软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(Suppl.(2)):136−146

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

Top-*k* 查询结果的有界多样化方法*

周 宇[1], 赵 威[2], 刘国华[1], 貟 慧[1], 翟红敏[1], 万小妹[1]

[1](东华大学 计算机科学与技术学院,上海 201620)

[2](国网黑龙江省电力有限公司信息通信公司,黑龙江 哈尔滨 150000)

通讯作者: 周宇, E-mail: zyawf810@163.com, http://www.dhu.edu.cn

摘 要: 查询结果重复率高是 top-*k* 查询处理过程中亟待解决的问题,已有的解决方法需要遍历初始结果集中所有的对象,因此,查询处理的效率较低.为了提高查询处理的效率,把初始结果集映射到欧氏空间中,根据拉式策略,可选用基于得分或基于距离两种方法之一从该空间选出差异最优子空间,在基于距离的方法中,对欧氏子空间进行分割并且利用探测位置和 Voronoi 图的几何特性减少二次查询对象的数目.在此基础上,提出了 top-*k* 查询结果有界多样化算法,并证明了算法的正确性.实验结果表明,所提出的算法提高了 top-*k* 查询处理效率.

关键词: top-*k* 查询;有界多样性;欧氏空间;拉式策略;Voronoi 图

中文引用格式: 周宇,赵威,刘国华,貟慧,翟红敏,万小妹.Top-*k* 查询结果的有界多样化方法.软件学报,2014,25(Suppl.(2)):136−146. http://www.jos.org.cn/1000-9825/14032.htm

英文引用格式: Zhou Y, Zhao W, Liu GH, Yun H, Zhai HM, Wan XM. Bounded diversification methods for top-*k* query results. Ruan Jian Xue Bao/Journal of Software, 2014,25(Suppl.(2)):136−146 (in Chinese). http://www.jos.org.cn/1000-9825/14032.htm

Bounded Diversification Methods for Top-*k* Query Results

ZHOU Yu[1], ZHAO Wei[2], LIU Guo-Hua[1], YUN Hui[1], ZHAI Hong-Min[1], WAN Xiao-Mei[1]

[1](School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

[2](State Grid Corporation of China Heilongjiang Electric Power Company Ltd. Information & Telecommunication Branch, Harbin 150000, China)

Corresponding author: ZHOU Yu, E-mail: zyawf810@163.com, http://www.dhu.edu.cn

Abstract: High repetition rate of query results is a problem needing a prompt solution in top-*k* query processing. Existing solutions require the traversing over all objects in initial result set which may cause a lower efficiency in query processing. To address the issue, this paper first maps initial result set to the Euclidean space and selects the optimal subspace using either the score-based method or distance-based method by adopting the pulling strategy. Applying the distance-based method, the Euclidean space is partitioned and the number of second query objects is reduced by incorporating geometric properties of Voronoi diagram. Further, the bounded diversification algorithm over top-*k* query results is developed and the soundness of the algorithm is proved. Experimental results demonstrate that the proposed algorithm improves the efficiency of top-k query processing.

Key words: top-*k* query; bounded diversification; Euclidean space; pulling strategy; Voronoi diagram

查询结果的重复率高是 top-*k* 查询处理过程中亟待解决的问题.以安居客网站检索为例,用户检索上海二手房价在 1~1.5 万之间的售房信息,搜索引擎给用户展示了 6 万多条搜索结果,传统 top-*k* 查询算法首先遍历所有 6 万多条记录,然后根据得分函数选出得分值最高的前 *k* 条记录,这不仅返回重复的记录而且大大降低了检索效率.产生该问题的根源是传统 top-*k* 查询算法仅关注评分,没有考虑元组间的差异性.

本文将元组的属性作为空间维度,从 *n* 维欧氏空间着手,把初始查询结果集中的元组按属性值映射到欧氏

* 基金项目: 国家自然科学基金(61070032)
收稿时间: 2014-05-07; 定稿时间: 2014-08-19

空间中,以 Voronoi 图为基准对欧氏空间进行分割,利用探测位置以及分割后空间的几何性质进行二次查询,求解差异最优子空间,该子空间中 *k* 条元组就是具有差异性的多样化查询结果.

1 相关工作

查询结果多样性概念及多样化方法由 Carbonell 等人在文献[1]中首次提出,在文中他们只给出一个权衡相关性和多样性的目标函数,并未给出具体的算法描述.在文献[2]中,Gollapudi 提出一个表征多样性系统的公理框架,提出了两个优化目标以及相应的算法 MaxSum 和 MaxMin.Borodin 改进了 MaxSum 多样性的研究成果,用结果集中每对元素的距离总和衡量多样性,分析如何在动态环境下改良结果集[3].现有实现多样性的方法要求重新排名初始结果集,适用于文档的局部多样性和结构化数据[4−10].文献[11]用一个复合的近似函数降低多样性维度,从而归约到文献[1]的算法解决了多维度下多样性问题.在文献[12]中,Angel 提出了算法 DivGen,该算法采用拉式策略,无需对所有初始结果集进行重新排名.Van Kreveld 在文献[13]中首次提出空间查询结果多样性,提出了多维散列排名这一结论,利用文本和空间相关性解决文档相关性问题.Turk 在文献[14]中用实验证明了用户更关注单一空间多样性排名.Piero 在文献[15]中首次提出有界多样性问题,并给出了形式化定义,提出相应的算法,但其数据对象仅仅是地理数据,研究结果不适用于位置多样性以外的需求.

2 定义和记号

定义 1(*n* 维向量[16]). *n* 个有序的数 $\{a_1,a_2,...,a_n\}$ 组成的序列称为 *n* 维向量,这 *n* 个数称为该向量的 *n* 个分量,第 *i* 个数 a_i 称为第 *i* 个分量.分量全为实数的向量称为实向量.

