基于 MapReduce 的增广动态 Skyline 查询处理方法

丁琳琳,崔子强,尹显坤,王俊陆,宋宝燕
(辽宁大学信息学院,辽宁沈阳 110036)

摘 要: Skyline 查询能够计算大规模的数据集中满足多个标准的最优解,被广泛应用于多目标决策等领域. 动态 skyline 查询作为 skyline 查询的一种重要变体,其结果随着查询点的不同而动态改变,为用户在指定查询要求方面提供了更大的灵活性. 然而,随着数据量的不断增加,动态 skyline 查询会产生大量的查询结果,忽略了查询点的维度方向性和数据的全局整体性,给用户的选择带来极大困难. 因此,需要进一步优化动态 skyline 查询的结果集,提高全局整体性,过滤冗余数据. 针对上述问题,提出一种基于 MapReduce 的增广动态 skyline 查询处理方法. 该方法将原始数据按照维度信息进行分区,在多个节点并行计算动态 skyline,优化传统动态 skyline 结果集,同时提供全局更优的结果供用户选择. 在此基础上,针对用户给出某些维度的容忍度的情况,提出一种引入用户容忍度的增广动态 skyline 查询处理方法. 该方法可以根据用户容忍度缩减增广动态 skyline 查询的原始数据集,很大程度上减少中间结果的比较次数,并且提高了结果集的准确度. 大量实验证明,基于 MapReduce 的增广动态 skyline 查询处理方法具有更好的有效性、准确性和可用性.

关键词: 动态 skyline 查询; MapReduce; 用户容忍度; 大数据
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2018)05-1062-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.05.006

Augmented Dynamic Skyline Query Processing Method Based on MapReduce

DING Lin-lin, CUI Zi-qiang, YIN Xian-kun, WANG Jun-lu, SONG Bao-yan
(*School of Information, Liaoning University, Shenyang, Liaoning* 110036, *China*)

Abstract: Skyline query can compute the optimal solution which meets the multiple standards in large-scale dataset. It has been widely applied for multi-objective decisions. Dynamic skyline query, as an important variant of skyline, its result can be dynamically changed with choosing different query points, providing more flexibility when the users make some specified needs. However, dynamic skyline query can return a large number of query results and ignore the directionality of query point and data integrality, making difficult for users to choose. It is necessary to optimize the result set of dynamic skyline, improving the whole data integrality and filtering a large number of redundant data. Focusing on these problems, we propose the augmented dynamic skyline query method based on MapReduce. The algorithm partitions the original data according to dimensional information, parallel computes dynamic skyline points in multiple nodes, optimizes the result set of the traditional dynamic skyline and at the same time provides the more global optimal results for the user to choose. In addition, when the users provide the tolerance of some dimensions, we propose the augmented dynamic skyline query with user tolerance. The algorithm reduces the original dataset according to the user tolerance and reduces the comparison times of intermediate results with improving the accuracy of the result set. The experiment results show that the augmented dynamic skyline query method based on MapReduce is efficient, accurate and scalable.

Key words: dynamic skyline query; MapReduce; user tolerance; big data

收稿日期:2017-02-21;修回日期:2017-09-18;责任编辑:蓝红杰
基金项目:国家自然科学基金(No. 61472169, No. 61502215, No. 61472069, No. 61402089, No. 61572119);辽宁省教育厅科学研究一般项目(No. L2015193);辽宁省博士科研启动基金(No. 201501127);辽宁大学青年科研基金(No. LDQN201438);国家重点研发计划项目(No. 2016YFC0801406)

1　引言

数据规模的扩大和数据维度的增加使得大数据环境下的多维数据处理面临着很多新的问题和挑战. Skyline 查询及其变体能够在大规模的数据集中计算满足多个标准的最优解,被广泛应用于多目标决策等领域.

传统的 skyline 查询是静态的,只考虑数据点各个维度的静态属性信息. 如果给定一个查询点 q,通过比较数据对象间相对于 q 的支配关系,计算出数据集中所有维度都距离 q 较近的数据点,这样的 skyline 查询称为动态 skyline 查询,其结果随着查询点的不同而动态改变. 如图 1 所示,数据集 $D=\{p_1,p_2,\cdots,p_9\}$ 有两个维度:酒店价格 price 和到机场的距离 distance. 假设游客理想的查询对象是酒店价格在 100 元上下并且距离机场 5 公里左右的数据点,那么该游客就可以将 $q=(100,5)$ 作为查询点,进行动态 skyline 查询. 动态 skyline 可以返回所有在各个维度上距离 q 点最近的点,把所有数据点映射到同一象限后(q 为原点的新的坐标空间),可以发现 p_1,p_2,p_3,p_6 是在各个维度上距离 q 最近的点,所以该动态 skyline 的结果集为 $\{p_1,p_2,p_3,p_6\}$.

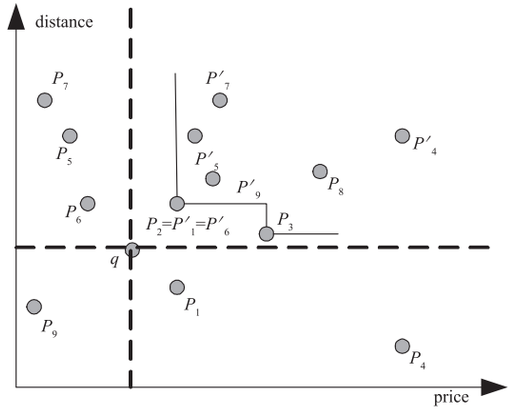

图1　动态skyline

虽然动态 skyline 在为用户指定查询要求方面提供了更大的灵活性,但是随着数据量的不断扩大,动态 skyline 的结果集也会增大,能够返回所有在各个维度上距离 q 点最近的点,忽略了查询点的维度方向性和数据的全局整体性. 因此,需要对动态 skyline 的结果集进行优化.

