Journal of Computer Applications
计算机应用,2021,41(6):1686 - 1693
ISSN 1001-9081
CODEN JYIIDU
2021-06-10
http://www.joca.cn

文章编号:1001-9081(2021)06-1686-08
DOI:10.11772/j.issn.1001-9081.2020091453

距离–关键字相似度约束的双色反k近邻查询方法

张　豪*,朱　睿,宋秩尧,方　鹏,夏秀峰
(沈阳航空航天大学 计算机学院,沈阳 110136)
(*通信作者电子邮箱13623212292@163.com)

摘　要:针对空间关键字双色反k近邻查询返回结果质量较低的问题,提出了基于距离–关键字相似度约束的双色反k近邻查询方法。首先,通过设置一个阈值将查询结果中质量较低的用户给过滤掉,从而避免了查询结果中出现空间距离相对较远的用户,保证了查询结果质量;然后,为支持该查询,提出了一种关键字多分辨率网格矩形树(KMG-Tree)索引来管理数据;最后,提出了基于Six-region算法的Six-region-optimize算法来提高查询处理效率。Six-region-optimize算法的查询效率相较baseline和Six-region算法分别平均提高了约85.71%和23.45%。基于真实时空数据进行实验测试和分析,实验结果验证了Six-region-optimize算法的有效性和高效性。

关键词:关键字;双色反k近邻查询;空间距离;相似度约束;查询效率
中图分类号:TP311　　**文献标志码**:A

Bichromatic reverse k nearest neighbor query method based on distance-keyword similarity constraint

ZHANG Hao*, ZHU Rui, SONG Fuyao, FANG Peng, XIA Xiufeng
(*College of Computer Science*, *Shenyang Aerospace University*, *Shenyang Liaoning* 110136, *China*)

Abstract: In order to solve the problem of low quality of results returned by spatial keyword bichromatic reverse k nearest neighbor query, a bichromatic reverse k nearest neighbor query method based on distance-keyword similarity constraint was proposed. Firstly, a threshold was set to filter out the low-quality users in the query results, so that the existence of users with relatively long spatial distance in the query results was avoided and the quality of the query results was ensured. Then, in order to support this query, an index of Keyword Multiresolution Grid rectangle-tree (KMG-tree) was proposed to manage the data. Finally, the Six-region-optimize algorithm based on Six-region algorithm was proposed to improve the query processing efficiency. The query efficiency of the Six-region-optimize algorithm was about 85.71% and 23.45% on average higher than those of the baseline and Six-region algorithms respectively. Experimental test and analysis were carried out based on real spatio-temporal data. The experimental results verify the effectiveness and high efficiency of the Six-region-optimize algorithm.

Key words: keyword; bichromatic reverse k nearest neighbor query; spatial distance; similarity constraint; query efficiency

0　引言

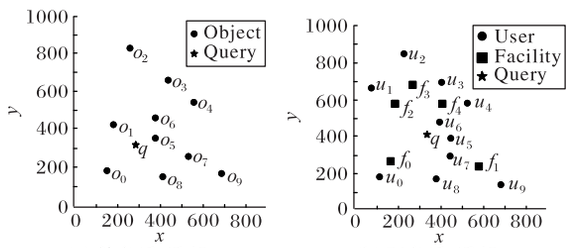
随着移动互联网技术的不断发展,基于位置的社交网络(Location-Based Social Network, LBSN)服务应用越来越广泛。许多学者研究了面向LBSN的查询处理问题应对不同服务类型下的业务请求。

在众多查询处理问题中,面向反k近邻空间关键字(Reverse Spatial Keyword k Nearest Neighbor, RSKkNN)[1]查询是一类重要问题,在市场分析、决策支持和交通信息等领域具有重要应用。具体地,在客户推荐系统中,该查询可以帮助商家根据客户偏好有针对性地利用短信等手段为客户推荐商品信息,从而增加营业收入。在交通信息管理领域,它可以通过分析乘客的用车偏好和位置信息,为乘客推荐车辆。

传统反k近邻查询可分为单色反k近邻查询(简称单色查询)和双色反k近邻查询(简称双色查询)。具体地,给定一组对象o和一个单色查询q,查询返回所有以q作为其k近邻的对象。与之不同,给定一组用户U、一组设施F和一个双色查询设施q,查询返回以设施q为k近邻的所有用户。反k近邻查询主要应用于推荐场景,其中单色反k近邻查询主要应用于同类型对象之间的推荐。例如,为某位用户推荐一些与他距离比较近的志趣相投的人。而双色反k近邻主要应用于两种不同类型对象之间的推荐,相较于单色反k近邻查询的应用更加广泛,比如推荐场景中更多的是为商场提供潜在的消费者,为司机提供潜在的乘客等,这些都是单色反k近邻查询无法做到的,所以本文重点研究双色反k近邻。接下来本文

收稿日期:2020-09-18;**修回日期**:2020-12-09;**录用日期**:2020-12-10。　**基金项目**:国家自然科学基金资助项目(61702344)。
作者简介:张豪(1996—),男,河北石家庄人,硕士研究生,主要研究方向:大数据;　朱睿(1982—),男,辽宁沈阳人,副教授,博士,CCF会员,主要研究方向:大数据;　宋秩尧(1995—),男,辽宁鞍山人,硕士研究生,主要研究方向:大数据;　方鹏(1994—),男,甘肃平凉人,硕士研究生,主要研究方向:新媒体大数据;　夏秀峰(1964—),男,山东胶南人,教授,博士,主要研究方向:数据库、数据仓库、数据挖掘。

以$k=2$为例简单说明传统反k近邻的查询规则,如图1(a)所示,对象$\{o_0,o_1,o_5,o_6\}$的k近邻对象包含q,所以对象q的单色反k近邻查询返回结果为$\{o_0,o_1,o_5,o_6\}$;如图1(b)所示,用户$\{u_0,u_5,u_6,u_7,u_9\}$的k近邻设施包含q,所以设施q的双色反k近邻查询结果为$\{u_0,u_5,u_6,u_7,u_9\}$。本文研究双色反k近邻查询。



