



文章编号:1001-9081(2018)01-0152-07

DOI:10.11772/j.issn.1001-9081.2017051219

基于耦合相关度的空间数据查询结果自动分类方法

毕崇春^{1*}, 孟祥福¹, 张霄雁¹, 唐延欢¹, 唐晓亮², 梁海波¹

(1. 辽宁工程技术大学 电子与信息工程院, 辽宁 葫芦岛 125105; 2. 辽宁工程技术大学 软件学院, 辽宁 葫芦岛 125105)

(*通信作者电子邮箱 marxi@126.com)

摘要:由于空间数据库通常蕴含海量数据,因此一个普通的空间查询很可能会导致多查询结果问题。为了解决上述问题,提出了一种空间查询结果自动分类方法。在离线阶段,根据空间对象之间的位置相近度和语义相关度来评估空间对象之间的耦合关系,在此基础上利用概率密度评估方法对空间对象进行聚类,每个聚类代表一种类型的需求;在线查询处理阶段,对于一个给定的空间查询,在查询结果集上利用改进的C4.5决策树算法动态生成一棵查询结果分类树,用户可通过检查分类树分支的标签来逐步定位到其感兴趣的空间对象。实验结果表明,提出的空间对象聚类方法能够有效地体现空间对象在语义和位置上的相近性,查询结果分类方法具有较好的分类效果和较低的搜索代价。

关键词:空间数据库;聚类;耦合关系;查询结果分类

中图分类号: TP274.2 **文献标志码:**A

Coupling similarity-based approach for categorizing spatial database query results

BI Chongchun^{1*}, MENG Xiangfu¹, ZHANG Xiaoyan¹, TANG Yanhuan¹, TANG Xiaoliang², LIANG Haibo¹

(1. College of Electronic and Information Engineering, Liaoning Technical University, Huludao Liaoning 125105, China;

2. College of Software, Liaoning Technical University, Huludao Liaoning 125105, China)

Abstract: A common spatial query often leads to the problem of multiple query results because a spatial database usually contains large size of data. To deal with this problem, a new categorization approach for spatial database query results was proposed. The solution consists of two steps. In the offline step, the coupling relationship between spatial objects was evaluated by considering the location proximity and semantic similarity between them, and then a set of clusters over the spatial objects could be generated by using probability density-based clustering method, where each cluster represented one type of user requirements. In the online query step, for a given spatial query, a category tree for the user was dynamically generated by using the modified C4.5 decision tree algorithm over the clusters, so that the user could easily select the subset of query results matching his/her needs by exploring the labels assigned on intermediate nodes of the tree. The experimental results demonstrate that the proposed spatial object clustering method can efficiently capture both the semantic and location relationships between spatial objects. The query result categorization algorithm has good effectiveness and low search cost.

Key words: spatial database; clustering; coupling relationship; query result categorization

0 引言

随着移动网络的普遍应用,对于兴趣点(Point Of Interests, POI),如饭店、宾馆、旅游景点等的空间Web对象(简称空间对象)的查询已成为当前空间数据库和信息检索领域的研究热点。空间对象一般包含两类信息:空间信息(Spatial Information,如经纬度)和描述信息(Descriptive Information,如对象名称、特征等信息)。空间查询条件的基本形式为:{位置,〈关键字1,关键字2,...,关键字m〉},其中“位置”代表空间信息查询条件(通常用一对经纬度或一个地理范围表示);关键字代表文本查询条件^[1]。研究发现,目前大约有五分之一的Web查询都与空间位置相关^[2],并且这种

类型查询的比例还在不断增长。然而,由于空间数据库通常蕴含海量数据且空间查询又通常是试探性的,因此一个普通查询往往返回大量结果,这种情况被称为“信息过载”问题。例如,当用户向Yahoo房地产网站提交了一个寻找“位于西雅图市中心(位置查询)、价格较低(关键字查询)”的房屋查询条件后,系统会返回数以千计的房产信息。为了找到真正感兴趣的信息,用户需要逐条检查每个结果对象的相关性,这将浪费大量时间和精力。