定义 2(*n* 维实数向量空间 $\mathbf{R}^n$). *n* 维实数向量空间为一个二元组 (R^n,C),其中,$\mathrm{R}^n=\{(a_1,a_2,...,a_n)\mid a_1,a_2,...,a_n\in\mathrm{R}\}$ 是由全部 *n* 维实向量构成的集合.*C* 是定义在 R^n 上的运算,R^n 在 *C* 上是封闭的:

向量加法:$\mathrm{R}^n\times\mathrm{R}^n\to\mathrm{R}^n$,把 R^n 中两个向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 映射到 R^n 中另一个向量,记作 $\boldsymbol{u}+\boldsymbol{v}$;

标量乘法:$\mathrm{R}\times\mathrm{R}^n\to\mathrm{R}^n$,把实数集 R 中一个实数 μ 和 R^n 中一个向量 $\boldsymbol{v}$ 映射到 R^n 中另一个向量,记作 $\mu.\boldsymbol{v}$.

R^n 本身是无限不封闭的,*n* 称为空间的维度,为了下文叙述方便,把 (R^n,C) 简写成 R^n.

定义 3(*n* 维欧氏空间). 在 R^n 中,对任意两个向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,引入它们的标准内积 $\langle\boldsymbol{u},\boldsymbol{v}\rangle$:

$$\langle u,v\rangle=\sum_{i=1}^{n}u_iv_i=u_1v_1+u_2v_2+...+u_nv_n \tag{1}$$

即 R^n 中任意两个向量对应一个实数值,把 R^n 及这样的内积,称为 R^n 上的欧几里德结构,此时 R^n 也被称为 *n* 维欧几里德空间,简称 *n* 维欧氏空间.下文中所有 R^n 都表示 *n* 维欧氏空间.

定义 4(*n* 维欧氏子空间). 如果欧氏空间 R^n 的一个非空子空间 $\boldsymbol{W}(\boldsymbol{W}\subseteq\mathrm{R}^n)$,对于 R^n 的向量加法和标量乘法都封闭($\boldsymbol{W}$ 中向量相加或标量相乘之后仍在 $\boldsymbol{W}$ 中),则将 $\boldsymbol{W}$ 称为 R^n 的线性子空间,简称子空间.

定义 5(*n* 维欧氏子空间最小边界框 *B*). $\boldsymbol{W}$ 是 R^n 的一个子空间,*B* 是包含 $\boldsymbol{W}$ 中所有向量的最小边界框.形式化为

$$B=\{x\in\mathrm{R}^n\mid p_i\leqslant x_i\leqslant q_i,i=1,...,n\} \tag{2}$$

定义 6(差异最优子空间 $\mathbf{R}^n_{k^*}$). 给定 $\boldsymbol{W}$ 和 *B*(同定义 4 和定义 5),若 $\boldsymbol{W}$ 的一个子空间 $\mathrm{R}^n_{k^*}$ 使集合选择函数 *f* 的值最大且 $|\mathrm{R}^n_{k^*}|=k$(即集合 $\mathrm{R}^n_{k^*}$ 的基数为 *k*),则称 $\mathrm{R}^n_{k^*}$ 为差异最优子空间,其中 $\mathrm{R}^n_{k^*}$ 由公式(3)计算求得.

$$|\mathrm{R}^n_{k^*}|=\arg\max_{\mathrm{R}^n_k\subseteq W,|\mathrm{R}^n_k|=k}f(\mathrm{R}^n_k) \tag{3}$$

f 为目标函数,是函数 MaxMin 和 MaxSum 的混合形式[17].其中,函数 *w*(*u*)表示向量 *u* 的权重,函数 *d*(,)表示两个向量的欧氏距离.

$$f(\mathrm{R}^n_k)=(1-\lambda)\sum_{u\in\mathrm{R}^n_k}w(u)+\lambda\sum_{u,v\in\mathrm{R}^n_k}d(u,v) \tag{4}$$

添加到 $\mathrm{R}^n_{k^*}$ 的向量 *u**使多样性加权得分 σ 有最大值:

$$\sigma\left(u;\mathrm{R}_k^n\right)=(1-\lambda)\cdot w(u)+\lambda\cdot\min_{u'\in\mathrm{R}_k^n} d(u,u') \tag{5}$$

$$u^*=\arg\max_{u\in W\setminus\mathrm{R}_k^n}\sigma\left(u;\mathrm{R}_k^n\right) \tag{6}$$

定义 7(关系模式). 关系模式由关系名 S 和属性集 $\{A_1,A_2,...,A_m\}$ 构成,记为 $S(A_1,A_2,...,A_m)$. 对于属性 $A_i(1\leqslant i\leqslant m)$, $Dom(A_i)$表示 A_i 的可能取值集合,称为域.令 $W=Dom(A_1)\times Dom(A_2)\times...\times Dom(A_m)$, W中的元素称为 m 元组(简称为元组),$P(W)$表示 W 的幂集,$P(W)$中元素称为关系模式 $S(A_1,A_2,...,A_m)$ 的实例,则 S 的数学含义是定义域为 $P(W)$的一个谓词.$P(W)$中令 S 为真的实例称为关系模式 $S(A_1,A_2,...,A_m)$ 的关系.数据库模式是非空有限关系模式的集合,记作 $\boldsymbol{S}$.数据库模式 $\boldsymbol{S}$ 上的数据库实例[18]是指存在一个域为 $\boldsymbol{S}$ 的映射 $\boldsymbol{I}$,使得对于每个关系模式 $S\in\boldsymbol{S}$,$\boldsymbol{I}(S)$是 S 的关系.

以表 1 中 Houses 为例,模式为 Houses(Name,Region,Price,Area),其中 Name,Region 等为属性.Name 和 Region 属性的域为字符串的集合,Price 和 Area 属性的域为实数集合,(Seascape,Jiading,14000,100)为一条元组,表 1 为关系模式 Houses 的实例,由于其中的每一条元组都使得关系模式 Houses 为真,因此,表 1 又称为关系模式 Houses 的关系.

Table 1 Table Houses

表 1 Houses 表

Price	Area
14000	100
15000	140
10000	130
14000	100
12000	120

定义 8(查询映射(查询)). 这是定义域为特定关系模式或数据库模式上所有实例组成的集族,值域为一个数据库模式或关系模式实例构成的集族的一个映射,一般简称为查询.

