软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(12):3223−3240 [doi: 10.13328/j.cnki.jos.005244]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

泛化双向相似连接*

王昶平, 王朝坤, 汪 浩, 王 萌, 陈 俊

(清华大学 软件学院,北京 100084)

通讯作者: 王朝坤, E-mail: chaokun@tsinghua.edu.cn

摘 要: 相似连接是数据管理领域的一个热门话题,已在社会生产生活中得到广泛应用.然而,现有的相似连接方法并不能满足真实世界不断增长的客观需求.通过引入定义在多种数据类型上的满足操作符和每条数据的独立阈值,定义了一种相似连接——泛化双向相似连接.这种连接扩展了相似连接的应用范围.同时,还提出了两种高效的解决泛化双向相似连接问题的方法:子连接集算法和映射-过滤-验证算法.通过真实与合成数据集上的大量实验,得出了所提方法的正确性和有效性.

关键词: 双向相似连接;泛化数据;独立阈值;数据映射;过滤验证

中图法分类号: TP311

中文引用格式: 王昶平,王朝坤,汪浩,王萌,陈俊.泛化双向相似连接.软件学报,2017,28(12):3223−3240. http://www.jos.org.cn/1000-9825/5244.htm

英文引用格式: Wang CP, Wang CK, Wang H, Wang M, Chen J. On generalized bisimilarity join. Ruan Jian Xue Bao/Journal of Software, 2017,28(12):3223−3240 (in Chinese). http://www.jos.org.cn/1000-9825/5244.htm

On Generalized Bisimilarity Join

WANG Chang-Ping, WANG Chao-Kun, WANG Hao, WANG Meng, CHEN Jun

(School of Software, Tsinghua University, Beijing 100084, China)

Abstract: Similarity join is one of the hottest topics in the field of data management, and it has been widely applied in many fields. However, existing similarity join methods cannot meet the increasing demands in the real world. This paper define generalized bisimilarity join as a new similarity join to expend the applications of the similarity join research by introducing the satisfaction operator on various data types with individual thresholds. Two efficient methods, SJS (sub-join set) and MFV (mapping-filtering- verification), are proposed to solve this problem. A large amount of experiments conducted on both real-world and synthetic datasets demonstrate the correctness and the effectiveness of the proposed methods.

Key words: bisimilarity join; generalized data; individual threshold; data mapping; filter and verification

相似连接旨在从两个或一个给定的数据集中找出满足预定连接条件的所有数据对.作为数据库应用中的一个重要操作,相似连接受到学术界和工业界的普遍关注,并在重复检测[1]、同源检索[2]和下一代基因测序[3]等众多应用场景中发挥着越来越重要的作用.

数据库研究者已针对不同类型的数据进行了大量的相似连接研究工作,如关系数据[4]、实体[5]、集合[6]以及字符串[7].近年来,研究者们更是将相似连接问题拓展到图数据等更为复杂的数据类型上[8].然而,现有的相似连接研究成果仍不能很好地满足现实世界中不断增长的客观需求.

场景 1(交友):假设有两位女士{Alice,Carol}和两位男士{Bob,Dave}希望结交异性.见表 1,他们分别给出了

* 基金项目: 国家自然科学基金(61373023)
Foundation item: National Natural Science Foundation of China (61373023)
收稿时间: 2016-08-01; 修改时间: 2016-10-26; 采用时间: 2016-11-19; jos 在线出版时间: 2017-03-24
CNKI 网络优先出版: 2017-03-24 17:11:02, http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1711.014.html

在各个属性上的自身条件(事实)以及对于交友对象的要求(期望).表中每一列分别对应事实(期望)属性:“◇~★”表示“从◇到★”.教育属性中,B 代表学士,M 代表硕士,D 代表博士;有房属性中,Y 代表有房,N 代表无房;婚姻属性中,S 代表单身.例如,Alice 身高事实是 166cm,对于交友对象的身高期望是 168 cm ~172cm.从表 1 可知,事实和期望可以是数值、数值范围、布尔和枚举等不同类型的数据.同时,假定 Alice 和 Bob 都比较挑剔,他们要求交友对象的事实完全符合自己的期望,也就是说,匹配程度(在所有属性中,潜在交友对象的事实满足希望交友者所提对应期望的百分比)是 100%.与此不同的是,Carol 和 Dave 相对友好,仅需要交友对象的事实大部分满足自己提出的期望.表 1 中,Carol 要求的匹配程度不低于 80%,Dave 的要求类似.

Table 1 Dating information

表 1 交友信息

(a) 男士集合

姓名	年龄	身高	教育	有房	婚姻	阈值
Bob	26 (21~22)	174 (165~167)	D (B|M)	N (N)	S (S)	100%
Dave	27 (20~22)	175 (165~169)	B (B|M)	Y (N)	S (S)	80%
…	…	…	…	…	…	…

(b) 女士集合

姓名	年龄	身高	教育	有房	婚姻	阈值
Alice	23 (24~27)	166 (168~172)	D (M|D)	N (Y)	S (S)	100%
Carol	24 (25~27)	167 (171~173)	B (B|M|D)	N (Y)	S (S)	80%
…	…	…	…	…	…	…

场景 2(求职招聘):求职和招聘是工作市场中的一个重要话题.招聘者会提供薪水、假期和工作地点的事实以及对于应聘者的专业技能和工作经验等的期望.同时,求职者会对应地提供他们对于薪水、假期、工作地点的期望以及他们的专业技能和工作经验的事实.与交友场景类似,不同的求职者和招聘者会有着不同的对于匹配程度的最低标准.

此外,还存在房屋租赁等一系列类似应用场景及问题.显然,现有的相似连接方法不能直接用来解决这些问题.其主要原因在于:

(1) 现有的相似连接方法通常仅考虑一种数据类型,如字符串或向量,并根据这种数据类型的特点提出特殊的处理方法以提高连接效率.然而,在上述问题中存在着多种数据类型,如有房属性中的布尔类型数据,年龄、身高属性中的数值范围和数值类型数据;

(2) 现有的相似连接方法采用全局阈值,但是在场景 1 中,采用全局阈值不能得到{(Carol,Dave)}这样的理想连接结果.当全局阈值被设置为 100%时,连接结果为空集,而当全局阈值被设置为 60%时,连接结果为两个集合的笛卡尔积.显然,这两个结果都与理想结果存在差距;

(3) 现有的所有相似连接方法只在同一个属性维度上考虑了两个比较对象的相等关系或者某种偏序关系.但是在上述场景中,我们需要在事实类属性和对应的期望类属性上考虑上述关系.同时,更复杂的关系(如“包含”)也可能被考虑.

我们认为,上述问题存在以下 3 个挑战:(1) 每个比较对象都存在自己独立的匹配标准,虽然这类标准可以通过阈值进行刻画,但是无法采用当前相似连接查询处理方法中普遍使用的全局阈值;(2) 上述问题涉及数值、数值范围、枚举、布尔、字符串等多种类型的数据,这使得对象的比较方式更加多样化;(3) 每个比较对象都有事实类属性和期望类属性,具体比较时,事实属性和期望属性的交叉匹配将取代在同一属性上的简单比较,使得比较方式更为复杂.

针对以上挑战,本文提出泛化双向相似连接的概念及相关查询处理算法,能够较好地解决上述问题.本文的主要贡献如下.

(1) 提出了一种新的相似连接概念——泛化双向相似连接.这种连接支持泛化数据类型(包括数值、数值范围、枚举、布尔、字符串等)的事实属性与对应期望属性的交叉比较;通过为每个比较对象设置独立阈值,使得连接结果更加符合用户客观需求;

(2) 针对泛化双向相似连接查询处理问题,提出子连接集算法和映射-过滤-验证算法.对于映射-过滤-验证算法,还提出了 3 种映射方法.其中,启发式映射方法在性能上优于单射和等步长映射,能进一步提高算法效率;

(3) 在真实和合成数据集上的实验结果表明,本文所提的两种算法相对于基准算法有效提高了运算效率.同时,实验结果显示了为用户设置独立阈值得到的连接结果要优于仅通过全局阈值获得的连接结果.

本文第 1 节介绍相关工作.第 2 节给出泛化双向相似连接的形式化定义.第 3 节提出泛化双向相似连接查询处理算法并分析其时间复杂度,同时论述算法的正确性.第 4 节给出两种改进的映射方法.第 5 节对实验结果进行展示和分析.第 6 节总结全文.