为了解决信息过载问题,一些研究工作提出根据空间对象与查询条件的位置相近度和文本相关度对查询结果进行top-k排序^[3-7]。然而,这种方法返回的空间对象的排序次序固定,不能为用户提供多样选择,而不同用户的需求可能是不

收稿日期:2017-05-19;修回日期:2017-07-17。

基金项目:国家自然科学基金面上项目(61772249);辽宁省教育厅一般项目(LJYL018);辽宁省自然科学基金资助项目(20170540418)。

作者简介:毕崇春(1992—),男,辽宁丹东人,硕士研究生,CCF会员,主要研究方向:空间数据分析与查询;孟祥福(1981—),男,辽宁朝阳人,副教授,博士生导师,博士,CCF会员,主要研究方向:Web数据库查询、空间数据分析;张霄雁(1983—),女,山东烟台人,工程师,博士研究生,主要研究方向:时空数据库查询、城市计算;唐延欢(1992—),男,广东汕头人,硕士研究生,主要研究方向:空间数据挖掘、推荐系统;唐晓亮(1980—),男,辽宁阜新人,讲师,博士,主要研究方向:机器学习;梁海波(1995—),男,广西北海人,主要研究方向:数据挖掘、数据库查询。



同的。文献[10]指出,分类与排序是解决多查询结果问题的两个互补手段。本文提出一种空间对象查询结果分类方法,该方法能够在查询结果集上动态产生一棵分层的分类树,树的每个叶节点包含一类在位置和语义上都相近的空间对象,用户通过检查分类树中间节点的标签,可以逐步定位到其感兴趣的信息。下面用一个简单例子来说明本文分类方法的基本思想。

例1 对于Yahoo房地产搜索网站,图1给出了利用本文方法在查询条件“位于西雅图市中心且价格 $\leq 250\,000$ ”的查询结果上生成的一棵分类树。该树的每个节点上都带有一个描述房产特征信息的标签,用户可根据树各节点标签进行导航,进而找到他们所感兴趣的房产。

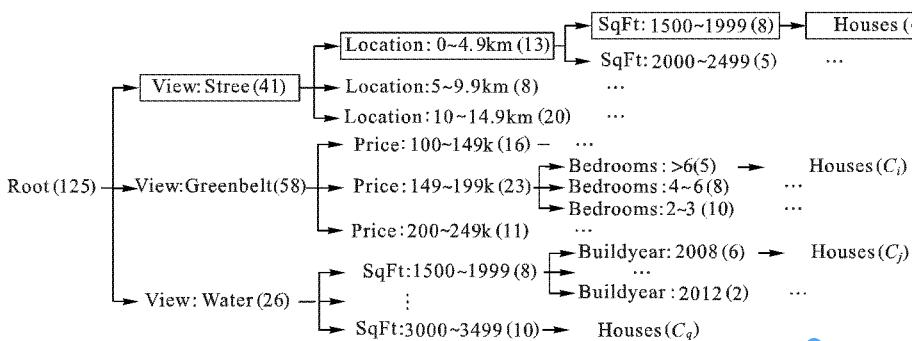


图1 Yahoo房产网站上的空间查询结果分类树实例

Fig. 1 Instance of category tree for spatial query results from Yahoo real estate website

本文的分类方法由两个步骤构成:第一步是在离线阶段分析空间对象之间的耦合关系(本文的耦合关系是指空间对象之间在位置信息和描述信息上存在的显式/隐式关联关系),并据此对空间对象进行聚类。现实中,如果用户对某个空间对象感兴趣,那么他通常也会对与该对象位置和语义相近的一类对象感兴趣,因此每个空间对象聚类都对应一种类型的用户需求。第二步,对于一个用户查询,根据在离线阶段产生的空间对象聚类,在查询结果集上利用改进的C4.5决策树算法自动生成一棵分类树,用户通过检查分类树标签最终导航到其感兴趣的结果对象。