(a) 单色反k近邻　　(b) 双色反k近邻

图1　传统单色和双色反k近邻

Fig. 1　Traditional monochromatic and bicolor reverse k nearest neighbors

传统反k近查询不能根据设施和用户需求针对性地返回查询结果,而带关键字约束的反k近邻查询可以很好解决此问题。鉴于此类问题的重要性,许多学者针对此类问题展开研究。例如:Lu等[1]首次研究了基于关键字约束的反k近邻查询问题。给定查询点q和对象集合D,查询计算q和D中对象空间上的距离以及关键字间的相似性。以此为基础,算法利用打分函数评价各对象对于查询点q的重要性,并返回得分最高的k个对象。Zhao等[2]提出了一种相关反近邻布尔空间关键字查询(Ranked Reverse boolean Spatial Keyword Nearest Neighbors, Ranked-RSKNN)问题。该查询返回一定数量与查询点相关程度最高的对象。

然而,这些研究存在的共性问题是没有考虑查询点与查询结果的距离对查询结果造成的影响,很多距离查询点很远但是关键字相似度较高的对象也将返回给用户。显然,和一些距离查询点较近但关键字相似度较低的对象相比,这些查询结果的质量较低。

因此,本文研究距离-关键字相似度约束的双色反k近邻(Distance-Keyword Reverse k Nearest Neighbor, DKRkNN)问题。和以往研究不同,DKRkNN通过引入阈值φ,过滤掉一些距离查询点相对较远的用户。为高效支持此类查询,本文需面对以下挑战:

1)高效的过滤能力。空间文本数据同时具有空间位置和关键字属性。因此,如何同时根据空间位置和关键字对数据进行筛选是极具挑战的。

2)高效的验证能力。对每个用户进行验证都需要遍历一次设施集。因此,在验证时如何尽可能少地遍历设施是极具挑战的。

针对上述挑战,本文提出了一种基于多分辨率网格树查询处理框架支持查询。本文主要工作如下:

1)提出了一种关键字多分辨率网格矩形树(Keyword Multiresolution Grid rectangle-Tree, KMG-Tree)索引管理设施和用户数据。该索引同时具有以下优点:①通过对网格中的单元格添加关键字信息,以达到通过这两个属性批量过滤数据的目的;②相对平衡的结构,通过对传统的多分辨率网格进行改进,KMG-Tree用分辨率较高网格划分数据密集的区域;用分辨率较低网格划分数据稀疏的区域。这样一来,它可以保证索引各节点维护的对象数目大致相同从而保证了索引的平衡性。

2)提出了基于Six-region-optimize的过滤算法。该算法将整个空间等分为6个区域,根据位置关系定义过滤区域。这样一来,被过滤区域覆盖的网格节点可被安全过滤。此外,与查询点关键字无交集的网格节点也可被安全过滤。

3)提出了基于Six-region-optimize的验证算法。在验证用户时,只需要访问该区域及相邻区域的对用户可能有影响的设施,这样可以快速找到用户的最近邻及k近邻,一定程度上提高了验证的效率。

1　相关工作

LBSN平台的数据规模越来越庞大,在做大数据查询时首先要对大数据进行分析,例如对于大数据分类问题,Xia等[3]提出了粒度球邻域粗糙集(Granular Ball Neighborhood Rough Set, GBNRS)分类方法,该方法自适应地为每个对象生成不同的邻域,从而具有更大的通用性和灵活性,并提高了在公共基准数据集上的分类精度和性能。对于大数据"k-均值"问题,Xia等[4]提出了一种称为"Ball k-means"的方法,用球来描述每个簇,减少了点质心距离的计算,提高了算法的性能。对于大数据噪声监测问题,Xia等[5]提出了一种完整有效的随机森林方法,通过模拟网格的生成和扩展来检测类噪声,提高了噪声监测的性能。在这些大数据分析的基础上许多学者根据大数据推荐问题研究了反k近邻查询。

区域剪枝是处理空间反k近邻查询的方法之一,其主要思想是将空间分为几个区域,并针对每块区域的空间距离对用户进行剪枝。Stanoi等[6]提出了一种基于Six-region剪枝的方法。该方法以查询点为中心将空间分为6个相等的区域。在每个区域中找到查询点的k近邻设施,以该设施到查询点距离为半径定义剪枝区域。

半空间剪枝法的核心思想是找到查询对q与设施对象f的中垂线。以此为基础,Tao等[7-8]提出了半空间剪枝的算法,给定一个设施f和一个查询q,f和q之间的垂直平分线$B_{f:q}$将空间分成两部分。让$H_{f:q}$表示包含f的半空间,$H_{q:f}$表示包含q的半空间。$H_{f:q}$中的每个用户u满足$dist(u,f)<dist(u,q)$。换句话说,位于$H_{f:q}$中的每个用户u被f剪枝,被k个这样的设施剪枝的用户可以被过滤。Cheema等[9-11]提出影响区概念,影响区是指当且仅当用u位于该区域内时,u为q的RkNN。建造影响区的方法是绘制所有设施的半空间,被小于k个设施过滤的区域为影响区。Lee等[12]提出了相关反最近邻查询,该查询通过对数据对象与查询对象q的相关性进行排序,返回受查询对象q影响最大的t个数据对象。

Yiu等[13]提出了在大型图上的反k近邻查询,并提出了两种剪枝方法:一种是在网络节点被访问后立即进行修剪的急切算法;另一种是在发现数据点时对搜索空间进行修剪的懒惰算法,实现了在大型图上的反k近邻查询。李佳佳等[14]提出了面向时间依赖的路网中的反k近邻查询,该查询提出了

计算节点到达时间的方法，利用该到达时间查询出多个候选k近邻结果。Hidayat等[15]提出了近似反最近邻查询，该查询将一些不是查询点的反最近邻但距离查询点也很近的点返回到了查询结果中。

以上几个算法都只是在空间或路网中反k近邻查询应用很好的算法，并不能够解决反k近邻空间关键字查询问题。Lu等[1]首次提出了反k近邻空间关键字查询问题，将空间位置和关键字通过度量公式结合在一起，并设计了一种空间文本混合索引结构IUR-Tree（Intersection-Union R-Tree）。Zhao等[2]提出了一种相关反近邻空间关键字查询，简称Ranked-RSkNN。该查询可以返回固定数量的对象，并将这些对象按照查询相关程度的大小进行排序。该查询能保证找到一定数量的结果返回。

