Journal of Computer Applications
计算机应用,2012,32(3):625-628,642
ISSN 1001-9081
CODEN JYIIDU
2012-03-01
http://www.joca.cn

文章编号:1001-9081(2012)03-0625-04
doi:10.3724/SP.J.1087.2012.00625

基于三层存储模型的 RFID 数据压缩存储方法

夏秀峰,赵　龙*
(沈阳航空航天大学 计算机学院,沈阳 110136)
(*通信作者电子邮箱 changchuan618@163.com)

摘　要:针对物联网技术中亟待解决的海量数据存储问题,提出了一种基于射频识别(RFID)的三层数据存储压缩模型。该模型将数据分为当前数据层、临时数据层和历史数据层,利用每一层中数据的特点分别设计了相应的数据汇总算法,最终实现 RFID 数据的压缩存储。在该模型的基础之上,提出了针对路径的编码算法,用于对路径进行压缩存储。实验结果表明,该三层存储模型可以有效地压缩存储 RFID 数据,同时数据汇总算法具有较低的时间复杂度与较高的数据压缩比。

关键词:射频识别;海量数据;路径编码;数据压缩;三层存储模型
中图分类号:TP301.6;TP311　　**文献标志码**:A

RFID data compression storage method based on three-level storage model

XIA Xiu-feng, ZHAO Long*
(*School of Computer, Shenyang Aerospace University, Shenyang Liaoning* 110136, *China*)

Abstract: Concerning the problem of massive data storage in the technology of the Internet of things, this paper proposed a three-level Radio Frequency IDentification (RFID) data compression storage model. This model divided the data into the current level, the temporary level and the historical level. Corresponding data collection algorithm was designed for every level according to the features of data in each level. And a coding algorithm for paths based on the model was proposed to store the compressed paths. The experimental results show the three-level storage model can effectively store compression data, and demonstrate the algorithm of data gathering has higher data compression rate as well as lower time complexity.

Key words: Radio Frequency IDentification (RFID); massive data; path coding; data compression; three-level storage model

0　引言

所谓物联网就是物物相连的互联网,是指通过射频识别、红外感应器等信息传感设备把物品与互联网相连接,进行信息交互和通信的一种网络[1-3]。物联网这个概念在 1999 年被提出之后,并没有引起人们广泛的关注,由于其包含技术的复杂性,社会普遍质疑物联网大规模实施的可行性。但随着构建物联网的电子芯片费用的不断降低与电子标签(Electronic Product Code, EPC)技术的日渐成熟[4-5],物联网的普及逐渐变得切实可行。

普遍认为,射频识别(Radio Frequency IDentification, RFID)的特性为:时空关联性、海量性、不确定性、实时性等[6-7]。随着物联网技术的日益发展,如何有效并快速地存储与查询 RFID 数据逐渐引起人们的重视。如果电子标签被放置在每个物品上,那么类似于沃尔玛这样的大型超市将会在一天之内得到 7 TB 左右的数据,所以像 Oracle、IBM、Teradata 和一些其他的数据库公司不得不考虑将 RFID 信息整合到企业级数据库中[8]。

在物联网被广泛应用的背景下,RFID 数据存储与管理逐渐成为物联网技术的研究方向之一[9-10]。之前的研究工作主要采用单一数据层的方式存储 RFID 数据,较少涉及压缩存储与历史数据的处理。如文献[11]首次给出了一般意义上的 RFID 数据存储结构与数据管理的体系结构,但是其结构并不能完全适应当前的 RFID 应用系统;文献[12]提出了一个以位置为关键字的 RFID 数据存储模型,并给出了在这个模型之上的查询语句,但是对于历史数据却并没有进行处理;文献[8]提出了一个简单的 RFID 数据压缩方法,其主要思想是将一个固定的编号来代表一连串情境相关的 EPC 编号(如一箱牛奶等),但是对于这个编号的尚未完成的划分方式却是实施这一方法的阻碍;文献[13]提出了一个 RFID 数据压缩方法,该方法的主要思想是通过合并与连接那些用户不感兴趣的路径片段进行路径的语义压缩,但是如何确定哪些路径用户不感兴趣是一个很大的难题。

