

文章编号:1001-9081(2006)04-0880-03

## 数据仓库 ETL 中相似重复记录的检测方法及应用

张永<sup>1,2</sup>,迟忠先<sup>2</sup>,闫德勤<sup>1</sup>

(1. 辽宁师范大学 计算机系,辽宁 大连 116029; 2. 大连理工大学 计算机系,辽宁 大连 116024)

(ayong\_zh@163.com)

**摘要:**检测和消除数据仓库中的相似重复记录是数据清洗和提高数据质量要解决的主要问题之一。将位置编码技术引入到数据仓库 ETL 中,提出了一种相似重复记录的检测算法 PCM(位置编码方法)。该算法不仅可以应用到西文字符集中,而且也成功的应用到了中文字符集中,实例表明该算法具有很好的效果。

**关键词:**位置编码;数据仓库;ETL;相似重复记录

**中图分类号:**TP311.13    **文献标识码:**A

### Approximately duplicated records examining method and its application in ETL of data warehouse

ZHANG Yong<sup>1,2</sup>, CHI Zhong-xian<sup>2</sup>, YAN De-qin<sup>1</sup>

(1. Department of Computer, Liaoning Normal University, Dalian Liaoning 116029, China;

2. Department of Computer Science and Engineering, Dalian University of Technology, Dalian Liaoning 116024, China)

**Abstract:** Examining and eliminating approximately duplicated records is one of main problems needed to solve for data cleaning and improving data quality. The position-coding technology to ETL of data warehouse was introduced, a novel examining algorithm named Position-Coding Method(PCM) of approximately duplicated records was presented. The algorithm was applied to Chinese character set, as well as Western character set. Experiment comparison with the previous work indicates that the method is effective.

**Key words:** position-coding; data warehouse; ETL; approximately duplicated records

## 0 引言

越来越多的部门将自己的数据集成在一起,构建数据仓库,从而更好地帮助管理者进行决策。然而由于许多异构数据的大量集成,数据仓库中的数据质量难以保证,反过来又影响了数据仓库中数据的查询,导致了决策支持的可靠性降低。当今世界中的数据库极易受到噪声数据、空缺数据和不一致性数据的侵扰,因此在数据仓库的构建中,ETL 是相当重要的一个环节。ETL 是数据采集(Extraction)、数据转换(Transformation)、数据载入>Loading 的简称。ETL 过程就是从数据源采集所需数据,经过数据转换和清洗等预处理过程,最终按照预先定义好的数据仓库数据模型,将数据加载到数据仓库中。

数据预处理主要包括数据清洗、数据集成和变换、数据归约等技术<sup>[1]</sup>,而检测和消除数据仓库中的相似重复记录是数据清洗和提高数据质量要解决的主要问题之一。所谓相似重复记录是指客观上表示现实世界中的同一实体,但由于表述方式不同或拼写问题而使 DBMS 不能识别其为重复的记录。这些重复的记录可能导致建立错误的数据挖掘模型,给后期数据的决策分析产生很大的影响。因此,在 ETL 中,判断两条记录是否相似重复尤为重要。

造成相似重复记录的原因一般可分为两类:其一是拼写错误,比如某数据表中包含三个字段:姓名,职业,出生日期,

则对于两条记录 R1 = { "Zhang Yong", "Teacher", "1975-06-06" }, R2 = { "Zhang Yong", "Teahcer", "1975-06-06" },我们可以认为 R2 中的 "Teahcer" 是拼写错误,他们是重复记录;其二是缩写所引起的等价错误,比如 "Dept" 和 "Department"。目前许多数据表中都包含了一些诸如单位名称、地址、场所位置、客户姓名等这样的字段信息。为了满足管理和决策上的需要,几乎每个行业或机构都以某种形式采集、处理和传播这些信息。在这种情况下,尤其是在数据仓库中进行查询,对数据的综合利用和共享就可能会产生一些问题。

