

周期性时间序列数据聚类算法的改进研究

郭秀珍¹, 陆建峰¹, 汤九斌²

(1 南京理工大学 计算机科学与技术学院, 江苏 南京 210094;

2 中国电信江苏公司, 江苏 南京 210037)

摘 要: 聚类分析是时间数据序列分析的一种常用手段, 现有的聚类算法通常从相似性度量方面进行改进. 实际的时间序列数据往往具有一定的周期性和连续性, 现有的算法往往忽略时间序列数据周期性和连续性特点对聚类算法的影响. 对此问题进行了研究, 尝试采用延拓的方法来解决该问题, 从而改善聚类效果. 初步的实验结果表明了该方法的可行性和有效性.

关键词: 时间序列; 聚类; 周期性; k 均值; 延拓

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2011)10-0118-04

Research on Improvement of Clustering Algorithm for Periodic Time-series Data

GUO Xiu-zhen¹, LU Jian-feng¹, TANG Jiu-bin²

(1 School of Computer Science and Technology Nanjing University of Science and Technology, Nanjing 210094, China;

2 China Telecom Jiangsu Corp., Nanjing 210037, China)

Abstract: Clustering analysis is a common means of time-series data analysis. The existing clustering algorithm for the time-series data is improved from the similarity measure aspects, but the actual time-series data tend to have a certain periodicity and continuity. The existing algorithms often ignore the impact of periodicity and continuity on time-series data clustering algorithms. This issue is researched in this paper and extension method is used to solve this problem. Preliminary results show the feasibility and effectiveness.

Key words: time-series; clustering; periodicity; k -means; extension

1 引言

从统计上来讲, 由不同时间重复测量得到的值或时间的序列组成的数据就是时间序列数据. 时间序列挖掘是从大量的时间序列数据中提取人们事先不知道的, 但又是潜在有用的与时间属性相关的信息和知识^[1].

时间序列聚类是将一组时间序列集合根据其相似性分类为几组, 组与组之间的相似性尽可能小, 组内各序列之间的相似性尽可能大. 时间序列聚类的重要性引起了大家的重视, 近年来也得到广泛的研究. 翁颖钧^[2]、段江娇^[3]等人将分层聚类应用于时序

聚类; 黄超等^[4]研究了基于多重分形的时间序列聚类. K 均值聚类也被用于时序聚类^[5], 邵丹^[6]等人应用模糊 K 均值对电力负荷时序进行聚类.

在对周期性时间序列数据的聚类中, 常用的方法是将时间点上的数据先进行统计累加再进行聚类, 这样就丢失了时间轴在聚类过程中的影响. 如对于每小时的上网人数进行统计, 第一天的 24 时和第二天凌晨的 1 时从时间轴来说是非常接近的, 但是, 如果用时间轴 24 小时制来表示进行统计的话, 24 时和 1 时的时间距离为 1 小时, 但是在聚类中, 这两点的距离通常会被计算为 23 小时, 因此其距离值被不合理地扩大了, 从而影响了聚类的效果. 为解决这

收稿日期: 2011-04-30; 修回日期: 2011-06-28

基金项目: 江苏省自然基金项目(BK2009489); 江苏省青蓝工程

个问题,本文以 k 均值聚类方法为例,应用延拓的方法对时间轴进行延拓,进而在延拓的时间轴上,建立现有聚类中心的映像,即把现有的 k 个聚类中心扩充为 $2k$ 个聚类中心,以解决时间连续性和周期性所带来的距离计算问题,从而改善聚类结果.

2 k -均值算法

假定聚类的对象共 n 个,样本集为 $X = \{x_1, x_2, \dots, x_n\}$, 算法目的是把 n 个样本对象分为 k 个类. 具体的步骤如下:

① 输入原始的数据,为每个聚类选取一个初始聚合中心 $Z_j(I), j = 1, 2, \dots, k$;

② 计算每个样本与聚类中心的距离 $D(x_i, Z_j(I)), i = 1, 2, \dots, n; j = 1, 2, \dots, k$. 若 $D(x_i, Z_j(I)) = \min_{j=1,2,\dots,k} \{D(x_i, Z_j(I)), i = 1, 2, \dots, n\}$, 则 $x_i \in \omega_c$;

③ 使用每个聚类中所有样本的均值作为 k 个新的聚类中心:

$$Z_j(I+1) = \frac{1}{n_j} \sum_{k=1}^{n_j} x_k^j, j = 1, 2, \dots, k;$$

④ 判断:若 $Z_j(I+1) \neq Z_j(I), j = 1, 2, \dots, k$, 则 $I = I + 1$, 返回 ②, 否则算法结束;

⑤ 最后得到的 k 个聚类中心即聚类的结果.