定义 9(top-k 查询). 设 q 为一个查询,其定义域为 N,值域为 M,top-k 查询是定义域为 N,值域为 M 中元组个数为 k 的关系实例构成的集族的映射.

针对某个 top-k 查询 q',定义域为所有属性域笛卡尔积的子集,从定义域中取出一个关系 $I(S)$,首先找出满足查询 q'的所有元组,然后通过映射关系-得分函数 g,把满足查询条件的每一条元组 $o\in I(S)$映射到实数 $g(o)$作为得分值,把所有得分值重新降序排列,取前 k 个得分值最大的元组组成的集合作为值域. $O=\{o_1,\ldots,o_m\mid o_i\in O,1\leqslant i\leqslant m\}$ 表示初始查询结果集,O_i 为结果中一条元组,这里称其为对象.

定义 10(差异最优子集 $\boldsymbol{O_k^*}$). 差异最优子空间 $\mathrm{R}_{k^*}^n$ 中所有向量对应的数据库中对象组成的集合称为差异最优子集 O_k^*.

将初始查询结果集 $\boldsymbol{O}$ 中每一个对象映射到 n 维欧氏空间 R^n 中某个向量,其中,n 为查询值域所有属性的个数,属性值作为向量坐标值,对于不为实数形式的属性值,可将其映射为实数,所有向量组成 n 欧氏子空间 $\boldsymbol{W}$,即 $\boldsymbol{W}$ 中每个向量或点对应于数据库中一条元组或一个对象.对于 $\boldsymbol{W}$ 中每个向量 $u(u\in W)$, 都有一个权值 $w(u)$且 $w(u)\to[0,1]$.$w(u)$表示向量 u 对应的元组与查询的相似程度,权值越大越相似.$d(u,v)$计算向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的欧氏距离,用以衡量相应的两条元组的差异性,距离的值越大,差异性越大.从初始查询结果集 $\boldsymbol{O}$ 中选差异最优子集 O_k^* 转化为给定子空间 $\boldsymbol{W}$,有界区域 B,目标函数 f,求在区域 B 下子空间 $\boldsymbol{W}$ 的差异最优子空间 $\mathrm{R}_{k^*}^n$.

给定查询 q 为查询房价 1~1.5 万,面积 100~150m^2 的记录,查询 q 的定义域为所有属性域笛卡尔积的子集,从定义域中取出一个关系 Houses(见表 1),表 2 为查询 q 的值域.若 top-k 查询 q'在查询 q 的基础上限定查找前两条最优记录(即 k=2),此时 q'的定义域与 q 一致,值域为表 2 中元组个数为 2 的关系集合,表 2 作为 top-k 查询 q'的初始查询结果集有 5 条元组 $O=\{o_1,o_2,o_3,o_4,o_5\}$. 首先将 $\boldsymbol{O}$ 中所有元组映射到 $\mathbf{R}^n$,以其中一条元组 o_1(14000,100)为例,把元组o_1映射到 2 维欧氏空间中,设中心坐标为(12500,125),归一化处理后该条元组映射到 $\mathbf{R}^n$ 中向量 $\boldsymbol{u}_1$(0.3,−0.5).o_2 映射到 $\boldsymbol{u}_2$(0.5,0.3),$d(\boldsymbol{u}_1,\boldsymbol{u}_2)$=0.82 为向量 $\boldsymbol{u}_1$ 和 $\boldsymbol{u}_2$ 的欧氏距离,表示 o_1 和 o_2 的差异程度.

设这 5 个点的权值 $w(\cdot)$分别为[0.5,0.6,0.9,0.5,0.7],权值的计算可有多种方式,假设只考虑查询值域面积这一属性,不妨设面积为 125m^2 时权值为 1,则距离 125 越近的点权值越大,当然也可基于房价属性考虑,或者同时考虑这两个属性,权值越大的对象与给定查询越相似.所有 5 条元组映射到 $\mathbf{R}^n$ 的情况如图 1 所示,由于 o_1 和 o_4 这两个属性的值完全一致,因此取其一即可.

Table 2 Results of price in 1~1.5 ten thousand
表 2 房价为 1~1.5 万元结果集表

Name	Region	Price	Area
Seascape	Jia ding	14 000	100
Riverside Park	Chang ning	15 000	120
Chinatown	Putuo	10 000	98
Seascape	Jia ding	14 000	100
Newtown	Pu dong	28 000	115
Milan	Baoshan	17 500	64
Star World	Song jiang	12 000	120

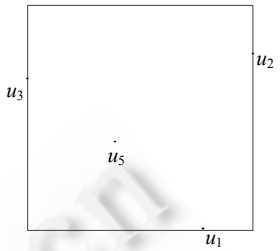

Fig.1 Euclidean space of Table 2
图 1 表 2 对应的欧氏空间图

3 二次查询方法

3.1 基于得分的方法

基于得分的二次查询方法只考虑向量的权重 $w(\cdot)$.$w(\cdot)$降序排列,子空间 $\boldsymbol{W}$ 中向量按 $w(\cdot)$的排列顺序进行二次查询,即 $w(\cdot)$值越大的点,越容易被查询.每次查询时,添加一个或几个点到集合 M(二次查询临时对象集合)中.设阈值 τ(公式见 3.2.3 节)有 μ 位小数,若 $\alpha<\lambda\cdot n$(α 是 τ 基于距离下降的比率,$n=10^{-\mu}$),则在获得下一个添加到 $\mathrm{R}_{k^*}^n$ 的对象之前,一直基于得分进行二次查询,此时可根据 σ 的计算公式估算出权重 $w(\cdot)$的最小值 w';若 $\alpha\geqslant\lambda\cdot n$,用比率 α 估算最小权重的差值 Δw,当 $w^{last}-w(\boldsymbol{u})>\Delta w$ 时,一直基于得分进行二次查询.

3.2 基于距离的方法

与基于得分方法降序排列不同的是,基于距离二次查询时,按 $d(\cdot,p_i)$升序的排列顺序进行二次查询,其中 p_i 为探测位置.