1 相关工作

与本文最相似的工作是相似连接.数据库研究者已针对不同类型的数据进行了大量的相似连接的研究工作,如关系数据[4]、实体[5]、集合[6]、字符串[7]以及图[8]等.

在以上这些相似连接工作中,字符串相似连接(SSJ)是最具典型性的一个代表,也是近些年数据库领域非常热门的话题.它被用于查找两个字符串集合中所有相似的字符串.现有的 SSJ 研究成果可以分为如下 3 类:第 1 类是基于索引的方法,它们使用了针对字符串的不同的索引结构,如 Trie 树[9]、倒排索引[10,11]、B+树[7,12]、HS 树[13]等,以此在搜索空间中进行减枝,从而使 SSJ 的计算更为高效;第 2 类是基于数字签名的方法,基于不同的数字签名机制,研究者提出了许多高效的 SSJ 算法,比如 FastJoin[14]和 VChuckJoin[15];最后一类是基于过滤的方法,为了避免相似连接算法对所有可能的字符串对都计算相似度,研究者们设计出许多过滤器,如前缀过滤[4,16,17,18,19]和位置过滤[1].为了处理海量数据,近些年,人们还提出了许多应用 MapReduce 框架并行计算字符串相似连接的方法[18,19].

然而,上述方法都不能直接应用于泛化双向相似连接.原因可以归纳为如下 3 点:首先,大多数已有的研究工作集中于考虑字符串,而很少考虑同时包含字符串、数值、枚举等多种数据类型的泛化数据;其次,现有的字符串相似连接方法仅仅考虑点对点的比较,没有处理数值范围的能力,同时也不支持交叉比较;最后,现有的字符串相似连接方法均采用全局统一的阈值,没有将独立阈值纳入研究范围.

2 基本概念

本节首先定义一种新的操作符,在此基础上给出泛化双向相似连接的形式化定义.论文用到的符号及其含义见表 2.为了方便叙述且不失一般性,论文中讨论的数值范围都为整数.对于实数,可根据不同场景采用多种方法将其转化为整数.

Table 2 Symbols and meanings

表 2 符号及其含义

符号	含义
T	阈值
R,S	数据集
R,s	数据集中的一条记录
$\propto$	“满足”操作符
$\bowtie$	泛化双向相似连接运算
$F(r)$	记录 r 的所有事实数据
$E(r)$	记录 r 的所有期望数据
$T(r)$	记录 r 的阈值

2.1 “满足”操作符

“满足”(∝)操作符定义在事实和对应的期望上.对于不同类型的数据,“∝”的判定标准不尽相同.举例来说,如果事实 f 是一个数值类型的数据,而期望 $e=a\sim b$ 是一个数值范围类型的数据,那么 $f\propto e$ 当且仅当 $f\geqslant a\wedge f\leqslant b$.例如在表 1 中,166∝165~167.同时,若事实 f 是集合中的一个元素,而期望 $e=\{e_1,e_2,\ldots,e_n\}$ 为一个集合,则 $f\propto e$ 当且仅当 $f\in e$,如 $M\propto\{B,M,D\}$.

2.2 泛化双向相似连接的定义

定义 1(泛化双向相似连接). 假设数据集 R 和 S 中的每条记录都包含事实数据、期望数据、一个阈值数据和其他数据,且事实数据和期望数据的数据类型可以是数值、数值范围、枚举、布尔、字符串等在内的多种类型,形式化描述为

$$R=\{(r_1^f,\ldots,r_u^f,r_{u+1}^e,\ldots,r_{u+v}^e,r_{u+v+1},\ldots,r_q,T(r))\},S=\{(s_1^f,\ldots,s_v^f,s_{v+1}^e,\ldots,s_{u+v}^e,s_{u+v+1},\ldots,s_w,T(s))\},$$

其中,$u+v\leqslant q$ 且 $u+v\leqslant q$. r_i^f (i=1,2,…,u)代表 r 的 u 个事实数据, r_i^e (i=u+1,u+2,…,u+v)代表 r 的 v 个期望数据,r_i (i=u+v+1,u+v+2,…,q)代表 r 的其他数据,$T(r)$是 r 的阈值数据.同样地, s_i^f (i=1,2,…,v)代表 s 的 v 个事实数据, s_i^e (i=v+1,v+2,…,v+u)代表 s 的 u 个期望数据,s_i(i=u+v+1,u+v+2,…,w)代表 s 的其他数据,$T(s)$是 s 的阈值数据.

$R\bowtie S=\{(r,s)|r\in R,s\in S,ExSim(r,s)\geqslant T(s),ExSim(s,r)\geqslant T(r)\}$,其中,

(1) $ExSim(r,s)=|\{r_i^f\propto s_{(i+v)}^e\,|\,1\leqslant i\leqslant u\}|/u$;

(2) $ExSim(r,s)=|\{s_j^f\propto r_{(j+u)}^e\,|\,1\leqslant j\leqslant v\}|/v$.

如图 1 所示,r 的事实数据对应 s 的期望数据,s 的事实数据对应 r 的期望数据.为便于叙述,在本文余下部分,$F(r)$代表记录 r 的所有事实数据的集合,$E(r)$代表 r 的所有期望数据的集合.同时,L 表示函数 $ExSim$ 的分母.于是,对 $ExSim(s,r)$有 $L=v$,对 $ExSim(r,s)$有 $L=u$.

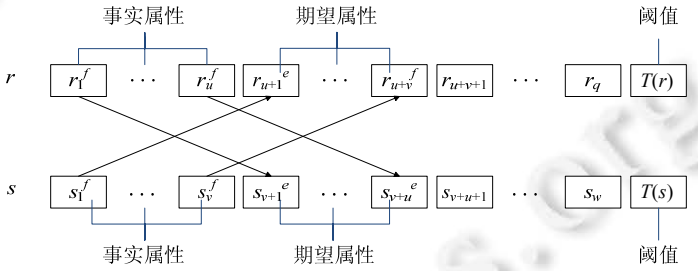


Fig.1 An illustration of bisimilarity join, $r\in R$, $s\in S$

图 1 泛化双向相似连接示意图,$r\in R,s\in S$

表 1 中,F(Dave)={27,175,M,Y,S},E(Alice)={24~27,168~172,M|D,Y,S},于是有:

$$ExSim(\text{Dave},\text{Alice})=|\{27\propto 24\sim 27,\text{Y}\propto\text{Y},\text{S}\propto\text{S}\}|/5=60\%.$$

类似地,可计算出其他 $ExSim$ 值.

根据上述计算结果,因为 $ExSim$(Dave,Alice)=60%<100%=T(Alice),$ExSim$(Carol,Bob)=60%<100%=T(Bob)且 $ExSim$(Bob,Alice)=60%<100%=T(Alice),所以,(Dave,Alice),(Bob,Alice)和(Bob,Carol)都不满足要求.与此同时,考虑到 $ExSim$(Dave,Carol)=80%≥T(Carol)且 $ExSim$(Carol,Dave)=80%≥T(Dave),因此,(Dave,Carol)满足限制条件,要被放入结果集中.

3 算 法

本节首先给出解决泛化双向相似连接的嵌套循环(nested loop)算法,接着提出两种采取不同策略的更高效的算法:一种是基于分治思想的子连接集(sub join set)算法,其针对不同的属性采取不同的处理方式;另一种是

基于归一化思想的映射-过滤-验证(mapping filtering verification)算法,其将泛化数据类型统一为符号表示进行处理.最后,我们论证了 3 种算法的正确性,并且比较了它们的优缺点与适用场景.

3.1 嵌套循环算法

如前所述,现有的专用于相似连接查询处理的算法都不能直接用来解决泛化双向相似连接问题.注意到,朴素的嵌套循环算法是可以应用于任何连接问题的.于是,本节介绍如何将嵌套循环算法应用于泛化双向相似连接,并将其作为后继方法的比较基准.

嵌套循环算法的思路简单而有效,它比较 R 和 S 两个数据集中的所有记录,计算得出它们的相似程度并与各自的阈值进行比较.如果它们彼此间的相似程度不小于各自的阈值,那么这对数据就被认做相似对并被放入结果集合中.

