数据仓库中的相似重复记录检测方法

李星毅[1,2]，包从剑[2]，施化吉[2]

(1.北京交通大学电子信息学院 北京 海淀区 10004；2. 江苏大学计算机科学与通信工程学院 江苏 镇江 212013)

【摘要】针对检测和消除数据仓库中的相似重复记录问题，提出了数据仓库中的相似重复记录检测方法。该方法先通过等级法计算每个字段的权值；然后，按照分组思想，选择关键字段或字段某些位将大数据集分割成许多不相交的小数据集；最后，在各个小数据集中检测和消除相似重复记录，为避免漏查，再选择其他关键字段或字段某些位重复多次检测。理论分析和实验表明，该方法不仅具有好的检测精度，而且具有很好的时间效率,能够有效地解决大数据量的相似重复记录检测问题。

关 键 词 相似重复记录； 数据仓库； 分组； 等级法； 数据加权
中图分类号 TP311 文献标识码 A

A Method for Detecting Approximately Duplicate Database Records in Data Warehouse

LI Xing-yi[1,2]，BAO Cong-jian[2]，SHI Hua-ji[2]

(1. School of Electronics and Information Engineering, Beijing Jiaotong University Haidian Beijing 100044;
2. School of Computer Science and Telecommunications Engineering, Jiangsu University Zhenjiang Jiangsu 212013)

Abstract Detecting and eliminating approximately duplicated records is one of the main problems needed to be solved for data mining and data quality improvement. An algorithm for detecting approximately duplicated database records is presented based on rank group. Firstly, each property of the data is endowed with certain weight according rank-based weights method. Secondly, in term of group thought, large data sets are divided into many non-intersect small data sets. Finally, approximately duplicated records are detected and eliminated in each small data set. To avoid missing, the above steps can be repeated. The theory analysis and experiment show that this algorithm has a good detecting precision better efficiency of time, and therefore is an effective approach to solve approximately duplicate records of massive data.

Key words approximately duplicated records; data warehouse; group ; rank method; weighted data

数据质量是影响数据挖掘效果的关键因素之一。为提高被挖掘数据源的数据质量，数据清理变得很重要。不同数据源数据集成的一个重要问题是语法上相同或相似的不同记录可能代表现实世界中的同一实体，因此相似重复记录的检测成为数据清理中的一个关键环节。

面对海量数据的数据仓库，传统相似重复记录检测主要采用距离函数模型[1]，基于q-gram 算法[2]、“ 排序&合并 ”的方法[3-4]、以及标准的字符串度量方法[5]。用传统的相似重复记录检测方法在海量数据库中查找相似重复记录，会涉及很大的时间复杂度和空间复杂度，并且排序时由于字符位置敏感性并不能保证相似的记录排在邻近的位置，导致基于q-gram 算法或“ 滑动窗口 ”聚类[3]或“ 优先队列聚类 ”[4]不能取得很好的效果。为克服上述缺陷，本文提出数据仓库中的相似重复记录检测方法，其主要特征是：(1) 根据等级法计算每个字段的权值及设计多趟查找方法，目的为提高检测精度。(2) 分组的目的为提高时间效率。

1 基于等级分组的重复记录检测方法

1.1 基本定义

设数据集合$X=\{x_1,x_2,\cdots,x_n\}$，字段向量$\boldsymbol{F}=\{F_1,F_2,\cdots,F_p\}$，$F_k$ 表示数据表第k个字段，对于任意记录$x_i=\{x_{i1},x_{i2},\cdots,x_{ip}\}$，其中$1\le i\le n$，$x_{ip}$ 表示记录x_i第p维的值。用W_k表示字段F_k的权值，代表字段在对象中的重要程度，称为属性的权重，权重向量$\boldsymbol{W}=\{W_1,W_2,\cdots,W_p\}$。

定义 1 T_{i_k} 是第i个操作用户为字段 F_k 所指定的等级(从1开始，使用连续正整数表示等级，1表示

收稿日期：2007 – 09 – 07
基金项目：国家火炬计划项目(2004EB33006[0])；江苏省高校自然科学指导性计划项目(05JKD520050)
作者简介：李星毅(1969 –)，男，博士，副教授，主要从事数据挖掘、空间数据库、交通信息系统和控制方面的研究.

