关系数据库上基于非数值属性关键词的模糊查询*)

杨路明 王佳宜 谢 东
(中南大学信息科学与工程学院 长沙 410083)

摘 要 关系数据库上的关键词查找技术使得用户像使用搜索引擎一样获取数据库中的相关数据。然而,这种技术只实现了精确查询,还不能很好地实现模糊查询。本文通过引进分类学习中的Rocchio算法并对其做小部分修改,用于数据库的关键词查询中,结合不同类型对象之间相异度和相关度的量化计算,每次返回的结果集按照相关度降序排列,实现精确到模糊的查询。如果用户不满意初始查询结果集,利用Rocchio算法经过几次交互,便可不断满足需求。对权值优化的Rocchio算法反馈过程进行了实验测试,结果证明是比较令用户满意的,而且返回的结果集中少量的不相关集合可以提高查询的性能。

关键词 关系数据库,关键词,模糊查询

Fuzzy Query Based on Non-numeric Attribute Keywords over Relational Databases

YANG Lu-ming WANG Jia-yi XIE Dong
(Institute of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract KSORD (keyword search over relational database) techniques allow users to obtain information from databases, which is just like using search engines. However, the advanced techniques only realize exact queries, but not for fuzzy queries. The Rocchio algorithm of learning classification is introduced which is made a little changed to achieve keyword search over relational databases. Connected with dissimilarity and correlation calculated, returned result sets are ranked in descendant order according to correlation. Thus, the system realizes both exact and fuzzy queries. If users are not satisfied with the initial result sets, they can utilize the Rocchio algorithm to do several relevance feedbacks in order to make results better. We experiment with the optimal Rocchio algorithm and the results prove to satisfy the requirements of users. In addition, few non-relevant result sets could improve the performance of searching.

Keywords Relational databases, Keywords, Fuzzy query

1 引言

KSORD技术允许普通用户或Web用户在对数据库做相关数据处理时,简单地就像在Web上使用搜索引擎查找信息一样,避免去学习复杂的数据库模式知识和SQL结构化查询语言。目前,国外利用KSORD先进技术发展的系统有DBXplorer[1],DISCOVER[2],IR-Style[3]等。在国内,王姗等人在中国人民大学与NCR Teradata数据仓库及商务智能联合实验室开发出系统SEEKER[4]。

这些系统的基本思路是一致的,即将数据库看作是由数据库中的元组(顶点)通过主码-外码(边)的关系连接而成的图。当用户给出一个关键词查询时,则通过全文索引从图中找出含全部关键词的最小子图作为查询结果。而且几个系统有着类似的定义,认为查询结果是含有至少一个关键词且每个叶结点都含有至少一个关键词的连接元组树。不过,SEEKER不要求查询结果包含全部关键词,只要求查询结果包含关键词的一个非空子集即可。总之,这些系统都必须返回基于关键词(部分)的查询序列。

比较上述系统,只是能够提供精确查询。如果关系数据库中没有包含关键词的信息,就不返回任何结果,没有很好的交互性。很多用户希望在除了得到精确结果以外可以产生一些不是很精确的结果,通过判断哪些对自己有意义,哪些没有,尝试更高级的查询。

因此,我们引进分类学习方法中的Rocchio算法[5],该算法主要用于文档(文摘)的查询。它的一大优点是在整个查询过程中,通过不停地相关反馈,不断提供用户更有用的信息。我们对该算法做了修改,就可以用于对数据库的查找,实现模糊查询。执行该算法过程中,如果用户不满意初始的查询结果,则可通过反馈来优化查询结果。同时,执行查询的过程中,系统在判断输入的关键词对象是否匹配数据库表或某一属性的实例基础上,再根据对象的类型计算两者之间的相异度,从而可以得到相关度。返回的结果集则根据相关度的大小进行降序排列,也就是最相关的查询信息位于最前面,这样有利于用户对相关与不相关信息的区分。如果相关度等于1,系统就意味实现了精确查询,否则实现了模糊查询。相关度越小,模糊性越大。

