基于属性值分布的关系数据库对象级别检索结果排序算法

邓承刚 张 俊 刘 宁
(大连海事大学信息科学与技术学院 大连 116026)

摘 要 目前关系数据库的相关性排序方法有很多。对象级别检索可以更好地将分散在各个元组中的信息进行整合,得到完整的信息。由于每个对象是唯一的,对象之间的区别不仅仅是通过关键词来体现,而且是通过它们包含的属性值来体现的。因此介绍的方法通过统计包含关键词的对象中的属性值的分布情况,运用信息熵的方法为每个属性分配权值,由此计算每个对象针对单关键词的相关性得分,进而以对象为单位将针对每个关键词的得分求和,得到最终的排序得分。

关键词 关系数据库,对象级别,排序算法,属性值,信息熵
中图法分类号 TP311.132 **文献标识码** A

Attribute-Value-Distribution Based Result Ranking Algorithm for Object-level Keyword Search over Relational Databases

DENG Cheng-gang ZHANG Jun LIU Ning
(Department of Information Science and Technology, Dalian Maritime University, Dalian 116026, China)

Abstract Currently, there are many ranking methods sorted by correlation in relational databases. Object-level retrieval can integrate information which is dispersed in many tuples into complete information. As each object is unique, the difference among the objects is not only embodied in the keywords, but also embodied in the attribute values which they contain. Thus, this article described a method which gathers statistics of the distribution of attribute values in the object which contains the keyword, and makes use of the information entropy method to assign weights for each attribute value to calculate the correlation score of each tuple to each keyword. In the end, the object's score for each keyword is summed up to get the final ranking score for the object.

Keywords Relational databases, Object-level, Ranking algorithm, Attribute value, Information entropy

1 引言

随着互联网的发展,以Google为代表的网页搜索引擎取得了巨大的成功,用户可以用简单的关键词找到自己需要的信息。而关系数据库是当前数据库的主流形式,它采用结构化查询语言进行内容检索,并要求用户掌握一定的查询语言和数据库模式知识。由此就产生了一个很自然的需求,即让关系数据库支持高效的关键词查询[1]。相比于网络搜索引擎,关系数据库关键词检索又有了新的特点:

(1) 元组之间拥有语义关系。即连接表之间的主外键是有类型的,不同类型的语义强度是不一样的。

(2) 数据库中的属性值隐藏着等价和传递关系[2]。

(3) 数据库中的文本都是短文本[3-6],一些信息检索的方法并不适合于关系数据库。例如基于词频-逆文档频率的模型算法并不适合于短文本的计算方法,需要研究一种适合于关系数据库本身特点的短文本相关性计算方法[4-6]。

直接在元组上进行关键词检索有一定的缺陷。因为这类数据库查询是在通过主外键联系起来的表之间进行的,这样得到的信息是分散的,包含关键词的信息可能会分散在不同的表里,从而造成相关信息的丢失。而对象级别的数据库查询就是要把分散在各表中的信息以对象为单位进行整合。对于构建好的对象,除了要考虑通过主外键相连这个外部联系外,还需要考虑隐藏的属性值之间的内部联系。而后者也是本文研究的重点。

相关性排序算法就是按照一定的方法计算检索结果与用户查询之间的相关性,然后排序检索结果,尽可能把与用户查询最相关的检索结果排在前面。相关性排序算法可以融合在检索算法中,也可以作为一个独立过程执行。在关系数据库中,通过关键词检索出与检索词相关的对象信息,并进行排序筛选,选出最相关的信息。

本文提出的方法首先要找到属性值之间的传递关系。在一定程度上,一个属性值出现的次数越多,则该属性值可能与关键词的联系会越紧密一些。利用信息熵[2]的方法为属性分配权值。信息熵的大小与数据分布的情况有密切的联系,可

到稿日期:2012-06-03 返修日期:2012-09-21 本文受国家自然科学基金面上项目(61073057,60972090),中央高校基本科研业务费专项资金项目(2011JC007)资助。

邓承刚(1986—),男,硕士生,CCF学生会员,主要研究方向为数据库与信息检索,E-mail:dcg19861205@126.com;张 俊(1971—),男,博士,副教授,CCF会员,主要研究方向为数据库与信息检索;刘 宁(1956—),女,硕士,教授,主要研究方向为智能信息处理。

以通过计算信息熵来反映当前属性值分布的情况，进而找到最相关的属性值和不相关的属性值。最终，通过相关性得分进行排序。

根据以上的讨论可知，相比于之前的方法，本文侧重于研究对象内部属性值相对于查询关键词之间的关系，更适合于区分包含相同关键词的对象之间的区别。因适合于进行相关性的排序，对象级别相关性排序（Object-level Correlation Sort）简称 OCS，这也是本文提出的算法的名称。

表 1 描述的是一个关于手机的数据库 D。表中的每一个元组都代表一款手机的详细信息。每一个元组有时间、制造商、型号、颜色、容量 5 个属性。t_1 到 t_5 是通过查询“IPHONE”返回的检索结果，这几个检索结果并没有进行排序。通过观察可以发现，制造商这个属性中的属性值均为“苹果”。在这个简单的例子中，两个属性值（IPHONE 与苹果）之间在统计学上有着很强的相关性。基于这种想法，本文研究了通过数据统计来计算属性值之间的相关性，并在此基础上进行排序。