对于相似重复记录,也已经有了一些研究<sup>[2~8]</sup>。但很多算法目前都是针对西文字符集,对于中文字符集的处理还有待进一步提高。本文首先分析了在相似重复记录检测方面的一些研究,指出了一些不足,然后针对西文字符集提出了一种新的检测方法,并对此做了一些改进,使该算法也能成功地应用到中文字符集中。同时对于匹配的阈值,也提出了一种动态的方法来确定。

## 1 相关工作

清除相似重复的记录可以针对两个数据集或者一个合并后的数据集。首先,需要识别出标识同一个实体的相似重复记录,即记录匹配过程。判定记录是否重复是通过比较记录对应的字符串之间的相似度来决定记录是否表示现实中同一实体。与领域无关的记录匹配方法主要思想是:利用记录间

收稿日期:2005-10-24;修订日期:2006-01-07    基金项目:辽宁省教育厅基金资助项目(2004C031)

**作者简介:**张永(1975-),男,四川阆中人,讲师,博士研究生,主要研究方向:数据仓库、数据挖掘与知识发现;  迟忠先(1939-),男,辽宁大连人,教授,博士生导师,主要研究方向:对象建模方法及应用、数据仓库及数据挖掘技术;  闫德勤(1962-),男,辽宁大连人,教授,博士,主要研究方向:人工智能,知识发现,粗集理论。

的文本相似度来判定两条记录是否相似,如果两条记录的文本相似度大于某个预先指定的阈值,那么则判定这两条记录是重复的,反之则不是<sup>[2~6]</sup>。记录的匹配方法更多的是以记录中的字段为基础进行的匹配,常见的方法有:编辑距离方法<sup>[7]</sup>,文本相似度度量方法<sup>[2]</sup>,基于 N-gram 的字符串匹配方法<sup>[5]</sup>和 Cosine 相似度方法<sup>[3,5]</sup>。

文献[3]中提出了 RFMA 算法,其基本思想是:两个字段  $A$  和  $B$  是匹配的,设定匹配分值为 1,当且仅当它们具有相同的原子串或者一个是另一个的缩写;否则匹配分值为 0。根据不同情况,可以将字段划分成若干个单词来处理,甚至可以将一个单词再划分成单个字母来处理。 $A$  与  $B$  匹配的相似度为:

$$\text{Sim}(A, B) = \frac{\sum_{i=1}^n \text{MAX(score}(a_i, B))}{|A|}$$

这里  $\text{score}(a_i, B)$  表示  $A$  中的原子串  $a_i$  在与  $B$  中的每个原子串匹配的分值, $0 \leq \text{score}(a_i, B) \leq 1$ ,如上述所定义; $|A|$  表示  $A$  的长度。

如果  $\text{Sim}(A, B)$  的值大于某个指定的阈值,则认为  $A$  和  $B$  是相似重复情况,可以进行下一步清理工作。

该方法简单,易于实现。但是该方法往往也引入了过多的错误匹配情况,比如:两个表示姓名的字串  $A = "John Johnson"$ , $B = "Johns Micheline"$ ,根据文献[3]的算法,首先将  $A$  和  $B$  按单词分解, $A$  中的子串"John"与  $B$  中的"Johns"匹配, $\text{score}("John", B) = 1$ ;同样  $\text{score}("Johnson", B) = 1$ 。从而  $\text{Sim}(A, B) = (1 + 1)/2 = 1$ ,即  $A$  和  $B$  相似度为 1。

## 2 相似重复记录的位置编码检测方法

为了判断一个数据表中的记录是否是相似重复的,我们可以先判断记录所拥有的各个属性是否匹配,或者说计算出他们的匹配程度,然后再考虑整个记录的相似匹配程度。下面描述的算法思想首先是针对属性级的。当计算出每个属性的匹配分值后,将这些匹配分值进行有权相加,根据预先设定的记录级匹配阈值来判断记录是否是相似重复的。当然,对于不同属性,考虑到属性在记录中的重要程度不一样,可以给属性赋不同的权值。