故本文根据 RFID 数据的特点,提出了 RFID 三层数据存储模型,并给出了相应数据层的数据汇总算法。

1　RFID 数据压缩存储模型

为了更好地区别与阐述当前数据与历史数据,给出 RFID 历史数据的定义。

定义 1　RFID 历史数据。RFID 历史数据为在某一特定事件驱动前的 RFID 数据,该特定事件与具体的 RFID 系统应用相关。

例如在一个超市 RFID 系统之中,一个物品在被卖给消费者之后,它之前被阅读器扫描得到的数据被称为历史数据。而在一个物流监控系统当中,在物品离开物流系统最终到达零售商店中之后,之前它在物流中产生的数据称为历史数据。

收稿日期:2011-09-14;**修回日期**:2011-11-10。

作者简介:夏秀峰(1964-),男,山东胶南人,教授,博士,CCF 高级会员,主要研究方向:数据库、RFID 数据管理;　赵龙(1986-),男,辽宁盘锦人,硕士研究生,主要研究方向:管理信息系统、数据库、RFID 数据管理。

如图 1 所示。

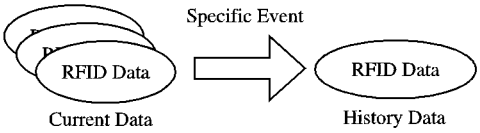

图 1 RFID 历史数据的产生

本文采用了三层存储结构,并给出了相应的数据汇总方法,以达到数据压缩的目的。本文的存储模型结构如图 2 所示。

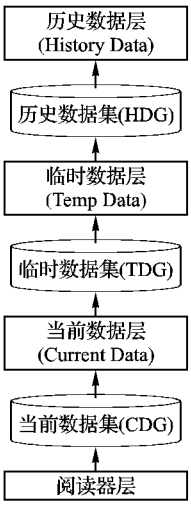

图 2 三层存储模型结构

图 2 中各层之间通过相应的数据汇总算法进行数据的汇总。本文 RFID 数据流的传递顺序是:阅读器层→当前数据层→临时数据层→历史数据层,并且高层数据层的数据量比低层数据层的数据量小。

1.1 当前数据层模型

在一个 RFID 系统中,将从阅读器得到的原始数据的集合称为观测数据集。其数据形式为 Observation{E,L,T},其中 E 表示被扫描的物品的 EPC 编码,L 表示物品被扫描的地点,T 表示物品被扫描的时间。其具体形式如表 1 所示。

表 1 Observation 集

E	L	T	E	L	T
epc_1	loc_1	t_1	epc_1	loc_2	t_5
epc_1	loc_1	t_2	epc_1	loc_2	t_6
epc_1	loc_1	t_3	epc_2	loc_1	t_7
epc_1	loc_2	t_4	epc_2	loc_1	t_8

随着时间的推移,观测数据集中的数据量将异常庞大,这时需要将观测数据向当前数据层进行汇总。当前数据层的数据形式为 CurrentData{EL,LocID,TS,TE,Count},其中 TS 与 TE 分别表示该物品在这个位置第一次被扫描到的时间与最后一次被扫描到的时间,Count 代表该物品在 TS 到 TE 这个时间段内在该位置出现的次数。

当观测数据集向当前数据集进行汇总时,首先会进行 EPC 编号的匹配。如果在 CurrentData 集中不存在这个 EPC 编号,则将这条信息存入 CurrentData 集中;否则将会进行位置信息的匹配,即查找该信息的 LocID 是否存在于 CurrentData 集中,如果存在则将计数增加,否则同样将这条信息存入 CurrentData 集中。

下面给出当前数据层的汇总算法:

算法 1 当前数据集(Current Data Gather, CDG)汇总算法。

输入 最低粒度集 Observation{E,L,T}。

输出 当前数据集 CurrentData{EL, LocID, TS, TE, Count}。

```
1)  lloc = ∅;
2)  for e_i ∈ E do
3)    if e_i ∈ EL then
4)      //检查 e_i是否存在于 CurrentData 集中
5)      //如果存在则检查位置信息是否相同
6)      if lloc_j = loc_i then
7)        //如果相同则更改 te_j并计数
8)        te_j = t_i;
9)        count_j++;
10)     end if
11)     //将不存在则将 e_i添加进去
12)     else then
13)       CurrentData = CurrentData∪{e_i, loc_i, t_i, t_i, 1};
14)   end if
15) end for
16) return CurrentData;
```

例 1 在一个物联网系统中,实体 epc_1 的标签在分别在时刻 t_1、t_2、t_3 在 loc_1 被读取到,t_4、t_5、t_6 在 loc_2 被读取到,epc_2 的标签在时刻 t_7、t_8 在 loc_1 被读取到,此时 Observation 集的内容如表 1 所示。则当运行完 CDG 算法之后,CurrentData 集合中的内容如表 2 所示。