然而，在反k近邻空间关键字查询问题中，没有考虑空间距离太远和关键字相似度过低的情况，得到的查询结果并非是查询点所期望的结果。因此，研究基于距离-关键字相似度约束的反k近邻空间关键字查询问题具有重要意义。

2 问题定义

前文已经提到现有的空间关键字双色反k近邻查询都没有考虑查询点与查询结果之间的距离对查询结果造成的影响，很多距离查询点很远但是关键字相似度较高的对象也将返回给用户。显然，和一些距离查询点较近但关键字相似度较低的对象相比，这些查询结果的质量较低。具体地，在商场推销商品的场景下，通过双色反k近邻查询找到了一些潜在的消费者。当$k \geqslant 2$时，这些消费者可能存在这样一种情况，商场a的位置在城市的中心，但是它的反k近邻中存在某个消费者u在该城市的郊区，在该郊区存在另一个商场b也可以满足u的消费需求，商场a对于u来说太远了，所以消费者u一定不会舍近求远，那么查询返回的消费者u对于商场a来说质量是非常低的。本文针对该问题设置了一个阈值φ_r，通过阈值φ_r可以将这些质量非常低的对象给过滤掉。下面详细介绍本文查询问题的形式化定义。

在介绍问题之前，本文首先定义一组符号。给定用户集合U和设施集合F，任意用户$u \in U$、任意设施$f \in F$均可用二元组$\langle l, key\rangle$表示。其中，$u.l$表示了u的位置信息，$u.key$表示了用户的关键字信息。给定任意查询q，它可用五元组$q\langle l, key, k, \varphi_r, \varphi_k\rangle$表示。$\varphi_r$表示用户到查询设施距离与其到最近邻设施距离比值的阈值；φ_k表示用户与查询设施关键字相似度阈值。给定用户u和设施f，本文利用欧氏距离$dist(u, f)$计算u与f之间的距离。给定u和其最近邻$NN(f)$，本文利用式(1)计算φ_r，利用式(2)计算u与q所对应关键字的相似程度。接下来介绍关键字-距离约束条件下的双色反k近邻查询。

$$Ratio(q,u)=\frac{dist(q,u)}{dist(u,NN(f))} \tag{1}$$

$$Sim(q,u)=\frac{(q.k \cap u.k)}{u.k} \tag{2}$$

定义1 距离-关键字相似度约束的双色反k近邻查询（DKRkNN）。给定一组带关键字的用户U、一组带关键字的设施F、一个带关键字的查询设施q、用户到查询设施距离与到其最近邻设施距离的比值阈值φ_r和关键字相似度阈值φ_k，返回满足条件$Ratio(q,u) \leqslant \varphi_r$和$Sim(q,u) \geqslant \varphi_k$，并以查询设施$q$为$k$近邻的每个用户$u$。

接下来，通过如下实例详细说明DKRkNN查询。通过设施和用户来区分不同的数据类型。图1(b)为每个对象在二维空间中的位置。图中五角星q表示商场在空间中的位置，正方形f_i表示其他商场在空间中的位置，圆形u_i表示用户在空间中的位置（后文图中的正方形点代表设施，圆形点代表用户）。表1分别为用户以及用户与商场q的关键字相似度、用户到商场q的空间距离。给定查询q，假设查询值$k=2$、$\varphi_r=2$、$\varphi_k=0.3$，关键字集合为{water, egg, meat, coffee, banana, milk}，q的R2NN为$\{u_0, u_5, u_6, u_7, u_9\}$（见图1(b)）。$u_0$的最近邻为$f_0$，$u_9$的最近邻为$f_1$，通过计算发现$Ratio(q,u_0)=3.05>2$，$Ratio(q,u_9)=3.47>2$（见表1）。因此，$u_0$和$u_9$不满足要求。进一步地，通过相似度计算发现用户$\{u_5, u_6, u_7\}$的关键字信息与商场$q$的关键字的信息相似度分别为{0.09, 0.71, 0.50}，显然$Sim(q,u_5)<0.3$，最终，查询结果为$\{u_6, u_7\}$。

表1 用户与设施之间的关系

Tab. 1 Relationship between users and facilities

用户	相似度	到查询点的距离	最近邻设施距离
u_0	0.50	$dist(q,u_0)$=220.91	$dist(f_0,u_0)$=72.3
u_1	0.63	$dist(q,u_1)$=360.69	$dist(f_2,u_1)$=142.13
u_2	0.25	$dist(q,u_2)$=451.22	$dist(f_3,u_2)$=167.63
u_3	0.80	$dist(q,u_3)$=288.62	$dist(f_4,u_3)$=100.05
u_4	0.36	$dist(q,u_4)$=261.73	$dist(f_4,u_4)$=110.00
u_5	0.09	$dist(q,u_5)$=118.80	$dist(q,u_5)$=118.80
u_6	0.71	$dist(q,u_6)$=92.20	$dist(q,u_6)$=92.20
u_7	0.50	$dist(q,u_7)$=161.55	$dist(f_1,u_7)$=148.66
u_8	0.21	$dist(q,u_8)$=225.61	$dist(f_1,u_8)$=206.16
u_9	0.55	$dist(q,u_9)$=424.38	$dist(f_1,u_9)$=122.07

3 基于多分辨率网格树的查询处理框架

3.1 框架概述

本文利用基于Six-region改进的Six-region-optimize过滤-验证框架支持基于距离-关键字相似度约束的双色反k近邻查询问题。为支持高效查询，本文首先提出了一种基于数据依赖的多分辨率网格树索引实现数据的高效管理，随后本文提出了一种基于Six-region-optimize过滤算法实现对用户和设施的过滤，最后本文提出了一种基于Six-region-optimize的验证算法实现对候选用户的验证。

