一种相似重复记录检测算法的改进研究

戴 颖,李兴国,赵启飞
(合肥工业大学 管理学院,安徽 合肥 230009)

摘 要:相似重复记录检测是数据清洗领域中的一个重要方面。文中研究了在数据模式与匹配规则不变的前提下,数据集动态增加时近似重复记录的识别问题,针对基于聚类数算法精度不高、效率低下等问题提出一种改进算法。该算法运用等级法给属性赋予相应权重并约减属性,通过构造聚类树对相似记录进行聚类,增设了一个阈值以减少不必要的相似度比较次数,提高了算法的效率和准确率。最后通过实验证明了该算法的有效性,并提出了进一步的研究方向。

关键词:相似重复记录;增量式;聚类树;等级法
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2010)07-0013-04

Improved Method for Detecting Incremental Approximately Duplicate Records

DAI Ying, LI Xing-guo, ZHAO Qi-fei
(School of Management, Hefei University of Technology, Hefei 230009, China)

Abstract: Cleaning approximately duplicate records is an important task in data cleaning. Problems of detecting approximately duplicate records when the data set is dynamically increased on the assumption of stable data model and matching rules are studied. An improved method is proposed to deal with problems in the method based on clustering tree. The proposed method appoints proper weight to each field of the record and reduces attributes through using ranked-based weights method; clusters duplicate records by creating a clustering tree. To improve the efficiency of this method, a limen is added into the arithmetic. Finally, the validity of this method is proved by experiment and further research directions are proposed.

Key words: approximately duplicate record; incremental; clustering tree; ranked-based method

0 引 言

数据库中的相似重复记录,是指那些客观上表示现实世界同一实体,但由于在格式和拼写上有差异而导致数据库管理系统不能正确识别的记录[1]。这类数据产生的主要原因是数据仓库中数据的来源不单一。它们不仅占用了数据仓库的空间,还降低了数据处理结果的正确性,因此对相似重复数据的清洗变得尤为重要。

现在对相似重复记录检测的研究很多。文献[2]提出了邻近排序法,先使用某个设定的键值对整个数据表进行排序,然后再对小范围的邻近记录进行检测找出相似重复记录。文献[3]提出了多趟邻近排序法,在多个键上对数据表进行排序并进行小范围的邻近记录检测,最后综合多次计算的结果。文献[4]提出了优先队列方法,将优先队列的思想运用于相似重复记录检测中。文献[5]采用等级法为记录各字段指定合适的权重,从而提高了相似重复记录的检测精度。

现有的大多数近似重复记录识别算法均以一个静态数据集为前提。如果系统接收到一个新的同构数据集,须将其与已处理的数据集进行合并,再对合并后的整个数据集进行处理,大部分时间都浪费在已处理过的数据集的重复计算上。如果数据模式和用于匹配记录的规则不变,仅需观察最近到达的增量数据,这是增量式重复记录识别处理的核心。文献[6]提出了一种基于优先队列的增量式重复记录识别算法,用优先队列算法对数据集进行聚类,提取每个聚类的特征记录组成特征记录集,将特征记录集与新进入的数据集拼接,再使用优先队列算法对之处理,如此循环。该方法能够处理增量式的数据集,但效率和精度都不高。文献[7]用了类似的方法。文献[8]提出了一种基于聚类树的增量式数据清洗算法,通过构建聚类树先对记录

收稿日期:2009-10-23;**修回日期**:2010-02-23
基金项目:国家自然科学基金项目(70871033)
作者简介:戴 颖(1985-),女,江苏扬中人,硕士研究生,研究方向为数据清洗、项目管理;李兴国,教授,研究方向为信息管理、企业管理及其信息化建设。

进行分区，然后在划分的区域内进行相似度的计算以识别出近似重复记录，从而完成了增量式相似重复记录的检测。该算法能较好地完成对增量式数据库的相似重复记录检测，但是在精确性和效率上还有可改进之处。