为了解决上述问题,首先,提出一种基于 MapReduce 的能够结合数据全局分布特点的动态 skyline 查询处理方法,称为增广动态 skyline,MR-Augmented Dynamic Skyline,MR-ADS,优化动态 skyline 结果集. 将原始数据按照维度信息进行分区,基于 MapReduce 框架在多个节点并行计算增广动态 skyline,增加数据的全局整体性. 其次,在此基础上,提出一种引入用户容忍度的增广动态 skyline 查询处理算法,MR-Tolerant Augmented Dynamic Skyline,MR-TADS. 根据用户容忍度缩减原始数据集,再计算增广动态 skyline. 另外,针对用户给出的容忍度可能导致缩减的原始数据集为空的情况,提出一种容忍度指针下移策略,在优先级低的维度上增加容忍,给出用户备选点,再求增广动态 skyline.

2　相关工作

2.1　基于 MapReduce 的 skyline 和动态 skyline

Zaman 等[1]提出将 skyline 查询应用于社交媒体领域以查询不被其他人支配的关键人物,并在 MapReduce 环境下实现大规模社交媒体数据的计算. Koh 等[2]提出 MapReduce 框架下两种 skyline 查询处理算法,采用基于网格和角度的空间划分方法过滤大量非 skyline 数据点,提高了并行处理 skyline 查询的效率. 王淑艳[3]等提出运用超平面投影方法划分低维和高维数据集,实现了在 MapReduce 上处理 skyline 查询的算法. Wang[4]等提出基于 MapReduce 框架解决大数据集上的空间 skyline 的有效方案. Park[5]等提出一种 MapReduce 框架下基于四叉树划分方法进行空间过滤的并行 skyline 查询处理算法. Li 等[6]提出一种基于 MapReduce 的动态 skyline 算法 MR-DSQ,将原始数据集分成许多小区域,通过比较各个区域相对于 q 的支配关系删除不存在动态 skyline 结果的区域,进而并行计算动态 skyline. Khandakar 等[7]提出一种有效的分布式动态 skyline 计算方法,应用于无线传感器网络. Md. Saiful 等[8]提出基于 Q + 树索引的动态 skyline 查询处理算法 MR-DSQ,基于 Q + 树索引进行分区,并预先删除不包括动态 skyline 结果的分区,减少了响应时间.

2.2　偏好 skyline 查询

Karim 等[9]针对多个用户给出的多种偏好定义了一种新的 skyline 选择标准,用来找到最符合这个集体偏好的结果,以最大限度同时满足这些用户的需求. Tassadit 等[10]提出一种增量式的 skyline 算法 EC^{2}Sky,在动态用户偏好下计算 skyline,减少了算法的执行时间和存储空间. Li 等[11]应用 MapReduce 框架计算基于用户偏好的 skyline,利用分布式的优势提高了算法的执行效率. Thanh 等[12]首次提出在不确定数据库中计算 top-k 典型 skyline,并基于用户偏好对此进行了研究. Li 等[13]提出基于用户偏好的子空间 skyline 查询,能够删除一些不符合用户偏好的 skyline 结果,减少了网络延迟和响应时间. M. Endres 等[14]提出一种基于网格的处理局部排序偏好 skyline 查询处理算法. Arun 等[15]提出一种大型数据库中处理不确定偏好的概率 skyline 查询处理算法,通过部分网格计算每个数据 skyline 概

率实现过滤. Qing Zhang 等[16]提出一种精确和近似的不确定偏好的概率 skyline 计算算法,并在真实和合成数据集下检验了算法的有效性.

3 增广动态 skyline 查询处理算法

3.1 问题定义

定义 1(Skyline) 给定一个数据集 $D=\{p_1,p_2,\cdots,p_{|D|}\}$,$p_i\in D$ 可表示为$\langle p_i(1),p_i(2),\cdots,p_i(d)\rangle$,$p_i(k)$表示 p_i 的第 k 维属性值. $p_i\in D$ 支配 $p_j\in D$,用符号记作 $p_i\prec p_j$,此时需要同时满足两个条件:p_i 在所有维度都不比 p_j 差,即

$$\forall k\in[1,d],p_i(k)\leqslant p_j(k) \tag{1}$$

至少有一个维度 p_i 比 p_j 优,即

$$\exists k\in[1,d],p_i(k)<p_j(k) \tag{2}$$

D 的 skyline 是 D 中不被其它数据点支配的所有数据点集合,记作 $SL(D)$,则

$$SL(D)=\{p_i\in D\mid \nexists p_j(\neq p_i)\in D \quad \text{s.t. } p_j\prec p_i\} \tag{3}$$

定义 2(动态 Skyline) 给定一个数据集 D 和一个查询点 q,相对于 q 点 $p_i\in D$ 动态支配 $p_j\in D$,用符号记作 $p_i\prec_q p_j$,此时需要同时满足两个条件:

相对于 q 点,p_i 在所有维度都不比 p_j 差,即

$$\forall k\in[1,d],|p_i(k)-q(k)|\leqslant|p_j(k)-q(k)| \tag{4}$$

相对于 q 点,至少有一个维度 p_i 比 p_j 优,即

$$\exists k\in[1,d],|p_i(k)-q(k)|<|p_j(k)-q(k)| \tag{5}$$

所有不被其他点关于 q 动态支配的点的集合组成了 q 的动态 skyline 集合,记作 $DSL(q,D)$,即

$$DSL(q,D)=\{p_i\mid p_i,p_j\in D,\nexists p_j(\neq p_i)p_j\prec_q p_i\} \tag{6}$$

定义 3(q 值分区及分区后节点 id 标识) 给定一个数据集 D 和一个查询点 q,把数据集 D 划分到 2^d 个分区中,位于分区 o 的数据点集合标记为 $Do(u)$,$u\in[0,2^d-1]$,每个分区 o 可表示为$(o(1)-,o(1)+)\cdots(o(d)-,o(d)+)$. 每个分区给予一个 id,由 $a_1a_2\cdots a_d$ 表示,id 是 u 的二进制表示. 如果$(o(i)-,o(i)+)=(-\infty,q(i))$,则 $a_i=0$;如果$(o(i)-,o(i)+)=(q(i),\infty)$,则 $a_i=1$.