3.2.1 凸多面体的分割和探测位置

假设查询 q 为查询上海房价 1~1.5 万,面积 100~150m^2 的记录,所有初始查询结果映射的向量组成子空间 $\boldsymbol{W}$.令 $X=\{x_i \mid x_i\in \mathrm{R}_t^n, i=1,\ldots,t\}$,子空间 R_t^n 中所有点已经被二次查询并且添加到差异最优子空间 $\mathrm{R}_{k^*}^n$ 中,当 $t=k$ 时,R_t^n 达到最终状态,即由 k 个向量组成的差异最优子空间 $\mathrm{R}_{k^*}^n$.

$H(x_i,x_j)$称为半空间,是一个集合,集合中所有点到点 x_i 的距离都比到点 x_j 的距离更近:

$$H(x_i,x_j)=\{x\in\mathbf{R}^n \mid d(x,x_i)\leqslant d(x,x_j), j\in[1,t]\text{且}j\neq i\} \tag{7}$$

x_i 点关联的 Voronoi 多边形为

$$V(x_i)=\bigcap_{1\leqslant j\leqslant t, j\neq i} H(x_i,x_j) \tag{8}$$

对于 X 中每个点 x_i 都可做一个 Voronoi 多边形,这样 t 个多边形组成的图称为 Voronoi 图,记作 $Vor(X)$,如图 2 所示.

$$Vor(X)=\bigcap_{1\leqslant i\leqslant t} V(x_i) \tag{9}$$

通过最小边框 B 限制 $Vor(X)$使其有界,即取 $Vor(X)$和 B 的交集,此时 $Vor(X)$就是封闭的 Voronoi 图,记作 $Dia(X,B)$,如图 3 所示.

$$Dia(X,B)=Vor(X)\cap B \tag{10}$$

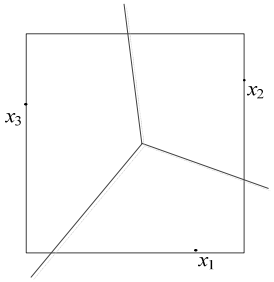

Fig.2 Unbounded Voronoi diagram

图 2 无界 Voronoi 图

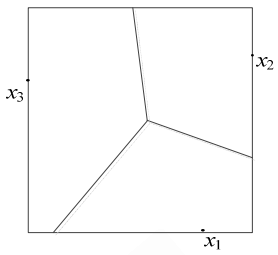

Fig.3 Bounded diagram of *Dia*(*X*,*B*)

图 3 有界图*Dia*(*X*,*B*)

图 2 和图 3 中点 x_1,x_2,x_3 为集合 X 中 3 个点,已经被二次查询并添加到 R_k^n. 中,此时 t=3 且空间维度 n=2(房价和面积).图 2 是集合 X 对应的 *Vor*(*X*)图,左半部分组成 $V(x_3)$,同理,右上角组成 $V(x_2)$,右下角是 $V(x_1)$.内部的 3 条线是 x_1x_2,x_2x_3 和 x_1x_3 这 3 条线段的垂直平分线,最外层四边形即为最小边界框 B.图 2 中每个 $V(x_i)$都是开放不封闭的,通过 B 限定使其成为一个有界封闭的图形,如图 3 所示.

探测位置局部最大化集合 X 中点 x_i(已添加到 O_k^* 中的对象)与 $\boldsymbol{W}$ 中所有未被二次查询到的点的最大距离.探测位置是 *Dia*(*X*,*B*)中顶点的子集.*Dia*(*X*,*B*)有如下 3 种类型的顶点:

- 最小边界框 B 的顶点.
- *Vor*(*X*)最初的顶点.此时在最小边界框 B 外的顶点抛弃,不予考虑.
- *Vor*(*X*)的边与最小边界框 B 的交点.

图 4 中顶点 $p_1,p_2,\ldots,p_8$ 为探测位置, p_1,p_2,p_3 和 p_4 是最小边界框 B 的顶点,p_5 是 *Vor*(*X*)最初的顶点, p_6,p_7,p_8 是 *Vor*(*X*)的边与最小边界框 B 的交点.

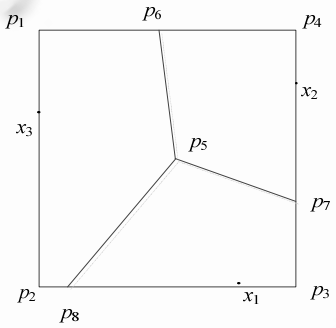

Fig.4 Probing locations in *Dia*(*X*,*B*)

图 4 *Dia*(*X*,*B*)中的探测位置

3.2.2 抛弃区域

令 $p_1,p_2,\ldots,p_v$ 为所有探测位置.若最后获得的点在以探测位置 p_u 为中心,r_u 为半径的多面体上,那么以 p_u 为中心,r_u 为半径的多面体 $\sum_u=\{x\in W\mid d(x,p_u)\leqslant r_u\}$ 内所有的点都已被二次查询,定义:

$$D=\bigcup_{u=1}^{v}\sum_u \tag{11}$$

为基于距离查询方法的抛弃区域,下次基于距离查询时就无需考虑抛弃区域中所有的点.

以探测位置 p_u 为中心,r_u 为半径,所做多面体 $\sum_u=\{x\in W\mid d(x,p_u)\leqslant r_u\}$ 如图 5 所示,所有 $\sum_u(u=1,\ldots,v)$ 的并集即为抛弃区域 D.

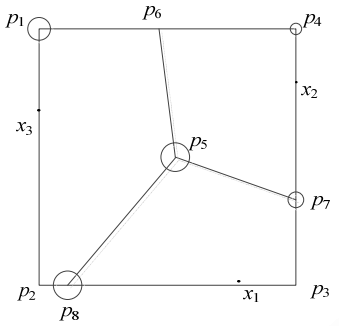

Fig.5 Discarded region *D* in *Dia*(*X*,*B*)

图 5 *Dia*(*X*,*B*)中抛弃区域 *D*

3.2.3 阈值模式(threshold mode)

阈值模式主要研究如何从已知探测位置中选取一个或几个最优探测位置进行二次查询.