1 原算法分析及改进措施

1)基于聚类树的增量式数据清洗算法分为属性约减和构造聚类树两个阶段，阶段一是通过利用公式 $SGF(a, A, D) = H(D \mid A) - H(D \mid A \cup \{a\})$（其中 A 是属性集，a 是属性，D 是决策属性）提取出对算法有重要影响的分区属性和判定属性。阶段二构造聚类树的主要过程为：

(1) 置根节点为空，即：置根节点为空（$T_0 = \varnothing$）。

(2) 按属性顺序读入第一条记录。

(3) 读入下一条记录，判断读入的记录是否为最后一条，是转(9)，否读入记录的第一个分区属性值。

(4) 将读入的属性值与聚类树相应层的节点的代表值进行相似度比较，若有代表值与之相似则转(5)；若没有代表值与之相似则转(6)。

(5) 将属性值写入相似代表值所在节点转(7)。

(6) 将属性值写入新建节点，剩下属性值分别放入新建节点，且上一个节点作为下一个节点的根节点。

(7) 判断是否是分区属性最后一个，是则转(8)；不是则读入下一个分区属性值转(4)。

(8) 读入判定属性值，将判定属性值放入聚类树的叶子节点，转(3)。

(9) 对同一分区的判定属性值进行相似度比较，一半以上的判定属性相似则写入相似记录集。

主要过程图如图 1 所示。

2)该算法能较好地完成对增量式数据库的相似重复记录检测，但是存在一些问题需改进。对这些问题的描述及改进如下：

(1)该算法对属性的顺序有很大依赖性。例如，对一个数据集属性约减后剩下五个属性（姓名，年龄，性别，出生年月，家庭住址）。设家庭住址这个属性的重要性比较低，如果将它放在聚类树的第一层，在构造聚类树的时候，某个记录若家庭住址因错误输入等原因被认为与另一个记录不相似，而其他四个属性值均相似，最后却被判定为不相似，这样的错误很容易产生。因此，应该在构造聚类树之前将约减后的属性按权重由大到小重新排列。原论文中属性约减时用到的公式 $SGF(a, A, D) = H(D \mid A) - H(D \mid A \cup \{a\})$（其中 A 是属性集，a 是属性，D 是决策属性）不能准确给出每个属性的权重，文中采用等级法[9]给每个属性设定权重，约减掉权重小于阈值的属性，将剩下的属性按权重由大到小重新排列。这样，在构造聚类树的过程中，权重大的属性自然比权重小的属性先接受相似度比较，减少了上面提到的错误。

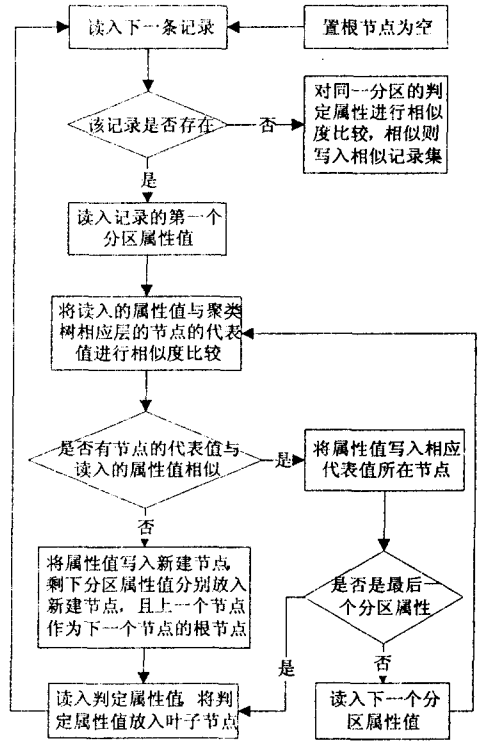

图 1 原算法主要过程图

(2)两条记录如果是相似的，原算法要到最后一步即对判定属性进行相似度比较之后才能得出结论。判断两条记录相似的最基本匹配规则是两条记录的相似度（记录各属性相似度与其权重乘积之和），超过一定阈值则视为相似。根据这一匹配规则，文中增设记录相似度阈值 H，在判断属性是否相似之前，先判断记录的相似度阈值是否超过 H，超过的话即认为两条记录相似，剩下的属性将不必再进行相似度比较，这样就减少了相似度比较次数。