如图 2 所示,给定一个二维数据集 D 和一个查询点 q,q 点左下方区域$(o(1)-,o(1)+)=(-\infty,q(1))$,$a_1=0$;$(o(2)-,o(2)+)=(-\infty,q(2))$,$a_2=0$. 该区域标记为分区 $Do(0)$,id 为 00. 同理其余三个分区分别标记为分区 $Do(1)$,id 为 01;$Do(2)$,id 为 10;$Do(3)$,id 为 11.

定义 4(增广动态 Skyline) 给定一个数据集 D 和一个查询点 q,D 的增广动态 skyline 记作 $ADS(q,D)$. ①$Do(0)$为全局最优分区,如果 $Do(0)$中存在数据,则删除 $Do(2^d-1)$,按照原始动态 skyline 的定义求得分区 $Do(1)$、$Do(2)$、$\cdots Do(2^d-2)$的动态 skyline 结果;如果 $Do(0)$中不存在数据,按照原始动态 skyline 的定义求得分区 $Do(1)$、$Do(2)$、$\cdots Do(2^d-1)$的动态 skyline 结果. ②$Do(0)$为全局最优分区,求 skyline 记作 $SL(Do(0))$. ③计算 $ADS(q,D)$. 00 分区存在数据时将①和②的结果合并,如式(7)所示. 00 分区不存在数据时的计算结果如式(8)所示.

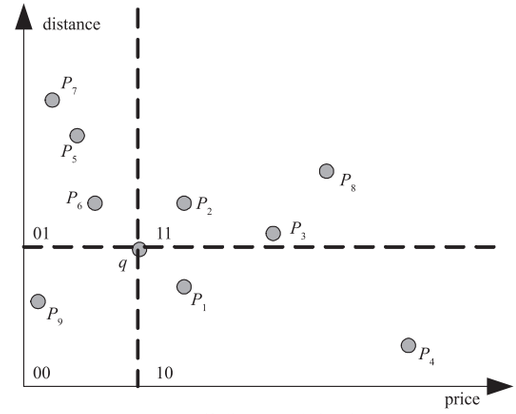

图2 q值分区o及节点id标识

$$ADS(q,D)=SL\{\bigcup_{u=1}^{u=2^d-2}DSL(q,D_o(u))\}\cup SL(D_o(0)) \tag{7}$$

$$ADS(q,D)=SL\{\bigcup_{u=1}^{u=2^d-1}DSL(q,D_o(u))\} \tag{8}$$

3.2 基于 MapReduce 的增广动态 skyline 查询

针对海量数据环境下处理增广动态 skyline 查询 $ADS(q,D)$,本文提出基于 MapReduce 的增广动态 skyline 查询算法 MR-ADS,该算法由两阶段 MapReduce 完成. 图 3 所示为 MR-ADS 算法的执行过程.

第一轮 MapReduce 的 Map 阶段,数据分区. 假设数据集 D 被随机分成了 n 个分片,由 n 个 Map 任务执行,输出每个点 p 所在的 id 号作为 key,p 点以及坐标作为 value.

第一轮 MapReduce 的 Reduce 阶段,全局最优分区 skyline 计算和非全局最优分区动态 skyline 计算. Map 任务得到的中间结果按照相同的 key 值进行分组,具有相同 key 值的 Map 输出结果输入到同一个 Reduce 任务,Reduce 任务接收到 Map 任务的中间结果后进行动态 skyline 查询. 在各个分区上进行动态 skyline 计算前,要进行如下判断:

(1)如果 id=00…0 的分区中存在数据点,则删除 id=11…1 的分区. 按照式(7)对 id=00…0 分区的数据计算 skyline,对其他分区计算动态 skyline.

(2)如果 id=00…0 的分区中不存在数据点,则按照式(8)计算其他分区的动态 skyline 结果集.

第二轮 MapReduce,生成结果集. 将非全局最优分区的动态 skyline 结果合并,对合并后的结果集进行 skyline 计算,并追加全局最优分区 skyline 计算结果.

MR-ADS 算法如算法 1 所示.