假定计算 *ExSim* 函数的时间复杂度为 $O(c)$,这是因为在实际情况中,一条记录中的数据类型可能比较复杂,计算 *ExSim* 的值会消耗较长的时间,不能简单地用 $O(1)$去衡量.例如:在交友场景中,在身高和年龄属性上存在着数值类型和数值范围类型的比较;在教育状况和婚姻状况属性上,存在着包含关系的判定(如果认为教育状况和婚姻状况都可以用枚举类型描述);而在是否有房属性上,存在着布尔类型的匹配.同时,假定每条记录占用 $O(1)$的存储空间,数据集 R 和 S 的规模分别为 m 和 n.因为嵌套循环算法比较数据集中的每对记录,所以时间复杂度为 $O(cmn)$.考虑到最终结果集可能是两个集合的笛卡尔积,因此,算法的空间复杂度为 $O(mn)$.

3.2 子连接集算法

提出一种基于分治策略的算法——子连接集算法,它将泛化双向相似连接分解成一系列的子连接问题来解决.每个子连接在单一属性上根据事实和期望对两个数据集合进行连接,对于每一个单一属性,都会建立适合其数据类型的索引结构来进行连接操作.通过合并每一个子连接的结果,我们可以得到最终的泛化双向相似连接的结果.具体算法见算法 1.

算法 1. 子连接集 SJS 算法.

输入:R,S——数据集;

输出:RS——查询结果数据集.

1. $RS\leftarrow\varnothing$

/*为数据集合 R 和 S 的每个属性建立索引*/

2. FOR ALL $A\in F(R)$ DO
3. $I_A\leftarrow SJIndexBuilding(A)$
4. FOR ALL $B\in F(S)$ DO
5. $I_B\leftarrow SJIndexBuilding(B)$

/*利用索引计算每一个子连接并合并*/

6. $M^R\leftarrow 0_{n\times m}$
7. FOR ALL $A\in E(R)$ DO
8. $M^R\leftarrow M^R+SubJoin(A,I_A,S)$
9. $M^S\leftarrow 0_{n\times m}$
10. FOR ALL $B\in E(S)$ DO
11. $M^S\leftarrow M^S+SubJoin(B,I_B,R)$

/*通过合并后的子连接的结果验证每一对数据*/

12. FOR ALL $r\in R$ DO
13. FOR ALL $s\in S$ DO
14. IF $M^S_{r,s}\geqslant T(r)\times|F(r)|$ AND $M^R_{s,r}\geqslant T(s)\times|F(s)|$
15. $RS\leftarrow RS\cup\{(r,s)\}$

16. RETURN *RS*

首先,我们为数据集 *R* 和 *S* 的每一个属性建立索引(算法 1,步骤 2~步骤 5).对于不同的数据类型,我们采用不同的索引结构.这样可以保证在每个单一属性上的子连接操作的效率.

然后,我们对 *R* 中的每一个属性进行子连接操作.M^R 是初始化为零矩阵的 $n\times m$ 的矩阵(步骤 6),其每一个元素 $M^R_{s,r}$ 的值代表的是 *s* 的期望属性可以被 *r* 的事实属性满足多少条.对于 *R* 数据集的每一个事实属性 *A*,我们以索引 I_A 为基础,执行一个 *R* 和 *S* 的子连接操作(步骤 8).子连接操作的具体步骤见算法 2.

算法 2. 子连接操作 Sub-join.

输入:*R*,*S* 数据集,*R* 的属性 *A*,*A* 对应的索引 I_A;

输出:BM^A.

1. $BM^A \leftarrow 0_{n\times m}$
2. FOR ALL $s\in S$ DO
3. $Bitmap_s^A \leftarrow Search(s,R,A,I_A)$
4. $BM^A \leftarrow AssignRow(BM^A,s,Bitmap_s^A)$
5. RETURN BM^A

我们使用比特矩阵 BM^A 记录对于属性 *A* 的子连接结果,它的每一个元素 $BM^A_{s,r}$ 代表的是仅考虑属性 *A* 时,*r* 是否可以满足 *s*.对于 *S* 数据集中的每一个记录 *s*,我们在数据集 *R* 上执行一个搜索操作,从而找到所有可以满足 *s*(仅考虑属性 *A*)的 *R* 中的记录(算法 2,步骤 3).搜索的结果为位图 $Bitmap_s^A$,其第 *i* 位表示 *R* 中的第 *i* 个记录是否可以在属性 *A* 满足 *s* 的期望.我们将位图中每一位的数据赋值给 BM^A 的对应行的每个元素(步骤 4).循环结束后,获得了完整的 BM^A.回到算法 1 中,将子连接的结果加至矩阵 M^R 上(算法 1,步骤 8),考虑完所有属性后,获得了最终的 M^R.

接下来,我们用相似的方式,对于数据集 *S* 的各个属性进行子连接操作(步骤 9~步骤 11).结果为另一个矩阵 M^S,它和 M^R 有着相似的含义.

最终,我们通过分析两个矩阵的值来验证每一个可能的结果对.对于结果对(*r*,*s*),如果可以满足步骤 14 中的条件,它就会被加入到结果集中.

用场景 1 中的例子来说明子连接集算法:首先,对于男士集合和女士集合的所有属性分别建立高效的索引,具体来说,对年龄和身高属性建立了 B^+树索引,对教育和婚姻属性建立了倒排索引,对有房属性建立了位图索引;接下来,在每个属性上进行子连接操作,以女士集合的教育属性为例,图 2(a)为该子连接操作对应的结果矩阵 *BM*,该矩阵显示,Alice 的教育属性不满足 Bob 和 Dave 的期望,而 Carol 的则可以满足;最终,在完成所有的子连接操作后,我们得到了两个结果矩阵 M^{Female} 和 M^{Male},如图 2(b)、图 2(c)所示,矩阵分别记录了两个集合中的对象的满足情况.以图 2(c)的 M^{Male} 为例,Bob 的事实属性一共满足了 Alice 的 5 个期望属性中的 3 个,而 Alice 的阈值是 100%,因此我们可知,(Bob,Alice)一定不属于结果集.通过对这两个矩阵的分析,我们找到了最终的结果集{(Dave,Carol)}.

	Alice	Carol
Bob	0	1
Dave	0	1

(a) 女士身高属性子连接结果

	Alice	Carol
Bob	3(×)	4(×)
Dave	3(×)	4(√)

(b) 合并多个子连接后的 M^{Female}

	Bob	Dave
Alice	3(×)	3(×)
Carol	3(×)	4(√)

(c) 合并多个子连接后的 M^{Male}

Fig.2 An example of the SJS algorithm

图 2 子连接集算法示例

3.3 映射-过滤-验证算法

本节提出另一种专门解决泛化双向相似连接问题的算法,它包含映射、过滤和验证这 3 个步骤,即映射-过滤-验证算法.首先,在映射阶段,所有记录的数据被映射到一个全局的符号集上;然后,过滤步骤排除掉不符合条件的数据对;最后,通过验证,得到最终的结果集.

算法 3. 映射-过滤-验证算法 MFV.

输入:R,S——数据集;

输出:RS——查询结果数据集.

1. $RS\leftarrow\varnothing,CR_1\leftarrow\varnothing,CR_2\leftarrow\varnothing$

/*映射步骤,将数据映射为全局符号*/

2. $R_m\leftarrow Map(R),S_m\leftarrow Map(S)$

/*过滤步骤:预处理阶段——在 R_m 和 S_m 的期望符号上建立倒排索引*/

3. $Sort(R_m,S_m)$

4. $I_r\leftarrow IndexBuilding(R_m),I_s\leftarrow IndexBuilding(S_m)$

/*过滤步骤:在映射后产生的符号记录上进行双向过滤获得候选集合*/

5. FOR ALL $r\in R_m$ DO

6. FOR ALL $w\in F(R)$ DO

7. FOR ALL $(s,w)\in I_s$ DO

8. $CR_1\leftarrow CR_1\cup\{(r,s)\}$

9. FOR ALL $(r,s)\in CR_1$ DO

10. FOR ALL $w\in F(s)$ DO

11. IF $(r,w)\in I_r$ THEN

12. $CR_2\leftarrow CR_2\cup\{(r,s)\}$

13. BREAK

/*验证阶段:对最终候选结果集执行双向验证获得最后的匹配结果*/

14. FOR ALL $(r,s)\in CR_1$ DO

15. IF $ExSim(s,r)\geqslant T(s)\wedge ExSim(r,s)\geqslant T(r)$ THEN

16. $RS\leftarrow RS\cup\{(r,s)\}$

17. RETURN RS

算法 3 详细描述了 3 个具体步骤.