(3)原论文中将属性分为分区属性和判定属性，先用分区属性对记录进行分区，再用判定属性判断记录的相似性。算法止于对判定属性进行相似度比较，而文中的改进算法止于记录相似度大于阈值 H，不需要用到判定属性。因此文中不区分分区属性和判定属性。

2 改进后的算法过程

阶段一用等级法给每个属性设定权重，约减掉权重小于阈值的属性，将剩下的属性按权重由大到小重新排列。阶段二为构造聚类树，主要过程如下：

(1) 置根节点为空即：置根节点为空（$T_0 = \varnothing$）

(2) 按属性顺序读入第一条记录。

(3) 读入下一条记录,判断读入的记录是否为最后一条,是算法终止;否读入记录的第一个属性值。

(4) 将读入的属性值与聚类树相应层的节点的代表值进行相似度比较,若有代表值与之相似则转(5);若没有代表值与之相似则转(6)。

(5) 将属性值写入相似代表值所在节点转(7)。

(6) 将属性值写入新建节点,剩下属性值分别放入新建节点,且上一个节点作为下一个节点的根节点。

(7) 判断是否是最后一个属性,是则转(3);不是则读入下一个属性值转(8)。

(8) 计算此记录与比较记录到现在比较属性为止的记录相似度,若相似度大于阈值 H 则表示两条记录相似,将其放入相似记录集,转(3),若相似度不大于阈值 H 则转(4)。

主要过程图如图 2 所示。

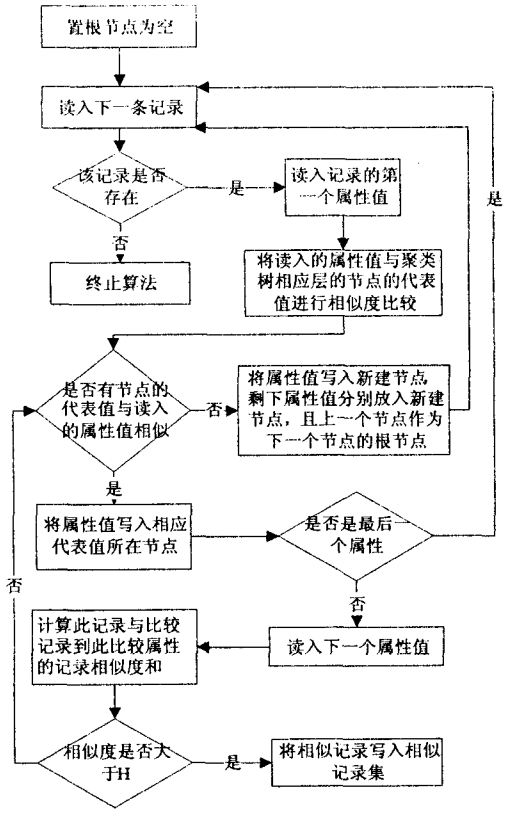

图 2　改进算法的主要过程图

3　改进后的算法描述

本改进算法与原算法一样,通过构造聚类树来实现相似重复记录的检测。聚类树中的节点(除首节点),表示按属性对记录划分分区,用三元组(data, m, LS)表示,data 是优先队列的代表值,m 是优先队列的长度,LS 是一个优先队列。这个优先队列用于储存相似属性的值,每个元素都是一个二元组(value, point),value 是属性的值,point 指向下一个子节点。假定记录的重复具有传递性,即若 R1 与 R2 相似,R2 与 R3 相似,则 R1 与 R3 相似,以一条记录来代表整个优先队列,选择最新进入队列的记录来代表这个优先队列。