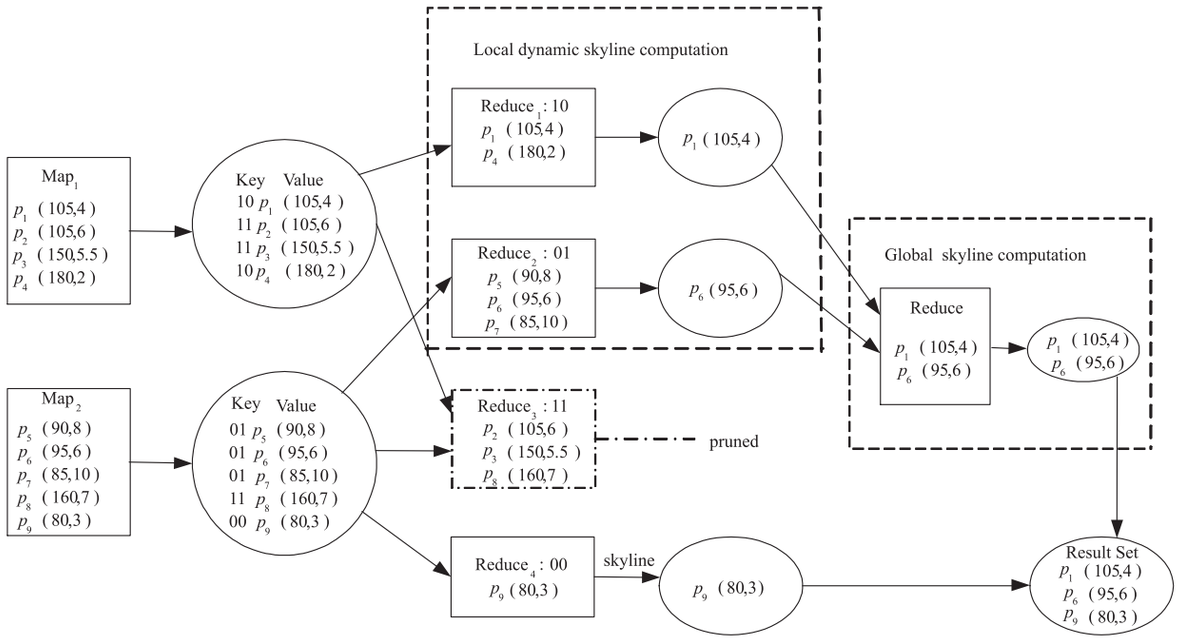

图3　增广动态Skyline查询

算法 1　MR-ADS Algorithm

输入:Data set D,query point q
输出:Result set R
1　for(each point $p \in D$) do
2　map function:point p mapped into its node;
3　if node(00…0) $\neq \varnothing$
4　prun node(11…1);
5　$s = SL(\text{node}(00\cdots0))$;
6　reduce function:$d = DSL(\text{else nodes}, q)$;
7　$R = s \cup SL(d)$;
8　end if node(00…0) $= \varnothing$
9　reduce function:$d = DSL(\text{else nodes}, q)$;
10　$R = SL(d)$;
11　emit R

4　引入用户容忍度的增广动态 skyline 查询处理算法

4.1　问题定义

定义 4(维度内部容忍度 T)　设用户给出的维度容忍度为一组数据集合,即 $T = \{t_1, t_2, \cdots, t_m\}$,其中 t_i 表示用户对于第 i 维最大容忍度为 t_i,超过 t_i 的数据点即不符合用户的需求.

定义 5(维度间优先级 P)　设用户给出的维度优先级 P 为一组数据的有序集合,即 $P = \{p_1, p_2, \cdots, p_m\}$,其中 p_i 代表 D 上的某维度 d_i,且有 $p_1 > p_2 > \cdots > p_m$. 此处“ > ”用于两维度之间,如 $p_1 > p_2$ 表示对于用户来说第一维比第二维重要. 可以从 d 个维度中选取 α 个维度加上容忍度,$\alpha \in [1, d]$,P 中的维度优先级顺序代表着用户对各维度的偏好.

定义 6(引入用户容忍度的增广动态 skyline)　根据用户给出的各维度容忍度和优先级,缩减原始数据集 D,缩减后的数据集记作 T-D,对 T-D 计算增广动态 skyline,得到引入用户容忍度的增广动态 skyline,记作 $ADS(q, T-D)$.

4.2　维度属性索引

为了提高计算效率,根据各个维度的容忍度可以提前把不满足容忍度的数据点删掉,减少不满足用户偏好的数据,因此,提出维度属性索引 DLink.

维度属性索引 DLink 为一组二维数据集合〈key, value〉,由维度属性值 key 和数据集 value 组成. 维度属性值 key 表示数据集 D 中某一维度所有可能的属性值,并按照 key 进行排序;数据集 value 表示某一维度有同一属性值的数据点集合. 数据集 D 有 d 维,则有 d 个 DLink.

仍以图 1 中的数据集为例,说明在 MapReduce 环境下属性索引 DLink 的构建过程. 为了更清晰地说明该索引的构建过程,加入第三维度,评价 Evaluation,如 p_1 的 Evaluation = 1/4,代表 p_1 的评价是四星评价. 图 4 为构建 DLink 的实例图.

Map 阶段:将有 d 个维度的原始数据集 D 随机分片,

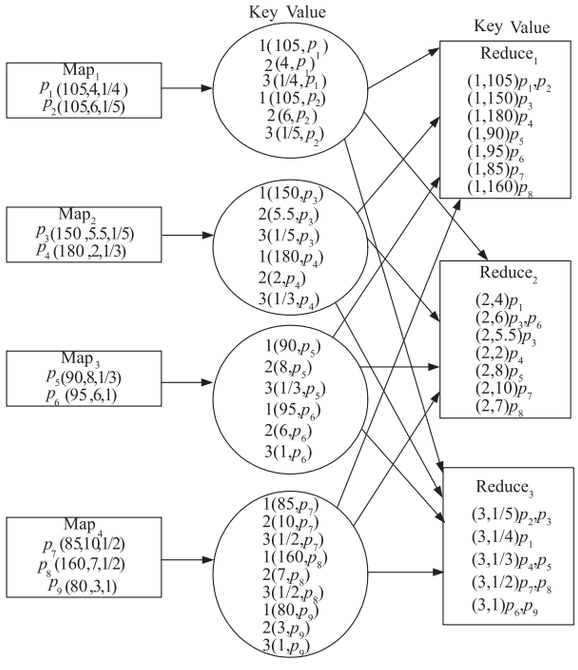

图4 构建DLink

假设分配到 n 个 Map 任务中执行,Map 任务输出格式是〈维度 i,(属性值,数据点)〉作为 key/value($1\leqslant i\leqslant d$).