表 2 CurrentData 集

EL	LocID	TS	TE	Count
epc_1	0001	t_1	t_3	3
epc_1	0002	t_4	t_6	3
epc_2	0001	t_7	t_8	2

通过这个例子可看出:经过 CDG 算法之后得到的数据集要比原始的 RFID 数据集的数据量小,即有效地进行了数据压缩。

1.2 临时数据层模型

临时数据层是中间数据层,其主要工作是将当前数据层中的位置点信息转化为路径信息进行存储,以达到压缩存储的目的。为了便于描述,下面给出 RFID 数据路径的定义。

定义 2 RFID 数据路径。RFID 数据路径是一串有序的位置点编号的集合,形如 $PathID_i\{locID_j \mid j \in \mathbf{N}\}$,其中,$PathID_i$ 表示路径的编号,$locID_j$ 表示出现在这条路径上的位置点。

为了更好地阐述路径之间的关系,下面给出 RFID 数据子路径与主路径的定义。

定义 3 RFID 数据子路径。对于两条路径 $trace_x$ 与 $trace_y$,如果对于任意按序的 $loc_i \in trace_x$(其中 $i \in \mathbf{N}$),$loc_i \in trace_y$,则说明 $trace_x$ 为 $trace_y$ 的子路径,记为 $trace_x \rightarrow trace_y$。

定义 4 RFID 数据主路径。RFID 数据主路径是指不包含父路径的路径,即如果 $trace_i$ 为主路径,则不存在 $trace_j \in Trace$,使得 $trace_i \rightarrow trace_j$。反之,我们称不是 RFID 数据主路径的路径为 RFID 非主路径。

对于路径的编号采用改进的二进制哈夫曼编码,其主要思想是将路径的前 $n/2$ 个码位置为其父路径编码的前 $n/2$ 个码位,而后 $n/2$ 个码位为其自身的自然序号。一条 RFID 非主路径编号编码如式(1)所示。

$$PathID_i = PathID_i.ParentID * 2^{\frac{n}{2}} + i \tag{1}$$

而一条 RFID 主路径的编号编码如式(2)所示。

$$PathID_i = i * 2^{\frac{n}{2}} \quad (2)$$

其中:$PathID_i$. ParentID 表示该路径的父路径编号,i 表示路径的自然序号。采用编号化的编码方式可以有效提高查询效率,同时压缩存储空间。

下面给出具体的路径编码算法:

算法 2　路径编码(Path Coding,PC)。

输入　路径集合 Trace$\{trace_i \mid i \in N\}$。

输出　路径编码 PathID$\{pid_i \mid i \in N\}$。

1)　计算 n 的值
2)　for $trace_i \in$ Trace do
3)　　检查 $trace_i$是否有父路径
4)　　if $trace_i$有父路径 then
5)　　　将 pid_i父路径的前 $n/2$ 个字段存入 pid_i的前 $n/2$ 个字段;
6)　　　将 i 存入 pid_i的后 $n/2$ 个字段;
7)　　else then
8)　　　将 i 存入 pid_i的前 $n/2$ 个字段;
9)　　　将 pid_i的后 $n/2$ 个字段设置为 0;
10)　　end if
11)　　PathID = PathID$\cup\{pid_i\}$;
12)　end for
13)　return PathID;

例 2　有这样一组路径 Trace$\{trace_1, trace_2, \cdots, trace_6\}$,其中,$trace_2 \rightarrow trace_1$, $trace_3 \rightarrow trace_5$, $trace_4 \rightarrow trace_2$, $trace_6 \rightarrow trace_1$。在这个例子中,可以看到 $trace_1$ 和 $trace_5$ 为主路径,根据 PC 编码算法,从 $trace_1$ 到 $trace_6$ 的编码应为:001 000,001 010,101 011,010 100,101 000,001 110。

根据 PC 编码,可以将当前数据层中的数据向临时数据层进行汇总。临时数据层的数据存储结构为 TempData{E, TraceID,TTS,TTL},其中 E 表示单个物品的编号,TraceID 表示路径的编号,TTS 表示该物品在该路径上的起始时间,TTL 表示该物品在该路径上的持续时间。具体汇总算法思想是将每个 EPC 编号所对应的 Location 信息进行合并,并最终合并为一条条不同的路径,并对每条路径进行 PC 编码。下面给出临时数据集汇总算法:

算法 3　临时数据集(Temp Data Gather,TDG)汇总算法。

输入　当前数据集 CurrentData{EL, LocID, TS, TE, Count}。

输出　临时数据集 TempData{E,TraceID, TTS, TTE},路径集 Path{PathID,Route}。

1)　Trace = ∅;
2)　for $el_i \in$ EL do
3)　　if $e_j \in$ E then
4)　　　//检查 el_i是否存在于 TempData 集中
5)　　　//如果存在则更新路径信息与时间信息
6)　　　$trace_j$ = $trace_j \cup \{locID_i\}$;
7)　　　tte_j = te_i;
8)　　　//将不存在于临时数据集中的 e_i添加进去
9)　　else then
10)　　TempData = TempData$\cup\{el_i, locID_i, ts_i, te_i\}$;
11)　　对 el_{i-1}所在的条目进行 PC 编码;
12)　　end if
13)　end for
14)　更新 Path 集
15)　return Path,TempData;

例 3　对表 1 中的 CurrentData 集合执行 TDG 算法,并对路径进行 PC 编码之后,TempData 集合中的内容如表 3 所示。

表 3　TempData 集

E	TraceID	TTS	TTE
epc_1	0001 0000	t_1	t_6
epc_2	0001 0010	t_7	t_8

Path 集合中的内容如表 4 所示。

表 4　Path 集

PathID	Route
0001 0000	0001,0002
0001 0010	0002

1.3　历史数据层模型

由于 EPC 编号的格式很固定,相同种类物品之间的 EPC 编号相连,故可以将不同的 EPC 汇总到不同的物品种类簇(EPC Cluster, EPCC)中,并用 EPCC 编号代替 EPC 编号。

根据定义 1,在经过某一特定的事件驱动之后,要将临时数据层数据汇总到历史数据层。历史信息层的数据存储结构为 History{EPCC, HTraceID, TS, TL, Count},其中,EPCC 为 EPC 的类别号,HTraceID 为其主路径编号,Count 为这类物品在这条路径上出现的次数。历史数据汇总的具体思想是将 EPCC 信息和路径的主路径信息代替 TempData 中的 EPC 信息和路径信息,从而达到压缩存储的目的。

下面给出历史数据汇总的算法。

算法 4　历史数据集(History Data Gather, HDG)汇总算法。

输入　临时数据集 TempData{E,TraceID,TTS, TTE}。

输出　历史数据集 History{EPCC, HTraceID, TS, TE, Count}。

1)　for $e_i \in$ E do
2)　　if e_i belongs to $epcc_j$then
3)　　　//查看 $epcc_j$是否存在于 History 中
4)　　　if $epcc_j \in$ EPCC then
5)　　　　//检查 TraceID 与 HTraceID 的匹配
6)　　　　for $htraceid_k \in$ HTraceID do
7)　　　　　if $htraceid_k$ < − $traceid_i$ then
8)　　　　　　$count_k$++;
9)　　　　　else then
10)　　　　　//如果路径不存在则将主路径添加
11)　　　　　History = History$\cup\{epcc_j, htraceid_i, tts_i, tte_i, 1\}$;
12)　　　　　end if
13)　　　　end for
14)　　　else then
15)　　　　//如果 $epcc_j$不存在则将主路径添加进去
16)　　　　History = History$\cup\{epcc_j, htraceid_i, tts_i, tte_i, 1\}$;
17)　　　end if
18)　　end if
19)　end for
20)　return History;

例 4　对例 3 中的 TempData 集合执行 HDG 算法之后,History 集中的内容如表 5 所示。

表 5　History 集

EPCC	HTraceID	TS	TE	Count
$epcc_1$	00010000	t_1	t_6	1
$epcc_2$	00010000	t_7	t_8	1

2 实验结果及分析

本部分对该三层存储模型及其数据汇总算法进行实验验证。本文的硬件环境是:2.1 GHz 的 Intel Core 2 Duo CPU,2.0 GB 的主存,160 GB 的硬盘。软件环境是:操作系统为 Windows XP,编程环境为 JDK1.6,数据汇总算法采用 Java 语言编写。实验主要分析算法的时间性能、数据压缩率、数据的失真率以及查询的响应时间。