表 1　手机数据库的实例

	时间	制造商	型号	颜色	容量
t_1	2010	苹果	IPHONE	黑色	32G
t_2	2011	苹果	IPHONE	黑色	16G
t_3	2010	苹果	IPHONE	黑色	8G
t_4	2010	苹果	IPHONE	白色	32G
t_5	2010	苹果	IPHONE	白色	16G

2　相关工作

为了帮助用户能够更方便地在数据库中检索，至今研究者们在关系数据库关键词检索领域开展了大量的研究工作。Retune[7] 通过主外键关联构造对象，利用词频逆文档频率方法和结构紧密度方法计分排序。BANKS[8,9] 通过主外键关联构造连接树，分别计算点和边的权值得到相关性结果。Object Summary[10-12] 以数据主题图的形式将数据库中的元组联合起来形成对象，再进行排序。本文提出的 OCS 算法侧重于通过研究关系数据库中的属性和属性值来对检索结果进行相关性排序。OCS 算法的研究意义主要有：在进行相关性排序的时候，如果两个或多个对象包含相同的关键词，那么它们之间的区别是无法通过传统的方法区分的。然而每个对象都是唯一的，都是与其它对象不相同的，即使都包含着相同数量的关键词，它们之间也是有差别的，而这种差别是通过对象中的其它属性值来体现的。算法正是通过考虑对象中的属性值的情况来区分对象之间的差别。

3　关系数据库对象构建

由关键词检索得到的元组是一个个单独的包含关键词的元组，它并没有表现出元组之间的关系，一些与该元组相关的信息就会丢失，这样得到的信息是分散的，因为包含关键词的信息可能会分散在不同的表里。而对象构建就是要把分散在各表中的信息进行整合，以对象为单位进行组织。使数据结构化，不仅有元组内部的描述，而且有元组之间的相互关系，更符合关系数据库的数据结构化的特点。

对象的构建，首先需要统计包含关键词的元组所属于的数据表。其次，将这些数据表按照数据库的模式图通过主外键联系起来，得到一个完整的数据图，这张图包含了数据表之间的联系，也就得到了元组之间的联系，由此就可以保留那些不包含关键词但却相关的信息。最后，将包含关键词的元组按照得到的数据图的模式来组织，得到一个个相关的对象。

4　排序算法概述

4.1　数据统计

数据统计[2] 是表示某一地理区域自然经济要素特征、规模、结构、水平等指标的数据，是定性、定位和定量统计分析的方法。表达形式有统计表格和统计地图两种。OCS 算法中的数据统计方法就是对关系数据库中进行关键词检索得到的元组中的各项属性值进行统计，对各个属性值的分布进行分析，并依据分析结果进行排序的方法。

例如，对表 1 中数据进行统计，表 1 中的 t_1 到 t_5 是通过输入“IPHONE”得到的一个对象集合。对其进行数据统计可以得到对象集合中所包含的属性值及其出现次数，结果为：(IPHONE,5),(2010,4),(2011,1),(苹果,5),(黑色,3),(白色,2),(32G,2),(16G,2),(8G,1)。

4.2　相关性

相关性是指搜索词和页面的相关程度[13]。在关系数据库中，相关性是指查询关键词与对象的相关程度。OCS 算法的相关性的衡量标准主要包括两方面：首先是由属性值的相互关系推出属性的相互关系；其次是基于信息熵来为属性分配权值。

4.2.1　属性值之间的传递关系

OCS 算法对包含关键词的对象中的其它属性值的分配情况进行统计，针对每列属性观察属性值是单一还是多样，以及每个值出现的频率，来决定与关键属性值最相关的属性列和属性值。如果是一一对应关系，那就是最理想的等价关系，而一般情况是具有隐藏的传递关系。OCS 算法为关系最紧密的属性分配较高的权值。

所以，属性权值的分配是依据非关键属性值的分布状态进行的。非关键属性值指的是不包含关键词的属性，那么关键属性指的就是关键词所在的属性。包含关键词的属性的权值为 1。更进一步，每次进行关键词检索时，针对每一个关键词，可分别进行单独统计，进而挖掘出隐藏的属性的关系。请注意，不同的关键词所对应的相关属性是不一样的，即同一属性值在对应不同的关键词时，其重要程度也是不一样的。

针对每一个对象，通过数据统计得到的信息来判别每个属性值与查询关键词的相关性。针对非关键属性，通过计算每个对象中的每个属性值本身的相关程度，可以得到每个对象的相关性得分。注意：每个属性值的相关程度是通过对所有包含关键词的对象进行统计得到的，而最后计算每个对象组的得分只需要进行后期的相加求和。若该对象包含关键词，则计算公式为：

$$Corr(k_i,ti_z)=1+\sum_{Au\in X-Ak_i} W_{Au}\cdot F(ti_z[Au],k_i) \quad (1)$$

式中，1 代表关键属性中包含关键词。若对象中不包含关键词，那么相关性计算公式为：

$$Corr(k_i,ti_z)=\sum_{Au\in X-Ak_i} W_{Au}\cdot F(ti_z[Au],k_i) \quad (2)$$

因为在这类对象的关键属性列中不包含关键词，关键属性列的得分即为 0。除此之外，其他的符号：Ak_i 是关键词 k_i 所属的属性，Au 是非关键属性，W_{Au} 是相应的权值，$tiz[Au]$