在二次查询过程中,目标函数阈值 τ会不断更新,每次更新后 τ值都会下降并且作为未查询对象的上界.只有当某个对象的多样性加权得分 σ大于阈值 τ时,才会把这个对象添加到子空间 R_k^n. 中. τ的计算公式如下所示:

$$\tau = (1-\lambda)\cdot w^{last} + \lambda \cdot d \tag{12}$$

$$d = \max_{y\in Y} \min_{x\in X} d(x,y) \tag{13}$$

其中,$Y=B-D$,不考虑探测位置 p_u 为中心、r_u 为半径的抛弃区域 $\sum_u (u=1,\ldots,v)$. w^{last} 是上一次基于得分二次查询时对象的权值.

在基于距离二次查询时,求出每个探测位置的 $\tau_u(u=1,\ldots,v)$ 值,整体阈值 τ取局部最大值:

$$\tau = \max_{u=1,\ldots,v} \tau_u \tag{14}$$

$$u^* = \arg\max_{u=1,\ldots,v} \tau_u \tag{15}$$

局部 τ_u 的计算公式如下:

$$\tau_u = (1-\lambda)\cdot w^{last} + \lambda \cdot d_u \tag{16}$$

$$d_u = \min_{x\in X} d(p_u,x) \tag{17}$$

综上所述,阈值 τ可由探测位置计算得到.值得注意的是,在计算 τ_u 时,若探测位置 p_u 存在抛弃区域 $\sum_u$, 此时 p_u 在抛弃区域中,在计算公式 $\min_{x\in X} d(p_u,x)$ 时应考虑 $\sum_u$和 B 的交点与 X 中点的最近距离,然后取所有距离的最大值作为 d_u.

3.3 拉式策略(pulling strategy)

有界多样化算法通过拉式策略选择基于得分或基于距离的二次查询方法.通常,选择能使 τ值下降更快的二次查询方法,保证能更快得到子空间 R_k^n. .令 $n_u(u=1,\ldots,v)$ 为基于距离的查询方法从探测位置 p_u 获得的对象数,令 n^s 为基于得分的查询方法获得的对象数.两种情况下阈值 τ的下降速率分别为 $\alpha = \left|\frac{\partial \tau}{\partial n_u}\right|$ 和 $\beta = \left|\frac{\partial \tau}{\partial n^s}\right|$.当 $\left|\frac{\partial \tau}{\partial n_u}\right| < \left|\frac{\partial \tau}{\partial n^s}\right|$ 时,基于得分的查询方法使 τ下降得更快,选择基于得分的查询方法.否则,选择基于距离的查询方法.

对于基于距离的二次查询,有

$$\frac{\partial \tau}{\partial n_u} = \frac{\partial}{\partial n_u}\left\{\max_{u=1,\ldots,v} \tau_u\right\} = \frac{\partial}{\partial n_u}\left\{(1-\lambda)\cdot w\left(n^s\right) + \lambda \cdot d_{u^*}\right\} = \begin{cases} 0 & u \neq u^* \\ \lambda \dfrac{\partial d_u}{\partial n_u} & u = u^* \end{cases} \tag{18}$$

又 $\frac{\partial d_u}{\partial n_u}=\frac{\partial d_u}{\partial r_u}\cdot\frac{\partial r_u}{\partial n_u}$,在对象分布不明确时,$\frac{\partial r_u}{\partial n_u}\approx r_u(n_u)-r_u(n_u-1)$,对于 $\frac{\partial d_u}{\partial r_u}$,可利用几何性质求其近似解.

对于基于得分的二次查询,有

$$\frac{\partial \tau}{\partial n^s}=\frac{\partial}{\partial n^s}\left\{\max_{u=1,\ldots,v}\tau_u\right\}=\frac{\partial}{\partial n^s}\left\{(1-\lambda)\cdot w(n^s)+\lambda\cdot d_{u^*}\right\}=(1-\lambda)\cdot\frac{\partial w(\cdot)}{\partial n^s} \tag{19}$$

其中,$w(n^s)$ 是基于得分查询时第 n^s 个点的权值,$\frac{\partial w(\cdot)}{\partial n^s}=w(n^s)-w(n^s-1)$.

3.4 Top-k查询结果的有界多样化算法(BDMMR)

算法 1 是 top-k 查询结果的有界多样化算法.每次二次查询时,选择基于距离或基于得分的方法,添加一个或几个对象到集合 P 中,对 w^{last} 或 D 进行更新,最后返回的结果集中对象与查询有一定的相似性,且彼此之间又有一定的差异性,是一个多样性结果集.利用空间索引结构,整个算法时间复杂度为 $O(N\log N)$.

算法 1. Top-k 查询结果有界多样化算法(O,k,λ).

输入:结果集 $\boldsymbol{O}$,top-k 的 k 值,平衡因子λ;

输出:差异最优子集 O_k^*.

主要变量:二次查询到所有对象组成的集合 P,二次查询临时对象集合 M,抛弃区域 D,上一个基于得分二次查询时对象的权值 w^{last},最优多样性加权得分σ^*,最优对象 o^*,阈值上界τ,各个探测位置的阈值 $\tau_1,\tau_2,\ldots,\tau_v$,距离上界 d,各个探测位置的距离 $d_1,d_2,\ldots,d_v$,全局变量标志 *flag*.

参数:初始化策略 IS,拉式策略 PS,阈值模式 TM.