1 相关工作

近年来,随着Web上的空间对象日益增多和移动网络的普遍应用,空间关键字的top- k 查询和排序方法受到了广泛关注。根据文献[8~9],这些空间查询方法可分为三类。1)布尔kNN(k Nearest Neighbor)查询^[5]:该类方法用于检索那些距离查询位置最近且文本描述包含所有查询关键字的前 k 个空间对象。2)top- k 范围查询^[6]:用于检索那些与查询关键字具有最高文本相关度且位于查询区域内的前 k 个空间对象。3)top- k kNN查询^[7]:该类方法根据空间对象的位置相近性和文本相关性进行top- k 检索和排序,排序分数根据对象到查询位置的距离和空间对象的文本描述与查询关键字的文本相关度的权重进行评估。

就我们所知,目前还没有对空间数据查询结果分类方法的研究工作。在关系数据库查询领域,文献[10]和[11]分别提出了基于贪心算法(Greedy Algorithm)和基于决策树分类算法的查询结果分类方法。Chakrabarti等^[10]提出了一种利用贪心算法构建查询结果分类树的方法,该方法利用系统中

的历史查询数据推断大多数用户的偏好,据此作为当前用户对每个分类属性感兴趣的概率。文献[11]利用了C4.5决策树算法,提出了一种两步解决方法:第一步对用户的历史查询数据聚类;第二步构建一棵搜索树来实现用户的个性化查询。然而,使用该方法构建分类树存在以下两个缺点:1)元组聚类仅依赖于用户历史查询数据,若历史查询数据不充分或不准确,就会导致算法失效;2)数值属性值域的划分采用二元划分,然而这种划分在实际应用中通常会导致不合适(过大或过小)的数值区间分割。本文方法借鉴了上述在关系数据库查询领域对查询结果进行分类的思想来对空间查询结果进行分类。需要指出的是,空间数据所包含的信息比关系数据更为复杂(包含位置和描述信息),对聚类和结果分类需要同时考察空间对象在位置和描述信息方面的相关性。

在文本分类^[12~14]和Web查询结果分类^[15~16]方面,研究者也进行了大量研究,但是分类对象与本文的分类对象不同:本文是对空间对象进行分类,空间对象同时包含了空间位置信息和文本描述信息,而文本分类方法仅能对文本信息进行分类。本文提出的分类树构建方法目的在于搜索代价最小化,而现存的文本分类方法仅考虑如何准确划分数据。

2 相关定义和解决方案

本章首先给出分类树及其搜索代价定义,然后描述解决方案。

2.1 查询结果分类树

空间对象包含位置信息和描述信息,本文将描述信息转换成〈属性,值〉的形式,这样每个空间对象的描述信息将构成一个关系元组。例如,在例1的Yahoo房产信息中,一个房屋的描述信息将由〈Price,14 999〉,〈View,water〉,〈Bedrooms,5〉,〈SqFt,2000〉等数据单元构成。

用 D 表示一个包含 n 个空间对象的数据集, A 表示 D 中空间对象描述信息的属性集 $\{A_1, A_2, \dots, A_m\}$ 。在此基础上定义查询结果分类树和预计搜索代价。

定义1 分类树。一个分类树 $T(V, E, L)$ 是由一个节点集合 V 、一个边集合 E 和一个标签集合 L 三部分构成。每个节点 $v \in V$ 都有一个描述该节点特征的标签 $lab(v) \in L$,它需要满足如下条件:1)该标签是指定在某个属性上的等值或范围查询条件(如 $Price < 100\,000$ 或 $Price$ between $100\,000$ and $150\,000$);2)节点 v 中包含的一组空间对象 $N(v)$ 必须满足它所有前驱(包括它自身)上的标签;3)一个中间节点的所有孩子节点上的标签都指定在同一属性上(该属性被称为分类属性),这些标签构成了对节点 v 中所有 $N(v)$ 个对象的不相交划分。

