第 38 卷 第 4 期
Vol.38 No.4

计 算 机 工 程
Computer Engineering

2012 年 2 月
February 2012

• 软件技术与数据库 •　　文章编号：1000—3428(2012)04—0040—03　　文献标识码：A　　中图分类号：TP311

基于时序聚类的北斗位置冗余数据压缩算法

赵恩来，郝文宁，刘 航，戎 誉，朱耀华

(解放军理工大学工程兵工程学院，南京 210007)

摘 要：在北斗用户机的位置数据采集过程中，容易出现数据冗余现象。为此，分析导致数据冗余的原因，提出一种基于时序聚类的冗余数据压缩算法。该算法采用基于密度的聚类方法将数据集进行分簇，把属于同一类运动特征的位置数据归为一类，根据簇直径判断该簇是否为冗余数据，并对冗余数据进行压缩。实验结果表明，该算法可以正确标识冗余数据，实现数据压缩。

关键词：冗余数据；时序数据；聚类；数据压缩

Compression Algorithm of Beidou Position Redundant Data Based on Time Series Clustering

ZHAO En-lai, HAO Wen-ning, LIU Hang, RONG Yu, ZHU Yao-hua

(Engineering Institute of Engineering Corps, PLA University of Science & Technology, Nanjing 210007, China)

【Abstract】Aiming at data redundancy problems appeared in the data collection process of Beidou user machine position, the paper analyzes the reason caused by data redundancy. Compression algorithm of redundant data based on time series clustering is proposed. The algorithm which adopts the clustering method based on density puts the data sets into the same cluster, which have the same movement characteristics. According to the cluster diameter to determine whether the cluster is redundant data, then compress the redundant data. Experimental results show the algorithm can correctly identify the redundant data and implement data compression.

【Key words】redundant data; time series data; clustering; data compression

DOI: 10.3969/j.issn.1000-3428.2012.04.013

1 概述

在北斗信息应用系统使用过程中，后台数据库每时每刻都在采集北斗设备的定位数据。然而在北斗设备位置信息采集过程中，有些北斗设备在很长一段时间内，根本就没有运动过(本文称为不动点)，它的存在给后台数据库带来了大量冗余数据，严重影响到数据的精确性和可靠性，甚至数据库的性能。冗余数据就是一个数据表中，这个表中的行包含一些相同的值，即重复元组。

数据冗余是资源浪费、数据不一致的根源。由于北斗卫星导航系统在定位过程中，经度纬度会存在一定的偏移量[1-2]，这种固有的测量误差是无法避免的，因此不动点采集来的数据在经度纬度值上并不完全一样，并非重复元组。但从现实意义上讲，不动点采集回来的若干位置数据已构成了冗余数据。

时间序列是指按时间顺序排列的观测值集合。在对时间序列进行数据挖掘的过程中，必须考虑数据间存在的时间关系，这类数据挖掘称为时间序列数据挖掘。对时间序列数据进行聚类的算法有基于相似性(或距离)[3]、基于特征[4]、基于模型[5]和基于分割[6]的聚类分析。本文基于时序聚类的方法解决北斗设备位置数据冗余问题。

2 问题描述

北斗设备位置数据存储在定位设备轨迹表中，结构如表 1 所示。

由于北斗卫星固有测量误差和不动点的存在，定位设备轨迹表中将积累大量的经度值、纬度值相近，采样时刻不一样的同一目标的定位数据，这些数据就是本文所要解决的冗余数据。经过调研，根据目标运动特性，本文把目标分为 4 类：

(1)不动点

该目标运动速度为 0，待在原地未曾发生经度和纬度上的变化。但由于北斗卫星固有测量误差(ΔR: m)，采集入库的同一目标的位置数据并非相同。

(2)慢速运动点

该目标运动速度比较慢，运动步长小于 ΔR。时间上先后 2 次采集的位置数据，其距离甚至小于北斗卫星固有测量误差 ΔR。

(3)迂回点

该目标运动速度正常，并沿着固定的轨迹来回巡逻，在不同的时刻可能位置数据完全一致，或者非常相近。这类数据并非冗余数据，不可删除，必须保留。

(4)正常运动点

该目标运动速度正常，运动步长大于 ΔR。即时间上先后 2 次采集的位置数据，其距离大于北斗卫星固有测量误差 ΔR。

本文算法的目的就是要从海量北斗设备数据中，找出不动点冗余数据，并根据策略进行压缩；用不同的簇编号标识出其他 3 类点形成的轨迹；并避免把“慢速运动点”和“迂回点”数据误判成冗余数据。