(1) 映射阶段

首先将数据集中每条记录的每个属性的数据映射到全局的符号上,得到符号字典,并在映射结束后,通过统计和排序得到一个全局的按照出现次数递增排序的符号顺序 O_t.而后,每条记录被映射成为一条由全局符号集合中的符号组成的生成记录,这些记录组成了映射后的数据集,记做 R_m 和 S_m(步骤 2).本节所用映射方法为单射,即,各个属性的每个数值被映射到唯一的全局符号上.改进的映射方法将在第 4 节介绍.

(2) 过滤阶段

首先,对于 R_m 和 S_m 中的生成记录依照 O_t 进行排序(步骤 3);然后,采用全局符号作为关键词分别对 R_m 和 S_m 中记录的期望部分建立倒排索引 I_r 和 I_s(步骤 4).根据我们对文献[4]的过滤原则改述后所得到的定理 1,对于排序后的 R_m 和 S_m 中的全局符号记录,只需要索引其前 $L-T\times L+1$ 个期望符号即可,其中,t 代表这条记录的阈值.需要注意的是:在定理 1 中,原过滤原则中的操作符∩被满足操作符∝所替代.

定理 1(过滤原则). 考虑 2 个数据集 R_m 和 S_m,其符号顺序按照 O_t 排序.同时 $ExSim(r,s)\geqslant T(s)$ 且 $r\in R_m$ 和 $s\in S_m$.令 $p=L-T(s)\times L+1$,则 $\{r_i\propto s_j|r_i\in F(r),s_j\in E(s),i\leqslant p,j\leqslant p\}\neq\varnothing$.

证明:若 $\{r_i\propto s_j|r_i\in F(r),s_j\in E(s),i\leqslant p,j\leqslant p\}=\varnothing$,则 $ExSim(r,s)\leqslant(L-p)/L=(T(s)\times L-1)/L<T(s)$.

这与 $ExSim(r,s)\geqslant T(s)$矛盾.故$\{r_i \propto s_j | r_i \in F(r), s_j \in E(s), i \leqslant p, j \leqslant p\} \neq \varnothing$.证毕. □

之后,过滤步骤将依据过滤原则生成候选结果对(步骤 5~步骤 13).具体地,算法先枚举 R_m 数据集中的生成记录 r,找到那些 r 的事实属性对应的全局符号在索引 I_s 中对应的记录 s,并将初始候选对(r,s)放入初始候选集合 CR_1 中(步骤 5~步骤 8).而后,算法遍历 CR_1 中所有的初始候选对(r,s),判定 s 的事实属性对应的全局符号在倒排索引 I_r 中是否存在记录 r:如果存在,则将其放入最终候选结果集 CR_2;如果不存在,则进行剪枝(步骤 9~步骤 13).

(3) 验证阶段

将检验最终候选结果集 CR_2 中的每个候选对,将符合条件的作为最终结果输出(步骤 14~步骤 16).

例如,针对表 1 中的数据(男士集合 R、女士集合 S),在映射阶段,若采用表 3 的单射规则及前述映射-过滤-验证算法,则可得表 4 中的映射后的由事实数据和期望数据构成的符号记录.以 Bob 为例,因为 T(Bob)=100%,同时 L=5,所以 Bob 对应的前缀长度 p(Bob)=5−100%×5+1=1.根据 Bob 的映射后的期望全局符号记录和过滤原则,只有在字典中的全局符号 B 和 C 应该被索引.也就是说,在倒排索引 I_r 中,全局符号 B 和 C 对应的出现列表里应加上 Bob.与此类似,Dave 的前缀长度 p(Dave)=2,因此,Dave 应被加入全局符号 A~C 以及 I~M 的倒排列表内.对于倒排索引 I_s,因为 p(Alice)=1,所以 L~P 这些全局符号的倒排列表中应包含 Alice.同时,因 p(Carol)=2,故全局符号 O~Q,W 对应的倒排列表中应有 Carol.最终得到图 3 所示的倒排索引结构.

Table 3 Injective mapping rule

表 3 单射规则和频率统计

属性(值)	映射符号	频率(事实属性中)
年龄(20,21,…,27)	A,B,C,…,H	0,0,0,1,1,0,1,1
身高(165,166,…,174,175)	I,J,K,…,S	0,1,1,0,0,0,0,0,0,1,1
教育(B,M,D)	T,U,V	2,0,2
有房(Y,N)	W,X	1,3
婚姻(S)	Y	4

Table 4 Injective mapping result

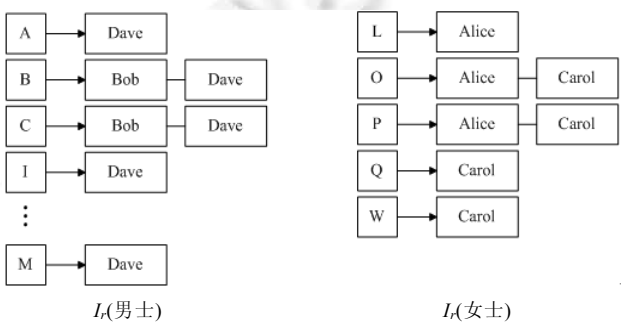
表 4 单射规则和频率统计

(a) 男士集合

姓名	映射结果	排序后结果
Bob	GRVXY ({B~C}{I~K}{T~U}XY)	GRVXY ({B~C}{I~K}{T~U}X Y)
Dave	HSTWY ({A~C}{I~M}{T~U}XY)	HSWTY ({A~C}{I~M}{T~U} X Y)

(b) 女士集合

姓名	映射结果	排序后结果
Alice	DJVXY ({E~H}{L~P}{U~V}WY)	DJVXY ({L~P}W{U~V}{E~H}Y)
Carol	EKTXY ({F~H}{O~Q}{T~V}WY)	EKTXY ({O~Q}W{F~H}{T~V}Y)



I_r(男士) I_r(女士)

Fig.3 Index structure for injective mapping

图 3 应用单射后的索引结构

在过滤阶段,首先,只有(Dave,Carol)被选入初始候选集合 CR_1 中.这是因为 Dave 的事实符号 W 在 I_s 的符号字典内,且 Carol 在 W 对应的倒排列表中.接着,因为 I_r 的符号字典里包含的 Carol 的事实符号 K 所对应的倒排列表中也含有 Dave,所以候选对(Dave,Carol)通过了过滤.在验证阶段,最终,候选结果集中的每个候选对都要被进行检验,结果表明,{(Dave,Carol)}即为最终结果.

与嵌套循环算法相比,映射-过滤-验证算法通过提前过滤掉不可能的匹配对,避免了记录之间不必要的相似性比较,从而有效提高了泛化双向相似连接查询处理的效率.在上例中,验证阶段前有 75%的数据对被过滤掉,这明显提高了查询处理效率.

3.4 算法讨论

本节提出的 3 个算法都是正确的.这里的正确性指不产生错误结果且不漏掉正确结果.嵌套循环算法通过遍历所有数据对,来得到满足相似度阈值的全部结果.采用该方式,嵌套循环方法不会引入错误结果,也不会漏掉正确结果.子连接集算法同样遍历所有的数据对,但是用一种比嵌套循环算法更高效的方式验证每一个结果对,因此它也是正确的.对于映射-过滤-验证算法而言,过滤步骤对一部分错误数据对提前进行剪枝操作,而验证阶段则精炼最终候选结果集,剔除剩余的错误数据对.同时,因为在过滤过程中正确结果都会被保留(根据定理 1),所以不会产生正确结果的遗漏.综上所述,本文提出的嵌套循环算法、子连接集算法和映射-过滤-验证算法都具有正确性.

嵌套循环算法相比于子连接集算法和映射-过滤-验证算法,由于没有使用索引结构来加速,在算法的时间效率上一定逊于后两者.不过,索引结构本身是有空间开销的,因此在空间限制极为严格的场景中,嵌套循环算法还是有其意义的.子连接集算法和映射-过滤-验证算法采用了不同策略来解决我们的泛化双向相似连接问题:子连接集算法采用分治的思想,对不同的属性建立不同类型的有针对性的索引结构,分开处理每个属性;映射-过滤-验证算法采用了归一化的思想,将不同类型的属性值都统一映射成符号记录,并建立一个包含所有属性的倒排索引,可以同时处理所有属性.因此:当属性数较少时,由于索引结构更有针对性,子连接集算法将具有一定优势;而当属性较多时,受益于可以同时处理所有属性,映射-过滤-验证算法会有更强的过滤能力,从而更加高效.