2.2 预计搜索代价

给定一个空间查询结果分类树 T ,用 v 表示一个叶节点,该节点包含了 T 中一组空间对象 $N(v)$, C_j 是集合 C 中的一个聚类(该聚类中包含了一组位置相近和语义相关的空间对象



集合), $C_j \cap v \neq \emptyset$ 表示叶节点 v 中包含了聚类 C_j 中的空间对象, $Anc(v)$ 表示节点 v 的前驱集合(包括 v 本身在内),但不包含根节点; $Sib(v)$ 表示节点 v 的兄弟集合(包含 v 本身在内)。令 K_1 和 K_2 分别表示访问叶节点中一个空间对象以及访问一个中间节点标签的代价, P_j 表示用户对于聚类 C_j 感兴趣的概率。

定义2 预计搜索代价。用户使用分类树找到所有相关对象的预计搜索代价为:

$$Cost(T, C) = \sum_{v \in leaf(T)} \sum_{C_j \cap v \neq \emptyset} P_j \left(K_1 N(v) + K_2 \sum_{v_i \in Anc(v)} |Sib(v_i)| \right) \quad (1)$$

该搜索代价由两部分构成:访问叶节点 v 中空间对象的代价和访问从根到 v 的中间节点的代价。用户首先需要检查从根节点到叶节点 v 路径上的每个中间节点及其所有兄弟上的标签,这一过程用户总共需要访问 $\sum_{v_i \in Anc(v)} |Sib(v_i)|$ 个中间节点;当用户到达叶节点 v 时,他需要检查 v 中所有的 $N(v)$ 个空间对象。 P_j 代表用户对叶节点 v 感兴趣的概率,后面将讨论 P_j 的计算方法。

例如,假设某用户对图1所示的分类树最上面的叶节点(“SqFt:1500 ~ 1999”)感兴趣的概率是100%,再假设 $K_1 = K_2 = 1$,则用户访问叶节点“SqFt:1500 ~ 1999”的搜索代价为:3(检查根下3个子节点标签的代价)+3(检查中间节点“View: Street”下3个孩子标签的代价)+2(检查中间节点“Location: 0 ~ 4.9 km”下2个孩子标签的代价)+8(检查叶节点“SqFt: 1500 ~ 1999”下8个空间对象的代价)=16。

2.3 解决方案

分类树构建的解决方案如下:

1) 空间对象耦合相关度评估与聚类。

首先利用欧氏距离计算两个空间对象在空间位置上的距离;然后利用语义相关度评估方法计算空间对象在描述信息上的语义相关度;最后,用式(2)将位置关系和语义关系相融合得到两个空间对象 o_i 和 o_j 之间的耦合相关度:

$$Sim(o_i, o_j) = \alpha \cdot E_sim(o_i, o_j) + (1 - \alpha) \cdot S_sim(o_i, o_j) \quad (2)$$

其中: $\alpha \in [0, 1]$ 是一个调节参数, $E_sim()$ 和 $S_sim()$ 分别代表 o_i 和 o_j 之间的位置相近度和语义相关度。

根据空间对象之间的耦合关系,利用基于概率密度估计的聚类方法把空间对象划分为 q 个类别 $\{C_1, C_2, \dots, C_q\}$,使得每个类别中包含的空间对象在位置和语义上都比其他类中的对象更为接近。

2) 分类树构建。

对于一个空间数据库 D 、一个聚类集合 C 和一个空间查询 Q ,本文目的是构建一棵分类树 $T(V, E, L)$,使得:①该树中的叶节点包含了 D 中满足查询 Q 的所有空间对象;②不存在另外一棵树 T' 既满足条件1)又使得搜索代价 $Cost(T', C) < Cost(T, C)$ 。算法1给出了构建查询结果分类树的过程。