### 2.1 算法思想

通过上述的分析,本文仍利用相似度的概念,重新定义了相似度的计算方法,并提出了一种动态方法来确定匹配阈值。相似重复记录的位置编码检测方法如下。

第一步:将两个字段或字符串  $A$  和  $B$  按照基于单词的方式进行标记(划分)。为了便于后续步骤的解释,并假定  $A$  的标记个数(记为  $|A|$ )比  $B$  的标记个数(记为  $|B|$ )少。

第二步:如果  $|A| = 1$ ,将其按照基于字母的方式进一步标记。如  $A = "DSS"$ ,可以将  $A$  标记为  $A = \{"D", "S", "S"\}$ 。如果  $A$  中字母的个数恰好等于  $B$  中基于单词的标记个数,则将  $A$  中的每个字母与  $B$  中的每个单词的首字母比较,如果匹配,则表明  $A$  是  $B$  的缩写形式。比如对于上述的  $A$ ,若有  $B = "Decision Support System"$ ,则  $A$  与  $B$  匹配。

第三步: $A$  中的每个基于单词的标记  $a_i$ ,分别与  $B$  中的每个基于单词的标记进行比较,如果在  $B$  中存在某个标记  $b_j$ ,使得相似度  $\text{Sim}(a_i, b_j)$  大于或等于设定的阈值,则匹配分值为 1。同时,标记出在  $B$  中的匹配位置  $p$ ,该次比较结束。 $A$  中取出

下一个标记,从  $B$  中上一次标注位置  $p$  开始向右匹配,直到全部比较结束为止。

算法的关键在于第三步中单词间的比较,即如何来确定两个基于单词标记的相似度问题。本文在解决此问题时,借鉴了生物学中的序列比对方法,引入了罚分机制。基本思想是:两个基于单词的标记进行比较,首先从第一个标记的首字符开始依次与第二个标记的字符比较,如果有匹配的字符,则标注该位置,并设定该处的匹配分值为 1;如果第一个标记中两个连续的字符在第二个标记中的匹配位置有间隔,则引用罚分。第一次出现匹配间隔,每个间隔位罚 0.2 分;第二次出现匹配间隔,每个间隔位罚 0.4 分;以此类推,每出现一次间隔的罚分是前次的 2 倍。

### 2.2 算法描述

对于上面描述的算法思想,给出了形式化的算法描述,记为 PCM( Position-Coding Method) 算法,如下所示:

```

输入:两个待匹配的字串 A,B
输出:匹配分值
方法:
(1)将字串 A 和 B 按照基于单词的方式标记;
  If (|A| < |B|) A↔B;
  /* 如果字串 A 的长度小于 B 的长度,则字串 A 和 B
  交换 */
(2)If (|A| = 1 and |B| > 1) {
  将 A 基于字母进行标记;并计 A 中字母个数为 na;
  If (na == |B|) {
    For 每个字母 ∈ A 将其与 B 中每个单词的首字母进
    行比较是否相等;
    If 完全相等 RETURN 匹配分值 1;
    /* 说明 A 与 B 匹配 */
  }
  (3)Else {
    设初始匹配位置 p0 = 0;
    For 每个单词 a_i ∈ A
    For 每个单词 b_j ∈ B
    {
      /* 从 p0 位置开始进行匹配比较 */
      计算相似度 Sim(a_i, b_j);
      If (Sim(a_i, b_j) > T)
        匹配成功,并标记新的位置 p0 = j;
    }
    /* T 为阈值 */
    RETURN A 和 B 的匹配分值;
  }
}
2.3 算法举例
例如:A = "dept", B = "department"
B:   d   e   p   a   r   t   m   e   n   t
A:   d   e   p                   t
score  1   1   1                   1
penalty -0.2 -0.2
score(A,B) = 1 + 1 + 1 - 0.2 - 0.2 + 1 = 3.6
Sim(A,B) = 3.6/4.0 = 0.9
如果设定阈值为 0.7,由于 0.9 > 0.7,所以 A 与 B 是匹配
的,是相似重复记录。
再例如:A = "damn", B = "department"
B:   d   e   p   a   r   t   m   e   n   t
A:   d           a           m       n
score  1           1           1       1
penalty -0.2 -0.2 -0.4 -0.4 -0.8
score(A,B) = 1 - 0.2 - 0.2 + 1 - 0.4 - 0.4 + 1 - 0.8 +
1 = 2.0