4 改进的映射方法

本节提出两种新的映射方法,以优化映射-过滤-验证算法中的映射阶段.同时,对这些映射方法的优缺点进行分析比较.

在上一节中,映射阶段统一采用单射方法.然而我们认为,不同数据类型有各自的特性,比如数值类型的属性就要比布尔类型的属性取值范围大.因此,针对不同的数据类型,我们可以考虑采用不同的映射策略,以适应其特点,并进一步提高索引结构的效率.下面,我们将分析针对数值类型的映射策略.

前述单射方法在处理数值类型的属性时,严格准确地将每个不同数值映射到一个唯一符号.但是,如图 3 所示,单射通常会导致非常庞大的索引结构,同时需要很长的索引建立时间.例如,若采用单射方法将薪水的期望 3000~4999 映射到全局符号,则数值范围 3000~4999 中的每一个整数都会被映射到一个唯一的符号.于是,单射将会产生比原始记录膨胀 2 000 倍的符号记录.显然,理想情况下,索引结构的规模应越小越好,以确保建立索引结构的时间开销较小.为此,本节提出等步长映射方法和启发式映射方法用以处理数值类型的属性.这两种方法均能显著减少映射后符号记录的数量,从而明显缩减索引结构的规模、减少索引结构建立的时间代价.

4.1 等步长映射方法

等步长映射方法的基本思想是,通过固定的步长来均匀分割数值范围数据.表 6 展示了对表 1 中的数据采用表 5 中的等步长映射规则后得到的映射结果.与表 4 中的映射结果相比,采用等步长映射方法产生的符号记录数量远小于采用单射方法产生的符号记录数量.在表 6 中,Dave 的期望记录仅仅被映射到 AC{H~I}XY,生成的符号记录只有 2 条.而在表 4 中,Dave 的期望记录被映射为{A~C}{I~M}{T~U}XY,生成的符号记录有 3×5×3×

1×1=45 条.

Table 5 Homogeneous mapping rule

表 5 等步长映射规则

属性(值)	映射符号	频率(事实属性中)
年龄(20~23)	A	1
年龄(24~27)	B	3
身高(165~169)	C	2
身高(170~174)	D	1
身高(175~180)	E	1
教育(B,M,D)	H,I,J	2,0,2
有房(Y,N)	W,X	1,3
婚姻(S)	Y	4

Table 6 Homogeneous mapping result

表 6 等步长映射方法结果

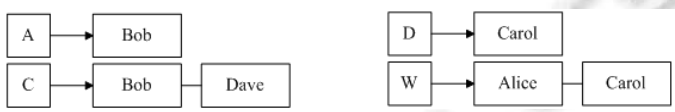
(a) 男士集合

姓名	映射结果	排序后结果
Bob	BDJXY (AC{H~I}XY)	DJXBY (AC{H~I}XY)
Dave	BEIWY (AC{H~I}XY)	IEWBY (AC{H~I}XY)

(b) 女士集合

姓名	映射结果	排序后结果
Alice	ACJXY (B{C~D}{I~J}WY)	ACJXY (W{I~J}B{C~D}Y)
Carol	BCHXY (BD{H~J}WY)	CHBXY (DWB{H~J}Y)

图 4 显示了对于表 1 采用等步长映射方法后的索引结构.显然,相对于图 4,基于等步长映射方法产生的符号记录构建的索引结构的规模大为减小,对应的索引建立时间也缩短了很多.



I_r(男士) I_r(女士)

Fig.4 Index structure for homogeneous mapping

图 4 应用等步长映射后的索引结构

然而,等步长映射方法的剪枝能力较单射的差.对表 1 而言,采用等步长映射方法后,首先遍历男士集合的事实符号,在女士期望的索引结构中找匹配对时,{(Dave,Alice),(Bob,Carol),(Dave,Carol)}会被保留.而在男士期望的索引结构中查找女士事实符号时,这 3 对继续保留在候选集合之中,导致最终将要验证 3 对,过滤效果不明显.这是因为等步长映射方法将多个不同的属性值映射到同一个符号,这种做法降低了剪枝能力.

总体来说,等步长映射方法通过将多个值映射到同样的符号,减小了索引结构的规模和建立索引结构的时间开销.但是,这会降低全局符号对于原始记录属性数据的区分性,进而导致映射-过滤-验证算法在过滤阶段剪枝能力的下降.

4.2 启发式映射方法

4.2.1 优化目标分析

基于前述讨论易知,好的映射方法需要满足以下两个优化目标.

(1) 最小化映射后产生的符号记录数量;

(2) 最大化索引结构的剪枝效果(即,最小化在过滤步骤后最终候选结果集合的大小).

事实上,上述两个目标之间存在不一致性:一方面,如果希望产生的符号记录最少,那么通过将一个属性上的所有值都映射到同一个符号,可以使得映射后产生的符号记录的数量与原始数据中记录的数量保持一致,然而这样做,过滤阶段将失去剪枝能力,并导致最终候选结果集中的数据对与采用嵌套循环算法需要比较的数据对完全相同;另一方面,若要最大化索引结构的剪枝效果,则应采用单射方法,以使索引结构对于符号的区分性最强,但这样会产生最大规模的符号记录集.

针对上述情况,本文提出一种启发式映射方法的折衷方案,尝试获得近似最优的映射方法.

定义 2(集合的划分)[20]. 集合 X 的一个划分 Π 是指 X 的一系列非空且互不相交的子集.换句话说,集合族 P 是 X 的一个划分当且仅当满足如下条件.

(1) P 不包含空集;

(2) P 中所有集合的并集为 X;

(3) P 中任意两个集合之间的交集为空集.

P 中的集合被称为划分的区域、块或者单元.

在本节中,集合 X 代表期望数据里的数值范围,即,从最小值(min)到最大值(max).划分 $\Pi=\{a_i\sim b_i|b_i\geqslant a_i,a_{i+1}=b_i+1,i=1,2,\ldots,k,a_1=\min,b_k=\max\}$.其中,$a_i\sim b_i$ 表示一个划分块 P_i,k 为划分块的数目.

根据定义 2,一般情况下,生成的符号记录的数量随着 k 值的增加而增长.当 k=1 时,所有的原始记录都被映射到同一个生成记录,此时,生成结果的数量与原始记录数量一致;当 k=max−min+1 时,原始记录的每个不同的属性值被映射到唯一的全局符号,于是,一个原始记录将被映射成为多条生成记录,其数量与单射方法产生的映射记录数目相同.这两种极端情况在上述讨论中都已进行阐述,其效果均不理想.

通过上述分析,可以得到如下结论:k 值越大,产生的生成记录数量就会越多.为了满足第一个优化目标,需要将生成的符号记录量限制在一个较小范围内.于是,本文设置 k 值的上限为 k_0.显然,k_0 表示某个数值范围所接受的最大划分块的数目.通过这种方法,生成记录的数目可得到控制.

接下来考虑索引结构的剪枝效果和划分 Π 之间的关系.容易发现,一个划分通常会延展期望数据的数值范围.例如,对于一个给定的划分块 $P_i=a_i\sim b_i$,若 $c_i\geqslant a_i$ 且 $d_i<b_i$,则原始的期望数值范围 $c_i\sim d_i$ 仅是 P_i 的一个部分.表 5 中,在年龄属性上,一个划分块为 20~23.于是,划分块 20~23 延展了 Bob 的期望数值范围 21~22,其延展数值为 2.类似地,该划分块也延展了 Dave 的期望数值范围 20~22,其延展数值为 1.显然,这些划分引起的延展会使得索引结构剪枝能力有所下降,进而导致最终候选结果集的增大.从趋势看,当延展总量增加时,最终候选结果集中数据对的数量也会增长.与此同时,因为划分块数上限 k_0 的限制,延展的情况很难被避免.综上可得,最终优化目标为:在划分块不大于 k_0 的情况下,最小化由于划分所导致的所有延展数值的和.

令 $e(r)=a\sim b$ 为期望的数值范围,$Ext(e(r))$ 为划分对于 $e(r)$ 的延展,即,$Ext(e(r))$ 让 $e(r)$ 能够适合划分的一个或者多个划分块.也就是说,$Ext(e(r))=a_j\sim b_k$.这里,$a_j=\mathrm{Max}(a_i|a_i\leqslant a)$,$b_k=\mathrm{Min}(b_i|b_i\geqslant b)$.