1. $O_k^*=\{IS.initialObject()\}$; /*取权重最大的对象*/
2. $D=\varnothing; w^{last}=1; P=O_k^*; flag=0;$
3. $while(|O_k^*|<k)$ /*查找剩余的 k–1 个对象*/
4. $\tau=\infty;\sigma^*=-\infty;$
5. $\text{if}\left(P\setminus O_k^*\neq\varnothing\right)\text{ then } o^*=\arg\max_{o\in P\setminus O_k^*}\sigma(o;O_k^*);\sigma^*=\sigma(o^*;O_k^*);$ /*对σ^*和 o^*进行更新*/
6. while($\sigma^*<\tau$ and $D\subseteq B$ and $P\subseteq O$)
7. $M=PS.getNextObjects()$; /*基于得分或基于距离获取下一个对象集*/
8. $P=P\cup M;$
9. while($M\neq\varnothing$) /*对新查询到的每个对象计算其σ值以更新σ^**/
10. $o=getobject(M);$
11. $\text{if}\left(\sigma(o;O_k^*)>\sigma^*\right)\text{ then }\sigma^*=\sigma(o;O_k^*);o^*=o;$
12. $M=M\setminus\{o\};$
13. $(D,w^{last})=PS.updateRegionAndWeight(D,w^{last});$ /*更新抛弃区域或上一个基于得分二次查询时对象的权值*/
14. $(d;\tau;d_1,d_2,\ldots,d_v;\tau_1,\tau_2,\ldots,\tau_v)=TM.updateThreshold(P,D,w^{last});$ /*更新最大距离和阈值*/
15. $O_k^*=O_k^*\cup\{o^*\};$

16.return O_k^*;

定理 1. BDMMR 是正确的.

证明:BDMMR 与 MMR[1]算法相比是正确的,用归纳法证明 BDMMR 每次选出的对象都与 MMR 一致.当 k=1 时,两种算法的初始化策略一致,都是选择权重最大的对象,此时对象一致.设 k–1 步也成立,则第 k 步时,BDMMR 从 $P\setminus O_{k-1}^*$ 和内层循环中选取使多样性加权得分最大的对象.在如下两种情况下退出循环,要么当前的

最优多样性加权得分σ^*比τ大,要么最小边界框 B 中所有对象都被二次查询,然而无论基于何种情况,选出来的对象 o^*保证多样性加权得分σ有最大值,这与 MMR 一致.定理得证. □

对于算法 1 中第 7 步根据拉式策略选用基于得分或基于距离的方法之一进行二次查询,具体过程见算法 2.

算法 2. *PS.getNextObjects*().

输入:差异最优子集 O_k^*,二次查询到所有对象组成的集合 P,抛弃区域 D,上一个基于得分二次查询时对象的权值 w^{last},平衡因子λ,阈值上界τ,距离上界 d,各个探测位置的距离 $d_1,d_2,\ldots,d_v$,各个探测位置的阈值 $\tau_1,\tau_2,\ldots,\tau_v$,最优多样性加权得分$\sigma^*$,全局变量标志 *flag*,比率界限值 n;

输出:二次查询临时对象集合 M,全局变量标志 *flag*.

主要变量:Δw 权重的最小差值,探测位置集合 V,基于距离的下降比率α,基于得分的下降比率β,最小权重值 w',查询方法 m.

1. $p_1,p_2,\ldots,p_v$ = vertices of $Dia(X,B)$, where $X=\{x_i \mid x_i \in R^n_{(k^*)}, i=1,\ldots,t\}$;
2. α=*distanceRate*( );β=*scoreRate*( );$V=\varnothing$;i=1;$M=\varnothing$;
3. if($\alpha<\beta$ and *flag*==1) /*基于得分的二次查询方法*/
4. if($\alpha \geqslant \lambda \cdot n$)
5. $\Delta w=\alpha/\lambda$;
6. o=*m.getScoreObject*( );
7. $M=M\cup\{o\}$;
8. while ($w^{last}-w(o)>\Delta w$)
9. $w^{last}=w(o)$;o=*m.getScoreObjece*( );$M=M\cup\{o\}$;
10. else
11. $w'=\left[\sigma^*-(1-\lambda)\times d\right]/\lambda$;
12. o=*m.getScoreObject*( );$M=M\cup\{o\}$;
13. while ($w(u)>w'$)
14. if $\left(\sigma(o;O_k^*)>\sigma^*\right)$ then $\sigma^*=\sigma(o,O_k^*)$; $o^*=o$; $w'=[\sigma^*-(1-\lambda)\times d]/\lambda$;
15. o=*m.getScoreObject*( );$M=M\cup\{o\}$;
16. else /*默认情况下基于距离进行二次查询或基于距离二次查询时使τ下降得更快*/
17. $(d;\tau;d_1,d_2,\ldots,d_v;\tau_1,\tau_2,\ldots,\tau_v)=TM.updateThreshold\left(P,D,w^{last}\right)$; $flag=1$;
18. for (i=1;$i\leqslant v$;i++) /*将满足条件的探测位置都作为最优探测位置*/
19. if $(\tau-\tau_i<n\cdot 5)$ then $V=V\cup\{p_i\}$;
20. M=*m.getDistanceObjects*(V);
21. return M;

以表 1 所示 Houses 为例,查询房价 1~1.5 万,面积 100~150m^2 的记录,表 2 为初始结果集 $O=\{o_1,o_2,o_3,o_4,o_5\}$,对应向量为 $\boldsymbol{u}_1(0.3,-0.5)$,$\boldsymbol{u}_2(0.3,-0.5)$,$\boldsymbol{u}_3(-0.5,0.1)$,$\boldsymbol{u}_4(-0.1,-0.1)$,权值为[0.5,0.6,0.9,0.5,0.7],设参数 k=2,λ=0.75,映射到欧氏空间的情况如图 1 所示.

首先,初始化时 $w(\boldsymbol{u}_3)$=0.9 最大,所以 $O_k^*=\{o_3\}$且添加到集合 P 中,抛弃区域 D 初始化为$\varnothing$,w^{last} 初始化为 1,且 *flag* 初始化为 0,表示未使用基于距离的方法进行二次查询.因为 k=2,所以需要执行 1 次循环以获得第 2 个对象.τ和σ^*分别被赋予初值∞和$-\infty$(第 4 步),由于 $P\setminus O_k^*=\varnothing$, 不执行后面的更新操作(第 5 步).此时满足循环条件(第 6 步),执行子算法 *PS.getNextObjects*()(第 7 步),根据拉式策略,选择基于得分或基于距离的方法进行二次查询,由于 *flag*=0,没有使用基于距离的方法查询对象,默认情况下选择基于距离的方法.由于 O_k^* 中只有一个对象