Reduce 阶段:

(1)将相同 key 的数据点分配到同一个 Reduce 任务中,将同一维度同一属性值的数据点合并,输出格式是〈(维度 i,属性值),维度 i 有此属性值的数据点集合)〉作为新的 key/value. (2)将每个 Reduce 中的数据按照维度属性值升序排列成一个索引列表并输出 DLink. 每个 Reduce 任务输出的数据按照 key 值升序排序,得到属性 Price、Distance 和 Evaluation 的 DLink,如表 1 所示.

表 1 价格、距离和评价的 DLink

Price		Distance		Evaluation	
Key	Value	Key	Value	Key	Value
80	$\{p_9\}$	2	$\{p_4\}$	1/5	$\{p_2,p_3\}$
85	$\{p_7\}$	3	$\{p_9\}$	1/4	$\{p_1\}$
90	$\{p_5\}$	4	$\{p_1\}$	1/3	$\{p_4,p_5\}$
95	$\{p_6\}$	5.5	$\{p_3\}$	1/2	$\{p_8,p_7\}$
105	$\{p_1,p_2\}$	6	$\{p_2,p_6\}$	1	$\{p_6,p_9\}$
150	$\{p_3\}$	7	$\{p_8\}$		
160	$\{p_8\}$	8	$\{p_5\}$		
180	$\{p_4\}$	10	$\{p_7\}$		

4.3 基于维度属性索引的值缩减及结果获取

定义 7(维度容忍度指针 DPoint) 设选取的每一个 DLink 都有一个维度容忍度指针 DPoint,初始指向该维容忍度 t 所对应 DLink 的 key,如果没有对应 t 的 key,则 DPoint 初始指向小于 t 的最大 key.

得到维度属性值索引后,需要根据给出的维度容忍度预先把不满足容忍度的数据点删除. 设 R 集合存储当前 DLink 根据用户偏好删除后剩余的数据点. 用户给出容忍度 T,Reduce 阶段将每一维度的 DLink 中大于该维容忍度的数据删除,对得到的所有 R 集合取交集得到缩减后的结果. 如图 5 所示,给出 $q(95,4,1/3)$,price 的容忍度是 105,distance 的容忍度是 6,evaluation 的容忍度是 1/4,各维的优先级相同,各维的 DLink 分别表示为 DLink_p,DLink_d,DLink_e. $R_p=\{p_1,p_2,p_5,p_6,p_7,p_9\}$,$R_d=\{p_1,p_2,p_3,p_4,p_6,p_9\}$,$R_e=\{p_1,p_2,p_3\}$,$R_p\cap R_d\cap R_e=\{p_1,p_2\}$.

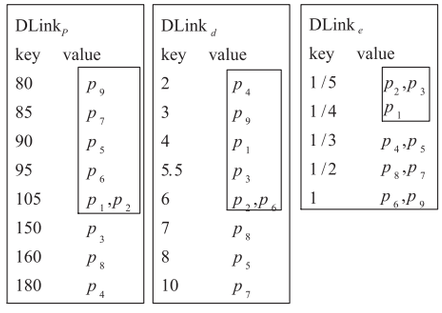

图5 基于维度属性索引的值缩减

最后,在获得根据用户容忍度进行数据缩减之后的数据集上进行增广动态 skyline 计算,即可得到引入用户容忍度的动态 skyline 查询处理结果. 如对上述根据用户容忍度缩减后的结果集$\{p_1,p_2\}$进行增广动态 skyline 查询,得到的最终结果仍是$\{p_1,p_2\}$,至此本次查询结束. MR-TADS 算法如算法 2 所示.

算法 2 MR-TADS Algorithm

输入:T,P,α,DLink,DPoint
输出:$ADS(q,T\text{-}D)$
1 map function:for all points,emit〈维度 i,(属性值,数据点)〉
2 reduce function:construct $\mathrm{DLink}_1\cdots\mathrm{DLink}_\alpha$
3 $R=\mathrm{DLink}(0,\mathrm{DPoint})$;
4 $T\text{-}D=R_1\cap R_2\cap\cdots\cap R\alpha$
5 if ($T\text{-}D\neq\varnothing$)
6 emit $ADS(q,T\text{-}D)$
7 else while($T\text{-}D\neq\varnothing$)
8 $\mathrm{DPoint}_\alpha=\mathrm{DPoint}_\alpha+1$;
9 $R_\alpha=\mathrm{DLink}_\alpha(0,\mathrm{DPoint}_\alpha)$;
10 $T-D=R_1\cap R_2\cap\cdots\cap R\alpha$;
11 emit $ADS(q,T-D)$ using Algorithm1

4.4 容忍度指针下移策略

针对用户给出的容忍度可能导致的缩减原始数据集为空的情况,提出一种容忍度指针下移策略,在优先级低的维度上增加容忍,给出用户备选点,避免根据用户容忍度无法得到结果的情况. 需要根据用户

给出的各维度优先级通过放松容忍来得到结果,从优先级 F 最低的维度开始进行逐级放松. 如给出优先级 price > distance > evaluation,三个维度的 DLink 中分别有一个 DPoint 指向初始容忍度所对应的索引,如图 6 所示.