最高等级，数值越大，等级越低)，T_k 表示第k个字段的最终统一等级，$k\in\{1, 2,\cdots, p\}$，$i\in\{1, 2, \cdots, N\}$，T_k 字段的最终统一等级：

$$T_k = \left\lfloor (\sum_{i=1}^{N} T_{ik}) / N \right\rfloor \tag{1}$$

定义 2 T_k 表示 F_k 最终统一的等级，T表示最低等级(即数值最大的等级)，$k\in\{1, 2,\cdots, p\}$，如果任意两字段的最终统一等级不相同，那么$T=p$，采用RC(Rank-Centroid)转换方法[6]，字段F_k的权重可以表示为：

$$W_k(\mathrm{RC}) = \frac{1}{T}\sum_{t=T_k}^{T}\frac{1}{t} \tag{2}$$

如果存在两个或两个以上的字段，它们的最终统一等级相同，则式(2)应变成：

$$W_k = W_k(\mathrm{RC}) / W' \tag{3}$$

式中 $W' = \sum_{k=1}^{p} W_k(\mathrm{RC})$。

定义 3 对任意记录x_i与x_j，它们的第k维字段为x_{ik}与x_{jk}，x_{ik}与x_{jk}的字段相似度[7]：

$$\mathrm{SimField}(x_{ik}, x_{jk}) = \frac{\sum_{t=1}^{q}\max(\mathrm{score}(a, x_{jk}))}{|x_{ik}|} \tag{4}$$

式中 score(a，x_{jk})表示x_{ik}中的原子串a与x_{jk}中的每个原子串匹配的分值，0≤score(a, x_{jk})≤1，如上述所定义；$|x_{ik}|$表示x_{ik}的长度；q表示x_{ik}的原子串的数量。

定义 4 给定两条记录x_i和x_j ，则x_i和x_j的记录相似度：

$$\mathrm{SimRecord}(x_i, x_j) = \sum_{k=1}^{p}\mathrm{SimField}(x_{ik}, x_{jk})W_k \tag{5}$$

定义 5 X_a代表原数据集实际的重复记录集合，X_b代表识别出来的重复记录集合，查准率是正确识别出来的重复记录占识别出作为重复记录的比率，则查准率表示为：

$$\mathrm{ScanAccuracy}(X)=|X_a\cap X_b|/|X_b| \tag{6}$$

查全率是正确识别出来的重复记录占数据集中实际的重复记录比率，则查全率表示为：

$$\mathrm{ScanComplete}(X)=|X_a\cap X_b|/|X_a| \tag{7}$$

1.2 基本思想

数据库中记录的属性描述了实体的特征，各个属性决定实体身份的重要程度为权重，这有必要计算每个属性的权值。为提高检测精度，本文设计等级法计算权值。面对大数据量的数据仓库，为提高检测效率，先对大数据集作一定处理。本文运用分组思想，即把大的数据集分割成很多不相交的小数据集，再在小数据集中分别查找重复记录。为提高检测精度，实行多趟查找，基本算法描述如下：

(1) 用户根据实际经验给每个字段指定等级，系统根据规则计算每个字段的最终统一等级，并把最终统一等级转化为相应的权值。

(2) 选择关键字段甚至关键字段的某些位,把大数据集分割成不相交的小数据集。

(3) 分别在各个小数据集中检测相似重复记录，并进行处理。

(4) 选择另外关键字段或字段的某些位，重复第二和第三步，实行多趟查找，避免漏查。

1.3 基本步骤

1.3.1 等级法计算权值

对于记录间字段，如果它们具有的相同属性越多且相同属性的权值越大，则它们越相似。本文采用RC等级转换法[6]计算各字段的权重。

等级法思想：首先各用户根据实际经验为各个字段指定等级；然后根据式(1)计算各字段的最终统一等级；最后根据式(2)或式(3)再计算它们相应的权重。表1是字段等级表，F_k 表示第k个字段；T_{ik}是第i个操作用户为字段F_k所指定的等级；T_k表示第k个字段最终统一的等级。

表**1** 字段等级表

字段名	用户指定等级						等级
	U_1	U_2	…	U_i	…	U_N	
F_1	T_{11}	T_{21}	…	T_{i1}	…	T_{N1}	T_1
F_2	T_{12}	T_{22}	…	T_{i2}	…	T_{N2}	T_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
F_k	T_{1k}	T_{2k}	…	T_{ik}	…	T_{Nk}	T_k
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
F_p	T_{1p}	T_{2p}	…	T_{ip}	…	T_{Np}	T_p

权值算法如下：

输入：字段向量$\boldsymbol{F}$；

输出：字段权值向量$\boldsymbol{W}$；

WeightProcess($\boldsymbol{F}$)