2 Rocchio算法及其优化

参与Rocchio算法的查询向量,每一维值的大小用对象之间的距离来衡量。文献[6]给出原始Rocchio算法的过程,但是并未考虑用户对结果集中相关子集与非相关子集的满意程度。我们在原算法的基础上,引进三个参数,用来量化用户

*)湖南省教育厅科研基金(05C671);中南大学重点资助创新项目(ZB018)。**杨路明** 博士生导师,博士,主要研究方向为数据库系统和网络通信;**王佳宜** 硕士研究生,主要研究方向为数据库和数据管理;**谢 东** 博士研究生,主要研究方向为数据管理。

对初始查询向量和初始查询集的满意程度(用户可以选择四种满意程度,分别是很满意、一般满意、不满意、很不满意),动态优化反馈过程。

2.1 文档向量与距离度量

Rocchio算法的基本思想是把每一个文档 d 表示为一个向量 $\vec{d}=(d_1,d_2,\cdots,d_n)$。每一维对应关键词集合 V 中的一个特征词,该词的权重即为 d_i 的值。文献[5]认为 d_i 的大小与关键词向量在文档中的映射向量 $TF(w_i,d)$ 和 $IDF(w_i)$ 有关,即常用的 $TF*IDF$ 求距离(权重)方法来检索文摘或文档。$TF*IDF$ 函数表达式如下:

$$d_i=TF*IDF=TF(w_i,d)*IDF(w_i)$$

其中 TF 表示关键词频率,即某一关键词 w_i 出现在文档 d 中的次数;IDF 表示逆文档频率,它可以由 $DF(w_i)$(文档频率,关键词 w_i 在文档 d 所在的基文档中出现的次数)得到:

$$IDF(w_i)=\log\left(\frac{|D|}{DF(w_i)}\right)$$

$|D|$ 表示的是所有文档的总数。

2.2 Rocchio算法

Rocchio算法通过几次反馈或交互从而查找符合用户的信息。在反馈的过程中,少量的模糊信息也会同时和用户交互。文献[6]指出,Rocchio算法主要是用于对文档的查询和反馈,关系数据库中存在除文本这样的非数值型变量以外,还有大量的数值型变量。因此,把Rocchio算法用于数据库的查询,它的优点就在于用户输入非数值型关键词序列时,整个查询及相关反馈过程执行的准确率和满意度会高于用户对数值型变量的关键词查询。如在人员信息数据库中,属性或字段如"姓名"是非数值型的变量,而"年龄"是数值型的变量。

算法的三个重要参数分别是初始查询向量、用户指定的相关集和不相关集。Rocchio反馈算法描述:

$$Q_{i+1}=Q_i+\frac{1}{p}\sum_{x=1}^{p}R_x-\frac{1}{q}\sum_{y=1}^{q}S_y$$

其中:$R=\{R1,R_2,\cdots,R_P\}$,是返回结果集中用户认为相关的子集;

$S=\{S_1,S_2,\cdots,S_q\}$,是返回结果集中用户认为不相关的子集。

反馈过程的具体描述见图1。首先,考虑用户输入的初始向量 Q_0,由2.1节中的方法计算 d_i 值,返回第一次的结果集。用户对初始结果集做出判断后,再利用Rocchio算法得到新的查询向量。这样重复几次之后,停止与用户的交互。

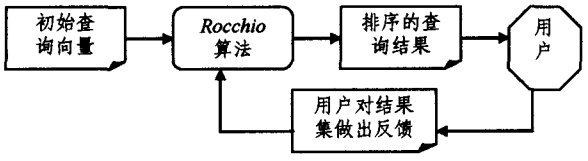

图1 相关反馈

2.3 权重优化Rocchio算法

原始反馈策略中,参与新查询向量计算的相关子集 R 和 S,在一定程度上不能体现出用户对其满意的程度,因此我们提出一种权重优化的相关反馈策略。新修改的Rocchio算法如下:

$$Q_i^{new}=aQ_i^{old}+b\frac{1}{|\text{rel docs}|}\sum_{\text{rel docs}}R_x-c\frac{1}{|\text{nonrel docs}|}\sum_{\text{nonrel docs}}S_y$$