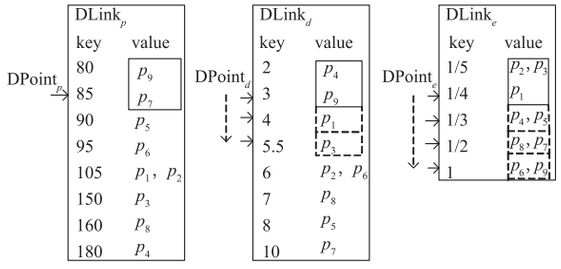

图6　容忍度指针下移示例

此例中,evaluation 为优先级最低的维度,DPoint$_e$ 下移一位,此时 $R_e = \{p_1, p_2, p_3, p_4, p_5\}$, $R_d = \{p_4, p_9\}$, $R_p = \{p_7, p_9\}$, $R_e \cap R_d = \{p_4\}$,但 $R_e \cap R_d \cap R_p = \varnothing$. 此时从优先级次低的维度开始,即后维度 distance 和 evaluation 的 DPoint 同时下降一个索引,此时仍有 $R_e \cap R_d \cap R_p = \varnothing$. distance 和 evaluation 的 DPoint 再同时下降一位,此时有 $R_e \cap R_d \cap R_p = \{p_9\}$,故基于维度属性索引缩减后的数据集 $T\text{-}D = \{p_9\}$. 由于缩减后的数据集只有 p_9,不需再计算增广动态 Skyline,最终查询结果为 $\{p_9\}$,至此本次查询结束.

5　实验测试与分析

5.1　实验环境和数据集

本文实验环境是由 12 台高速千兆网络连接的 PC 机组成,每个 PC 机的配置为 Intel Pentium G3220 3.00GHz CPU,4GB 内存,操作系统为 ubuntu14,其中 1 台 PC 机作为 Master 节点,其它 PC 机作为 Slave 节点. 采用的 Hadoop 平台版本为 2.6.

实验采用合成数据集和真实数据集进行测试. 合成数据集采用文献[17]中的生成工具生成不同维度和不同大小的均匀分布数据集. 数据集的大小变化区间为 $1\times10^8 \sim 6\times10^8$ 条数据,默认数据条数为 3×10^8,维度变化区间为[2,12],默认维度是 4,每一维的值域为[1,100]. 查询点 q、各维度容忍度和优先级在值域内由随机函数给出. 真实数据集采用文献[18]的地理位置属性进行测试,包括北美地区 33696 个人口稠密地点和文化地标的属性维度为 6 的独立分布数据.

实验选取 MR-DSQ[6] 算法和 MR-QDSQ[8] 同本文的 MR-ADS 算法和 MR-TADS 算法从算法响应时间、算法结果集大小、算法比较次数以及用户容忍度对 MR-TADS 算法结果集大小的影响 4 个方面进行对比.

5.2　实验结果与分析

5.2.1　响应时间的对比分析

图 7 表示默认维度下四种算法的响应时间随数据量变化的情况,图 8 表示在默认数据量下四种算法的响应时间随维度的变化情况. 随着数据量和维度的增加,本文算法比另外两种算法的优势更加明显. MR-DSQ 算法仅在 MapReduce 的环境下处理动态 skyline. MR-QDSQ 算法在 Q + 树索引中提前删除了不包含动态 skyline 结果的节点,一定程度上减少了响应时间. 由于 MR-TADS 算法和 MR-ADS 算法都删除了大量的冗余数据,所以响应时间都低于其它两种算法.

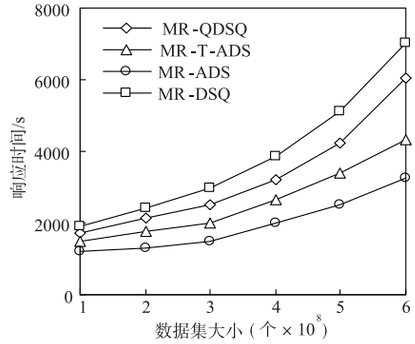

图7　数据集大小对响应时间的影响

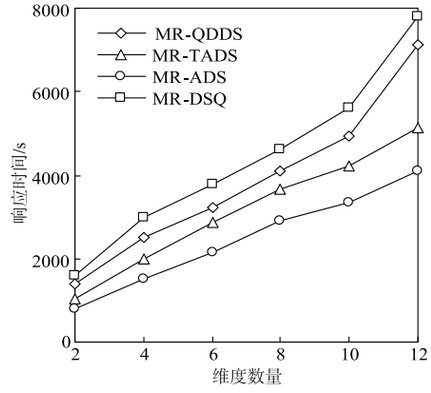

图8　数据集维度对响应时间的影响

5.2.2　结果集大小的对比分析

图 9 表示默认维度下四种算法的结果集大小随数据量变化的情况,图 10 表示数据量为 1×10^8 时四种算法的结果集大小随维度的变化情况. MR-TADS 算法比 MR-DSQ 算法和 MR-ADS 算法的结果集都要小很多. 原因是 MR-TADS 算法按照容忍度缩小了增广动态 skyline 查询的数据集. MR-ADS 算法删除了大量冗余数据,但是由于算法没有根据用户容忍度缩减原始数据集,所以 MR-ADS 算法的结果集要少于 MR-DSQ 和 MR-QDSQ 算法,多于 MR-TADS 算法.

5.2.3　比较次数的对比分析

图 11 表示默认维度下四种算法的数据点间支配关系比较次数随数据量变化的情况. MR-TADS 算法数

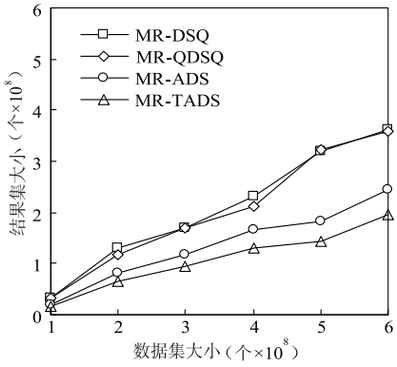

图9 数据集大小对结果集大小的影响

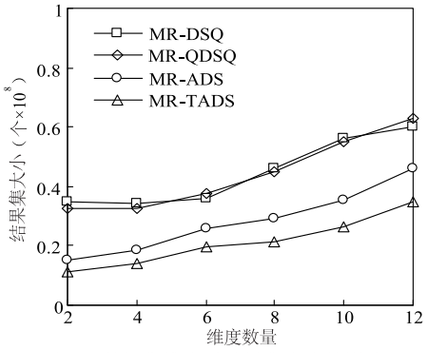

图10 数据集维度对结果集大小的影响

据点之间支配关系的比较次数比其余算法小. MR-TADS 算法根据用户容忍度删除了不符合用户需求的点,缩减了原始数据集. MR-DSQ 算法、MR-ADS 算法的原始数据集并没有变化. MR-QDSQ 算法基于 Q + 树数据索引预先删除不包括动态 skyline 结果的分区,减少了数据点间无意义的支配计算.