假设划分 $\Pi=\{15\sim17,18\sim22,23\sim26,27\sim32,33\sim36\}$,$e(r_1)=20\sim30$,$e(r_2)=23\sim27$,则有 $Ext(e(r_1))=\mathrm{Max}\{15,18\}\sim\mathrm{Min}\{32,36\}=18\sim32$,同时,$Ext(e(r_2))=\mathrm{Max}\{15,18,23\}\sim\mathrm{Min}\{32,36\}=23\sim32$.

于是,上述最终优化目标可形式化表示为

$$\left.\begin{array}{l}\min\sum_{r\in R}f(e(r),\Pi)\\ \text{sub.to}\ \ |\Pi|\leqslant k_0\end{array}\right\} \tag{1}$$

其中,$|\Pi|$ 是划分 Π 中的块数;f 是一个计算在划分 Π 情况下,$Ext(e(r))$ 的势 $|Ext(e(r))|$ 和 $e(r)$ 的势 $|e(r)|$ 之差的函数.

在上述例子中,$|\Pi|=5$,$f(e(r_1),\Pi)=|Ext(e(r_1))|-|e(r_1)|=4$,$f(e(r_2),\Pi)=|Ext(e(r_2))|-|e(r_2)|=5$,于是:

$$\sum_{r=r_1,r_2}f(e(r),\Pi)=4+5=9.$$

本文称通过优化目标(1)获得的映射方法为启发式映射方法,并将这种映射方法采用的划分 Π 叫做启发式映射规则.

4.2.2 划分算法

为了获得能达到优化目标的划分Π,最直接的方法是枚举.枚举方法是指先遍历所有可能的划分,然后比较它们的延展和,最终获得最优的划分Π.假设 l 表示给定属性中最大值(max)和最小值(min)的差值,那么在集合 R 的该属性上,采用枚举方法获得最优划分Π的时间复杂度为 $O(|R|l^{k_0})$,这是一个很大的时间开销.

针对枚举方法的不足,本节提出了动态规划的方法来获得最优划分Π.首先,需要找到最优子结构.用记号 $P_{i,j,k}$ 表示将数值范围 i~j 划分为 k 个划分块的一个划分,其中,$i\leqslant j,k>0$.$P_{i,j,k}$ 的代价被定义为该划分导致的所有期望数据的延展和.显然,计算 $P_{i,j,k}$ 在 $i<j$ 且 $k>1$ 的情况下是非平凡的.具体地,$P_{i,j,k}$ 首先在第 1 个划分位置 m ($i\leqslant m<j$)上将这个划分分为两个子划分,分别记为 $P_{i,m,1}$ 和 $P_{m+1,j,k-1}$.更进一步地,如果 $P_{i,j,k}$ 是将数值范围 i~j 划分为 k 个划分块的最优划分,那么 $P_{m+1,j,k-1}$ 肯定也是将数值范围 $m+1$~j 划分为 $k-1$ 个划分块的最优划分.这是因为如果又存在一个划分使得数值范围 $m+1$~j 划分为 $k-1$ 个划分块的代价更小,那么这个划分和 $P_{i,m,1}$ 组合之后就可以产生一个新划分,该划分将数值范围 i~j 划分为 k 个划分块的代价将小于 $P_{i,j,k}$ 的代价,于是 $P_{i,j,k}$ 就不是一个最优划分,这与假设矛盾.故 $P_{m+1,j,k-1}$ 也是将数值范围 $m+1$~j 划分为 $k-1$ 个划分块的最优划分.令 $c[i,j,k]$为 $P_{i,j,k}$ 的代价,那么可以得到如下的递归表达式:

$$c[i,j,k]=\begin{cases}0, & i=j\\ c[i,j,1], & k=1,i<j\\ \min\limits_{i\leqslant m<j} c[i,m,1]+c[m+1,j,k-1], & k>1,i<j\end{cases} \tag{2}$$

得到上述最优子结构后,就可以采用动态规划的方法获得最优划分,对应的时间复杂度为 $O(|R|k_0l^2)$.与枚举方法相比,动态规划的方法能显著提高计算最优划分的效率.

4.2.3 例 子

表 7 展示了对表 1 中数据采用动态规划方法获得的启发式映射规则.通过启发式映射方法获得的符号记录和排序后的符号记录见表 8.可以看到:对于年龄属性,存在数值范围期望 20~22,21~22,24~27 和 25~27.在划分块数上限 $k_0=3$ 时,采用启发式划分算法可获得好的划分结果{20~22,23~23,24~27}.最终,经过启发式映射规则,Dave 的原始记录也仅被映射为 4 条符号记录.虽然与等步长映射相比,启发式映射生成记录条数可能会有所增加,但是相较于单射而言,采用启发式映射规则产生的生成记录的条数的规模要显著变小.从这个角度看,启发式映射方法能够减小建立索引的开销,提高建立索引的速度.同时,当采用启发式映射方法时,映射-过滤-验证算法在过滤阶段能够提前排除掉(Bob,Alice),(Bob,Carol)和(Dave,Alice)这 3 个不是最终结果的数据对,这种剪枝效率远远优于等步长映射方法,能够与单射的效率相媲美.因此我们可以看出,启发式映射方法效率较高.

总的来说,基于动态规划生成最优划分的启发式映射方法兼顾了映射后尽量少的生成符号记录和过滤阶段较好的剪枝效率.通过上述分析,可以得到如下结论:启发式映射方法优于等步长映射方法和单射映射方法.事实上,它是根据第 4.2.1 节提出的优化目标得到的一类最优的映射方法.下一节的实验结果也展示了启发式映射方法比其他两种映射方法有着很大的优势.

Table 7 Heuristic mapping rules

表 7 启发式映射规则

属性(值)	映射符号	频率(事实属性中)
年龄(20~22)	A	0
年龄(23~23)	B	1
年龄(24~27)	C	3
身高(165~167)	D	2
身高(168~170)	E	0
身高(171~173)	F	0
身高(174~180)	G	2
教育(B,M,D)	H,I,J	2,0,2
有房(Y,N)	W,X	1,3
婚姻(S)	Y	4

Table 8 Heuristic mapping results.

表 8 启发式映射结果

(a) 男士集合

姓名	映射结果	排序后结果
Bob	CGJXY (AD{H~I}XY)	GJCXY (AD{H~I}XY)
Dave	CGHWY (A{D~E}{H~I}XY)	WGHCY (A{D~E}{H~I}XY)

(b) 女士集合

姓名	映射结果	排序后结果
Alice	BDJXY (C{E~F}{I~J}WY)	BDJXY ({E~F}W{I~J}CY)
Carol	CDHXY (CF{H~J}WY)	DHCXY (FWC{H~J}Y)

5 实 验

5.1 实验设置

5.1.1 实验环境

采用 Java 语言实现了本文提出的算法,JDK 为 1.8.0.实验机器的配置为 Xeon E5-2650 2.00GHz CPU,256GB 内存,操作系统为 Windows Server 2008 64 位.

5.1.2 实验数据集

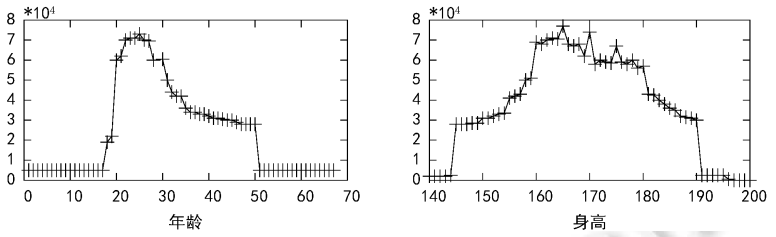
实验中采用了真实数据集 DATING;同时,依据 DATING 中的数据分布规律生成了合成数据集 L-DATING.

- DATING 是交友信息数据集,包括 10 430 条男性交友信息和 9 831 条女性交友信息.这些信息是从交友网站(http://www.eharmony.com/)上爬取的,并进行了去除隐私和归一化等操作.每条记录包括 8 个事实属性和 8 个对应的期望属性,它们分别是居住城市、收入、子女和在表 1 中列举的 5 条属性(不包括阈值).在事实信息中:属性值的数据类型可能是一个数值或一个字符串;期望信息的属性对应的数据类型可能是数值范围或枚举类型(后者可转化为字符串);
- L-DATING 是合成数据集,包括 1 500 000 条记录(男士和女士的信息各 750 000 条).每条记录里包含 12 个事实属性和 12 个期望属性.除了 DATING 原有的属性外,L-DATING 还通过重复年龄、身高、教育和婚姻增加了 4 个额外的虚拟属性.L-DATING 中的数据是根据 DATING 中对应属性上所有数据值的分布情况生成的.