2.1 实验数据

由于场地与应用的限制,本文采用了模拟的数据集,即通过程序模拟了物品跟踪系统产生的 10^5 条 RFID 数据。为了更全面地验证算法的可靠性,将这 10^5 条数据划分为 4 个子集,分别包含 1×10^4,2×10^4,3×10^4 和 4×10^4 条数据,分别记为数据集 1、数据集 2、数据集 3 和数据集 4。

2.2 算法的时间性能

本文分别针对上述 4 个数据集进行 3 层数据汇总算法,实验得到算法消耗时间如图 3 所示。

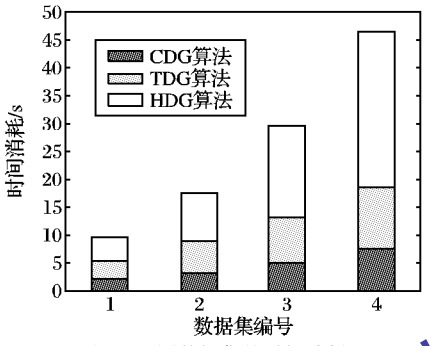

图 3 不同数据集的时间消耗

通过图 3 可看出:随着数据集数据量的增大,时间消耗将随之增加,并且大部分的时间均消耗在第 3 层数据汇总即 HDG 算法之中,因此可以考虑改进该算法以进一步降低时间开销。

2.3 算法的数据压缩率

数据的压缩率是指每个算法运行结束之后得到的数据集的大小与原始数据集的大小之比。本实验的数据压缩率如图 4 所示。

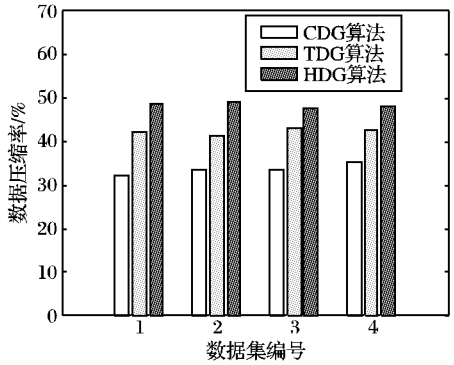

图 4 数据压缩率

通过图 4 可看出:HDG 算法的数据压缩率最高,TDG 算法次之,而 CDG 算法的数据压缩率最低。同时,随着数据集的不断增大,这 3 个算法的数据压缩率变化不大,即这 3 个算法的数据压缩率趋于固定值。

2.4 数据的失真率

由于只有第 3 层数据汇总即 HDG 算法采用的是有损压缩方法,所以本实验过程只考虑 HDG 算法的失真率。RFID 数据失真率(Data Lost, DL)的计算公式为:

$$DL = lost\_colums/N \tag{3}$$

其中:*lost_colums* 表示已经失效的数据条数,*N* 表示总的数据条数。

通过在 4 个数据集上运行 3 层汇总算法之后,实验得到 HDG 算法的失真率如图 5 所示。

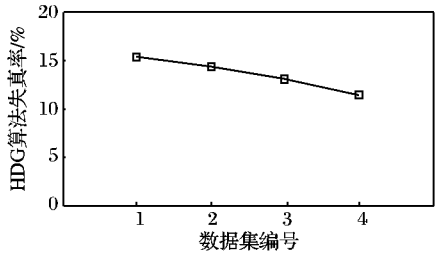

图 5 数据的失真率

从图 5 可看出:随着数据集数量的增大,HDG 算法的失真率将会变小,这是由于主路径数量的增加随着数据集的增大而趋于缓慢所造成的。该实验数据表明,随着主路径数据量的增加,使用主路径替代原始路径将使数据更加真实。

2.5 查询的响应时间

分别在本文提出的三层模型与原始数据集下运行 1 000 条标准查询语句来检验模型的查询响应时间,实验结果如图 6所示。

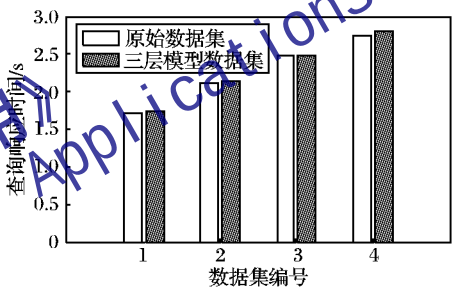

图 6 查询的响应时间

从图 6 可看出:随着数据集数据量的不断增加,查询的响应时间也相应增大。但是在不同的数据集中,运行在原始数据之上的查询响应时间与运行在本文提出的模型数据之上的查询响应时间基本相同,说明本文数据的压缩存储结构对数据的查询影响并不明显。