$\{o_3\}$,无需对图进行分割,探测位置为最小边界框 B 的 4 个顶点,根据公式(16)计算可得,τ_1=0.55,τ_2=0.70,τ_3=1.13,τ_4=1.06,因此 p_3 为最优探测位置,又由于 $\tau_3-\tau_i>5\cdot10^{-2}(i=1,2,4)$,所以只基于探测位置$p_3$进行二次查询,得到对象 $M=\{o_1\}$. $P=\{o_3,o_1\}$ (第 8 步), $\sigma^*=\sigma_1=0.88, o^*=o_1$,并且对 D,d,τ等进行更新,因为只基于探测位置 v_3 进行查询,所以其他探测位置τ和 d 值不变,只有 τ_3 由于抛弃区域而有所下降,τ_1=0.55,τ_2=0.70,τ_3=1.06,τ_4=1.06(第 13 步、第 14 步).又 $(\sigma^*=0.88)<(\tau=1.06)$, 仍执行内部循环(第 6 步),执行子算法时(第 7 步)由于使用过两种查询方法,因此选择使τ值下降得更快的查询方法,由第 3.3 节中拉式策略的计算公式计算得到 $\alpha=0.15,\beta=0.25,\alpha<\beta$ 且 *flag*=1,选择基于得分的方法查询对象.由$\alpha>\lambda\cdot10^{-2}$,选择分支 1 继续执行得$\Delta w$=0.2,查询得到对象 $o_5, M=\{o_5\}$,由于 $w^{last}-w(o)=0.3>0.2$, 继续执行子算法第 8 步的循环查询到对象 o_2,添加到集合 $M=\{o_5,o_2\}$ 后跳出子算法, $P=\{o_3,o_1,o_5,o_2\}$, 更新 $\sigma^*=\sigma_2=0.915$, $o^*=o_2$, $w^{last}=0.6$ 以及 $\tau=\tau_3=\tau_4=0.96$, 由于内循环条件 $P\subsetneq O$, 因此,跳出循环,将 $o^*=o_2$ 添加到 O_k^* 中,得到差异最优子集 $O_k^*=\{o_3,o_2\}$.

4 实 验

4.1 实验环境

实验运行的硬件环境为 Intel 酷睿 i5,主频 1.8GHz,内存 4GB,Mac OS X 10.9 操作系统,集成开发环境为 Netbeans 8.0,编程语言为 Java 8.

4.2 数据集

实验共有 3 种不同类型的数据集.第 1 个数据集是合成的二维数据,对象满足面积为 1 的均匀分布,其中每一个对象都指定了一个随机的权重值 $w(\cdot)$.对于每一种参数,都取了 20 组不同的数据,最后将它们的平均值作为最终实验结果.第 2 个数据集是真实二维数据集,通过从安居客网站爬取相关网页,将房价和面积两个属性的数据作为坐标值,权重值是基于房价或者基于面积的得分函数值.第 3 个数据集是合成的三维数据,满足体积为 1 的均匀分布,其中每个对象也都指定了一个随机的权重值 $w(\cdot)$.

4.3 实验结果与分析

图 6 是人工合成二维数据下的实验结果,图 7 是真实二维数据下的实验结果,图 8 是三维人工合成数据下的实验结果,这 3 种情况同时考虑了两种二次查询方法(即同时考虑了基于得分和基于距离的得分方法),与前面 3 个图不同的是,图 9 是只基于距离方法的二次查询情况.图中横坐标为 top-k 的 k 值,纵坐标为二次查询的对象个数与初始查询结果集中对象个数的百分比($|P|/|\boldsymbol{O}|$),$|P|$为算法中二次查询到的对象个数,$|\boldsymbol{O}|$为初始查询结果集中对象总个数.总体而言,算法明显减少了二次查询到的对象个数.

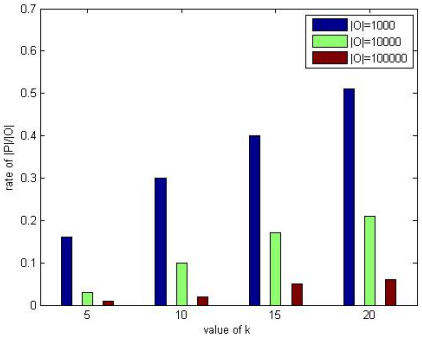

Fig.6 Artificial two-dimensional data

图 6 人工模拟二维数据的实验结果

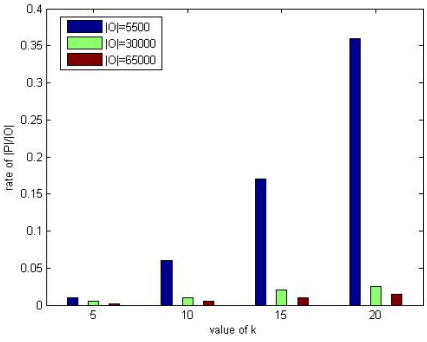

Fig.7 Real two-dimensional data

图 7 二维真实数据的实验结果

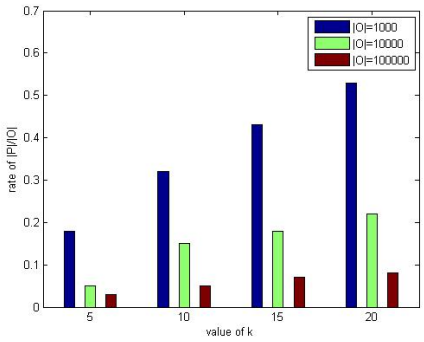

Fig.8 Artificial three-dimensional data

图 8 三维人工模拟数据的实验结果

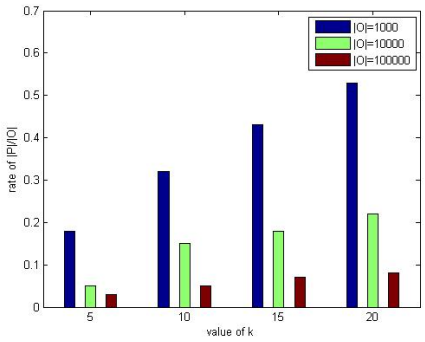

Fig.9 Only based on distance method

图 9 只基于距离方法的二维模拟数据结果

数据集的大小|***O***|:当 *k* 固定不变时,随着|***O***|值的增加,|*P*|/|***O***|的比率反而减小.例如,当 *k*=10,|***O***|=1000 时,需要查询 30%的对象,然而当|***O***|=10000 时,比率缩小到 10%,即只需要查询 1 000 个对象就能得到查询结果.总体而言,初始查询结果集基数越大,二次查询的对象数目越小.