{For (k=1;k<p;k++) T_k=0; //等级初始化，p表示字段数量

For (i=1;i<N;i++) //N表示用户数量

{For (k=1;k<p;k++)

{第i个操作用户根据经验指定字段F_k的等级T_{ik};} }

For (k=1;k<p;k++) 根据式(1)计算T_k；

For (k=1;k<p;k++) 根据式(2)或式(3)计算W_k ；

Retrun ($\boldsymbol{W}$);}

1.3.2　数据分组

面对海量数据库，按照传统方法查找相似重复记录要有大量磁盘空间及内存空间，并且运算时间复杂度相当大，则需采取一种方法来避免这种缺陷。本文根据分组思想，把大的数据集分割成不相交的小数据集，然后在各个小数据集中查找相似重复记录，样本数据如表2所示。

基本思想：

(1) 选择能明显区别记录间特征的字段或字段的特定几位，把大数据集分割成很多个不相交的小数据集合。如取姓名的姓氏，《中国家谱总目》共收录了643个姓氏，把大数据集分割成643个不相交的集合。

(2) 分割后，某些数据集仍然十分庞大，则选择另外关键字段或字段的特定几位，对这些数据集再次分割。如根据全国人口普查知李姓占13%等，这就可能造成李姓数据集仍然十分庞大，则对这些数据集进行二次分割，如取出生日期的年月，把比较大的数据集再次分割成几千个小数据集。

(3) 如有些数据集仍很大，可重复第二步，直到数据集分割比较合理为止。

但简单根据一个关键字段或关键字段的部分对大数据集进行划分可能会产生问题，如根据表2的数据，假定以姓名中的姓氏简单划分，则数据集中很明显的相似记录“ 吴宏明… ”“ 胡宏明… ”会被划分到两个数据集。为了解决这个问题，引入多趟查找技术，即把数据集划分成合理的小数据集，并查找相似重复记录，这一轮结束后，再选定另外关键字段或关键字段某些位，重新对数据集进行划分，并查找相似重复记录，根据实际情况决定是否进行下一轮划分查找，直至结果满意。设X_t为第t个数据集，$X_t \subset X$，SubArea算法如下：

输入：数据集X；

输出：小数据集$X_1, X_2,\cdots, X_t$；

SubArea(数据集X，阀值a)

{选择关键字段F_j或关键字段某些位F_{jh}；

While (从数据集X中依次取记录x_i)

{根据x_i的F_j或F_{jh}值，找到相应数据集X_k;

X_k=X_k∪{x_i};}

While (任意数据集X_k)

{If (|X_k|>a) then SubArea(数据集X_k,阈值a)；}

thenReturn(小数据集X_1,X_2,⋯,X_t); }

表**2**　样本数据表

姓　名	出生日期	工作单位	住　址
黄建华	1976-04-07	浙江大学	东吴路18号
刘胜利	1981-11-04	浙江百利达有限公司	秧田路136号
黄剑华	1976-04-07	浙大	东吴路18号
吴宏明	1964-08-09	浙江第一医院	中山路5号
⋮	⋮	⋮	⋮
杨小兵	19791208	浙江圣达有限公司	解放路67号
胡宏明	19640809	浙江第一医院	中山路5号

1.3.3　字段匹配算法

对于式(4)，字段类型会有多种类型，先进行字符化处理。字符化后，字段组成可能会分成三类情况：(1) 西文和汉字混合。(2) 西文组成(包含数字等)。(3) 汉字组成。

对于第一类情况可以采用分割的方法，把混合字段分割成西文和中文字段。如两地址字段：杭州市东吴路18号A座301室和杭州市西湖区东吴路18号A座309室，自然分割后：杭州市东吴路号座室－杭州市西湖区东吴路号座室，18A301－18A309，再分别对它们进行比较。

对于第二类情况的分割现有技术比较成熟，假设对任意记录x_i与x_j，它们的第k维字段分别为西文字段x_{ik}与x_{jk}，它们的相似函数SimFieldEnglish(x_{ik}，x_{jk})，定义参照式(4)，按照西文规律分别把x_{ik}和x_{jk}分割成合适的原子串，再进行两字段相似度比较，函数内容参照文献[7-8]或其他资料，也可以直接调用高级语言的相关函数。

对于第三类情况，现有的技术也比较成熟，假设对任意记录x_i与x_j，它们的第k维字段分别为中文字段x_{ik}与x_{jk}，它们的相似函数为SimFieldChina (x_{ik},x_{jk})，定义参照式(4)，按照汉文规律分别把x_{ik}和x_{jk}分割成合适的原子串，再进行两字段间相似度比较，函数内容参照文献[7-9]或其他资料，也可以直接调用高级语言相关函数。