给定n个带关键字的用户，首先，根据用户空间位置将空间等分为$\sqrt{n}\times\sqrt{n}$个带关键字信息的单元格。然后，根据单元格中数据的密度对单元格进行合并和分裂操作，从而得到一个多分辨率网格结构。最后，通过多叉树维护网格之间的空间位置关系。该索引可以利用网格的特性降低维护代价，保证索引的平衡性，并在访问过程中根据单元格的关键字信息对用户实现批量过滤。

给定查询$q\langle q.l, q.k\rangle$，过滤算法以q为中心将整个空间划分为6个区域$\{S_0, S_1, \cdots, S_5\}$。对于每一个区域，算法利用用户和设施之间空间位置关系与关键字信息分别对设施和用户进

行过滤,得到候选用户集合R和用于验证用户的设施集合V。这样做的好处是可以快速过滤掉一些用户和设施,从而减少计算代价。

根据过滤算法得到候选用户集R和用于验证用户的设施集V,根据验证算法对候选集R中的每个用户进行验证。验证每个候选用户需要遍历设施集V,将满足查询要求的用户存储到结果集*Result*中。

3.2　多分辨率网格树索引KMG-Tree

本文提出了基于关键字多分辨率网格矩形树(KMG-Tree)索引分别管理用户和设施。该索引是一种以多分辨率网格为基础的数据结构。它对数据密集的区域使用分辨率较高的单元格进行划分,对数据稀疏的区域使用分辨率较低的单元格进行划分,从而保证每个单元格存储用户或设施数量基本相同。

首先,介绍一组KMG-Tree索引用到的符号。KMG-Tree索引的每个节点t使用一个5元组$\langle id, h, c, k, leaf \rangle$表示。KMG-Tree每个节点$t$唯一标识用$t.id$表示;$t.c$是一个数组,用来存储节点$t$的所有孩子节点;$t.leaf$是一个标记,它标记节点$t$是否为KMG-Tree索引的叶子节点,值为true表示为该索引的叶子节点,值为false表示为该索引的非叶子节点;$t.k$用来存储每个节点中所有对象的关键字信息,该关键字信息是一个基于哈夫曼树构造的一个bit串。算法可以根据每个节点t存储的bit串快速过滤对象。使用哈夫曼编码表示关键字信息可以有效地降低空间代价。与直接使用节点存储关键字方式相比,基于哈夫曼编码的bit串存储策略可以有效压缩存储空间。

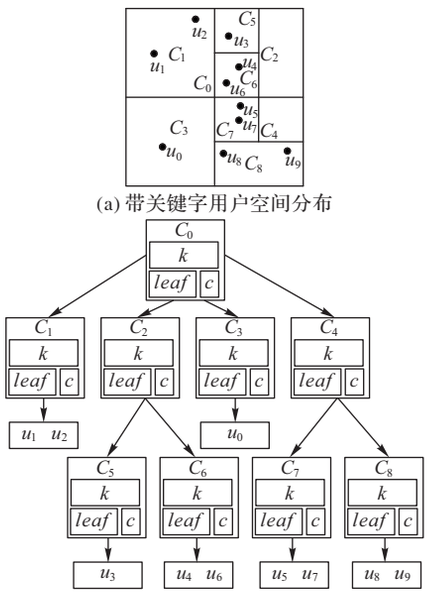
接下来,举一个例子详细说明索引的构建过程。本文首先指定一个阈值m用来表示每个单元格存储对象数量的最大值。以$m = 2$时为例,整个空间存在10个对象,将整个空间划分为$\sqrt{10} \times \sqrt{10}$个单元格,根据每个单元格中对象的数量判断单元格是否需要进行合并或者分裂。合并和分裂的条件如下:

1)当单元格中对象数量大于m时进行分裂;

2)当单元格中对象数量小于m时进行合并。

经过合并与分裂之后得到的逻辑空间如图2(a)所示,C_0包含四个节点C_1、C_2、C_3和C_4;C_1包含两个对象u_1和u_2;C_2包含两个节点C_5和C_6;C_3包含一个对象u_0;C_4包含两个节点C_7和C_8;C_5包含一个对象u_3;C_6包含两个对象u_4和u_6;C_7包含两个对象u_5和u_7;C_8包含两个对象u_8和u_9。这时所有单元格都满足对象数量的限制。将C_1、C_3、C_5、C_6、C_7和C_8的*leaf*值设置为true。构建完成后的KMG-Tree索引结构如图2(b)所示,索引中每个节点以及每个单元格集成对应的关键字集合存储在$t.k$中。

接下来,介绍构建KMG-Tree时间复杂度。构建KMG-tree索引需要遍历所有的对象,构建用户索引首先遍历所有用户时间代价为$O(|U|)$,遍历过程需要将用户存储到网格节点中。根据网格划分得到$\sqrt{|U|} \times \sqrt{|U|}$个单元格,索引的高度为$\log \sqrt{|U|} \times \sqrt{|U|}$,分裂与合并的时间代价为$O(\log \sqrt{|U|} \times \sqrt{|U|})$,所以构建用户KMG-Tree索引的总代价为$O(|U| + \log \sqrt{|U|} \times \sqrt{|U|})$。同理构建设施KMG-Tree索引的总代价为$O(|V| + \log \sqrt{|V|} \times \sqrt{|V|})$。



(a) 带关键字用户空间分布

(b) KMG-Tree结构

图2　索引结构示意图

Fig. 2　Schematic diagram of index structure

3.3　基于Six-region-optimize的过滤算法

Six-region算法在处理双色反k查询时,不仅没有考虑关键字因素,而且只是对用户进行了过滤,这样在验证用户时需要遍历整个设施R-tree。针对以上问题本文提出了基于Six-region改进的过滤算法Six-region-optimize,该算法结合设施空间位置与关键字结合对用户和设施进行过滤。在介绍算法之前先介绍一个引理和一个定理。

引理1　以查询点q为中心将空间划分为6个相等区域,在区域$i(i = 0, 1, \cdots, 5)$中,找到距离设施q第k近的设施f_k。以q为圆心、$r_i = dist(q, f_k)$为半径做一个扇形sec_i,位于sec_i外面的用户可以被过滤。