是元组 ti_z 在属性 Au 中的值，$F(x,y)$是衡量 x 与 y 之间相关性的函数。

$F(x,y)$主要涉及两部分的计算，包括关键词的词频和相关度。下面将分别进行详细解释。

(1) 词频

给定一个关键词 k_i，通过检索得到直接包含 k_i 的元组集 U_i，共有元组 x 个 $T_i=\{t_{i1},t_{i2},\cdots,t_{ix}\}$。对元组集 U_i 进行数据统计，总共有 u 个不同的非关键属性值 $A=\{a_1,a_2,\cdots,a_u\}$。定义每个非关键属性值在相关对象集中出现的次数为 f_j，f_j 越大，表明该属性值相对于关键词越重要。f_j 可能会比较大，此处对其进行处理，得到关于 tf_j 的公式：

$$tf_j=1+\ln f_j \tag{3}$$

(2) 相关度

再来考虑相关度(Relevancy)的影响。相关度是指两个事物间存在相互联系的百分比。通过相关度可以衡量属性值与关键词之间的相关程度。首先通过关键词检索，得到包含关键词的元组的集合，其是一个经过筛选后的“文档集”，每一个元组都不同程度地与关键词相关，所以，属性值 j 出现的次数越大，与关键词的相关度就越大。可以采用相关度来表示属性值的相关程度。相关度定义为：

$$Relevancy_j=\frac{tf_j}{x} \tag{4}$$

式中，tf_j 为词频，查询关键词所在的元组数 x。很明显 $0<Relevancy_j\leqslant 1$，$Relevancy_j$ 越大，它对应的属性值与关键词越相关。

由此得到 $F(x,y)$的函数公式：

$$F(a_j,k_i)=Relevancy_j=\frac{(1+\ln f_j)}{x} \tag{5}$$

4.2.2 基于信息熵分配属性的权值

信息论之父 C. E. Shannon 在 1948 年发表的论文“通信的数学理论(A Mathematical Theory of Communication)”中指出，任何信息都存在冗余，冗余大小与信息中每个符号(数字、字母或单词)的出现概率或者不确定性有关。Shannon 借鉴了热力学的概念，把信息中排除了冗余后的平均信息量称为“信息熵”，并给出了计算信息熵的数学表达式。本文利用信息熵来衡量属性值对查询的重要性。

在求信息熵的时候是基于属性值的出现频率的，之前的方法将词频的计算直接应用在全部的元组集合当中。OCS 算法是统计出通过关键词查询得到的检索出的对象，将这些对象构造成一个整体，称之为相关对象集。统计相关对象集中每个属性值出现的频率，以此来进行权值的计算。在相关对象集这个范围内，当一种信息出现的概率更高时，表明它传播得更广泛，或者说，被引用的程度更高。所以可以用其来衡量属性值相对于关键词的相关程度。

设 V 是一个有限集合上的离散随机变量，表示为：$V=\{v_1,v_2,\cdots,v_n\}$。$P(v_i)$是 v_i 的概率。可以把 v_i 看作是在属性 A 上的属性值分配，同一属性中的属性值之间是相互独立的。通过统计属性值出现的频数来计算属性值 v_i 的概率 $P(v_i)$，则在属性 A 上的信息熵的计算公式如下：

$$E(A)=-\sum_{v_i\in t_i[A]}^{n}p(v_i)\log_b p(v_i) \tag{6}$$

根据香农的信息论理论，当变量的不确定性越大时，信息熵也就越大，把它搞清楚所需要的信息量也就越大；当变量确定时，人们不需要任何信息量就可以得到确定的结果，因此信息熵也就越小。在进行属性权值的计算中，属性值的分布越不均匀，信息熵越小，某个属性值发生的可能性越大。因此在为属性设定权值的时候应该取 $1/E(A)$，考虑到 $E(A)$可以取 0 值，将其改进为 $1/(1+E(A))$。

4.2.3 相关性得分计算

综上所述，相关性与属性值的分布有着重要的联系。不论是 $F(x,y)$函数还是属性的权值，都是通过对属性的统计得来的。OCS 算法得到最终的相关性得分表达式。当对象中包含关键词时，具体的公式为：

$$Corr(k_i,ti_z)=1+\sum_{Au\in X-Ak_i}\frac{1}{1+E(Au)}\frac{(1+\ln f_j)}{x} \tag{7}$$

式中，Au 是非关键词的属性值。当对象中不包含关键词时，具体的公式为：

$$Corr(k_i,ti_z)=\sum_{Au\in X-Ak_i}\frac{1}{1+E(Au)}\frac{(1+\ln f_j)}{x} \tag{8}$$

例如：利用此公式计算表 1 中的 t_1 与苹果之间的相关性得分。首先统计出 Au={iphone,2010,黑色,32G}，而每一项出现的次数为 5、4、3、2。首先分别计算出每一项的词频为 1+ln5、1+ln4、1+ln3、1+ln2，以及相关度为 0.52、0.48、0.42、0.34。接下来计算属性的权值，根据数据统计，得到属性权值为 1、0.909、0.774、0.458。因为 t_1 包含关键词，用式(8)以及以上数据得到 t_1 与“苹果”关键词之间的相关性得分为 2.43712。