3 结语

本文建立了一种针对 RFID 数据的三层压缩存储模型,并给出了相应的数据层的数据汇总算法。对数据汇总算法的复杂度分析及实验数据分析,表明本文提出的三层数据存储结构可以有效地压缩数据,具有较低的时间复杂度和较少的查询响应时间,同时,存储模型的第三层压缩数据具有较低的数据失真率,说明该模型可适应大规模 RFID 数据集。

参考文献:

[1] GUSTAVO R, MARIO M, CARLOS D. Early infrastructure of an Internet of things in spaces for learning [C]// Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies. Piscataway, NJ: IEEE Press, 2008: 381 - 383.

[2] HU YING, SUNDARA S, CHORMA T, *et al*. Supporting RFID-based item tracking applications in Oracle DBMS using a bitmap datatype [C]// Proceedings of the 31st International Conference on Very Large Data Bases. New York: ACM Press, 2005: 1140 - 1151.

[3] COCCI R, TRAN T, DIAO Y, *et al*. Efficient data interpretation and compression over RFID streams [C]// Proceedings of the 24th International Conference on Data Engineering. Piscataway, NJ: IEEE Press, 2008: 1445 - 1447.

(下转第 642 页)

有最好的聚类效果，聚类性能绝对地优于原始 RPCL 算法和魏立梅算法。对含有噪声的人工模拟数据集可以实现最好的聚类，而不受噪声点的影响，因此该算法具有很强的抗噪声能力。

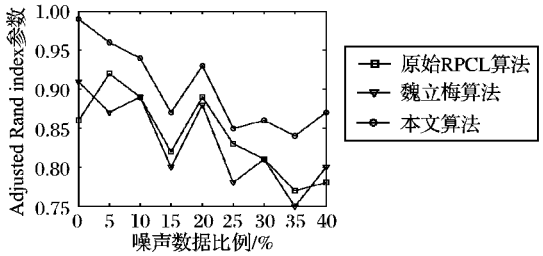

图 8 人工模拟数据集上 Adjusted Rand index 参数的比较

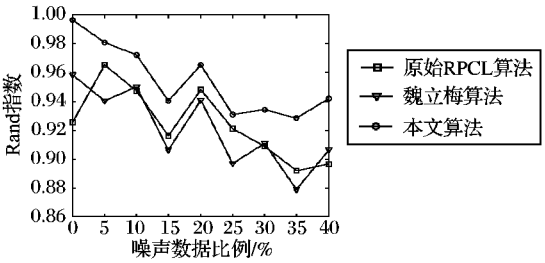

图 9 人工模拟数据集上 Rand 指数的比较

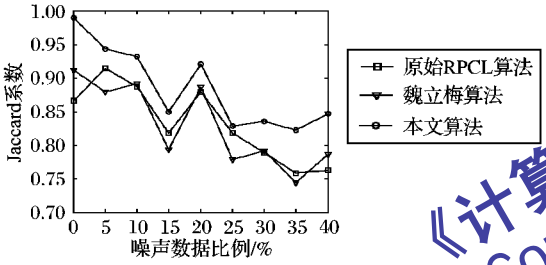

图 10 人工模拟数据集上 Jaccard 系数的比较

4 结语

在分析现有 RPCL 算法不足基础上，提出一种基于样本空间分布密度的改进 RPCL 算法，引入数据集的几何结构，利用数据集样本的自然分布信息定义样本密度，将该密度引入到 RPCL 算法的节点权值调整，解决现有 RPCL 算法没有考虑数据集几何结构对节点权值调整的影响，或者考虑不足的问题。UCI 机器学习数据库数据集和随机生成的带有不同比例噪声的人工模拟数据集上的实验共同表明：本文算法能够有效确定数据集的合适类簇数目和初始类簇中心。聚类时间、聚类误差平方和，以及 Rand 指数、Jaccard 系数和 Adjusted Rand index 参数 3 个聚类有效性指标参数的比较分析显示，本文算法收敛速度快，聚类效果好，对噪声数据有很强的抗干扰性能。不足之处是：本文算法依然是对球形数据进行分析的，关于非球形数据的分析有待进一步研究。

参考文献：

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

[2] HAN J W，KAMBER M. 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，2000.