证明　在区域$i(i = 0, 1, \cdots, 5)$中,位于sec_i外面的任意用户u的kNN一定不包含q,因为在sec_i里面一定存在k个设施满足$dist(u, f) < dist(u, q)$。

定理1　以查询点q为中心将空间划分为6个相等区域,在区域$i(i = 0, 1, \cdots, 5)$中,以q为圆心、$2R_i$为半径做一个扇形SEC_i,位于SEC_i外面的设施可以被过滤。其中R_i为区域i及其相邻区域中r的最大值。r是根据引理1得到的每个区域的sec_i的半径。

证明　在区域$i(i = 0, 1, \cdots, 5)$中,位于SEC_i外面的任意设施f一定不会影响查询点q成为任何候选用户u的kNN。因为每个区域中的任意候选用户u一定满足$dist(q, u) \leqslant R_i$,而位于i区域SEC_i外面的任意设施f一定满足$dist(f, u) \geqslant R_i$。

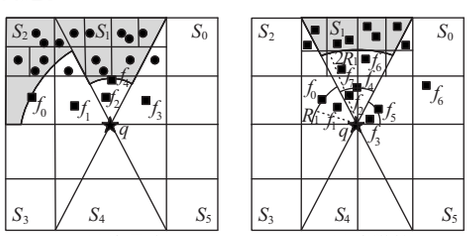
接下来,介绍基于Six-region-optimize的过滤算法。该算法过滤过程分为批量过滤和局部过滤。针对用户的批量过滤,在遍历用户KMG-Tree过程中根据定理1对用户在空间位置上进行过滤,在区域$i(i = 0, 1, \cdots, 5)$中位于sec_i外面的网格节点可以直接被过滤。然后,根据关键字信息对用户进行过

滤，没有被直接过滤的网格节点 t 需要检查该节点的关键字信息，如果 t 满足 $t.k \cap q.k=\varnothing$，则 t 可以被直接过滤。针对用户的局部过滤，访问那些没有被直接过滤网格节点的用户，根据定理 1 过滤掉那些在区域 $i(i=0,1,\cdots,5)$ 中位于 sec_i 外面的用户。

例如图 3(a)所示，以查询 q 为中心的整个空间被划分为六个相等的扇形区域 S_0~S_5。在区域 S_1 和 S_2 中，查询点 q 的 2 近邻设施分别为 f_4 和 f_0，所以基于这两个设施定义了这两个区域的过滤空间，如图中阴影部分。被阴影部分完全覆盖的网格节点可以直接被过滤；被部分覆盖的网格节点将访问所包含的网格节点或用户，并将位于过滤空间中的网格节点或用户过滤。这样一来，算法无需要遍历所有用户便可批量过滤掉不满足要求的用户。

针对设施的批量过滤，在遍历设施 KMG-Tree 过程中根据定理 1 对设施在空间位置上进行过滤，在区域 $i(i=0,1,\cdots,5)$ 中位于 SEC_i 外面的网格节点可以直接被过滤。针对设施的局部过滤，访问那些没有被直接过滤网格节点的设施，根据定理 1 过滤掉那些在区域 $i(i=0,1,\cdots,5)$ 中位于 SEC_i 外面的设施。

例如图 3(b)所示，在区域 S_1 和与其相邻的两区域 S_0 与 S_2 中，f_4、f_5 和 f_0 分别在各子区域中为查询 q 的 2 近邻。它们满足不等式 $dist(q,f_0) > dist(q,f_4) > dist(q,f_5)$。设 $R_1 = dist(q,f_0)$，在区域 S_1 中以 R_1 为半径做扇形 SEC_1，SEC_1 定义了过滤空间，如图中阴影部分。被阴影部分完全覆盖的网格节点可以直接过滤；被部分覆盖的网格节点将访问所包含的网格节点或设施，并将位于过滤空间中的网格节点或设施进行过滤。这样一来，算法无需要遍历所有设施便可批量过滤掉不影响候选用户的设施。



(a) 过滤用户 (b) 过滤设施

图 3 空间属性过滤用户和设施

Fig. 3 Spatial attributes filtering users and facilities

基于 Six-region-optimize 的过滤算法见算法 1。利用两个队列分别缓存用户节点和设施节点如算法 1 第 2)行。对用户进行过滤，不能被过滤的用户存储到候选用户集合 R 中如算法 1 的第 2)~7)行。对设施进行过滤，不能被过滤的设施存储到验证集合 V 中如算法 1 中第 8)~13)行。

算法 1 基于 Six-region-optimize 的过滤算法。

输入 查询设施 q，k 值，用户树 *u-root* 和设施树 *f-root*，每个区域扇形半径 r；

输出 集合 R，集合 V。

1) 初始化 $R \leftarrow \varnothing$, $V \leftarrow \varnothing$;
2) Enqueue(*u-root*) to U, Enqueue(*f-root*) to F;
3) while $U \neq \varnothing$ do
4) $u \leftarrow$ dequeue(U);
5) $i \leftarrow$ region(u);
6) if $dist(u, q) < r[i]$ and $u.k \cap q.k \neq \varnothing$;
7) $R \leftarrow u$;
8) while $F \neq \varnothing$ do
9) $f \leftarrow$ dequeue(F);
10) $i \leftarrow$ region(f);
11) $max[i] \leftarrow i$ 及相邻区域扇形最大半径；
12) if $dist(f, q) < 2 * max[i]$
13) $V \leftarrow f$;
14) return R, V;

接下来，介绍 Six-region-optimize 算法过滤阶段的时间复杂度。该阶段首先需要访问设施 KMG-Tree 索引进行空间划分，划分需要的时间代价为 $O(\log|V|)$。然后，再同时访问用户和设施 KMG-Tree 索引，根据划分空间对用户和设施进行过滤，该阶段的时间代价为 $O(\log|V| + \log|U|)$。整个过滤阶段的总代价为 $O(2\log|V| + \log|U|)$。

3.4 基于 Six-region-optimize 的验证算法

Six-region-optimize 的验证算法通过过滤得到候选集 R 中的每个用户 u 进行验证，如果用户 u 同时满足条件 1)~3)时将用户存储到最终的结果集 *Result* 中。