其中:a,b,c 是引进的三个权重因子,$|\text{rel docs}|$ 和 $|\text{nonrel docs}|$ 分别表示相关子集与非相关子集(以0.5作为相关与非相关子集的区分度)的元素个数。

在算法的执行过程中,对 a,b,c 的取值可以由系统自动生成(对应四种满意程度的选择,分别设置不同的 a,b,c 等级和值),分别对应用户对上一次查询向量和查询结果的满意度。查询反馈的过程中,用户对前一次结果集的判断大大影响后一次查询的结果。如果初始查询的模糊集合中非相关子集过多,用户对此不满意,在交互时,就像简单地使用"Google"上高级检索的下拉列表框一样,用户选择"不满意",c 值由系统自动生成,同时优化的Rocchio算法计算新的查询向量并组织新一轮的查询。该过程可重复迭代几次,直至用户满意。

3 模糊查询的实现

本文在新修改的Rocchio算法的基础上,提出一种通过计算关键词对象和数据库中实例对象之间相异度的方法,实现查询过程。为能迅速匹配上关键词对象与数据库关系或属性对象,我们在数据库中建立起两张匹配表并建立全文索引,支持关系名和属性名上的关键词查询。

3.1 相异度和相关度

Rocchio算法中,$\sum_{\text{rel docs}}R_p$ 和 $\sum_{\text{nonrel docs}}S_q$ 是以对象之间的相异度(距离)量化值表示的。2.1节中介绍的 $TF*IDF$ 方法主要是测量文档类型的对象之间的距离,然而数据库记录中包含了丰富的类型的变量,比如区间标度变量、二元变量、序数型变量或这些变量类型的组合。因此,传统聚类中的欧氏距离和测量文摘或文档的 $TF*IDF$ 方法不再适用,需要考虑一种新的方法用于这种"混合类型变量"的两两对象的相异度(距离)计算。

我们首先定义对象 i 和对象 j 之间的相异度和相关度。

定义1(相异度) 对象 i 和对象 j 之间相异性的量化表示,用 $d(i,j)$ 表示。它是一个非负数。当对象 i 和 j 越相似或接近,其值越接近0;两个对象越不同,其值越大。特殊地,$d(i,i)=d(j,j)=0$。

定义2(相关度) 对象 i 和对象 j 之间相关性的量化表示,简单地记做 $c(i,j)=1-d(i,j)$。

我们将数据库记录按变量的类型分别标量化,并将所有有意义的变量转换到共同的值域区间[0.0,1.0]上。

假设数据集包含 p 个不同类型的变量,对象 i(用 x_i 表示)和 j(用 x_j 表示)之间的相异度 $d(i,j)$ 定义为:

$$d(i,j)=\frac{\sum_{f=1}^{p}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f=1}^{p}\delta_{ij}^{(f)}}$$

其中:用 f 来表征对象变量的类型,当某个记录中变量缺失即不存在时,$\delta_{ij}^{(f)}=0$,否则 $\delta_{ij}^{(f)}=1$;$d_{ij}^{(f)}$ 的计算与具体的类型有关。给出如下主要的几种类型:

(1)f 是二元变量或标称变量(二元变量的推广)。如果在该变量上完全相同,则 $d_{ij}^{(f)}=0$,否则为1。

(2)f 是区间标度变量(粗略线性标度的连续变量)。$d_{ij}^{(f)}=\frac{|x_i^{(f)}-x_j^{(f)}|}{\max_h x_h^{(f)}-\min_h x_h^{(f)}}$,$h$ 遍取变量 f 的所有非空值。

(3)f 是序数型或比例标度变量。变量由 M 个有序状态构成,分别映射成1,2,…,M,按公式 $x_i^{(f)}=\frac{r_i^{(f)}-1}{M-1}$ 分别计算对应记录的标量值。其中 $r_i^{(f)}$ 和 M 分别表示该变量的序数值和序数最大值。然后将 $x_i^{(f)}$ 作为区间标度变量类型对待。