作者简介：赵恩来(1985－)，男，硕士研究生，主研方向：数据挖掘；郝文宁，副教授、博士研究生；刘 航、戎 誉，学士；朱耀华，硕士研究生

收稿日期：2011-08-08　**E-mail：**hwnbox@163.com

表 1　定位设备轨迹

序号	中文描述	字段名	类型	空值	主键	描述
1	auto_ID	auto_ID	int(11)		√	表记录的自增 id
2	定位设备 ID	position_equ_id	varchar(50)			对应北斗机器的 ID 号码
3	定位时间	position_equ_id	datatime(0)	√		对象位置数据采集时间
4	定位类型	position_type	smallint(6)	√		对象位置类型
5	经度	longitude	double(0)	√		经度，11.8 格式
6	纬度	latitude	double(0)	√		纬度，10.8 格式
7	高程	altitude	double(0)	√		位置高度
8	高程异常	altitude_uncon	float(9)	√		高度异常为 0
9	精度等级	precision_level	int(11)	√		精确度数等级
10	记录者	registrar	varchar(38)	√		发送位置信息的北斗机号码
11	标识	mark	tinyint(3)			标识为 0

3　基于时序聚类的冗余数据压缩算法

根据目标运动速度特性，本算法主要分为 3 个步骤：

(1)基于时序的聚类：遍历数据表，对每条元组进行标号，把属于同一个簇的元组用同一个簇标号进行标记。这里的某一个簇可能是不动点点集，或者某个运动目标的轨迹，亦或是某个噪声点。

(2)判断簇类型：对已经聚类好的簇进行判断。分辨出不动点点集、噪声点集和运动目标轨迹。

(3)压缩数据：根据策略选择方法对不动点数据进行压缩。

本文将算法步骤(2)和步骤(3)进行合并，将在下文详细分析基于时序的聚类算法和簇判断及压缩冗余数据算法。

3.1　基于时序的聚类算法

对于同一个运动目标，在时间和空间上应具有一定的连续性，进而形成运动轨迹，而不动点会以经度、纬度的真值为圆心，随机散布成一个类圆区域。所以，在同一个簇中，采集时刻上相邻的 2 个元组在时间差、空间距离差应小于给定的参数。子算法从第一个没有被标记簇编号的元组开始遍历余下的数据集合(整个数据集已按时间先后顺序排序)，顺序寻找簇中的下一个点，并以当前元组的簇标号标记下一个点；直至遍历完整个数据集合，以新的簇编号开始新一轮的遍历。该子算法每一次遍历都会得到一个簇，直至整个数据中的元组都标记上簇编号。

对于地球上任意 2 个点 Pi(longitude_i, latitude_i)、Pj(longitude_j, latitude_j)，假设经度、纬度的单位为度，不计高程，地球为规则球体，其半径为 R。则这 2 个点的经度距离差、纬度距离差和球面距离差分别为：

$$\Delta longitude = \frac{|longitude\_i - longitude\_j|}{360} \cdot 2\pi R \cdot \cos(longitude\_i)$$

$$\Delta latitude = \frac{|latitude\_i - latitude\_j|}{360} \cdot 2\pi R$$

$$P_iP_j = R \cdot \arccos[\cos latitude\_i \cdot \cos latitude\_j \cdot \cos(longitude\_i - longitude\_j) + \sin latitude\_i \cdot \sin latitude\_j]$$

基于时序的聚类算法，根据定位设备 ID、定位时间、经度、纬度 4 个属性值，以簇中元组在时间、空间上的连续性为依据，对数据集进行聚类，并给每个元组标记上相应的簇编号。基于时序的聚类算法流程如图 1 所示。