4.3 排序得分计算

通过上文的分析，OCS 算法在计算排序的得分时，每个对象的得分是由相关性计算得到的。得到的公式为：

$$Score(Q,t)=Corr(K,t) \tag{9}$$

因此，每个对象关于 k_i 的得分公式为：

$$Score(k_i,ti_z)=Corr(k_i,ti_z) \tag{10}$$

然而，这只是对单关键词 k_i，最终需要将每个对象针对每个关键词的得分相加，得到对象关于关键词查询的相关性得分。整体求和公式为：

$$Score(K,t_h)=\sum_{i=1}^{s}Score(k_i,t_h) \tag{11}$$

按照此得分进行排序，得到最终结果。与 Retune 算法的计算公式相比较，主要有以下几点区别：

(1) Retune 算法考虑的是检索关键词出现的频率以及逆文档频率。而 OCS 算法不仅考虑对象中是否包含关键词，而且需要考虑对象中元组结点的属性值分布。

(2) OCS 算法通过统计每个属性列中属性值的分布，利用信息熵的方法为属性列分配权值。当对象中包含的关键词相同的时候，可以通过其它属性值的不同来区分这类对象的区别。每个对象由于都是唯一的，因此必定是有区别的。而 Retune 可以很好地检索出包含关键词的对象，但是不能区分包含相同关键词的对象的区别。

(3) OCS 算法除了要进行关键词检索外，还需要统计属性值的信息，因此在时间上会差于 Retune。

5 OCS 算法框架

OCS 算法的总体框架分为 3 个模块，如图 1 所示。其中 1 代表输入以及预处理模块，2 代表处理模块，3 代表输出模块。

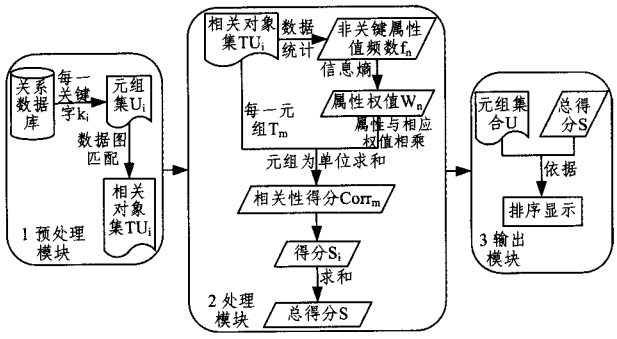


图 1　OCS 算法框架

5.1　OCS 算法流程

设关系数据库中有 n 个元组 $T=\{t_1,t_2,\cdots,t_n\}$，每个元组中有 m 个属性 $X=\{A_1,A_2,\cdots,A_m\}$。用户输入关键词序列 $K=\{k_1,k_2,k_3,\cdots,k_s\}$，针对每一个关键词 $k_i(0<i\leqslant s)$，数据库都会检索出只包含 k_i 的元组集合 U_i，共有 x 个元组 $T_i=\{ti_1,ti_2,\cdots,ti_x\}$。接下来的排序都是在此集合上进行的。对元组集 U_i 进行数据统计，总共有 u 个不同的非关键属性值，$A=\{a_1,a_2,\cdots,a_u\}$，得到非关键属性值的频数 $f_j(0<j\leqslant u)$。对 f_j 进行操作，通过 f_j 来计算相对属性的信息熵来为属性分配权值 W_k，与每一元组相乘得到相关性得分 $Corrz(0<z\leqslant x)$，最终通过相关性得分得到每个对象针对 k_i 的得分 $S_{hi}(0<h\leqslant n)$，通过求和得到总得分 S_h，并根据得分 S_h 进行排序。具体的系统流程图如图 2 所示。

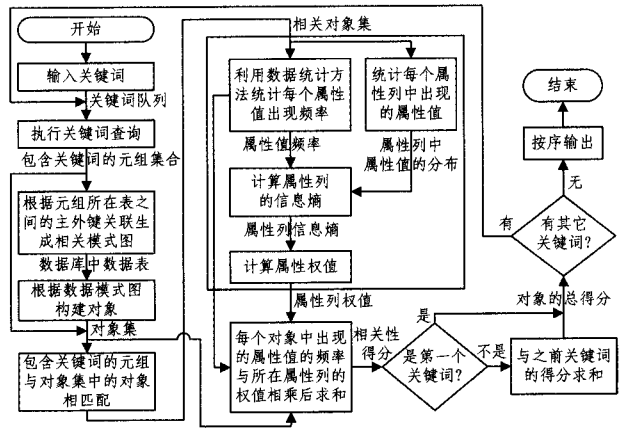


图 2　OCS 算法流程图

5.2　OCS 算法设计

根据具体的计算流程，得到 OCS 算法的具体算法伪代码，如图 3 所示。

SCORE(tu_i)
1. for each query term k_n
2. do retrieval tuple which conclude k_n
3. for each tuple t_i
4. do build the Object-level tupleunit tu_i
5. float Scores[tu_i]=0
6. do statistics a_j which was concluded in tu and calculate f_{aj}
7. for each Attribute Value a_j
8. do calculate $tf_{aj}=1+\ln f_{aj}$
9. do calculate $Relevancy_{aj}=tf_{aj}/i$
10. do calculate $F(a_j,k_j)=Relevancy_{aj}$
11. for each Attribute A_m
12. do statistics the Attribute Value's distribution
13. do calculate the Information Entropy $E(A_m)$
14. do calculate the Attribute Weights $W(A_m)=1/[1+E(A_m)]$
15. for each TupleUnit tu_i
16. do calculate the Correlation Scores $Corr(k_n,tu_i)$
17. for each TupleUnit tu_i
18. do calculate the Scores $Scores(K,tu_i)$
19. return Top K components of $Scores(K,tu_i)$