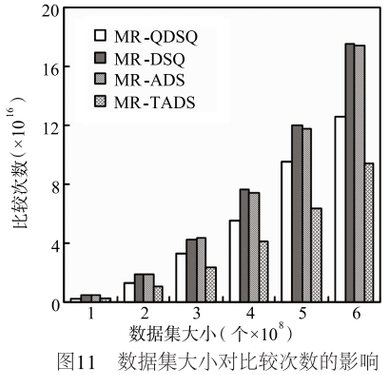

图11 数据集大小对比较次数的影响

5.2.4 容忍度对 MR-TADS 算法结果集大小的影响

本实验分析容忍度对 MR-TADS 算法结果集大小的影响. 数据集的大小为 1×10^8 条,维度是 6. 从 6 个维度中选取 α 个维度加上容忍度,各维容忍度初始值为 min + β(max-min),min 表示该维度最小属性值,max 表示该维度最大属性值,$\beta\in(1/8,2/8,\cdots,6/8)$,$\alpha\in[1,6]$.

图 12 表示在上述数据量和维度下,MR-TADS 算法的结果集大小随 α 变化的情况. 随着加入容忍度的维度数量 α 的增加,MR-TADS 算法结果集越来越小. 当 $\alpha=5$ 时,MR-TADS 算法结果集为空. 原因是在多个 DLink 中删除数据后,经过多次求交集运算,得到的集合可能为空. 当 $\alpha=5$ 和 $\alpha=6$ 时,分别降低优先级最低维度的容忍度,MR-TADS 算法的结果集数据不为空.

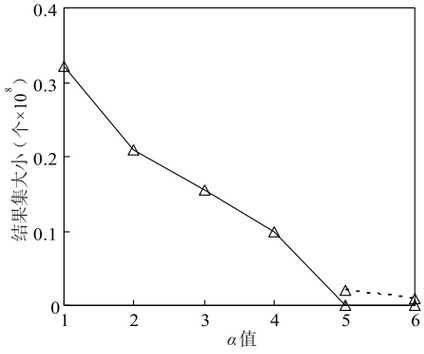

图12 α值对MR-TADS算法结果集大小的影响

图 13 表示在图 12 中的数据量和维度下,MR-TADS 算法的结果集大小随 β 变化的情况. 当 $\alpha=5$ 时,随着 β 增加,优先级最低维度的容忍度随着增加,MR-TADS 算法的结果集也随之增加. 说明当计算结果集为空时,从最低优先级的维度开始增加容忍度,可以给用户提供备选数据.

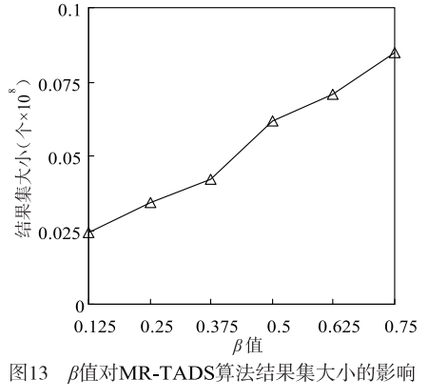

图13 β值对MR-TADS算法结果集大小的影响

5.2.5 真实数据集

本实验在真实数据集上对 MR-ADS 算法和 MR-DSQ 算法、MR-QDSQ 算法的响应时间进行测试. 查询点 q、优先级在值域内由随机函数给出. 由于真实数据集无用户容忍度,故对 MR-TADS 算法没有测试.

图 14 表示在真实数据集上,三种算法的响应时间随节点数变化的情况. 随着集群中节点个数的增加,三种算法的响应时间接近,并且均有降低. 由于数据集较小,三种算法的响应时间在 MapReduce 环境下相差不

多,但随着节点数量增加,均有减少,表明算法均有良好的并行性. 真实数据集的规模不足够大导致响应时间主要受 MapReduce 框架性能的限制.

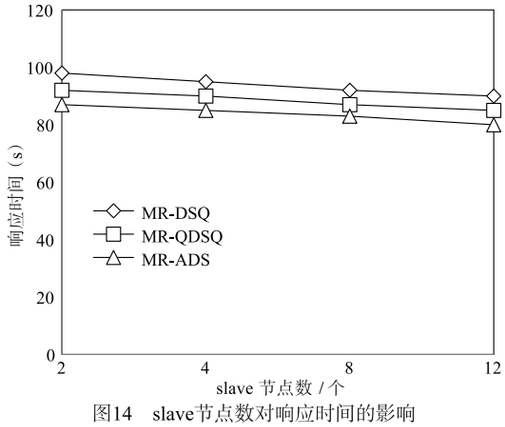

图14　slave节点数对响应时间的影响

6　结束语

本文针对基于 MapReduce 框架如何实现海量数据的增广动态 skyline 查询进行深入研究. 本文提出了增广动态 skyline 算法-MR-ADS,将原始数据按照维度信息进行分区,基于 MapReduce 框架在多个节点并行计算动态 skyline,提供给用户全局更优的数据,优化了动态 skyline 的结果集. 在此基础上,为了满足用户对某些维度有容忍度的情况,本文提出一种引入用户容忍度的增广动态 skyline 查询算法 MR-TADS,该算法根据用户容忍度缩小原始数据集,很大程度上减少中间结果的比较次数,并且提高了结果集的准确度. 最后,本文通过实验对提出的增广动态 skyline 算法进行了测试和评估. 实验结果表明,本文提出的增广动态 skyline 算法具有良好的有效性、准确性和可用性.