函数内容主要考虑两种情况：(1) 输入错误。(2) 缩写。对于第一类情况，输入的错字可能主要是同音、近音字或字型相似的字，因此，可以按照此规律将它们组成一个“ 相似汉字表 ”，供汉字比较时使用。对于第二类情况，如表2中，“ 浙 ”表示浙江缩写，“ 大 ”表示大学缩写等，根据汉字的各种缩写习惯和词组构成规律组成一张字词连环图[10]，通过查找字词连环图来判断它们是否是习惯用语的缩写。

输入：任意两字段x_{ik}和x_{jk}；
输出：两字段相似分值；
SimField(x_{ik}, x_{jk})
{If (字段x_{ik}或x_{jk}非字符类型) then
分别对字段x_{ik}和x_{jk}字符化处理；
If (字段x_{ik}或x_{jk}由中西文混合组成) then
{把x_{ik}分割成西文x_{ik1}和中文x_{ik2}两部分；
把x_{jk}分割成西文x_{jk1}和中文x_{jk2}两部分；
return ((SimFieldEnglish(x_{ik1}, x_{jk1})+
SimFieldChina (x_{ik2}, x_{jk2}))/2);}
Else If (字段x_{ik}和x_{jk}由西文组成) then
{return SimFieldEnglish(x_{ik}, x_{jk});}
Else If (字段x_{ik}和x_{jk}由汉字组成) then
{return SimFieldChina(x_{ik}, x_{jk});} }

1.3.4 算法总描述

输入：数据集X，字段向量$\boldsymbol{F}$，记录间相似度阈值b；
输出：重复数据；
SearchDupRecord(X,$\boldsymbol{F}$,b)
{WeightProcess($\boldsymbol{F}$);
flag–mulser=true;
给每一记录加字段flagdup=0； //标注记录是否与其他记录相似重复，0表示不重复
While (flag–mulser) //判断是否多趟查找
{确定分割后的数据集最大记录数的阈值a；
SubArea(X,a)； //对大数据集进行分组
For (k=1;$k\leq t$;k++) //t表示数据集的数量
while (依次取数据集X_k)
while (从数据集X_k中按序取记录x_i)
while (从数据集X_k中i位置后按序取记录x_j)
If (SimRecord(x_i, x_j)>阈值b) then
If ((x_i→flagdup=0) and (x_j→flagdup=0))then
{ x_i→flagdup=i; x_j →flagdup=i;}
Else If (x_i→ flagdup=0) then
x_i → flagdup= x_j → flagdup;
Else If (x_j → flagdup=0) then
x_j → flagdup= x_i → flagdup;
If (满足多趟查找结束条件) then
flag–mulser=false;}
根据flagdup值按一定规则输出重复记录;}

2 实 验

文献[4]提出的优先队列方法(Priority Queue Strategy，PQS)是传统算法中比较有代表性且较优的算法，具体策略是抽取一个或多个字段构成关键字进行排序，然后寻找数据库中各条记录在一个长度固定的子集队列中的匹配记录，采用类似最久未使用替换算法来控制队列的长度。

从数据库的角度来看，由于数据库记录是按一定的模式存放的，因此文献[4]提出的优先队列方法相当合理，本文选择文献[4]提出的算法作为本文提出算法的参照物。

为便于处理，数据仓库中的相似重复记录检测方法作为方法一，本文将其简称为等级分组，文献[4]的PQS方法作为方法二。为比较这两种方法优劣，本文用实验对其进行说明。

实验环境：P4 1.6 GHz CPU，物理内存512 MB，硬盘空间60 GB，操作系统Windows XP，数据库软件为SQL SERVER 2000，程序用JAVA语言编写。

实验数据来源于某市常住人口数据，属性有41项，基本数据格式见表2所示，数据量分别为：38.1、75.3、100.8、133.6万人，通过软件和人工等方式对以上数据分别处理成有0.32、0.78、1.25、1.67万人相似重复记录，用上述两种方法检测相似重复记录，采用有查准率、查全率和方法运行时间三个测试标准。