(4) f 是文本型变量。将 $IF*IDF$ 表示的数值作为该变量的标量值，采用余弦度量相异度：$d_{ij}^{(f)}=\cos(d_i^{(f)},d_j^{(f)})$，其中的 $d_i^{(f)},d_j^{(f)}$ 都是由 $TF*IDF$ 计算得到的数值。

将所有的对象变量归类为相应类型后，按上述方法转化为标量化形式，用于快速地计算相互对象之间的相异度。为方便起见，系统采用相异度矩阵存储关键词序列和实例对象(来自数据库)之间的相异度值，见图 2。假设来自数据库的一张二维表有 n 条记录，p 个属性。如果某一关键词的类型与数据表中某一属性的实例不匹配，则在相异度矩阵的相应位置上置零；否则，相应位置上置 $d(i^{(n)},j_p)$(也就是当关键词对象与表中实例对象类型匹配，相异度就按照上述方法归类计算)。$i^{(n)}$ 是用户输入的关键词对象去比照数据库表中的第 n 条记录，j_p 是表中第 p 个属性的实例对象。

$$\begin{bmatrix} d(i^{(1)},j_1) & d(i^{(1)},j_2) & \cdots & d(i^{(1)},j_p) \\ d(i^{(2)},j_1) & d(i^{(2)},j_2) & \cdots & d(i^{(2)},j_p) \\ \vdots & \vdots & \vdots & \vdots \\ d(i^{(n)},j_1) & d(i^{(n)},j_2) & \cdots & d(i^{(n)},j_p) \end{bmatrix}$$

图 2　相异度矩阵

3.2　查询过程的实现

当用户输入查询关键词后，系统首先要将关键词定位于某一关系表中，然后是属性上的匹配。为此，我们在人员信息数据库中建立了两个匹配表(见表 1 和表 2)，分别用于关系名和属性名的关键词匹配。这两个表上也建立了全文索引。关系匹配表的“表名”列存储数据库中所有关系的名称，“关键词”列给出描述关系的关键词，这两列都在全文索引中。属性匹配表的“属性名”列存储数据库中的所有属性的名称，“表名”列存储属性所属关系的名称，“关键词”列给出描述属性的关键词，“类型”列给出属性的类型，“所属类别”列是属性对象可以归类为上述所描述的何种分类，其中“属性名”列和“关键词”列在全文索引中。系统将在关系名、属性名和描述它们的关键词中进行匹配，匹配上的，计算相异度，匹配不上的则为“0”。

表 1　关系匹配表

表名	关键词
基本信息表	身份证号,…
家庭住址表	城镇,…
体检情况表	良好,…
…	…
…	…

表 2　属性匹配表

表名	属性名	关键词	类型	所属类别
信息表	行号	ID,…	INT	序数型
信息表	姓名	角色,…	VARCHAR	文本型
…	…	…	…	…
体检表	体检日期	年,…	INT	区间标度型
…	…	…	…	…

从相异度存储矩阵中我们容易找到前 k 个最小相异度值，即前 k 个最大相关度值。定位前 k 个值后，剩下的工作按值的大小降序返回这 k 个值所在矩阵中的行，转换为数据库中对应的记录返回给用户。至此，系统完成第一次查询，且返回的结果以相关度值为依据，实现精确至模糊的查询。开始第二次新的查询前，必须根据用户需要对初始结果给予一定的评价之后，利用新改进的 Rocchio 算法，进行新一轮以及多次的查询交互。整个查询过程及交互过程可以用图 3 来描述。