1)用户 u 与查询设施 q 的关键字相似度 $Sim(q,u) \geqslant \varphi_k$。

2)以用户 u 为圆心、$dist(q,u)$ 为半径做一个圆，遍历集合 V 中的设施，圆中的设施数量小于 k。

3)$dist(u,q)$ 与 $dist(u,NN(u))$ 的比值 $Ratio(q,u) \leqslant \varphi_r$。

以 $\varphi_k = 0.3$、$\varphi_r = 2$ 和 $k = 2$ 为例说明一下。首先，计算候选用户 $u_i(i=0,1,\cdots,3)$ 与查询设施 q 的关键字相似度 $Sim(q, u_i)$，如表 2 所示，$Sim(q,u_0) = 0.71$、$Sim(q,u_1) = 0.63$、$Sim(q, u_2) = 0.09$ 和 $Sim(q,u_3) = 0.67$，所以 u_2 不是查询结果。然后，检查满足关键字相似度要求的候选用户的 kNN 是否包含查询设施 q，例如图 4 所示，以 u_1 为中心、$dist(q,u_1)$ 为半径的圆中有两个设施 f_1 和 f_6，所以 u_1 不是查询结果。u_0 和 u_3 为 q 的 R2NN。最后，计算 $dist(q,u_0)$ 与 $dist(u_0,NN(u_0))$ 比值 $Ratio(q,u_0)$=0.7 和 $dist(q,u_3)$ 与 $dist(u_3,NN(u_3))$ 比值 $Ratio(q,u_3)$=2.5。u_3 不满足条件，u_0 为查询结果。

表 2 关键字相似度和最近邻距离比值

Tab. 2 Keyword similarity and nearest neighbor distance ratio

用户	$Sim(q, u_i)$	$Ratio(q, u_i)$
u_0	0.71	1.3
u_1	0.63	3.2
u_2	0.09	1.0
u_3	0.67	2.5

基于 Six-region-optimize 的验证算法见算法 2。访问候选用户集合 R 如算法 2 中的第 2)~3)行。验证关键字相似度是否满足查询条件如算法 2 的第 4)行。验证用户的 RkNN 是否包含查询设施 q，并计算是否满足距离约束如算法 2 的第 5)~11)行。

算法 2 基于 Six-region-optimize 的验证算法。

输入 集合 R 和集合 V，查询设施 q，k 值，阈值 φ_r 和 φ_k；

输出 集合 *Result*。

1) 初始化 *Result* $\leftarrow \varnothing$;
2) foreach $u_i \in R$ do
3) $u \leftarrow u_i$;

```
4)    if Sim(u,q) ≥ φ_k;
5)      temp ← ∅;
6)      foreach f_j ∈ V do
7)        if dist(u,f) ≤ dist(u,q)
8)          temp++;
9)        if temp < k and Ratio(u,q) < φ_r
10)         Result ← u;
11)   return Result;
```

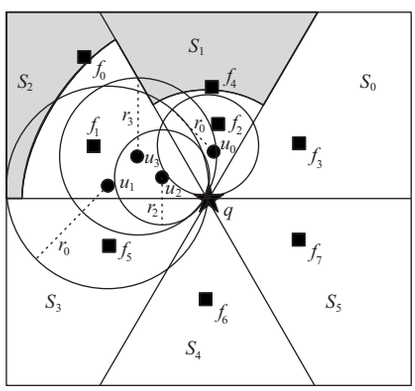


图4　空间范围验证用户

Fig. 4　Spatial scope authenticating users

接下来,介绍Six-region-optimize算法验证阶段的时间复杂度。在算法的过滤阶段已经得到了候选集R和验证集V。假设候选集的大小为K,验证集的大小为$2K$。可以明确的是$K \ll U$,并且$2K \ll V$。验证阶段需要或每个候选用户发出一个布尔范围查询,每个布尔范围查询大概需要访问2/3的验证集中的设施,所以该阶段的时间复杂度为$O\left(\frac{2}{3}K^2\right)$。

4　实验与结果分析

4.1　实验准备

实验的配置环境为:64位 Windows 10、操作系统,Java编程语言,实验内存为8 GB,CPU为Intel Core i5-4460处理器。采用的数据集如表3所示,包括两个真实数据集德国(GE)的部分地理位置信息和北京(BJ)的部分地理信息以及一个模拟地理位置的数据集(SD)。

4.2　结果分析

首先,对本文提出的DKRkNN查询与Ranked-RSKNN查询的返回结果进行比较验证该查询返回结果的有效性。然后,使用不同大小的数据集对本文提出的基于Six-region-optimize的过滤验证算法进行实验分析,将它与最基本的简称baseline的算法和Stanoi等[6]提出的Six-region算法进行对比,验证算法高效性。最后,通过改变参数的大小验证本文所提算法的稳定性。

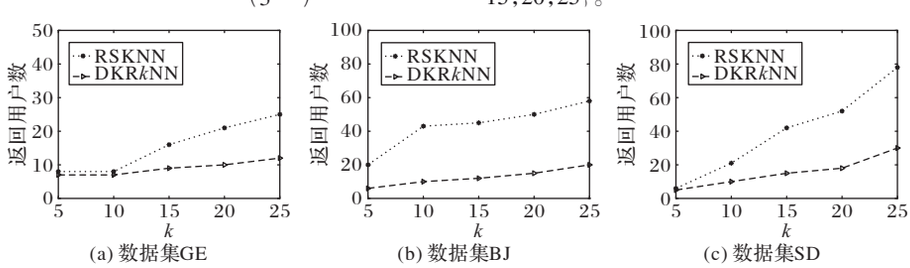
表3　实验数据集统计信息

Tab. 3　Statistical information of experimental datasets

数据集	实例数	数据集大小
德国(GE)	6 105	201 000
北京(BJ)	312 349	29 000 000
模拟数据(SD)	600 000	59 000 000

4.2.1　DKRkNN查询的有效性分析

通过比较DKRkNN查询和Ranked-RSkNN查询在三个数据集上的返回结果数量分析DKRkNN查询结果的有效性。如图5所示,设阈值$\varphi_r = 2$和$\varphi_k = 0.5$,设置不同的k值为{5,10,15,20,25}。