```

$$\text{Sim}(A, B) = 2.0 / 4.0 = 0.5$$

同样,如果设定阈值为 0.7,由于  $0.5 < 0.7$ ,所以  $A$  与  $B$  不匹配,不是相似重复记录。

#### 2.4 算法性能比较分析

与文献[3]中提到的 RFMA 算法相比,PCM 算法克服了匹配位置的错误记录现象,降低了错误匹配的概率,使得算法正确匹配的精确度得到了增强。为了对比分析算法的性能,从某个实际的数据仓库中选取了一个具有 5 个字段的数据表(包括姓名,职务,工作单位,邮政编码和联系电话)进行了实验。该数据表中共有 3 762 条数据信息,其中相似重复记录 15 对(组)。实验中选取了阈值  $T=0.5, 0.6$  和  $0.7$  进行了对比分析。实验结果如表 1 所示。从表 1 的实验数据可知,PCM 算法在匹配的精确度上有了很大的改进和提高。

表 1 实验对比分析

阈值 $T$	RFMA 算法精确度(%)	PCM 算法精确度(%)
0.5	43.3	29.8
0.6	55.4	46.7
0.7	72.9	53.8

### 3 算法在中文字符集中的改进及应用

上述算法能够很好地应用到西文字符集中,但是汉字字符的比较却有所不同。我们对该算法稍加改进,使其可以用于中文字符的匹配问题。考虑算法的第三步,仅针对汉字单个字符来做匹配,类似于算法中基于字符的匹配。

例如:有两个地址字段值  $A = " 大连市沙河口区黄河路 ", B = " 沙区黄河路 "$ 。首先用上述的方法来动态确定匹配阈值  $T$ , $T = 1 - (0.2 * (|A| + |B|) / 2) / |A| = 1 - (0.2 * (10 + 5) / 2) / 5 = 0.7$ 。匹配情况如下:

A= 大连市沙河口区黄河路  
 B= 沙区黄河路  
 Score= 1 -0.2 -0.2 1 1 1  
 $\text{Score}(A, B) = \sum s(A_i, B_i)$   
 $= 1 - 0.2 - 0.2 + 1 + 1 + 1 = 4.6$

(上接第 879 页)

### 2 结语

入侵检测采用的技术有多种类型,其中基于数据挖掘技术的入侵检测技术成为当前入侵检测技术发展的一个热点。基于数据挖掘的入侵检测虽在许多方面取得了成果,但现在还远没有达到能投入实际使用的程度,也没有形成完备的理论体系。因此,对解决数据挖掘的入侵检测实时性、正确检测率和误警率等方面问题加以研究,使入侵检测系统更完善并能投入实用阶段。总的来说应在系统体系结构、提高挖掘算法的检测性能、实时的入侵检测、与其他检测技术的融合等方面进行深入研究。