算法1 查询结果分类树生成过程。

输入:空间数据库 D ,聚类集合 C 和空间查询条件 Q 。

输出:分类树 T 。

1) 离线阶段。计算空间数据库 D 中所有空间对象之间的耦合相关度,在此基础上对 D 中的空间对象进行聚类,得到一

个不相交的空间对象聚类集合 $C = \{C_1, C_2, \dots, C_q\}$,其中每个聚类 C_j ($1 \leq j \leq q$) 都关联一个用户对其感兴趣的概率 P_j 。

2) 在线阶段。在查询 Q 返回的结果集 T 上,使用 C_1, C_2, \dots, C_q 作为类标签,利用改进的C4.5算法构建一棵具有最小预计搜索代价 $Cost(T, C)$ 的分类树。

算法1的复杂度分析 在离线阶段(步骤1))需要计算出 D 中任意一对空间对象之间的耦合相关度,如果 D 中有 n 个空间对象,则离线阶段的计算复杂度为 $O(n^2)$ 。为了降低离线计算的工作量,可以以固定时间间隔计算耦合相关度或当数据更新程度较大时再计算耦合相关度。对于在线阶段(步骤2))的计算复杂度,本文将在第3章的算法3中进行分析。

3 空间对象的耦合相关度评估与聚类方法

本章首先描述空间对象之间的耦合相关度评估方法,然后提出如何根据空间对象的耦合相关度对其进行聚类。

3.1 耦合相关度评估

空间对象之间的耦合相关度是由空间对象在位置上的相近度和描述信息上的语义相关度构成。

1) 空间对象的位置相近度。

欧氏距离是现有研究工作中最常用的评估空间对象之间距离的方法^[3,6,9],本文也采用欧氏距离评估空间对象之间的位置距离,空间对象 o_i 和 o_j 的距离定义如下:

$$Dist(o_i, o_j) = \sqrt{\sum_{k=1}^n d(o_i^{(k)}, o_j^{(k)})} \quad (3)$$

其中, n 表示空间对象的空间维度。基于式(3),空间对象 o_i 和 o_j 之间在距离上的相近度定义如下:

$$E_Sim(o_i, o_j) = 1 - Dist(o_i, o_j) / Maxdist \quad (4)$$

式(4)是对位置相近度的归一化处理,其中 $Maxdist$ 代表空间数据库中所有空间对象之间的最大距离。

2) 空间对象的语义相关度。

现有方法通常采用VSM(Vector Space Model)及其改进方法评估两个文本之间的相关度,但现实中空间对象的描述之间通常存在复杂的语义关系。本文提出一种语义相关度评估方法。该方法是对文献[18]中提出方法的一种改进,其基本思想是:对于同一属性下的两个值,如果它们在同一属性值域中出现的次数(或者发生频率)类似,则表示这两个值具有相关性,这反映了一个属性中不同属性值之间的内耦合关系(intra-coupling relationship)。如前文所述,空间对象的描述信息将首先被转换成〈属性,值〉形式的关系元组,因此可使用属性值的出现频率来计算同一属性下不同值之间的内耦合相关度。具体来讲,对于包含 n 个空间对象的空间数据集 D ,首先将所有空间对象的描述信息转换成一个具有 m 个属性 $\{A_1, A_2, \dots, A_m\}$ 和 n 条元组 $\{t_1, t_2, \dots, t_n\}$ 的关系表。在此基础上,属性 A_j 中的一对属性值 x 和 y 的内耦合度相关度(Intra-similarity)计算方法如下:

$$IaS_{A_j}(x, y) = \frac{N_{A_j}(x) \cdot N_{A_j}(y)}{N_{A_j}(x) + N_{A_j}(y) + N_{A_j}(x) \cdot N_{A_j}(y)} \quad (5)$$

其中, $N_{A_j}(x)$ 和 $N_{A_j}(y)$ 分别表示属性 A_j 中包含属性值 x 和 y 的元组个数。从式(5)可以看出,如果两个属性值在同一值域中的出现频率越高,则它们之间的内耦合相关度越大。