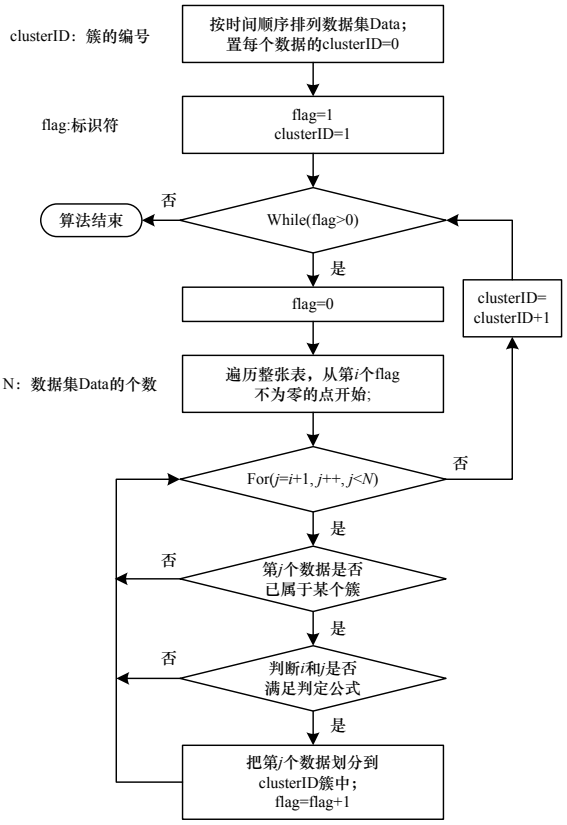

图 1　基于时序的聚类算法流程

3.2　簇判断及压缩冗余数据算法

根据 3.1 节的分析可知，不动点点集散布成类圆区域；而其他运动目标的点集形成轨迹。设北斗卫星经度精度为 ΔR_Longitude，纬度精度为 ΔR_Latitude，某不动点 O 的真实位置为点 P_0，在第 i 时刻位置为 P_i，下一采样时刻 j 位置为 P_j 或 P'_j 或 P''_j，如图 2 所示。

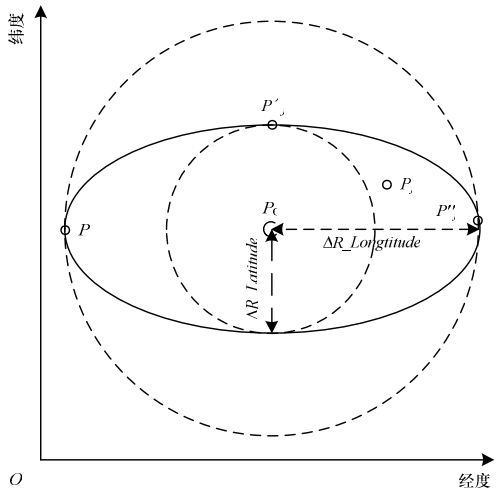

图 2　不动点散落区域

在图 2 中，2 个虚线圆分别以真实位置 P_0 为圆心，$\Delta R\_Longitude$ 和 $\Delta R\_Latitude$ 为半径；实线加粗椭圆为真实位置 P_0 为圆心，$\Delta R\_Longitude$ 和 $\Delta R\_Latitude$ 分别为长半轴和短半轴。故不动点的采集回来的数据应全部散落在该椭圆上及椭圆内部。经上述分析，不难得出重要结论：不动点的簇直径小于给定参数；其他运动目标轨迹的簇直径大于给定参数。簇判断及压缩冗余数据算法首先计算每个簇的直径，结