#### 参考文献:

- [1] 杨德刚. 入侵检测中数据挖掘技术应用研究分析[J]. 重庆师范大学学报(自然科学版), 2004, 21(4): 120-125.
- [2] 李鸿培, 王新梅. 基于神经网络的入侵检测系统模型[J]. 西安电子科技大学学报, 1999, 26(5): 667-670.
- [3] 郑宏, 陆阳, 徐朝农. 基于 BP 神经网络的入侵检测系统分类器的实现[J]. 合肥工业大学学报, 2003, 26(2): 150-155.
- [4] CHEN MS, HAN J, YU PS. Data Mining: An Overview from a Da-

$$\text{Sim}(A, B) = \text{Score}(A, B) / |A| = 4.6 / 5 = 0.92$$

因为  $\text{Sim}(A, B) = 0.92 > T$ , 所以可以认为  $A$  和  $B$  是相似重复的。

### 4 结语

在数据仓库 ETL 的构建中,提高数据质量是关键,而检测和消除数据仓库中的相似重复记录是数据清洗和提高数据质量要解决的主要问题之一。PCM 算法尽管在时间复杂度上并没有明显的改善,但是与 RFMA 算法相比,匹配的精度提高了。同时还针对不同层次的匹配,给出了匹配阈值的动态确定方法。该算法不仅可以应用到西文字符集中,而且也成功的应用到了中文字符集中,具有很强的通用性。

#### 参考文献:

- [1] INMON WH. Building the Warehouse[M]. 2nd Edition. New York: John Wiley and Sons Inc., 1996.
- [2] LEE ML, LU H, LING TW, et al. Cleaning Data for Mining and Warehousing[A]. Proceedings of the 10th International Conference on Database and Expert Systems Application[C]. London: Springer-Verlag, 1999. 751-760.
- [3] MONGE AE, ELKAN CP. The field matching problem: Algorithms and applications[A]. Proceedings of the 2nd international conference on knowledge discovery and databases[C]. London: Springer-Verlag, 1996. 267-270.
- [4] MONGE AE. Matching Algorithms within a Duplicate Detection System[J]. IEEE Data Engineering Bulletin, 2000, 23(4): 14-20.
- [5] GRAVANO L, IPEIROTIS PG. Using q-grams in a DBMS for approximate string processing[J]. IEEE Data Engineering Bulletin, 2001, 24(4): 28-34.
- [6] ANANTHANKRISHNA R, CHAUDHURI S, GANTI V. Eliminating Fuzzy Duplicates in Data Warehouses[A]. Proceedings of the 28th VLDB Conference[C]. Hong Kong, China, 2002. 586-597.
- [7] MASEK W, PATERSON MA. Faster Algorithm Computing String Edit Distance[J]. Journal of Computer System Science, 1980(20): 18-31.
- [8] 陈细谦, 迟忠先, 昂宗亮, 等. 地理编码在空间数据仓库 ETL 中的应用[J]. 小型微型计算机系统, 2005, 16(4): 628-630.

tabase Perspective[J]. IEEE Transaction on Knowledge and Data Engineering, 1996, 8(6): 862-883.

- [5] 王东龙, 李茂青. 基于遗传算法的数据挖掘技术应用[J]. 南昌大学学报, 2005, 27(1): 50-54.
- [6] 郑志军, 林霞光, 郑守淇. 一种基于神经网络的数据挖掘方法[J]. 西安建筑科技大学学报, 2000, 32(1): 28-30.
- [7] 刘勇国, 李学明, 张伟, 等. 基于遗传算法的特征子集选择[J]. 计算机工程, 2003, 29(6).
- [8] LEE WENKE, STOLFO SAL, MOK KUI. Mining Audit Data to Build Intrusion Detection Models[A]. New York: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining(KDD'98)[C]. 1998.
- [9] LEE WENKE, STOLFO SAL, MOK KUI. A Data Mining Framework for Building Intrusion Detection Models[A]. Oakland CA: Proceedings of the 1999 IEEE Symposium on Security and Privacy[C]. 1999.
- [10] LEE WENKE, STOLFO SAL. Data Mining Approaches for Intrusion Detection[A]. San Antonio, TX: Proceedings of the 7th USENIX Security Symposium[C]. 1998.
- [11] 云庆夏, 黄光球, 王战权. 遗传算法和遗传规则——一种搜索寻优技术[M]. 北京: 冶金工业出版社, 1997.