图 3　OCS 算法伪代码

6　算法对比

在这一节中，将通过一个简单的例子来对比 OCS 算法和 Retune 算法的区别。

表 2 是一个关于手机的数据库表。具体包含 5 个元组，每个元组包含 4 个属性。在该数据库上进行关键词为“iphone4”的查询。根据表中元组的情况，设定 5 个元组中有 4 个是相关的，其中又有 3 个是包含关键词的直接相关，有 1 个是没有包含关键词的间接相关。下面通过两种方法来计算，验证与预期结果的一致性。

表 2　手机数据库

	制造商	型号	颜色	容量
1	苹果	IPHONE4	白色	16G
2	苹果	IPHONE4	黑色	16G
3	苹果	IPHONE4	白色	8G
4	诺基亚	N8	黑色	8G
5	苹果	IPHONE4S	白色	16G

首先用 Retune 算法来计算。Retune 算法是基于词频逆文档频率的方法来计算查询得分的，它的计算公式如下：

$$SCORE_{IR}(k_i,u)=\frac{(1+\ln(1+tf(k_i,u)))*\ln(idf(k_i))}{(1-s)+s*ntl(u)} \tag{12}$$

式中，s 是一个常量，一般设置为 0.2。$idf(k_i)$ 称为 k_i 的逆文档频率，满足：

$$idf(k_i)=\frac{p+1}{O_{k_i}+1} \tag{13}$$

式中，O_{k_i} 是包含关键词 k_i 的对象的个数，$ntl(u)$ 称为标准化词项长度，满足：

$$ntl(u)=\frac{|u|}{\frac{\sum_{u'\in U}|u'|}{n}} \tag{14}$$

式中，$|u|$ 是 u 中词项的个数。

根据以上公式计算每个元组的最终得分并排序，如表 3 所列。

表 3　Retune 排序结果

	制造商	型号	颜色	容量	得分
1	苹果	IPHONE4	白色	16G	0.69
2	苹果	IPHONE4	黑色	16G	0.69
3	苹果	IPHONE4	白色	8G	0.69
4	诺基亚	N8	黑色	8G	0
5	苹果	IPHONE4S	白色	16G	0

通过结果可以发现，Retune 只能为包含关键词的元组进行计算，不能为不包含关键词的元组进行计算，并且包含相同关键词的元组的得分是相等的，没有体现出元组之间的差别。

接下来用 OCS 算法对该数据表进行计算。首先对包含关键词的集合进行数据统计，得到集合中所包含的属性值及其出现次数，结果为：(IPHONE4,3)，(苹果,3)，(黑色,1)，(白色,2)，(16G,2)，(8G,1)。由此得到每个非关键属性值的

tf={(苹果,2.1),(黑色,1),(白色,1.7),(16G,1.7),(8G,1)},以及每个非关键属性值的相关度:{(苹果,0.7),(黑色,0.33),(白色,0.57),(16G,0.57),(8G,0.33)}。其次,计算每个属性列的信息熵与权值分别为:{(制造商,0),(颜色,0.28),(型号,0.28)},{(制造商,1),(颜色,0.79),(型号,0.79)}。最后由以上数据计算得到5个元组的得分:{(1,2.60),(2,2.41),(3,2.41),(4,0.52),(5,1.60)}。最终的排序结果如表4所列。

表4 OCS排序结果

	制造商	型号	颜色	容量	得分
1	苹果	IPHONE4	白色	16G	2.60
2	苹果	IPHONE4	黑色	16G	2.41
3	苹果	IPHONE4	白色	8G	2.41
5	苹果	IPHONE4S	白色	16G	1.60
4	诺基亚	N8	黑色	8G	0.52

该算法成功地将不包含关键词"IPHONE4"但包含"IPHONE4S"的元组排到了第四位,因为它和包含关键词的元组都是由制造商为"苹果"的公司制造的,这也符合用户的查询需求。与Retune算法相比,其具有以下几点优势:

(1)能够检索出不包含关键词却又相关的元组。

(2)能够区别出包含相同关键词的元组。

7 实验结果

该算法是在MySQL上开发的,采用Java语言编写。实验测评是在自己构建的数据集上进行的,该数据集按照图4的模式分解成4个关系:Customers中有5000个元组;Phones中有100个元组;Communication Company中有30个元组;Use中有5267个元组。Customers、Phones和Communication Company上建立了全文索引,所有的外码和数字属性上也都建立了索引,实验是在一台1G内存的Core2 T5750 2.00GHz的计算机上进行的,操作系统是Windows XP。