结果集的大小 *k*:当|***O***|固定不变时,随着 *k* 值的增大,|*P*|/|***O***|的比率也随之增大,增大的速率取决于|***O***|的基数,基数越小,增大的速率越快.

维度的大小 *n*:由图 6 和图 8 对比可知,当|***O***|和 *k* 固定不变时,随着维度 *n* 的增大,|*P*|/|***O***|的比率也随之增大,增大的速率与|***O***|和 *k* 的大小并没有绝对关系.

方法选取上的不同:由图 6 和图 9 对比可知,在同时考虑基于得分和基于距离两种方法查询时,效果明显比在只基于距离的方法下进行查询的效果要好,同时考虑两种查询方法时,查询到的对象数远远小于初始结果集中的对象数.

5 总结与展望

本文针对查询结果重复率高这一问题,从 *n* 维欧氏空间着手,把初始查询结果集映射到欧氏空间中,以 Voronoi 图为基准对空间进行分割,选用基于得分或基于距离的方法进行二次查询,求解差异最优子空间,最终提出了 top-*k* 查询结果的有界多样化算法,消除了查询结果的重复率.实验验证了算法无需二次查询所有初始查询结果集,就能得到相应的结果集,提高了 top-*k* 查询处理效率.未来的工作将利用批处理方式继续完善本文提出的二次查询方法以及实验结果,同时考虑子算法中比率界限值 *n* 的最优值以及对象分布集中等极端情况,使得本文提出的算法得以进一步完善,以获得查询效率的最大提升.

致谢 本文是在导师刘国华教授的精心指导下完成的.在此特向刘老师的指导和帮助表示由衷的感谢.另外,感谢王伟同学,在本文研究方法的构思上提出了很多建设性意见.感谢所有同学们的帮助和鼓励,一起在讨论班上探讨与研究问题对研究能力的提高帮助很大.

References:

[1] Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proc. of the 21st Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 1998. 335−336.

[2] Agrawal R, Gollapudi S, Halverson A, *et al*. Diversifying search results.In: Proc. of the 2nd ACM Int'l Conf. on Web Search and Data Mining. ACM, 2009. 5−14.

[3] Borodin A, Lee HC, Ye Y. Max-Sum diversification, monotone submodular functions and dynamic updates. In: Proc. of the 31st Symp. on Principles of Database Systems. ACM, 2012. 155−166.

[4] Drosou M, Pitoura E. Search result diversification. ACM SIGMOD Record, 2010,39(1):41−47.

[5] Bansal N, Jain K, Kazeykina A, *et al*. Approximation algorithms for diversified search ranking. In: Automata, Languages and Programming. Berlin, Heidelberg: Springer-Verlag, 2010. 273−284.

[6] Capannini G, Nardini FM, Perego R, *et al*. Efficient diversification of Web search results. Proc. of the VLDB Endowment, 2011, 4(7):451−459.

[7] Rafiei D, Bharat K, Shukla A. Diversifying Web search results. In: Proc. of the 19th Int'l Conf. on World Wide Web. ACM, 2010. 781−790.

[8] Demidova E, Fankhauser P, Zhou X, *et al*. DivQ: Diversification for keyword search over structured databases. In: Proc. of the 33rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2010. 331−338.

[9] Liu Z, Sun P, Chen Y. Structured search result differentiation. Proc. of the VLDB Endowment, 2009,2(1):313−324.

[10] Vee E, Srivastava U, Shanmugasundaram J, *et al*. Efficient computation of diverse query results. In: Proc. of the ICDE 2008, IEEE the 24th Int'l Conf. on Data Engineering. IEEE, 2008. 228−236.

[11] Dou Z, Hu S, Chen K, *et al*. Multi-Dimensional search result diversification. In: Proc. of the 4th ACM Int'l Conf. on Web Search and Data Mining. ACM, 2011. 475−484.

[12] Angel A, Koudas N. Efficient diversity-aware search. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data. ACM, 2011. 781−792.

[13] Van Kreveld M, Reinbacher I, Arampatzis A, *et al*. Multi-Dimensional scattered ranking methods for geographic information retrieval. GeoInformatica, 2005,9(1):61−84.

[14] Tang J, Sanderson M. Evaluation and user preference study on spatial diversity. In: Advances in Information Retrieval. Berlin, Heidelberg: Springer-Verlag, 2010. 179−190.

[15] Fraternali P, Martinenghi D, Tagliasacchi M. Top-*k* bounded diversification. In: Proc. of the 2012 ACM SIGMOD Int'l Conf. on Management of Data. ACM, 2012. 421−432.

[16] 同济大学数学系.工程数学线性代数.第 5 版,北京:高等教育出版社,2007.

[17] Gollapudi S, Sharma A. An axiomatic approach for result diversification. In: Proc. of the 18th Int'l Conf. on World Wide Web. ACM, 2009. 381−390.

[18] Abiteboul S, Hull R, Vianu V. Foundations of Databases. Reading: Addison-Wesley, 1995.

周宇(1990－),女,江苏南通人,硕士,主要研究领域为数据库查询,top-*k* 查询,网页检索.
E-mail: zyawf810@163.com

贠慧(1992－),女,本科生,主要研究领域为数据库查询.
E-mail: luckyunhui@gmail.com

赵威(1974－),男,博士生,主要研究领域为数据库安全.
E-mail: 4752311@qq.com

翟红敏(1989－),女,硕士,主要研究领域为大数据多路连接优化.
E-mail: 844428365@qq.com

刘国华(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库理论,数据库安全,Web 数据管理,业务过程管理.
E-mail: 250160646@qq.com

万小妹(1990－),女,硕士,主要研究领域为以数据为中心的业务流程管理.
E-mail: 513118585@qq.com