(a) 数据集GE　(b) 数据集BJ　(c) 数据集SD

图5　不同k值时Ranked-RSKNN和DKRkNN返回用户数量

Fig. 5　Number of returned users by Ranked-RSKNN and DKRkNN with different k values

通过图5可以发现,在不同数据集上,k值越大两个算法返回对象数量差距越大,表明DKRkNN查询过滤掉的无效的对象越多,从而验证了DKRkNN查询返回结果的有效性。

4.2.2　Six-region-optimize算法的效率分析

通过评估Six-region-optimize算法、baseline算法和Six-region算法在数据集大小不同时CPU处理时间的消耗分析算法的效率。数据集的大小不同时实验结果如图6所示。由图6可以看出,在德国(GE)数据集的大小为6 105时每个算法在时间上的消耗都非常低,Six-region和Six-region-optimize算法表现得都不是很优秀。随着数据集的大小不断增加,例如在北京(BJ)数据集的大小为312 349和模拟(SD)数据集的大小为600 000时,很明显baseline算法在时间上的消耗明显增加,而Six-region和Six-region-optimize算法时耗增加都不是很明显,而Six-region-optimize算法时间消耗是最小的,验证了Six-region-optimize算法的高效性。

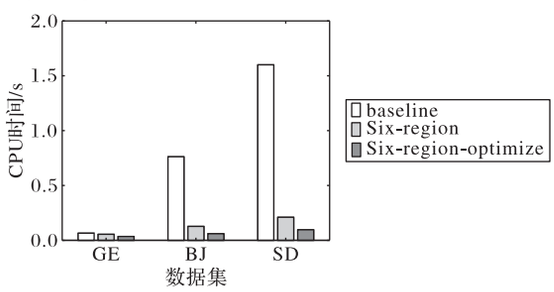


图6　数据集大小对不同算法时间消耗的影响

Fig. 6　Influence of dataset size on time consumption of different algorithms

4.2.3　Six-region-optimize算法的稳定性分析

接下来,通过改变参数的值比较Six-region-optimize算法、baseline算法和Six-region算法在三个不同数据集上的变化,

分析各个参数对于每个算法的影响。

1)k值对各算法的影响。

该组实验设阈值$\varphi_r = 2$和阈值$\varphi_k = 0.5$,设置不同的k值为{5,10,15,20,25},三个算法分别在数据集GE、BJ和SD上的变化情况如图7所示。由图7可以看出,随着k值增加三个算法在CPU时间上的消耗都在增加,这是因为k值越大满足条件的用户越多,从而过滤掉的数据就越少,需要验证的数据就越多。但是在数据集较大时,k值对Six-region和Six-region-optimize算法的影响很小,表明了k值对这两个算法影响很小,验证了算法的稳定性。

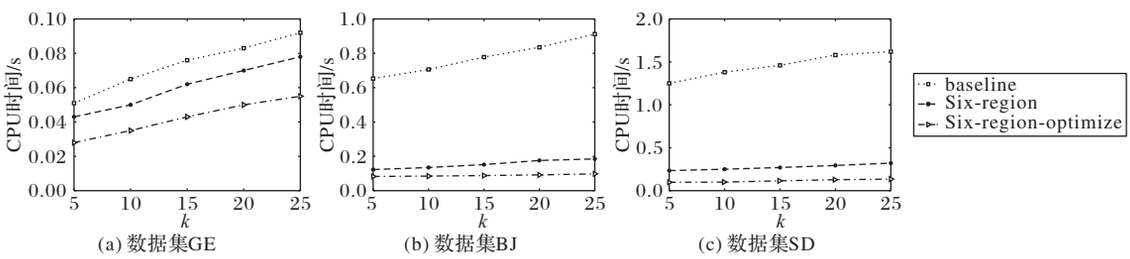
2)阈值φ_r对各算法的影响。

该组实验设$k = 5$和阈值$\varphi_k = 0.5$,设置不同的φ_r值为{1.5,2.0,2.5,3.0,3.5},三个算法分别在数据集GE、BJ和SD上的变化情况如图8所示。由图8(a)可以看出,三个算法在数据集GE上都基本不受影响,这是因为数据集GE非常小,距离阈值φ_r的变化对验证过程影响非常小,从而三个算法在CPU时间上的消耗也比较稳定。由图8(b)~(c)可以看出,baseline算法和Six-region算法随着阈值φ_r的增加在CPU时间上的消耗也有较明显增加,这是因为这两个算法在过滤阶段没有将大量设施过滤掉,这样使阈值φ_r增加时需要访问的设施也会大量增加。Six-region-optimize算法不会存在这样的情况,该算法在过滤阶段已经把大量不影响结果的设施给过滤掉了,表明了Six-region-optimize算法在CPU时间消耗上的稳定性。

3)阈值φ_k对各算法的影响。

该组实验设$k = 5$和阈值$\varphi_d = 9000$,设置φ_k值为(0.2,0.3,0.4,0.5,0.6),三个算法分别在数据集GE、BJ和SD上的CPU时间消耗变化情况如图9所示。

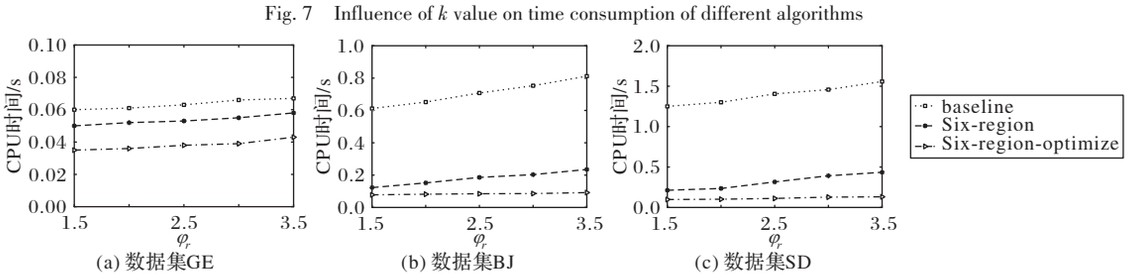
由图9可以看出,随着φ_k值的增加,三个算法的CPU时间消耗基本无变化,这是因为关键字相似度只需要在验证阶段计算一次。所以φ_k对三个算法都没什么影响,表明三个算法都是非常稳定的。