同一属性上的两个属性值之间除了具有内耦合关系外,它们的相关度还受到其他属性的影响,这里称之为间耦合相



关度。属性值之间的间耦合相关度评估的基本思想是:对于属性 A_j 中的两个属性值 x 和 y ,假设 U 是 x 和 y 在其他属性 A_k 上共同出现的属性值的交集。如果 U 非空,则称 x 和 y 在属性 A_k 上相关。

基于上述思想,属性值 x 和 y 在属性 A_k 上的间耦合相关度定义为:

$$\delta_{A_j|A_k}^{IS}(x,y) = \sum_{u \in U} \min\{P_{A_k|A_j}(u|x), P_{A_k|A_j}(u|y)\} \quad (6)$$

其中: U 是 x 和 y 在属性 A_k 上的交集, u 是 U 的一个元素(即, u 与 x 和 y 在相同元组中共同出现过), $P_{A_k|A_j}(u|x)$ (以及 $P_{A_k|A_j}(u|y)$)是 u 关于 x (和 y)的信息条件概率(Information Conditional Probability, ICP),计算方法如下:

$$P_{A_k|A_j}(u|x) = |T_{A_k}(u) \cap T_{A_j}(x)| / |T_{A_j}(x)| \quad (7)$$

其中: $T_{A_j}(x)$ 表示 D 中包含 $A_j = x$ 的元组集合, $T_{A_k}(u)$ 表示 D 中包含 $A_k = u$ 的元组集合。也就是说, u 对于 x 的ICP指的是当 x 在关系表中出现时, u 与 x 共同出现的概率。可以看出,对于同一个属性的两个值,如果它们在其他属性上的ICP越高,那么这两个值的间耦合相关度就越高。

式(8)给出了属性值 $\langle x, y \rangle$ 在所有其他属性上的间耦合相关度计算方法:

$$IeS_{A_j}(x,y) = \sum_{k=1, k \neq j}^m w_k \delta_{A_j|A_k}^{IS}(x,y) \quad (8)$$

其中, $\delta_{A_j|A_k}^{IS}(x,y)$ 已在式(6)中定义, $\sum_{k=1, k \neq j}^m w_k = 1$ 。注意,如果 $A_k (k = 1, 2, \dots, m, k \neq j)$ 是一个数值属性,需首先将其值域划分为若干数值区间,将每个区间都作为一个文本值看待,然后再用式(8)进行计算。

接下来,把属性值 x 和 y 之间的内耦合和间耦合相关度进行结合就可得到它们之间的语义相关度,即:

$$S_Sim_{A_j}(x,y) = \beta * IaS_{A_j}(x,y) + (1 - \beta) * IeS_{A_j}(x,y) \quad (9)$$

其中, $\beta (\beta \in [0,1])$ 是一个调节参数,用来调整内耦合和间耦合相关度在最终的语义相关度中的作用。

最后,空间对象 o_i 和 o_j 在描述信息上的语义相关度可定义为它们在所有属性上的内耦合和间耦合相关度的结合:

$$S_Sim(o_i, o_j) = \sum_{k=1}^m w_k * S_Sim_{A_k}(o_i, x, o_j, y) \quad (10)$$

其中, w_k 代表 A_k 的属性权重, m 是属性个数。

3) 空间对象的耦合相关度。

给定空间对象 o_i 和 o_j 在位置上的相近度和在描述信息上的语义相关度,它们之间的耦合相关度可以用式(2)来计算。接下来,本文将论述如何利用空间对象之间的耦合相关度对它们进行聚类。

3.2 空间对象聚类

本文提出的空间对象聚类的基本思想是:首先利用概率密度估计方法评估空间对象的典型程度,然后提取前 k 个最具有代表性的空间对象,最后把其他对象按其对代表性对象的耦合相关度分配到不同聚类中。

1) 概率密度估计方法。

高斯核函数是概率密度估算中的常用方法,当仅知道空间对象的耦合关系距离矩阵时,可以使用该方法找出空间对象集合中的代表性对象^[19]。对于一个空间对象集合 $D = (o_1,$