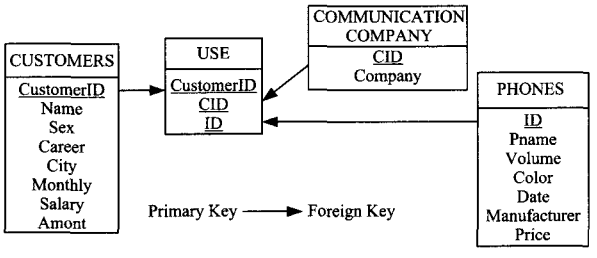


图4 数据库模式图

为了显示OCS算法所得结果的特点,采用Retune方法与该方法进行对比。同时为了突出结果的效果,分别采用两组策略同时进行检索排序,以进行更细微的对比。第一组策略是针对对象集中的"姓名、行业、居住地、手机型号、制造商、通信公司"共6个属性域进行统计;第二组策略在第一组的基础上加入了"性别的因素",这样可以凸显"性别"对检索结果排序的影响。

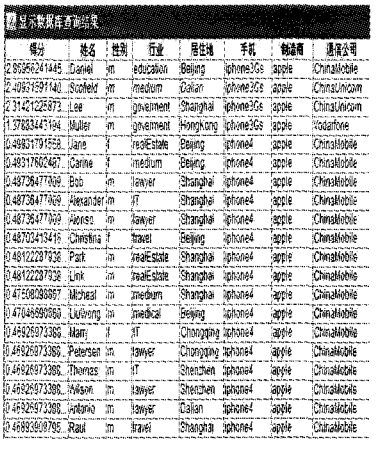
根据构建好的对象集,针对多关键词"iphone3gs,chinamobile"进行检索,检索结果都只取前20个结果,并对结果进行对比分析。

7.1 检索结果对比分析

该实验对比分别用Retune方法和OCS算法的两种策略在对象集上进行不同属性多关键词"iphone3gs,chinamobile"的检索排序,得到以下3组结果,分别如图5—图7所示。

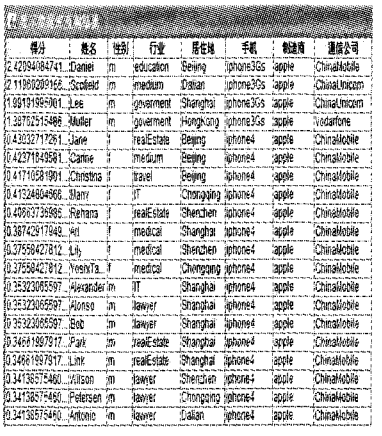
得分	姓名	性别	行业	居住地	手机	制造商	通信公司
5.37300397633...	Daniel	m	education	Beijing	iphone3Gs	apple	ChinaMobile
5.21102096575...	Lee	m	government	Shanghai	iphone3Gs	apple	ChinaUnicom
5.21102096575...	Muller	m	government	Hongkong	iphone3Gs	apple	Vodafone
5.21102096575...	Scofield	m	medium	Dalian	iphone3Gs	apple	ChinaUnicom
1.16198039057...	John	m	education	Beijing	I8700	samsung	ChinaMobile
1.16198039057...	Rehana	f	realEstate	Shenzhen	iphone4	apple	ChinaMobile
1.16198039057...	Park	m	realEstate	Shanghai	iphone4	apple	ChinaMobile
1.16198039057...	Wilson	m	lawyer	Shenzhen	iphone4	apple	ChinaMobile
1.16198039057...	Jean	f	education	Shanghai	I9100	samsung	ChinaMobile
1.16198039057...	Alexander	m	IT	Shanghai	iphone4	apple	ChinaMobile
1.16198039057...	Bush	m	travel	Shenzhen	iphone4	apple	ChinaMobile
1.16198039057...	YoshiTa...	f	medical	Chongqing	iphone4	apple	ChinaMobile
1.16198039057...	Alonso	m	lawyer	Shanghai	I9100	samsung	ChinaMobile
1.16198039057...	Pedro	m	medical	Xian	N8	nokia	ChinaMobile
1.16198039057...	Kaka	m	lawyer	Beijing	N8	nokia	ChinaMobile
1.16198039057...	Manuel	m	travel	Beijing	G10	htc	ChinaMobile
1.16198039057...	Marie	m	manufacture	Dalian	E63	nokia	ChinaMobile
1.16198039057...	Carlos	m	education	Shenzhen	N9	nokia	ChinaMobile
1.16198039057...	Raul	m	travel	Shanghai	iphone4	apple	ChinaMobile
1.16198039057...	Philip	f	medium	Hongkong	N9	nokia	ChinaMobile

图5 Retune—"iphone3gs chinamobile"截图



得分	姓名	性别	行业	居住地	手机	制造商	通信公司
2.86956241445...	Daniel	m	education	Beijing	iphone3Gs	apple	ChinaMobile
2.40637997140...	Scofield	m	medium	Dalian	iphone3Gs	apple	ChinaUnicom
2.31421225873...	Lee	m	government	Shanghai	iphone3Gs	apple	ChinaUnicom
1.37683447194...	Muller	m	government	Hongkong	iphone3Gs	apple	Vodafone
0.49831791958...	Jane	f	realEstate	Beijing	iphone4	apple	ChinaMobile
0.48317602467...	Carine	f	medium	Beijing	iphone4	apple	ChinaMobile
0.48736477009...	Bob	m	lawyer	Shanghai	iphone4	apple	ChinaMobile
0.48736477009...	Alexander	m	IT	Shanghai	iphone4	apple	ChinaMobile
0.48736477009...	Alonso	m	lawyer	Shanghai	iphone4	apple	ChinaMobile
0.48703413416...	Christina	f	travel	Beijing	iphone4	apple	ChinaMobile
0.48122287936...	Park	m	realEstate	Shanghai	iphone4	apple	ChinaMobile
0.48122287936...	Link	m	realEstate	Shanghai	iphone4	apple	ChinaMobile
0.47598088867...	Michael	m	medium	Shanghai	iphone4	apple	ChinaMobile
0.47046998868...	LiuYong	m	medical	Beijing	iphone4	apple	ChinaMobile
0.46926973386...	Mary	f	IT	Chongqing	iphone4	apple	ChinaMobile
0.46926973386...	Petersen	m	lawyer	Chongqing	iphone4	apple	ChinaMobile
0.46926973386...	Thomas	m	IT	Shenzhen	iphone4	apple	ChinaMobile
0.46926973386...	Wilson	m	lawyer	Shenzhen	iphone4	apple	ChinaMobile
0.46926973386...	Antonio	m	lawyer	Dalian	iphone4	apple	ChinaMobile
0.46893909795...	Raul	m	travel	Shanghai	iphone4	apple	ChinaMobile