5.2 数据集分析

在进行实验之前,我们先对数据集进行了分析.图 5 展示了数据集中和年龄与身高相关的两个期望属性的分布,我们可以看到,数据不满足均匀分布.对于年龄而言,18 岁和 50 岁处呈现了明显的起伏,期望交友对象的年龄在该区间内的人数显著多于期望交友对象的年龄在区间外的人数.对于身高属性,我们也能观测到类似的情况.

因此,如果我们使用等步长的映射方法,有些符号会对应非常长的倒排列表.与此同时,有些符号所对应的倒排列表又会很短.这种情况会严重影响倒排索引的效率.与此相对的是,启发式映射方法可以利用属性的分布特征来确定划分方式.举例来说,对于年龄属性,启发式映射方法会将 18~50 区间内的值进行细粒度划分,而对于区间外的值会采取粗粒度的划分.这种启发式映射方法使得各个符号所对应的倒排列表的长度相对平均,从而提高倒排索引的效率.



(a) 年龄期望属性 (b) 身高期望属性

Fig.5 Data distribution of attributes

图 5 属性数据分布

5.3 实验结果分析

5.3.1 MFV 算法生成符号记录数量和候选集大小比较

本节通过实验分析比较 MFV 算法映射阶段的 3 种可选映射方法.为方便叙述,用 MFV-I,MFV-O 和 MFV-R 分别代表采用单射的映射-过滤-验证算法、采用等步长映射的映射-过滤-验证算法和采用启发式映射的 MFV 算法.

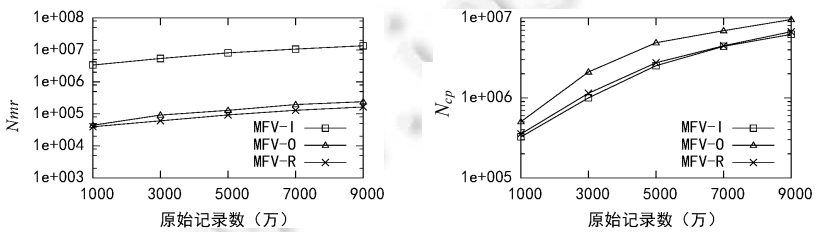
图 6 展示了采用 3 种映射方法的结果:生成的符号记录数量(用 N_{mr} 表示)、最终的候选结果对数量(用 N_{cp} 表示).根据第 4.2.1 节提出的优化目标,可以得到如下两个结论.

(1) N_{mr} 越小,建立索引结构所需要的时间开销就会越小,同时,在过滤阶段的搜索空间也相应减小;

(2) N_{cp} 越小,表明在过滤阶段的过滤效果越好.

如图 6(a)所示,对于 3 种映射方法,N_{mr} 随着原始数据量的增多而变大.一般来说,MFV-I 会比 MFV-O 和 MFV-R 产生多得多的生成记录数量.这将导致 MFV-I 花费大量时间建立索引和利用索引结构进行搜索.这也使得 MFV-I 算法将花费比嵌套循环算法更多的时间来获得最终结果.

图 6(b)显示了 N_{cp} 和原始记录数量的关系.与图 6(a)展示的关系类似,N_{cp} 随着原始记录数量的增多也存在上涨的整体趋势.从图中可以发现,N_{cp} 是原始记录的若干倍.但是需要注意的是:在没有映射的情况下,N_{cp} 的值是原始记录数量的平方,这个数量远多于采用 3 种映射方法的映射-过滤-验证算法生成的 N_{cp} 的数量.更进一步讲,图中也显示了 MFV-I 产生的最终候选结果集的规模最小.这是因为在 3 种映射方法中,单射对于不匹配的数据对有着最强的识别力.MFV-R 产生的候选结果数量与 MFV-I 相差无几.从产生最终候选结果集的规模这个评判指标上看,MFV-O 则是最差的.



(a) N_{mr}, DATING (b) N_{cp}, DATING

Fig.6 Records and candidate pairs comparison

图 6 生成记录数和候选集大小比较

5.3.2 时间开销

图 7 显示了不同算法的时间开销与原始记录数的关系.本实验设置原始记录数量从 5 000 增长到 9 000,步长为 1 000.

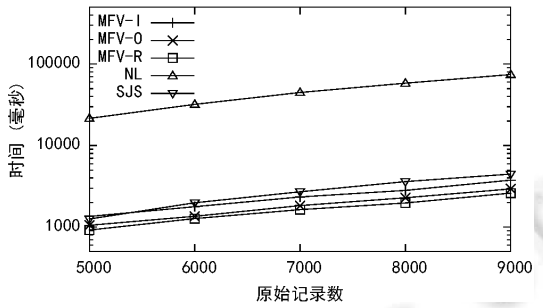


Fig.7 Time cost comparison on DATING

图 7 DATING 数据集时间开销对比

以上实验结果表明:SJS 算法和 3 个 MFV 算法在运行时间上都显著优于嵌套循环(NL)算法.这表明我们提出的用于解决泛化双向相似连接的算法都是非常有效的.在 3 种 MFV 算法中,MFV-R 又表现最好,这个结果显示了优化过的启发式映射方法比单射和等步长映射方法具有更好的效果.

5.3.3 全局阈值和独立阈值比较

直觉上说,独立阈值更能体现用户的个性化需求,例如,能更加准确地描述一个人挑剔的个性或者其相对包容的个性.本节尝试通过客观实验来分析验证设置独立阈值的必要性.

图 8 展示了泛化双向相似连接算法分别采用全局阈值(所有的记录采用相同的阈值)和独立阈值(每条记录都根据自身情况制定独有的阈值)时的对比结果.公平起见,采用独立阈值时,所有阈值的平均值为 0.8;相应地,也将全局阈值设置为 0.8.在 DATING 数据集上,采用独立阈值的结果集包含将近 200 000 对(199 428 对)结果;采用全局阈值的结果集也包括将近 200 000 对(197 958 对)结果.

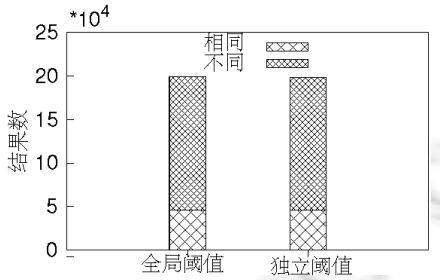


Fig.8 Unified threshold vs. individual thresholds

图 8 全局阈值和独立阈值结果比较

从结果集规模上看,这两个结果集的区别很小,独立阈值似乎可以被全局阈值所取代.值得注意的是,两个结果集中仅有 46 532 对结果是相同的,这个数量还不到全部结果集规模的 1/4.这表明采用两种不同的阈值设定方法会产生多达 150 000 对不同的结果.也就是说,采用全局阈值得到的结果集与采用独立阈值得到的结果集存在相当大的差异.

因此,为了满足真实世界的用户需求,简单地采用全局阈值代替独立阈值是不合适的.这也进一步验证了本文的观点:在相似连接问题中,为了满足不同用户的不同偏好,采用根据用户个人情况制定出的独立阈值是值得尝试的.

5.3.4 模拟数据实验结果分析

由于真实数据集的规模有限,所以我们采用合成数据进行实验以评价所提算法在较大规模数据集上的可扩展性.在合成数据集上的实验结果与在真实数据集上的实验结果有着相似的性质.

图 9 展示了论文所提算法在 L-DATING 数据集上的时间开销.因为前面的实验结果已经表明,MFV-R 算法

在 3 种过滤验证算法中效果最好,所以我们在图中没有展示另两种 MFV 算法.从图中可以看出:实验结果同第 5.3.2 节中的结果一致,SJS 算法和 MFV 算法在时间开销上均远小于 NL 算法.

