复杂度最低.

表 2　3种处理方法的结果比较

Table 2　Result comparison among 3 methods

		Cleanl			Isole5		
		方法 1	方法 2	方法 3	方法 1	方法 2	方法 3
第一类	聚类后第一类包含的元素个数	299	323	297	65	64	54
	真正属于第一类的元素个数	138	148	154	24	38	36
	由第二类误分来的元素个数	161	175	143	41	26	18
第二类	聚类后第二类包含的元素个数	177	153	179	54	55	65
	真正属于第二类的元素个数	108	94	126	19	34	42
	由第一类误分来的元素个数	69	59	53	35	21	23
	参加 K-means聚类的序列的长度	166	83	76	617	287	279

把 Isole5数据库与 Cleanl数据库相比较, Isole5数据库具有维数较高的特点. 两个数据库中的数据序列经过方法 3中的降维方法预处理后, 维数都得到约简, Isole5数据库的降维效果更加明显. 数据序列降维后的长度只有 279, 远小于原来的 617, 较大程度节省存储空间, 也降低下一步 K-means聚类的空间复杂度. 所以本文方法更擅长于维数较高的数据的降维.

由表 2中的结果, 计算出 3种方法的平均查准率和查全率, 如表 3所示.

从表 3可以看出, 无论是平均查准率还是平均查全率, 方法 3是最理想的. 这是因为方法 1直接对序列进行 K-means聚类, 而 K-means方法是利用序列之间的距离作为聚类的依据, 对孤立点和噪声数据很敏感. 有时会出现真正属于一类的两个序列由于某几维相差较大而没有被聚为同一类.

表 3　3种处理方法的平均查准率和查全率

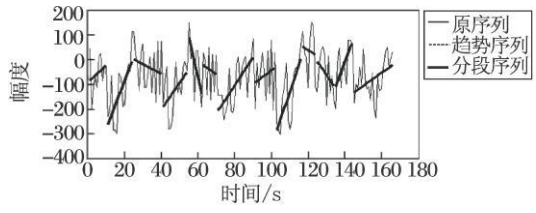
Table 3　Average precision and average recall of 3 methods

	Cleanl			Isole5		
	方法 1	方法 2	方法 3	方法 1	方法 2	方法 3
平均查准率	0.5359	0.5363	0.6112	0.3605	0.6060	0.6564
平均查全率	0.5341	0.5322	0.6062	0.3617	0.6054	0.6551

方法 2对序列进行分段, 然后把分段序列转换成由 {-1, 0, 1} 构成的齐序列, 再进行聚类, 可较好克服真正属于一类的两个序列由于某几维相差较大而没有被聚为同一类的现象, 可是直接对序列进行分段丢失数据序列中的有用信息.

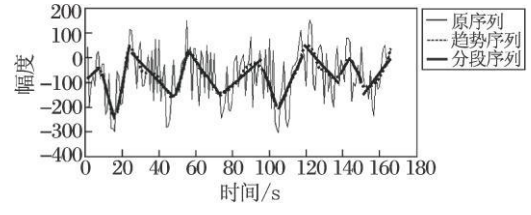
方法 3先对时间序列进行 EMD分解, 提取它们的趋势, 再对趋势序列进行分段、转换、聚类. 从图 1可以看出, EMD方法提取出的趋势序列准确反映原序列的趋势走向, 序列变得更加清晰, 而信息量丢失相对较少. 所以在趋势序列的基础上进行分段、转换、聚类, 一方面有效实现降维, 另一方法, 也提高对序列进行聚类时的查准率和查全率. 利用方法 3得到的由 {-1, 0, 1} 构成的齐序列可代替原序列用于时间序列聚类中.

实验 2　实验 1中, 方法 2和方法 3在聚类前, 都对序列进行降维处理, 它们的区别是方法 2对序列直



(a)方法 2

(a) Method 2



(b)方法 3

(b) Method 3

图 3　利用方法 2方法 3得到的同一个序列的分段序列

Fig. 3　Piecewise sequences of same series by method 2 and 3

接利用自底向上算法进行分段,方法 3首先对时间序列进行 EMD分解,然后再进行分段.方法 2得到的分段序列如图 3(a)所示,方法 3得到的分段序列如(b)所示.