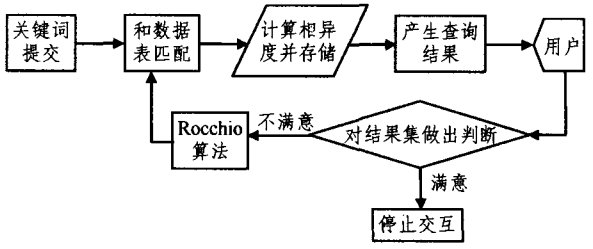

图 3　模糊查询实现过程描述

4　实验分析

实验环境是基于. NET 框架，通过与后台人员信息数据库的连接，利用 C# 语言编写查询过程与矩阵存储。简单的查询界面如图 4 所示。

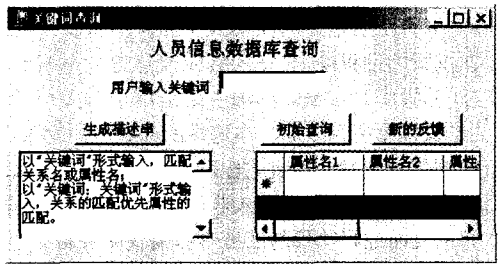

图 4　初始查询界面

用户输入关键词(序列)后，如果是以“关键词”的形式输入的，查找的过程中首先是和关系名匹配，然后再和属性名匹配。如果用户以“关键词:关键词”的形式输入的，就只匹配属性名。图 4 中生成描述串的作用在于系统的更加个性化。初始查询的结果用表格的形式(图 4 左下角的数据绑定控件)绑定返回给用户。因此基于关键词的查询技术就像使用常见的搜索引擎一样，避免用户学习复杂的相关 SQL 和数据库知识。

如果初始的查询并未满足用户对信息的需求，图 4 中提供了“新的反馈”按钮，点击之后出现图 5 所示的相关反馈界面。在用户对初始结果作出选择之后，本系统利用动态优化后的 Rocchio 算法重新组织查询，达到用户的更高要求。

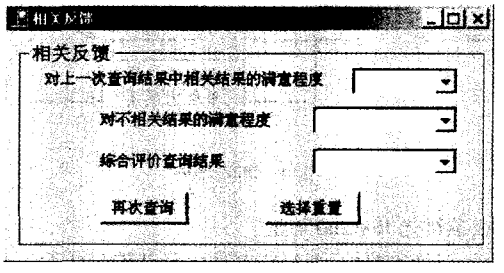

图 5　相关反馈界面

仔细考虑人员信息数据库，可以把“性别”归类为二元变量，把“出生年月”归类为区间标度变量等。输入关键词信息后，系统给出相关结果集。表 3 是当查询“出生地:河南”时数据库返回的前 5 条记录，其相关度分别是 1.0,1.0,0.8,0.8,0.8。分析初始的结果，我们对该结果集中的模糊记录不是很

满意(认为过多),经过新一轮的查询后,减少了这样的元组数。同时也进行了其他实验,如查询"本人成分:干部"、"健康状况:良好"等,结果都比较理想。

表 3 模糊查询结果

姓名	性别	出生年月	出生地	本人成分	健康状况
张 姚	女	1978.03.18	河南	干部	良好
刘美国	男	1983.10.12	河南	学生	良好
赵 婷	女	1975.12.14	河北	工人	良好
万佳成	男	1979.01.15	河北	工人	良好
王 伟	男	1985.08.19	河北	学生	虚弱

从表 3 可以看出,最匹配的记录在最前面。在比较大型的数据库中,如果系统一次呈现很多记录,从前往后找,也是很容易找到最相关的结果的。因此,该方法同时实现了精确和模糊查询。产生最不确定记录即相关度接近于"0"时,可以用来减少反馈的次数,加快查询。

结束语 本文在关键词实现精确查询的基础上,扩展了模糊查询。通过改进 Rocchio 算法和定义对象之间的相关度,实现了结果集的降序排列,同时也方便了用户的查看。模糊查询的过程大概分三步完成。首先是提交的关键词去匹配数据库表的关系名或属性名,然后计算相异度值并存储,最后返回非"0"相异度值所在的行对应的元组记录并按相关度值降序排列。改进的 Rocchio 算法可以增强用户与结果集的相互交互,不断满足更高要求的查询。

不足的是在实验过程中存在多种主观和人为因素。利用矩阵存储相异度,使得系统的响应时间过长。将来的工作中我们着重从算法的时间复杂度和空间复杂度入手,提高算法的效率和稳定性。