2.1 查准率和查全率对比

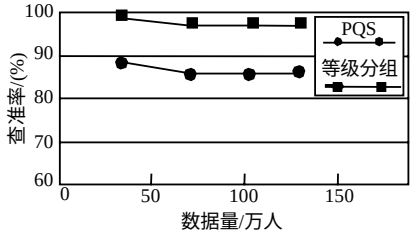

图1 等级分组和PQS查准率对比图

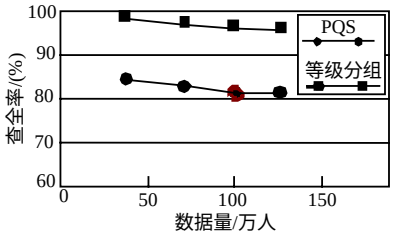

图2 等级分组和PQS查全率对比图

先进行两种方法的查准率和查全率比较，从图1和图2中可以看出，方法二的查准率和查全率远低于方法一，且随着数据量增大，方法二的查准率和查全率会下降，而方法一始终可以保持很高的查准率和查全率，这主要是因为方法二由于排序对字符位置敏感，不能保证将相似记录排在邻近位置，而方法一根据等级法给不同字段赋予不同权值，更能反

映现实实体特征，并利用多趟分组查找可以有效地提高精度，用方法一查找相似重复记录时就有更高的查准率和查全率。

2.2　时间对比

图3显示了两种方法在四个不同数据量的运行时间，方法一的运行时间分别为9、18、28、39 min；方法二的运行时间分别为10、27、52、99 min。从以上分析得知，方法一比方法二快很多。方法二的主要制约因素为外排序操作，它的时间复杂度大于$O(n\lg n)$，而方法一的数据分组的时间复杂度是为$O(n)$，分组后在各个小数据集中进行相似重复记录查找，总的时间复杂度接近$O(n)$，如采用分布式并行计算，时间复杂度更底。

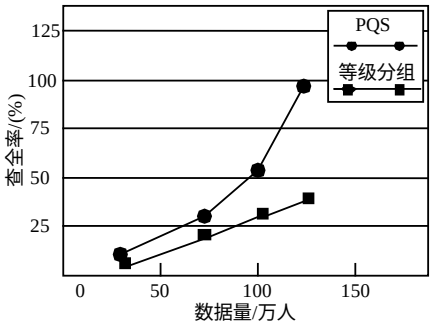

图3　等级分组和PQS运行时间对比

通过大量实验，本文得出结论：算法的复杂度、精度和理论分析是一致的。总体上说，数据仓库中的相似重复记录检测方法有以下显著优点：

(1) 算法复杂度仅为$O(n)$。

(2) 处理海量数据的能力相当强。

(3) 能有效地提高检测精度。

3　结 束 语

本文提出数据仓库中的相似重复记录检测方法来实现相似重复记录的检测。该方法的优点是时间复杂度小，检测精度较高，有效地解决了大数据量的相似重复记录识别问题。该方法在中西文字符集具有很强的通用性，可以很好地运用到实践生产中。

参 考 文 献

[1] BILENKO M, MOONEY R. Adaptive name matching in information integration[J]. IEEE Intelligent Systems，2003, 18(5): 16-23.

[2] LIANG J, CHEN L, MEHROTRA S. Efficient record linkage in large data sets[C]//Proc 8th Int Conf on Database Systems for Advanced Applications. Kyoto:[s.n.], 2003: 137-148

[3] MAURCIO H, SALVATORE J S. The merge/purge problem for large databases[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. ACM New York:[s. n.]1995: 127- 138.

[4] MONGE A. An adaptive and efficient algorithm for detecting approximately duplicate database records[EB/OL]. http://citeseer.ist.psu.edu/monge00adaptive.html, 2007-09-02.

[5] MINTON S, NANJO C, KNOBLOCK C, et al. A heterogeneous field matching method for record linkage[C]//Proceedings of the 5th International Conference on Data Mining (ICDM2005). Washington: IEEE Computer Society, 2005: 314-321.

[6] DEY D, SARKAR S, DE P. A distance-based approach to entity reconciliation in heterogeneous databases[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(3): 567-582.

[7] 张　永, 迟忠先. 位置编码在数据仓库ETL中的应用[J]. 计算机工程, 2007, 33(1): 50-52.

[8] MONG L, HONGJUN L. Cleansing data for mining and warehousing[C]//Int Conf on Database and Expert Systems Applications. Florence: [s.n.], 1999: 751-760.

[9] 程国达, 苏杭丽. 一种检测汉语相似重复记录的有效方法[J]. 计算机应用, 2005, 25(6): 1361-1365.

[10] 李先国, 梁　涌. 一种高效的适用于字词检索的数据结构[J]. 微电子学与计算机, 2006, 23(12): 157-160.

编　辑　张　俊