通过比较图 3(a)、(b),可以看出(b)中的分段序列更加符合原序列的趋势,而分段是进一步转换成齐序列,完成降维的基础.所以与方法 2相比,方法 3中的降维方法提高降维的准确性.

从表 2中两个数据库的数据序列分别经过方法 2、方法 3预处理后所具有的长度,及预处理后进行聚类所得的结果,也可以看出本文方法比方法 2中的降维方法更为有效.

5 结束语

在基于经验模态分解和 K-means的时间序列聚类算法中,首先对需要进行聚类的时间序列进行维度的约简.再利用 K-means方法进行聚类.

K-means对孤立点和噪声数据很敏感,有时会出现真正属于一类的两个序列由于某几维相差较大而没有被聚为同一类.本文方法对时间序列进行经验模式分解、分段、以及转换等处理,这些处理会导致部分信息的丢失,但更多的是噪声信号和孤立点的滤除,可以在一定程度上解决两个序列由于某几维相差较大不能聚为同一类的问题.实验结果也证明该方法一方面有效实现降维,另一方法,由于对原始序列中的噪声进行过滤,提高对序列进行聚类时的查准率和查全率.进一步的工作包括提高 EMD提取信号趋势运算模块的准确度,利用有效算法处理时间序列长度不一致的情况.

参 考 文 献

[1] Last M, Klein Y, Kandel A. Knowledge Discovery in Time Series Databases. IEEE Trans on Systems, Man and Cybernetics, 2001, 31(1): 160-169

[2] Moon Y S, Kim J. Efficient Moving Average Transform-Based Subsequence Matching Algorithms in Time-Series Databases. Information Sciences: An International Journal, 2007, 177(23): 5415-5431

[3] Kim S W, Park D H, Lee H G. Efficient Processing of Subsequence Matching with the Euclidean Metric in Time-Series Databases. Information Processing Letters, 2004, 90(5): 253-260

[4] Kontaki M, Papadopoulos A N, Manolopoulos Y. Adaptive Similarity Search in Streaming Time Series with Sliding Windows. Data & Knowledge Engineering, 2007, 63(2): 478-502

[5] Zhang Haijin, Cai Qingsheng. Time Series Similar Pattern Matching Based on Wavelet Transform. Chinese Journal of Computers, 2003, 26(3): 1-5. (in Chinese)
(张海勤,蔡庆生.基于小波变换的时间序列相似模式匹配.计算机学报, 2003, 26(3): 1-5)

[6] Zhang Min, Zhang Yanping, Cheng Jiaxing. Hierarchical Algorithm to Match Similar Time Series Pattern. Journal of Computer-Aided Design & Computer Graphics, 2005, 17(7): 1480-1485. (in Chinese)
(张 旻,张燕平,程家兴.时间序列相似模式的分层匹配.计算机辅助设计与图形学学报, 2005, 17(7): 1480-1485)

[7] Peng Z K, Tse P W, Chu F L. An Improved Hilbert-Huang Transform and Its Application in Vibration Signal Analysis. Journal of Sound and Vibration, 2005, 286(1/2): 187-205

[8] Keogh E, Chu S, Hart D, et al. An On-line Algorithm for Segmenting Time Series // Proc of the IEEE International Conference on Data Mining. San Jose, USA, 2001: 289-296

[9] Keogh E, Kasetty S. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration // Proc of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002: 102-111

[10] Ordonez C, Omiecinski E. Efficient Disk-Based K-means Clustering for Relational Databases. IEEE Trans on Knowledge and Data Engineering, 2004, 16(8): 909-921

[11] Zhang Jianpei, Yang Yue, Yang Jing, et al. Algorithm for Initialization of K-means Clustering Center Based on Optimized-Division. Journal of System Simulation, 2009, 21(9): 2586-2590. (in Chinese)
(张健沛,杨 悦,杨 静,等.基于最优划分的 K-means初始聚类中心选取算法.系统仿真学报, 2009, 21(9): 2586-2590)

[12] Zhang Yan, Sun Zhengxing, Li Wenhui. Texture Synthesis Based on Direction Empirical Mode Decomposition. Computers & Graphics, 2008, 32(2): 175-186