$o_2, \dots, o_n)$,对象 o 的概率密度 $f(o)$ 可以表示为:

$$f(o) = \frac{1}{m} \sum_{i=1}^m G_h(o, o_i) = \frac{1}{m \sqrt{2\pi}} \sum_{i=1}^m e^{-\frac{d(o, o_i)^2}{2h^2}} \quad (11)$$

其中, $d(o, o_i)^2$ 是指 o 与 o_i 之间的耦合关系距离(由于对象之间的耦合相关度是归一化的,因此用1减去耦合相关度就可以得到对象之间的耦合关系距离), $G_h(o, o_i) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^m e^{-\frac{d(o, o_i)^2}{2h^2}}$ 是高斯核函数。表达式的含义是,如果分布在某个空间对象周围的对象与其距离越近,则该对象的概率密度越大,越具有代表性,因此典型程度越高。

根据空间对象的典型程度,可以选出前 k 个典型程度最高的对象作为代表,其他对象按其对代表对象的耦合相关度进行聚类,这样可将空间对象划分成 k 个类别。

2) 空间对象的聚类算法。

给定一个空间数据库 D ,选取前 k 个代表性对象的准确方法是利用式(11)逐个计算每个空间对象的典型程度。然而,在数据规模很大的情况下,逐个计算空间对象的典型程度具有很高复杂度,因此需要考虑一种近似算法。本文提出一种基于淘汰思想的聚类方法,该方法的优点是简便且具有较高准确性,其聚类过程为:

a) 将空间数据库 D 随机划分成若干个小组,每个小组都包含 u 个对象,即将 D 划分成了 $\lceil n/u \rceil$ 个小组,接着在每个小组内计算所有对象的典型程度,选取每个小组中典型程度最高的对象构成一个新的集合,然后从 D 中去除其他对象。

b) 对于新得到的集合,重复上述过程,直到 D 中只剩下一个对象,将该对象放入 $top-k$ 候选对象集合中(上述过程记为一次选取过程)。

c) 为了保证选取对象的准确性,需要将上述选取过程重复执行 v 次(记为一轮),这样 $top-k$ 候选对象集合中最多会包含 v 个对象,接着在最初的空间数据库 D 上计算这 v 个对象的典型程度,最后将具有最高典型程度的对象作为当前轮次的选取结果,并从 D 中将该对象去除。

上述a)~c)过程重复 k 轮,这样就能得到 k 个近似于准确的对象。

d) 将 D 中剩余对象依次与这 k 个代表性对象分别比较耦合关系距离,各自划分到与代表性空间对象最相近的类别中。

算法2 top- k 代表性对象近似选取和聚类算法。

输入:空间数据库 D ,验证次数 v ,正整数 k ,小组大小 u 。

输出: $top-k$ 个代表性对象及其对应的聚类。

for $i = 1$ to k do

repeat

划分 D 成为若干小组 g ,每个小组有 u 个对象

for each 小组 g do

计算 g 中每个对象在 g 中的典型程度

从 g 中选出典型程度最高的对象,将 g 中其他对象从 D 中

移除

end for

until D 中仅有一个对象

把得到的典型程度最高的对象放入候选集合中

end for

在 D 上计算候选集合中每个对象的典型程度,输出一个典型程度最高的对象作为第 i 次选出的代表性空间对象

end for



将其他对象按其与代表性对象的距离进行聚类

return k 个代表对象及其聚类 $\langle \bar{O}_1, C_1 \rangle, \langle \bar{O}_2, C_2 \rangle \dots, \langle \bar{O}_k, C_k \rangle$

算法2的复杂度分析 计算每个小组中所有对象典型程度的时间复杂度为 $O(u^2)$, 每次选取过程要进行 $l = \log_u n$ 次小组划分。假设 $n = u^l$, 在每次选取过程中, 第一次划分可得到 n/u 个小组, 第二次划分可得到 $(n/u)/u = n/u^2$ 个小组, 以此类推, 这样每次选取过程总共划分的小组数将有

$$\sum_{1 \leq i \leq \log_u n} \frac{n}{u^i} = \frac{n}{u-1} \left(1 - \frac{n}{u}\right) = O\left(\frac{n}{u}\right) \text{ 个}, \text{ 所以每次选取过程}$$