图6 策略1—"iphone3gs chinamobile"截图



得分	姓名	性别	行业	居住地	手机	制造商	通信公司
2.42094984741...	Daniel	m	education	Beijing	iphone3Gs	apple	ChinaMobile
2.11980289168...	Scofield	m	medium	Dalian	iphone3Gs	apple	ChinaUnicom
1.99101996961...	Lee	m	government	Shanghai	iphone3Gs	apple	ChinaUnicom
1.38762514986...	Muller	m	government	Hongkong	iphone3Gs	apple	Vodafone
0.49032717261...	Jane	f	realEstate	Beijing	iphone4	apple	ChinaMobile
0.42271849581...	Carine	f	medium	Beijing	iphone4	apple	ChinaMobile
0.41710681901...	Christina	f	travel	Beijing	iphone4	apple	ChinaMobile
0.41324804966...	Mary	f	IT	Chongqing	iphone4	apple	ChinaMobile
0.40663735996...	Rehana	f	realEstate	Shenzhen	iphone4	apple	ChinaMobile
0.38742917949...	Ari	f	medical	Shanghai	iphone4	apple	ChinaMobile
0.37558427812...	Lily	f	medical	Shenzhen	iphone4	apple	ChinaMobile
0.37558427812...	YoshiTa...	f	medical	Chongqing	iphone4	apple	ChinaMobile
0.35323065597...	Alexander	m	IT	Shanghai	iphone4	apple	ChinaMobile
0.35323065597...	Alonso	m	lawyer	Shanghai	iphone4	apple	ChinaMobile
0.35323065597...	Bob	m	lawyer	Shanghai	iphone4	apple	ChinaMobile
0.34661997817...	Park	m	realEstate	Shanghai	iphone4	apple	ChinaMobile
0.34661997817...	Link	m	realEstate	Shanghai	iphone4	apple	ChinaMobile
0.34136575460...	Wilson	m	lawyer	Shenzhen	iphone4	apple	ChinaMobile
0.34136575460...	Petersen	m	lawyer	Chongqing	iphone4	apple	ChinaMobile
0.34136575460...	Antonio	m	lawyer	Dalian	iphone4	apple	ChinaMobile

图7 策略2—"iphone3gs chinamobile"截图

通过观察不难发现一些区别。首先对比图5和图6。在图5中,这20个检索排序结果都包含关键词,实现了关键词的检索功能,首先得到的是包含两个关键词的对象,因为该对象只有一个,所以起到了检索的目的。但观察其它仅包含关键词的对象的得分可以发现,包含相同的关键词的对象得分一样,相互之间没有区别。而它们之间的先后顺序是对象集中的相对先后顺序。包含"iphone3gs"的对象排到了包含关键词"chinamobile"的前面,因为"iphone3gs"出现的次数要少于"chinamobile"。因此Retune方法只实现了关键词的检索,但是没有实现检索结果的排序。接下来再观察图6,发现OCS算法和Retune一样,检索结果都是包含关键词的对象,同样也实现了关键词的检索。但与Retune不同也是最重要

的区别就是每个对象之间的得分是不一样的。同时，可以通过数据统计的方法把相对重要的结果排到结果的前面，注意观察图 6，属性制造商与关键词“iphone3gs”所在的属性相关性最高，通过统计将制造商为“apple”的对象排在了相对靠前的位置。其次，因为手机“iphone3gs”首先与“apple”直接相关，间接的每个“iphone4”手机的制造商都是“apple”。因此在仅包含“chinamobile”的对象中，将手机为“iphone4”的对象排到了结果的前面。这样挖掘出了属性与关键词的间接关系。表达的意思是一般应用“iphone4”的用户和应用“iphone3gs”的用户有某些共同点。

为了发现包含在属性中的隐藏关系，观察图 6 和图 7。发现图 7 的结果与图 6 不同的是，排在前面的结果都是性别为“f”的对象。这是因为性别这个属性只有两个属性值，属性值的变化不大，该属性的信息熵比较小，该属性的权值就比较大，因此对结果的影响也就比较大。这说明用“iphone4”手机的女用户比男用户多，因此排在了结果的前面。