3 延拓处理

延拓处理:将一个函数的定义域扩大的过程称为延拓. 在聚类初始时,采用延拓的方法将时间轴的范围由 $[0, 24)$ 延拓成 $[-12, 36)$. 之所以采用延拓的方法,其最主要的目的在于能够保持数据的连续性,而不像现在这样,在进行统计的时候,人为地把本来连续的数据强制地进行了截断,从而影响聚类的效果.

如图 1 所示,将时间范围由 $[0, 24)$ 延拓为 $[-12, 36)$, $[-12, 0)$ 这个区间对应于 24 小时中的 $[12, 24)$, 可以理解为是前一天的 $[12, 24)$, $[24, 36)$ 对应于 24 小时中的 $[0, 12)$, 可以理解为是后一天的 $[0, 12)$, 具体见式(1). 之所以这么做,是基于如下原因:从物理意义而言,以中午 12 点为分界线,每天的 24 点可以分为 $[0, 12)$ 和 $[12, 24)$ 两个区间, $[0, 12)$ 区间和前一天的 $[12, 24)$ 在物理上相连的, $[12, 24)$ 区间和后一天的 $[0, 12)$ 区间在物理是相连的,这样在进行聚类过程的距离计算的时候,就可以避免强制截断的情况了.

具体延拓的公式如下:

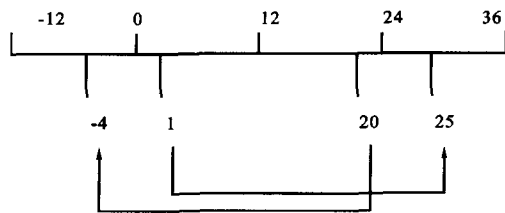


图 1 聚类中心的延拓示意图

$$y = \begin{cases} x + 24 & x \in [0, 12) \\ x - 24 & x \in [12, 24) \end{cases} \quad (1)$$

式中, x 代表延拓之前的值, y 代表延拓后的值, 此外, 我们仅对聚类中心采用上述延拓.

下面通过一个例子来具体说明.

在聚类过程中,假设有两类,其聚类中心分别 A 和 B (其类别我们也称为 A 和 B), 如果某个数据点 P 到 A 的距离小于 B , 那么 P 应该属于类 A . 进一步假设, A, B 的中心分别为 24 小时中的 1 点和 20 点, P 点为 23 点, 那么, 从实际物理意义而言, P 点到 A 点的距离应该小于 B 点的距离, P 点应该属于类 A . 但是, 一旦按照 24 小时进行统计处理后, P 点到 B 点的距离则小于 A 点的距离, P 点则属于类 B . 很显然, 这是不合理的. 但是, 如果采用了本文所提出的延拓方法以后, 上述问题就可以得到解决. 采用延拓之后, 我们可以把类 A 的中心进行延拓, 延拓后得到的中心就变为 25, 我们称之为 A' . 这时 P 到 A' 的距离小于 P 到 B 的距离, 因此就可以把 P 归为类 A' (即类 A), 很显然这样的结果是比较合理的.

因此, 把聚类中心按照上述方法进行延拓, 这样就可以充分利用时间数据的连续性特点, 从而可以提高周期数据聚类结果的准确性.

4 基于延拓的 k -均值算法

4.1 算法说明

通过上一节的描述, 可以知道, 本文的方法在聚类中心的更新方面与原来的方法也将有所差异.

仍以上一节的例子为例, 在上一节中, 我们知道最终的点 P 应该是属于类 A' (即类 A) 的, 但是, 如果直接利用点 P 的数据对类 A 中心进行更新, 显然是错误的. 因此, 在进行聚类中心更新的时候, 需要判断原先聚类中心点的范围, 如果聚类中心位于 $[24, 36)$ 区间, 则表明该聚类中心是经过延拓后得到的中心, 其真正的中心应该位于 $[0, 12)$, 因此把点 P 的坐标减去 24, 再用于聚类中心的更新; 如果聚类中心位于 $[-12, 0)$ 区间, 则表明该聚类中心是经过延拓后得到的中心, 其真正的中心应该位于

[12, 24), 因此把点 P 的坐标加上 24, 再用于聚类中心的更新。

进行聚类中心点更新的公式:

$$ppx = \begin{cases} px + 24 & \text{if } p \in c_i \text{ and } c_i \in [-12, 0) \\ px & \text{if } p \in c_i \text{ and } c_i \in [0, 24) \\ px - 24 & \text{if } p \in c_i \text{ and } c_i \in [24, 36) \end{cases} \quad (2)$$

式中, px 表示点 P 的坐标, c_i 表示某个类别, ppx 表示最终实际采用的 P 的坐标值。

和原算法相比, 改进的算法在以下几个方面有明显差异:

(1) 首先, 为了避免强制截断的情况, 把初始 k 个聚类中心按照式(1) 拓成 $2k$ 个。假设初始聚类中心为两个: 2 时和 23 时, 则经过延拓后, 初始聚类中心为 4 个: 2 时, 26 时, 23 时和 -1 时。

(2) 其次, 聚类中心的计算。在聚类算法迭代过程中, 计算新的聚类中心时不能简单的将样本的均值作为 k 个新的聚类中心。需要采用式(2) 进行计算, 然后再对聚类中心进行更新。尽管我们对聚类中心进行了延拓, 但是, 最终在更新聚类中心的时候, 实际上更新的聚类中心的个数还是最初的 k 个, 延拓的中心只是用于判断每个数据点属于哪个类别。

4.2 改进后的算法流程

算法流程:

① 输入原始的数据, 随机选择 k 个初始聚类中心;

② 对于每个聚类中心按照式(1) 进行延拓处理, 即聚类中心由原来的 k 个变为 $2k$ 个;

③ 计算每个样本与聚类中心的距离 $D(x_i, Z_j(I)), i = 1, 2, \dots, n; j = 1, 2, \dots, k$ 。若 $D(x_i, Z_i(I)) = \min_{j=1, 2, \dots, k} \{D(x_i, Z_j(I)), i = 1, 2, \dots, n\}$, 则 $x_i \in \omega_c$;

④ 使用每个聚类中所有样本的均值作为 k 个新的聚类中心:

$$Z_j(I+1) = \frac{1}{n_j} \sum_{k=1}^{n_j} x_k^j, j = 1, 2, \dots, k.$$

对于上式中每个 x_k^j 的具体计算方式见式(2)。

⑤ 若 $Z_j(I+1) \neq Z_j(I), j = 1, 2, \dots, k$, 则 $I = I+1$, 返回 ③, 否则算法结束。

4.3 算法分析

从算法复杂度的角度而言, 和原算法相比本文算法的复杂度是有所增加的。新算法增加的复杂度主要在类别的判断部分, 原先是和 K 个中心进行比较, 而新算法要和 $2K$ 个中心进行比较, 即原先要计

算 n 个点与 K 个中心的距离, 而现在需要计算 n 个点与 $2K$ 个中心的距离, 这是主要新增的复杂度所在。但是, 这个增加的算法复杂度是线性的。

5 实验与结果分析

实验采用的数据是电信一个月内每天每隔半个小时在线的人数统计, 想要通过聚类分析得到每天在线人数的分布特点。原始数据的分布如图 2 和图 3 所示, 其中横向坐标为 0~48, 代表每隔半个小时一个采样点, 共 48 个时间点, 纵向坐标是每个时间点的在线人数。在本实验中, 每个时间点的人数作为权重使用, 即执行加权聚类。该方法也可以扩展到其他具有周期性的时间序列数据。

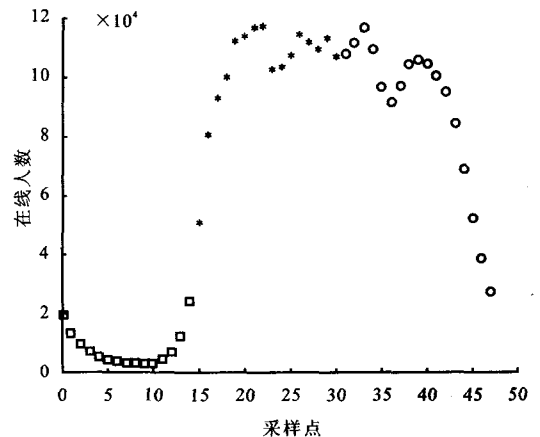


图 2 延拓前的聚类实验结果

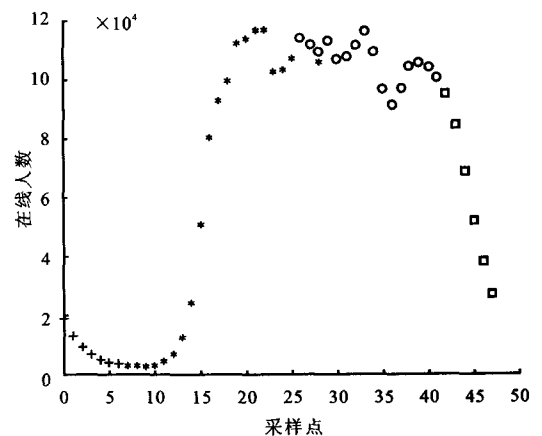


图 3 应用延拓后的聚类实验结果

图 2 和图 3 分别是延拓前和延拓后的聚类算法的结果, 假设要求把数据聚为 3 类。延拓前, $[0, 15]$ 区间的数据为一类, $[16, 30]$ 区间的数据为第二类, $[31 \sim 48]$ 区间的数据为第三类。而延拓后, 两边的数据同属于一类。在聚类过程中, $[0, 6]$ 和 $[40 \sim 48]$ 区间的数据聚在了一类, $[7, 25]$ 的数据聚在了一类, 剩余的数据为第三类。实验结果较好地验证了延拓的