[3] JAIN A K，DUBES R C. Algorithms for clustering data[M]. Upper Saddle River，NJ：Prentice Hall，1988：1－334.

[4] XU L，KRZYZAK A，OJA E. Rival penalized competitive learning for clustering analysis [J]. IEEE Transactions on Neural Networks，1993，4(4)：636－649.

[5] 李听，郑宇，江芳泽. 用改进的 RPCL 算法提取聚类的最佳数目[J]. 上海大学学报，1999，40(8)：120－122.

[6] 魏立梅，谢维信. 聚类分析中竞争学习的一种新算法[J]. 电子科学学刊，2000，22(1)：13－18.

[7] 张忠平，王爱杰，柴旭光. 简单有效的确定聚类数目算法[J]. 计算机工程与应用，2009，45(15)：166－168.

[8] 张惟皎，刘春煌，李芳玉. 聚类质量的评价方法[J]. 计算机工程，2005，31(20)：10－12.

[9] 王骏，程乾生. 模糊聚类方法中的最佳聚类数的搜索范围[J]. 中国科学 E辑，2002，32(2)：274－280.

[10] 王开军，李健，张军英，等. 聚类分析中类数估计方法的实验比较[J]. 计算机工程，2008，34(9)：198－199.

[11] 杨善林，李永森，胡笑旋，等. *K*-means 算法中的 *k* 值优化问题研究[J]. 系统工程理论与实践，2006(2)：97－101.

[12] PARK H S，JUN C H. A simple and fast algorithm for *K*-medoids clustering [J]. Expert Systems with Applications，2009，36(2)：3336－3341.

[13] 杨燕，靳蕃，KAMEL M. 聚类有效性评价综述[J]. 计算机应用研究，2008，25(6)：1631－1632.

[14] HUBERT L，ARABIE P. Comparing partitions [J]. Journal of Classification，1985，2(1)：193－218.

[15] VINH N X，EPPS J，NAILEY J. Information theoretic measures for clusterings comparison：Is a correction for chance necessary? [C]// Proceedings of the 26th International Conference on Machine Learning. New York：ACM Press，2009：1073－1080.

（上接第 628 页）

[4] DERAKHSHAN R，ORLOWSKA M，LI X. RFID data management：challenges and opportunities [C]// IEEE International Conference on RFID. Piscataway，NJ：IEEE Press，2007：175－182.

[5] FAZZINGA B，FLESCA S，MASCIARI E，*et al.* Efficient and effective RFID data warehousing [C]// IDEAS'09：Proceedings of the 2009 International Database Engineering & Applications Symposium. New York：ACM Press，2009：251－258.

[6] GONZALEZ H，HAN J W，LI X L. Mining compressed commodity workflows from massive RFID data sets [C]// Proceedings of the 15th ACM International Conference on Information and Knowledge Management. New York：ACM Press，2006：162－171.

[7] BAI Y，WANG F S，LIU P Y，*et al.* RFID data processing with a data stream query language [C]// ICDE 2007：IEEE the 23rd International Conference on Data Engineering. Piscataway，NJ：IEEE Press，2007：1184－1193.

[8] GONZALEZ H，HAN J W，LI X L，*et al.* Warehousing and analyzing massive RFID data sets [C]// ICDE'06：Proceedings of the 22nd International Conference on Data Engineering. Piscataway，NJ：IEEE Press，2006：83－92.

[9] 王霞. RFID 数据存储和管理技术综述[J]. 计算机应用与软件，2008，24(12)：175－176.

[10] 陈竹西，孙艳，胡孔法，等. 基于路径编码的 RFID 数据压缩技术研究[J]. 扬州大学学报：自然科学版，2008，11(2)：53－56.

[11] CHAWATHE S，KRISHNAMURTHY V，RAMACHANDRAN S，*et al.* Managing RFID data [C]// Proceedings of the 30th Very Large Data Bases Conference. Piscataway，NJ：IEEE Press，2004：1189－1195.

[12] WANG F S，LIU P Y. Temporal management of RFID data [C]// Proceedings of the 31st International Conference on Very Large Data Bases. New York：ACM Press，2005：1128－1139.

[13] GONZALEZ H，HAN J W，LI X L. FlowCube：Constructing RFID FlowCubes for multi-dimensional analysis of commodity flows [C]// VLDB'06：Proceedings of the 32nd International Conference on Very Large Data Bases. New York：ACM Press，2006：834－845.