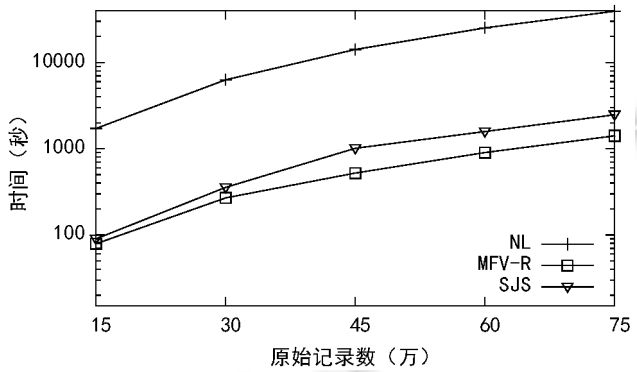
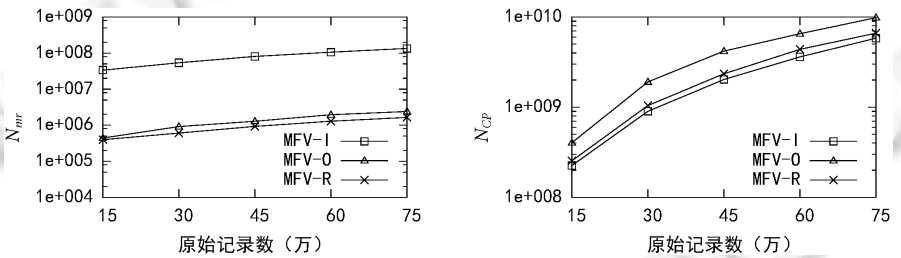


Fig.9 Time cost comparison on L-DATING

图 9 合成数据集时间消耗

图 10 展示了 3 种 MFV 算法在 L-DATING 上的生成符号记录数量和候选集大小.MFV-I 生成符号记录数量最多,过滤效果最好.MFV-O 和 MFV-R 生成符号记录数量相对较少.这二者的区别是:MFV-O 牺牲了很多过滤效果,而 MFV-R 过滤效果接近 MFV-I.



(a) N_{mr}, L-DATING (b) N_{cp}, L-DATING

Fig.10 Records and candidate pairs comparison on L-DATING

图 10 合成数据集上 MFV 算法的生成记录数和候选集大小比较

6 结束语

本文提出一种新的相似连接——泛化双向相似连接.它是对现有的大量相似连接工作的扩展,适用于更加广泛的应用场景,比如交友和求职招聘.同时提出了子连接集算法和映射-过滤-验证(MFV)算法来处理泛化双向相似连接查询,并对于 MFV 算法提出了等步长映射方法和启发式映射方法,以进一步提高算法效率.在真实和合成数据集上的大量实验结果,表明了论文所提算法的正确性和有效性.此外,对实验结果的深入比较分析显示,采用用户独立阈值比采用全局统一阈值更符合用户需求.这也证明了在泛化双向相似连接中考虑独立阈值的必要性.今后将继续改进算法,同时提高算法的可扩展性.

References:

[1] Xiao C, Wang W, Lin XM, Yu JX, Wang GR. Efficient similarity joins for near-duplicate detection. ACM Trans. on Database Systems (TODS), 2011,36(3). [doi: 10.1145/2000824.2000825]

[2] Papapetrou P, Athitsos V, Kollios G, Gunopulos D. Reference-Based alignment in large sequence databases. Proc. of the VLDB Endowment, 2009,2(1):205−216. [doi: 10.14778/1687627.1687651]

[3] Li YN, Patel JM, Terrell A. Wham: A high-throughput sequence alignment method. ACM Trans. on Database Systems (TODS), 2012,37(4):28. [doi: 10.1145/2389241.2389247]

[4] Chaudhuri S, Ganti V, Kaushik R. A primitive operator for similarity joins in data cleaning. In: Proc. of the 22nd Int'l Conf. on Data Engineering (ICDE 2006). 2006. [doi: 10.1109/ICDE.2006.9]

[5] Liu XL, Wang HZ, Li JZ, Gao H. Similarity join algorithm based on entity. Ruan Jian Xue Bao/Journal of Software, 2015,26(6): 1421−1437 (in Chinese with English abstract). http: //www.jos.org.cn/1000-9825/4610.htm [doi: 10.13328/j.cnki.jos.004610]

[6] Arasu A, Ganti V, Kaushik R. Efficient exact set-similarity joins. In: Proc. of the 32nd Int'l Conf. on Very Large Data Bases. 2006. 918−929.

[7] Zhang ZJ, Hadjieleftheriou M, Ooi BC, SrivastavaD. Bed-Tree: An all-purpose index structure for string similarity search based on edit distance. In: Proc. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data. 2010. 915−926. [doi: 10. 1145/1807167. 1807266]

[8] Zhao X, Xiao C, Lin XM, Wang W. Efficient graph similarity joins with edit distance constraints. In: Proc. of the 28th Int'l Conf. on Data Engineering (ICDE 2012). 2012. 834−845. [doi: 10.1109/ICDE.2012.91]

[9] Wang JN, Feng JH, Li GL. Trie-Join: Efficient trie-based string similarity joins with edit-distance constraints. Proc. of the VLDB Endowment, 2010,3(1-2):1219−1230.

[10] Li GL, Deng D, Wang JN, Feng JH. Pass-Join: A partition-based method for similarity joins. Proc. of the VLDB Endowment, 2011, 5(3):253−264.

[11] Li R, Ju L, Peng Z, Yu ZW, Wang CK. Batch text similarity search with MapReduce. In: Proc. of the 13th Asia-Pacific Web Conf. Beijing, 2011. 412−423.

[12] Lu W, Du XY, Hadjieleftheriou MM, Ooi BC. Efficiently supporting edit distance based string similarity search using B+-trees. IEEE Trans. on Knowledge and Data Engineering, 2014,26(12):2983−2996. [doi: 10.1109/TKDE.2014.2309131]

[13] Wang JN, Li GL, Deng D, Zhang Y, Feng JH. Two birds with one stone: An efficient hierarchical framework for top-*k* and threshold-based string similarity search. In: Proc. of the 31st Int'l Conf. on the Data Engineering (ICDE 2015). 2015. 519−530. [doi: 10.1109/ICDE.2015.7113311]

[14] Wang JN, Li GL, Feng JH. Fast-Join: An efficient method for fuzzy token matching based string similarity join. In: Proc. of the 27th Int'l Conf. on the Data Engineering (ICDE 2011). 2011. 458−469. [doi: 10.1109/ICDE.2011.5767865]

[15] Wang W, Qin JB, Xiao C, Lin XM, Shen HT. VChunkJoin: An efficient algorithm for edit similarity joins. IEEE Trans. on Knowledge and Data Engineering, 2013,25(8):1916−1929. [doi: 10.1109/TKDE.2012.79]

[16] Wang CK, Wang JM, Lin XM, Wang W, Wang HX, Li HS, Tian WP, Xu J, Li R. MapDupReducer: Detecting near duplicates over massive datasets. In: Proc. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data. 2010. 1119−1122. [doi: 10.1145/ 1807167.1807296]

[17] Deng D, Li GL, Feng JH. A pivotal prefix based filtering algorithm for string similarity search. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data. 2014. 673−684. [doi: 10.1145/2588555.2593675]

[18] Rong CT, Lu W, Wang XL, Du XY, Chen YG, Tung AKH. Efficient and scalable processing of string similarity join. IEEE Trans. on Knowledge and Data Engineering, 2013,25(10):2217−2230. [doi: 10.1109/TKDE.2012.195]

[19] Deng D, Li GL, Hao S, Wang JN, Feng JH. MassJoin: A MapReduce-based method for scalable string similarity joins. In: Proc. of the 30th Int'l Conf. on the Data Engineering (ICDE 2014). 2014. 340−351. [doi: 10.1109/ICDE.2014.6816663]

[20] Brualdi RA. Introductory Combinatorics. 5th ed., Pearson Education, 2009.

附中文参考文献:

[5] 刘雪莉,王宏志,李建中,高宏.基于实体的相似性连接算法.软件学报,2015,26(6):1421−1437. http://www.jos.org.cn/1000-9825/ 4610.htm [doi: 10.13328/j.cnki.jos.004610]

王昶平(1970－),男,陕西西安人,博士生,主要研究领域为社交网络,数据挖掘.

王朝坤(1976－),男,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为图和社交数据管理,音乐计算,大数据系统.

汪浩(1989－),男,硕士,主要研究领域为相似连接查询处理,时间序列分析.

王萌(1990－),女,硕士,主要研究领域为社交网络分析与挖掘,复杂网络理论.

陈俊(1989－),男,博士生,主要研究领域为数据挖掘,推荐系统,社交网络.