参考文献

[1] Asif Zaman, Md. Anisuzzaman Siddique, Annisa, Yasuhiko Morimoto. Finding key persons on social media by using MapReduce skyline[J]. International Journal of Networking and Computing, 2017, 7(1): 86 – 104.

[2] Jia-Ling Koh, Chia-Ching Chen, Chih-Yu Chan, et al. MapReduce skyline query processing with partitioning and distributed dominance tests[J]. Information Sciences, 2017, 375: 114 – 137.

[3] 王淑艳,杨鑫,李克秋. MapReduce 框架下基于超平面投影划分的 Skyline 计算[J]. 计算机研究与发展,2014,51(12):2702 – 2710.
Wang Shuyan, Yang Xin, Li Keqiu. Skyline computing on MapReduce with hyperplane-projections-based Partition [J]. Journal of Computer Research and Development, 2014, 51(12): 2702 – 2710. (in Chinese)

[4] Wenlu Wang, Ji Zhang, Min-Te Sun, et al. Efficient parallel skyline evaluation using MapReduce [A]. International Conference on Extending Database Technology[C]. Venice, Italy: Springer Press, 2017. 426 – 437.

[5] Yoonjae Park, Jun-Ki Min, Kyuseok Shim. Efficient processing of skyline queries using MapReduce [J]. IEEE Transactions on Knowledge & Data Engineering, 2017, 29(5): 1031 – 1044.

[6] Yuanyuan Li, Wenyu Qu, et al. Parallel dynamic skyline query using MapReduce[A]. International Conference on Cloud Computing and Big Data[C]. Wuhan, China: IEEE Press, 2015. 95 – 100.

[7] Khandakar Ahmed, Nazmus S. Nafi, et al. Enhanced distributed dynamic skyline query for wireless sensor networks [J]. J Sensor and Actuator Networks, 2016, 5(1): 2.

[8] Md. Saiful Islam, Chengfei Liu, J. Wenny Rahayu, et al. Q + Tree: an efficient quad tree based data indexing for parallelizing dynamic and reverse skylines[A]. International on Conference on Information and Knowledge Management [C]. Indianapolis USA: ACM Press, 2016. 1291 – 1300.

[9] Karim Benouaret, Djamal Benslimane, Allel HadjAli. Selecting skyline web services for multiple users preferences [A]. International Conference on Web Services[C]. Hawaii USA: IEEE Press, 2012. 635 – 636.

[10] Tassadit Bouadi, Marie-Odile Cordier, et al. Computing skyline incrementally in response to online preference modification [J]. Trans Large-Scale Data-and Knowledge-Centered Systems, 2013, 10: 34 – 59.

[11] Yuanyuan Li, Wenyu Qu, Zhiyang Li, et al. Skyline query based on user preference with MapReduce[A]. International Conference on Dependable, Autonomic and Secure Computing[C]. Dalian: IEEE Press, 2014. 153 – 158.

[12] Ha Thanh Huynh Nguyen, Jinli Cao. Preference-based top-k representative skyline queries on uncertain databases [A]. Pacific-Asia Conference on Knowledge Discovery and Data Mining [C]. Ho Chi Minh City, Vietnam: Springer Press, 2015. 280 – 292.

[13] Yuanyuan Li, Zhiyang Li, Mianxiong Dong, et al. Efficient subspace skyline query based on user preference using MapReduce[J]. Ad Hoc Networks, 2015, 35(C): 105 – 115.

[14] Markus Endres, Timotheus Preisinger: Beyond skylines: explicit preferences[A]. International Conference on Database Systems for Advanced Applications [C]. Suzhou China: Springer Press, 2017. 327 – 342.

[15] Arun K. Pujari, Venkateswara Rao Kagita, et al. Efficient computation for probabilistic skyline over uncertain preferences[J]. Information Sciences, 2015, 324(C): 146

－162.
[16] Qing Zhang, Pengjie Ye, Xuemin Lin, et al. Skyline probability over uncertain preferences[A]. International Conference on Extending Database Technology[C]. Genoa, Italy: ACM Press, 2013. 395－405.
[17] Stephan Børzsønyi, Donald Kossmann, Konrad Stocker: The skyline operator[A]. International Conference on Data Engineering[C]. Heidelberg, Germany: IEEE Press, 2001. 421－430.
[18] ChoroChronos[DB/OL]. http://www.chorochronos.org/.

作者简介

丁琳琳 女,1983 年生于辽宁阜新,辽宁大学信息学院副教授、硕士生导师,研究方向为大数据管理、分布式数据管理、图数据管理等.
E-mail: dinglinlin@lnu.edu.cn

崔子强 男,1992 年出生于山东威海,辽宁大学硕士研究生,研究方向为海量数据查询.
E-mail: ziqiang_c@126.com

尹显坤 男,1992 年生于湖南邵阳,辽宁大学硕士研究生,研究方向为大数据管理.
E-mail: jo12fjhh14@163.com

王俊陆 男,1988 年生于辽宁丹东,辽宁大学中级实验师,研究方向为大数据技术、图数据管理等.
E-mail: wangjunlu@lnu.edu.cn

宋宝燕(通信作者) 女,1965 年生于辽宁铁岭,辽宁大学信息学院教授、硕士生导师,CCF 高级会员、ACM 会员,研究方向为数据库理论和技术、RFID 数据流处理技术、大数据管理、图数据管理等.
E-mail: bysong@lnu.edu.cn