(a) 数据集GE (b) 数据集BJ (c) 数据集SD

图7 k值对不同算法时间消耗的影响

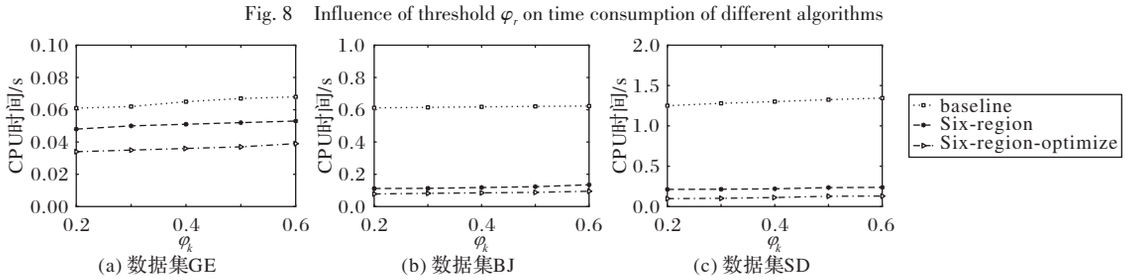
Fig. 7 Influence of k value on time consumption of different algorithms



(a) 数据集GE (b) 数据集BJ (c) 数据集SD

图8 阈值φ_r对不同算法时间消耗的影响

Fig. 8 Influence of threshold φ_r on time consumption of different algorithms



(a) 数据集GE (b) 数据集BJ (c) 数据集SD

图9 阈值φ_k对不同算法时间消耗的影响

Fig. 9 Influence of threshold φ_k on time consumption of algorithms

5 结语

本文研究了距离-关键字相似度约束的双色反k近邻查询(DKRkNN)问题,并提出了一种基于多分辨率网格树索引查询处理框架。该框架根据区域划分规定过滤区域,访问多分辨率网格树,得到用户候选集和影响用户成为查询结果的设施,对候选用户进行验证,得到最终的结果集。基于两个真实数据集和一个模拟数据集进行实验,实验结果验证了本文算法的有效性和高效性,表明所提框架能够有效处理该查询。但是随着社会的发展,查询的需求会越来越复杂,空间文本信息可能不再是以关键字的方式存在,有可能是一段文字。因此,未来的研究需要对空间文本做语义分析从而为用户提供更高质量的查询体验。

参考文献 (References)

[1] LU J, LU Y, CONG G. Reverse spatial and textual k nearest

neighbor search [C]// Proceedings of the 2011 ACM IGMOD International Conference on Management of Data. New York: ACM, 2011: 349-360.

[2] ZHAO P, FANG H, SHENG V S, et al. Monochromatic and bichromatic ranked reverse boolean spatial keyword nearest neighbors search [J]. World Wide Web, 2017, 20 (1): 39-59.

[3] XIA S, ZHANG Z, LI W, et al. GBNRS: a novel rough set algorithm for fast adaptive attribute reduction in classification [J]. IEEE Transactions on Knowledge and Data Engineering, 2020 (Early Access): 1-1.

[4] XIA S, PENG D, MENG D, et al. A fast adaptive *k*-means with no bounds [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020(Early Access): 1-1.

[5] XIA S, WANG G , CHEN Z, et al. Complete random forest based class noise filtering learning for improving the generalizability of classifiers [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(11): 2063-2078.

[6] STANOI I, AGRAWAL D, ABBADI A E. Reverse nearest neighbor queries for dynamic databases [C]// Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. New York: ACM, 2000: 44-53.

[7] TAO Y, PAPADIAS D, LIAN X. Reverse *k*NN search in arbitrary dimensionality [C]// Proceedings of the 2004 30th International Conference on Very Large Data Base. Amsterdam: Elsevier, 2004: 744-755.

[8] TAO Y, PAPADIAS D, LIAN X, et al. Multidimensional reverse *k*NN search [J]. The VLDB Journal, 2007, 16(3): 293-316.

[9] CHEEMA M A, LIN X, ZHANG Y, et al. Lazy updates: an efficient technique to continuously monitoring reverse *k*NN [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 1138-1149.

[10] CHEEMA M A, SHEN Z, LIN X, et al. A unified framework for efficiently processing ranking related queries [C]// Proceedings of the 2014 17th International Conference on Extending Database Technology. Berlin: Springer, 2014:427-438.

[11] CHEEMA M A, ZHANG W, LIN X, et al. Efficiently processing snapshot and continuous reverse k nearest neighbors queries [J]. The VLDB Journal, 2012, 21(5): 703-728.

[12] LEE K C K, ZHENG B, LEE W C. Ranked reverse nearest neighbor search [J]. IEEE Transactions on Knowledge and Data Engineering. 2008, 20(7): 894-910.

[13] YIU M L , PAPADIAS D, MAMOUIS N, et al. Reverse nearest neighbors in large graphs [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(4): 540-553.

[14] 李佳佳,李雨现,夏秀峰,等. 面向时间依赖路网的连续k近邻查询[J]. 计算机科学与探索,2019,13(5):788-799. (LI J J, LI Y X, XIA X F, et al. Continuous k-neighbor query for time dependent road network [J]. Journal of Frontiers of Computer Science and Technology, 2019, 13(5): 788-799.)

[15] HIDAYAT A, YANG S, CHEEMA M A, et al. Reverse approximate nearest neighbor queries [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(2): 339-352.

This work is partially supported by the National Natural Science Foundation of China (61702344).

ZHANG Hao, born in 1996, M. S. candidate. His research interests include big data.

ZHU Rui, born in 1982, Ph. D., associate professor. His research interests include big data.

SONG Fuyao, born in 1995, M. S. candidate. His research interests include big data.

FANG Peng, born in 1994, M. S. candidate. His research interests include big data of new media.

XIA Xiufeng, born in 1964, Ph. D., professor. His research interests include database, data warehouse, data mining.