给定一个记录集 D,内有 s 个属性 n 条记录。先用等级法给每个属性划分等级,并计算出权重,留下权重大于给定阈值的属性,假设留下的属性有 m 个。再将这 m 个属性按权重由大到小的顺序排列,然后将记录集以权重最大的属性为关键字重新排序。接下来就对记录集 D 用构造聚类树的方法进行相似重复记录检测。设 J_{pq} 为聚类树第 p 层第 q 个节点,A_{nm} 表示第 n 条记录第 m 个属性的值。构造聚类树的过程为:设聚类树的根节点为空即 $J_0=\varnothing$,读入第一条记录,将 A_{11} 放入 J_{11},其中 $J_{11}.data=A_{11}$,$J_{11}.m=1$,$J_{11}.LS[0].value=A_{11}$,$A_{12}$ 放入 J_{21} 中,其中 $J_{21}.data=A_{12}$,$J_{21}.m=1$,$J_{21}.LS[0].value=A_{12}$,将 J_{21} 作为 J_{11} 的子节点(即 $J_{11}.LS[0].point=J_{21}$),其余属性所在的节点分别作为上一属性所在的节点的子节点,按这个方法将第一条记录的其他属性值全部放入聚类树中。接着,按排序顺序依次读入剩下的 $n-1$ 条记录。这里以第 i 条记录第 j 个属性值 A_{ij} 为例。若 $j=1$,A_{i1} 与聚类树第一层所有节点的代表值进行相似度比较,若节点 J_{1k} 的代表值与 A_{i1} 的相似度大于阈值 T,则将 A_{i1} 写入节点 J_{1k} 即 $J_{1k}.data=A_{i1}$,$J_{1k}.m=m+1$,$J_{1k}.LS[m].value=A_{i1}$,若没有节点的代表值与 A_{i1} 的相似度大于阈值 T_1,则新建节点,将 A_{i1} 放入。如果 $j\neq1$,则将 A_{ij} 与以 $A_{i(j-1)}$ 为根节点的所有节点的代表值比较。若 A_{ij} 与 $J_{jk}.data$ 相似度乘以 A_{ij} 的属性权重与第 i 条记录 A_{ij} 前面所有属性的比较相似度和相应权重乘积的和大于设定阈值 T_2,说明 A_{ij} 所属记录与 $J_{jk}.data$ 所属记录相似,将其放入"近似重复记录集 M",后续属性不再比较,否则判断 A_{ij} 与以 $A_{i(j-1)}$ 为根节点的所有节点代表值的相似度,若与某节点代表值相似度大于 T_1,则将 A_{ij} 放入该节点,若与任何代表值的相似度都不大于 T_1,则新建节点将其放入。最后得到聚类树如图 3 所示。

4　实验结果

本实验在常配微机上,利用 Java 作为开发语言,进行了模拟实验。实验数据为网站 http://archive.ics.uci.edu/网站上的 adult 表,表中有 30000 条记录,14 个属性,实验中两个属性值之间的相似度比较使用的是 Smith－Waterman 算法[10]。分别使用原算法和本改进算法进行数据清洗,并对结果进行了比较。实

验结果见表 1,表中 T 表示算法时间开销,S 表示查出的相似记录数,R 为查出的正确的相似记录数,P 表示准确率,下标 1 和 2 分别表示原算法和改进算法。

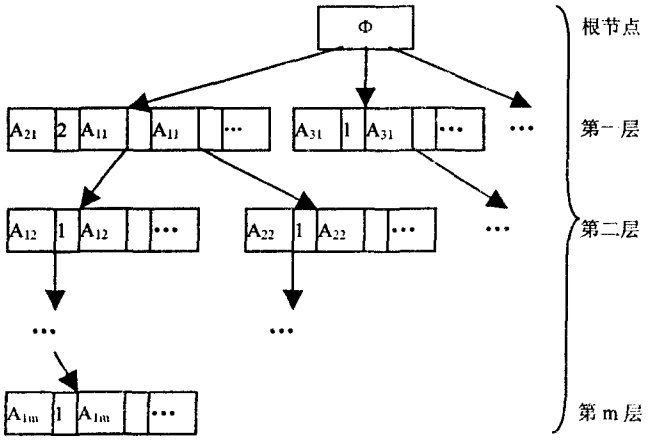

图 3 构造的聚类树[8]