合给定参数判断簇类型；然后根据压缩策略，选择压缩冗余数据的方法。子算法流程如图 3 所示。

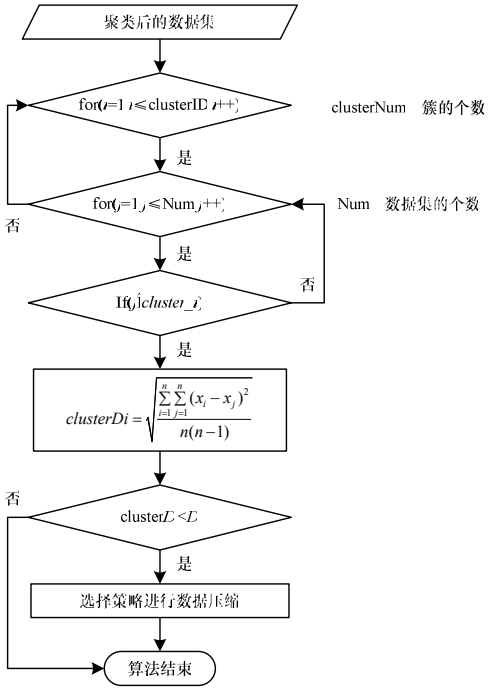

图 3 簇判断及压缩冗余数据算法流程

4 实验结果与分析

4.1 实验结果

本文用于实验的数据为某军分区某次演习所有北斗装备的位置信息。用于实验的数据主要包括 8 台北斗指挥机，共计 100 台手持或车载用户机，元组个数为 2.5×10^6 条，数据量为 240×10^6。通过本文提出的基于时序聚类的冗余数据压缩算法，对数据库中的冗余数据按策略进行压缩，并标记出运动目标的轨迹。并标记出若干个不同运动轨迹的对象。分析图 2 可知，不动点点集中任何 2 个元组的经度误差和纬度误差都小于等于 $2\times\Delta R\_Longitude$ 和 $2\times\Delta R\_Latitude$，且不动点簇的直径应小于等于 $2\cdot\sqrt{\Delta R\_Longitude^2+\Delta R\_Latitude^2}$。“北斗一号”定位误差在 10 m 左右，故在本实验中，经度误差、纬度误差和簇直径 3 个参数分别设为 30 m、30 m、45 m，此时数据集中不动点的检出准确率最高，超过 95%；冗余数据的压缩率 6.81%。系统运行结果如图 4 所示。

图 4 冗余数据压缩系统运行结果

4.2 算法分析

为解决北斗信息应用系统中出现的北斗位置数据冗余问题，本文从基于密度聚类算法可以发现任意形状簇[7-8]的角度提出了一种基于时序聚类的冗余数据压缩算法。本文算法首先根据北斗装备的 id、时间、经度、纬度，采用基于时序的聚类算法，把定位设备轨迹表中数据分成若干个簇：每个簇是一个运动目标形成的轨迹或不动点散布成的椭圆区域或噪声数据；计算每个簇的直径，根据参数判断簇的类型，并根据策略对不动点的元组进行压缩。本算法有 2 个特点：

(1)采用基于密度的聚类算法思想，可以发现任意形状的轨迹。根据目标的运动特性及输入参数，从当前点确定某轨迹上的下一点，每次遍历确定某一目标的运动轨迹数据，并把它们标记为同一个簇。算法基本时间复杂度是 $O(n\times$确立一个簇所需要的时间$)$，其中 n 是数据集的个数。在最坏情况下，时间复杂度是 $O(n^2)$，此时每个点都属于不同的簇；最好情况下，时间复杂度为 $O(n)$，此时每个点都属于同一个簇。

(2)采用基于时间裁剪的思想。由于运动目标轨迹中前后 2 点在时间上具有先后性，因此在标记某点是否是核心点时，不必计算待标记点与其他所有点的距离，只需计算与待标记点在采样时刻上相近的点，并且采样时刻值应大于待标记点的时刻值。

5 结束语

本文针对北斗信息应用系统使用过程中出现的定位数据冗余问题，首先分析导致数据冗余的原因，提出基于时序聚类的冗余数据压缩算法，并分析参数的设置。实验表明，该算法能正确聚类各类数据，并对冗余数据实行压缩。如何进一步提高算法的运行效率及聚类准确性是下一步研究方向。

参考文献

[1] 赵树强, 许爱华, 张荣之, 等. 北斗一号卫星导航系统定位算法及精度分析[J]. 全球定位系统, 2008, 33(1): 20-24.

[2] 何家滨, 刘小明. 北斗双星定位算法精度的研究[J]. 舰船电子工程, 2010, 30(3): 85-88.

[3] Kalpakis K, Gada D, Puttagunta V. Distance Measures for Effective Clustering of ARIMA Time-series[C]//Proc. of the IEEE Int'l Conf. on Data Mining. San Jose, USA: IEEE Press, 2001.

[4] Zhang Hui. A Non-parametric Wavelet Feature Extractor for Time-series Classification[C]//Proc. of the 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Berlin, Germany: [s. n.], 2004.

[5] Vatthyanathan S. Model-based Hierarchical Clustering[C]//Proc. of the 16th Int'l Conf. on Uncertainty in Artificial Intelligence. Stanford, USA: Morgan Kaufmann, 2000.

[6] 李爱国. 时间序列数据分割与时态模式挖掘研究[D]. 西安: 西安交通大学, 2003.

[7] Han Jiawei. 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, 2008.

[8] 赵恩来, 郝文宁, 赵水宁, 等. 改进的基于密度方法的态势聚类显示算法[J]. 计算机工程, 2010, 36(18): 35-37.

编辑 陈 文