7.2 查准率曲线图

通过细节上的对比可以看出，OCS 算法在结果效果上要优于 Retune 方法。接下来，从查准率上对两种方法进行评估。

这里构造了 20 个查询，这些查询都是查询手机型号与通信公司有关的对象。在实验中，采用的是两个关键词的检索，与单关键词相比，这样可以更好地筛选出相关的对象。通过对两种排序方法进行测试，取这些测试结果的查准率平均值，得到如图 8 所示的查准率平均值曲线图。

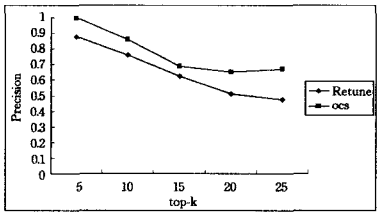


图 8 Top-k 平均查准率对比曲线图

通过曲线图可以发现，OCS 方法的平均查准率比 Retune 的要好。但是随着结果集的逐渐增大，OCS 方法的平均查准率趋于平稳，优于 Retune 的平均查准率。所以当结果集逐渐增大的时候，OCS 方法能更有效地检索出相关的对象。这是因为 OCS 算法可以通过属性值之间隐藏的关系找到与关键词相关的但是不包含关键词的对象。这一点是 Retune 所不具备的。因此在检索效果上，OCS 算法会优于 Retune 算法。

接下来，通过平均化技术来衡量排序算法的排序性能。平均准确率为每个查询的相关排序结果赋予一个评价数字，从多个查询中总结该排序算法性能的最简单的途径就是对这些数字进行平均。这种评价方法称为平均准确率(Mean Average Precision，MAP)[14]。按照该方法对 OCS 和 Retune 分别计算，得到的曲线图如图 9 所示。

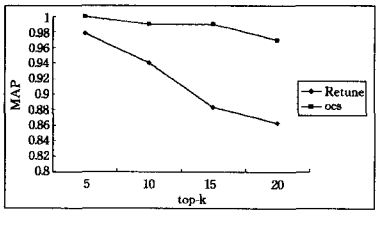


图 9 Top-k 的 MAP 对比曲线图

通过曲线图可以发现，OCS 的 MAP 要大于 Retune。OCS 在结果集较大的时候可以得到更多的相关对象，因为在结果集中存在一些不直接包含关键词但间接相关的对象。OCS 算法可以将其排到结果的前列。而 Rutune 算法只能将包含关键词的结果排到前面，那么那些不包含关键词的相关对象就会丢失。

结束语 本文提出了一种在关系数据库上进行排序的方法。该方法通过统计对象中的属性值分布情况进行权值分配和进一步查询。与 Retune 比较起来，其在关键词检索时可以划分出包含相同关键词的对象之间的区别，因为每一个对象都是唯一的，所以每个对象必定存在差异，得分也就必定与其它对象不同。

计划在未来的工作中完成以下工作：提高算法的效率，使系统能够更快地响应用户的请求；另外可以实现对属性值更细化的分析，将考虑数值型属性值的评判方法，尤其是在商业数据库中。例如一个关于超市的数据库中，出现频率大并且属性值大，即购买次数多并且每次购买的金额大的顾客肯定是重要的顾客；相反则不重要。然而对于那些出现频率多但是单值较小的以及单值较大但出现频率小的顾客该怎么衡量，他们的相关性该怎么评判，是接下来需要研究的问题。

参 考 文 献

[1] 林子雨，杨冬青，王腾蛟，等. 基于关系数据库的关键词查询[J]. 软件学报，2010(10)：2454-2476

[2] Park J，Lee S-G. Ranking Objects Based on Attribute Value Correlation[C]//DEXA. 2010：346-359

[3] 龚才春. 短算的关键技术研究[D]. 北京：中国科学院研究生院(计算技术研究所)，2008

[4] Ruíz N M，Martín-Bautista M J，de Reyes M A P，et al. Enhancing Short Text Retrieval in Databases[C]//FQAS. 2006：613-624

[5] 柴春梅. 互联网短文本信息分类关键技术研究[D]. 上海：上海交通大学，2009

[6] Oliva J，Serrano J I，del Castillo M D. SyMSS：A syntax-based measure for short-text semantic similarity[J]. Data & Knowle-dge Engineeringm，2011，70(4)：390-405

[7] Li Guol-iang，Feng Jian-hua，Zhou Li-zhu. Retune：Retrieving and Materializing Tuple Units for Effective Keyword Search over Relational Databases[C]//ER. 2008：469-483

[8] Aditya B，Bhalotia G，Chakrabarti S，et al. BANKS：Browsing and Keyword Searching in Relational Databases[C]//VLDB. 2002：1083-1086

[9] Bhalotia G，Hulgeri A，Nakhe C，et al. Keyword Searching and Browsing in Databases using BANKS[C]//ICDE. 2002：431-440

[10] Fakas G J，Cai Zhi. Ranking of Object Summaries[C]//ICDE. 2009：1580-1583

[11] Fakas G J. Automated Generation of Object Summaries from Relational Databases：A Novel Keyword Searching Paradigm[C]//ICDE. 2008：564-567

[12] Fakas G J. A novel keyword search paradigm in relational databases：Object summaries[J]. Data and Knowledge Engineering，2011，70(2)：208-229

[13] Grossman D A. Information Retrieval：Algorithms and Heuristics [M]. 北京：人民邮电出版社，2010

[14] Croft W B，Metzler D，Strohman T. 搜索引擎信息检索实践[M]. 北京：机械工业出版社，2010