表 1 实验结果表

记录数	T_1(ms)	S_1	R_1	P_1(%)
10000	35281	510	428	83.9
20000	145625	1315	1057	80.4
30000	314219	1574	1316	83.6
记录数	T_2(ms)	S_2	R_2	P_2(%)
10000	1891	524	452	86.1
20000	11172	1247	1014	81.3
30000	13937	1621	1379	85.1

从实验结果可知,与原算法相比,文中算法的总时间开销减少了很多,准确率略有提升。实验表明,本改进算法与基于聚类树的清洗算法相比优点为:避免了因属性排序不当使有些相似记录未被识别的情况,提高了精度;减少了一些不必要的相似度比较过程,在一定程度上提升了效率。但是本算法中优先队列的使用降低了结果的精度,这将是下一步的研究方向。

5 结束语

介绍并分析了数据清洗领域中针对增量式数据库的基于聚类树的相似重复记录检测算法,针对该算存在的不足提出了改进算法,并用实验证明了改进算法的有效性。

参考文献:

[1] 邱越峰,田增平,季文贝斌,等.一种高效的检测相似重复记录的方法[J].计算机学报,2001,24(1):69-77.

[2] Hernandez M, Stolfo S. The Merge/Purge Problem for Large Databases[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, USA: ACM,1995:127-138.

[3] Hernandez M,Stolfo S. Real-world data is dirty:data cleansing and the merge/purge problem [J]. Data Mining and Knowledge Discovery,1998,2(1):9-37.

[4] Monge A E. An adaptive and efficient algorithm for detecting approximately duplicate database records[EB/OL]. 2003-03. http://citeseer.nj.nec.com/monge00adaptive.html.

[5] 陈 伟,王 昊,朱文明.一种提高相似重复记录检测精度的方法[J].计算机应用与软件,2006,23(10):29-30.

[6] 佘春红.基于优先队列的增量式重复记录识别[J].计算机应用,2003,23(9):61-63.

[7] 许向阳,佘春红.近似重复记录的增量式识别算法[J].计算机工程与应用,2003,39(12):191-193.

[8] 刘 芳,何 飞.一种基于聚类树的增量式数据清洗算法[J].华中科技大学学报,2005,33(3):46-48.

[9] 李星毅,包从剑,施化吉.数据仓库中的相似重复记录检测方法[J].电子科技大学学报,2007,36:1273-1277.

[10] Smith T F, Waterman M S. Identification of common molecular subsequences[J]. Journal of Molecular Biology, 1981, 2(3):195-197.

(上接第 12 页)

techniques[C]//International Workshop on Passive and Active Network Meaurement. Berlin:Springer,2004:22-41.

[6] 孙海波.基于协议行为特征的协议识别方法[C]//全国网络与信息安全技术研讨会论文集(上册).北京:中国通信学会,2007:245-251.

[7] 张春男,龙 翔,高小鹏.面向应用的网络流控系统的设计与实现[J].微计算机信息,2008,24(10-3):88-90.

[8] 谭 炜,吴 健.基于半监督学习的 P2P 协议识别[J].计算机工程与设计,2009,30(2):291-293.

[9] 温 超,郑雪峰,戚 翔,等.基于流量分析的 P2P 协议识别方法的研究[J].微计算机应用,2007,28(7):714-717.

[10] Woo Kyoung-Gu, Lee Jeong-Hoon, Kim Myoung-Ho, et al. FINDIT:a fast and intelligent subspace clustering algorithm using dimension voting[J]. Information and software technology,2004,46:255-271.

[11] Basu S, Bilenko M, Mooney R J. A Probabilistic Framework for Semi-Supervised Clustering[C]//Proceeding of Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004). Seattle, WA, USA: ACM,2004:59-68.

[12] Grira N, Crucianu M, Boujemaa N. Active semi-supervised fuzzy clustering[J]. The journal of the pattern recognition society,2008,41:1834-1844.

[13] Basu S, Banerjee A, Mooney R. Semi-supervised Clustering by Seeding[C]//Proceedings of the 19th International Conference on Machine Learning (ICML-2002). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.,2002.