方法,在聚类过程中利用数据的连续性和周期性的作用。

关于时间的耗费和衡量聚类的准则函数值,如表1所示,表中的数据分别为未采用延拓方法和采用延拓方法各运行100次的时间耗费,分别为100 641 ms和199 000 ms,则运行一次的平均时间分别是1.006 s和1.99s,改进的算法在时间代价上有一定的增加。而聚类准则函数如式(3)的值在延拓后比延拓前有了明显的提高,基于这个改进步骤得到了比较好的聚类效果,所以相对而言,增加的运算量是值得的。

表1 算法运行比较

算法	延拓前	延拓后
运行100次时间	100 641 ms	199 000 ms
方差	3.3×10^9	1.53×10^9

6 结束语

实验应用延拓的方法来对聚类中心进行延拓,以充分利用时间序列数据的连续性和周期性,从而达到改善聚类效果的目的。实验结果表明,该方法可以较好地改善现有算法的不足。今后的工作,主要在初始聚类中心的选择上再进行改进,来更好地确保

该延拓方法的实用性及普遍性。

参考文献:

- [1] Jiawei Han, Micheline Kamber. 数据挖掘概念与技术[M]. 范明,孟小峰,译. 北京:机械工业出版社,2007.
- [2] 翁颖钧,朱仲英. 基于动态时间弯曲的时序数据聚类算法的研究[J]. 计算机仿真灰色理论,2004,21(3):37-40.
- [3] 段江娇,薛永生,林子雨,等. 一种新的基于隐Markov模型的分层时间序列聚类算法[J]. 计算机研究与发展,2006,43(1):61-67.
- [4] 黄超,吴清烈,武忠,等. 基于方差波动多重分形特征的金融时间序列聚类[J]. 系统工程,2006,24(6):100-103.
- [5] 秦亮曦,刘新峰,史忠植. 基于片段模式的多时间序列关联分析[J]. 计算机科学,2006,33(1):232-235.
- [6] 邵丹,陈平雁. 模糊C均值聚类在时间序列分析中的应用[J]. 中国卫生统计,2009,26(2):166-170.

作者简介:

郭秀珍 女,(1987-),硕士研究生. 研究方向为数据挖掘.
 陆建峰 男,(1969-),博士生导师. 研究方向为人工智能,数据挖掘.
 汤九斌 男,(1969-),高工. 研究方向为数据挖掘、企业信息化.

(上接第117页)

马鱼胚胎图像上提取的骨架线可能存在一些多余的分支,对分支是否冗余进行度量并去除掉也是骨架算法后期处理的关键,具体实现可以参考文献[7].

6 结束语

本文将一般的细化算法的轮廓收缩过程与距离场结合,用距离场保证了细化算方法收缩方向的正确性. 细化算法的连续性弥补了距离场无法提取连续骨架的缺点. 大量实验证明我们的方法提取出来的骨架准确,连续。

参考文献:

- [1] Harry Blum. Models for the perception of speech and visualForm [M]. Cambridge: Weiant Wathen - Dunn, 1967.
- [2] Blum H. Biological shape and visual science: part I[J]. Journal of Theoretical Biology, 1973,38(2):205-287.
- [3] 韩国强,田绪红,李志垣,等. 三维图像骨架化方法综述[J]. 小型微型计算机系统,2007,28(9):1695-1699.

- [4] 刘俊涛,刘文予,吴彩华,等. 一种提取物体线形骨架的新方法[J]. 自动化学报,2008,34(6):615-622.
- [5] 车武军,杨勋年,汪国昭. 动态骨架算法[J]. 软件学报,2003,14(4):818-821.
- [6] 李川. 斑马鱼胚胎图像的检索技术研究[D]. 南京:南京理工大学,2011.
- [7] Bai Xiang, Liu WenYu. Skeleton pruning by contour partitioning with discrete curve evolution [J]. IEEE Transactions on Pattern Analysis And Machine Intelligenc, 2007,29(3):449-486.
- [8] 吴作凌,陈雄,郑英杰. 一种近红外手背静脉图像识别算法[J]. 微电子学与计算机,2010,27(10):114-119.

作者简介:

李川 男,(1986-),硕士研究生. 研究方向为模式识别和智能机器人.
 彭甫镛 男,(1987-),博士研究生. 研究方向为模式识别和图像分析.
 陆建峰 男,(1969-),博士生导师,教授. 研究方向为模式识别、数据挖掘、智能机器人.