参 考 文 献

[1] Agrawal S, Chaudhuri S, Das G. DBXplorer: A System for Keyword-based Search over Relational Databases // Proc. of the 18th Int'l Conf on Data Engineering. San Jose, 2002: 5-16

[2] Hristidis V, Papakonstantinou Y. DISCOVER: Keyword Search in Relational Databases // Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong, 2002: 670-681

[3] Hristidis V, Gravano L, Papakonstantinou Y. Efficient IR-style Keyword Search over Relational Databases // Proc. of the 29th Int'l Conf. on Very Large Data Bases. Berlin, 2003: 850-861

[4] 文继军,王珊. SEEKER:基于关键词的关系数据库信息检索. 软件学报,2005,16(7): 1270-1281

[5] Rocchio J J. Relevance Feedback in Information Retrieval. in SMART Retrieval System Experiments in Automatic Document Processing, 1971: 313-323

[6] 战学刚,林鸿飞,姚天顺. 中文信息检索中的相关反馈. 计算机科学,2000,27(7): 39-41

(上接第 222 页)

OLAP 服务器结构,但其研究的内容主要停留在如何根据人们的主观想法来进行维度的更新,比如创建一个维度,增加一个维层次,删除一个维层次等,并没有涉及到由于一些外在因素的影响,如何对维结构进行自动调节的内容。文献[9]虽然也对维结构的变换问题进行了相关研究,并给出了相应的变换算法,但其在对维路径进行修正的过程中,是按照维层次的级别来进行的。这就导致了下述问题的出现:对于某一维路径,这次修改了其在维层次 l 上的取值,下次可能还得修改其在 $l'(\overline{l \prec_D^s} l')$ 上的取值,用关系表的形式来存储维实例数据的情况下,就导致了对同一条记录进行多次修改与提交。与文献[9]相比,本文采用规则的形式来描述异常情况给维结构带来的影响,并对描述异常的规则的作用次序进行了严格定义,很好地保证了在进行转换的过程中结果的正确性,而且本文所提出的变换算法花费的时间明显较短,具有更高的执行效率。

参 考 文 献

[1] Firestone J M. Dimensional Modeling and E-R Modeling in the Data Warehouse[R]. DKMS-White Paper No. Eight, June 1998

[2] 段云峰,等. 数据仓库基础 Data Warehousing Fundamentals[M]. 北京:电子工业出版社, 2004

[3] Pederson T B, Jensen C S. Multidimensional data modeling for complex data[A] // Proceedings of the 15th International Conference on Data Engineering (ICDE'99)[C]. Sydney, Australia: IEEE Computer Society, 1999: 336-345

[4] 李建中,高宏. 一种数据仓库的多维数据模型[J]. 软件学报, 2000,11(7): 908-917

[5] Jensen C S, Kligys A, Pedersen T B, et al. Multidimensional Data Modeling for Location-Based Services[J]. The International Journal on Very Large Data Bases, 2004, 13(1): 1-21

[6] 史忠植,等. 人工智能:复杂问题求解的结构和策略 Artificial Intelligence: Structures and Strategies for Complex Problem Solving [M]. 4E. 北京:机械工业出版社, 2004

[7] Antoniou G. A tutorial on default logics[J]. ACM Computing Surveys, 1999, 31(4): 337-359

[8] Agosta L. 数据仓库技术指南 The Essential Guide to Data Warehousing[M]. 潇湘工作室,译. 北京:人民邮电出版社, 2000: 118-141

[9] Minuto E M, Vaisman A. Efficient Intentional Redefinition of Aggregation Hierarchies in Multidimensional Databases[C] // Proceedings of the 4th International Workshop on Data Warehousing and OLAP. Atlanta, Georgia, USA, 2001: 1-8

[10] Hurtado C A, Mendelzon A O. OLAP Dimension Constraints [C] // Proc. ACM PODS. Madison, USA, 2002: 169-179

[11] Vaismana A A, Mendelzonb A O, Ruaroa W, et al. Supporting dimension updates in an OLAP server[J]. Information Systems, 2004, (29): 165-185