找到最典型对象的复杂度是 $O(u^2 * n/u) = O(un)$, 又因为每次淘汰有 v 次验证, 并且整个淘汰过程循环 k 次, 因此该算法的时间复杂度是 $O(kvun)$, 其中 k, v 和 u 都是很小的正整数, 因此, 算法2在实际应用中是可行的。

3.3 用户对聚类感兴趣的概率估计

本文借助查询历史评估用户对空间对象聚类感兴趣的概率, 其直觉是如果某个聚类中的空间对象被用户查询的次数越多, 说明用户对该聚类感兴趣的概率就越大。

给定一个空间对象聚类 C_j , 先统计查询历史 H 中能够返回 C_j 中任意一个对象的查询, 这些查询构成一个集合 S_j , 即 $S_j = \{Q \in C_j \mid \exists Q_i \in H, \text{使得 } o_i (o_i \in C_j) \text{ 能够由 } Q_i \text{ 返回}\}$ 。这样, 对于每一个聚类 C_j , 可通过计算 S_j 中包含的查询个数与查询总数之比得到用户对该聚类感兴趣的概率, 即 $P_j = S_j / H$ 中包含的查询个数 / H 中的查询总数。

4 查询结果分类树构建

本章首先提出空间对象查询结果分类树的构建算法;然后描述对于文本信息中的分类型属性和数值型属性的划分标准;最后给出分类树构建算法描述。

4.1 属性划分

本文提出了基于C4.5决策树算法的查询结果分类树构建方法。由于C4.5算法需要对属性进行划分, 空间对象的描述信息主要包含文本属性和数值属性, 因此下面分别描述文本和数值属性的划分标准。

1) 文本属性的划分。

对于每一个不同的文本值, 在查询结果分类树上产生一个新的子树, 并且在该子树上计算信息增益。

2) 数值属性的划分。

本文借助用户的查询历史划分数值属性值域。对于数值属性 A_i , 用 v_{\min} 和 v_{\max} 分别表示 $Dom(A_i)$ 的最小值和最大值。首先考虑简单划分的情况, 假设仅将 $Dom(A_i)$ 划分成两个互不相交的数值区间, 即 $[v_{\min}, v]$ 和 $(v, v_{\max}]$ 。如果在查询历史中, 大多数查询范围是以点 v 开始或结束, 则点 v 是一个最佳分割点。因为通过查询历史可知, 大多数用户是对以点 v 划分的两个数值区间其中之一感兴趣。

对于查询历史中指定在属性 A_i 上的范围查询, 令 $N_{v-\text{start}}$ 和 $N_{v-\text{end}}$ 分别代表这些查询中以点 v 开始或结束的查询个数, 然后用 $N_{v-\text{start}}$ 和 $N_{v-\text{end}}$ 之和代表点 v 的“分割能力”。很明显, 如果查询历史中以点 v 开始或结束的范围查询数量越多, 说明认为点 v 是一个好的分割点的用户越多, 那么点 v 的分割能力就越强。假设要将 $Dom(A_i)$ 划分为 k 个数值区间, 采用上述方法, 那么需要选取具有最高分割能力的 $(k-1)$ 个点。需要指出的是, 数值属性的候选分割点是通过在该属性值域区间内

以固定数值间隔获得的。

4.2 分类树构建算法

基于上述思想, 算法3给出了基于改进决策树算法的查询结果分类树构建过程。

算法3 查询结果分类树构建算法。

函数: $\text{BuildTree}(A, R, C, \lambda)$

输入: 空间对象描述信息的属性集 A , 结果对象集合 R, C 是 R 中每个对象在聚类中对应的标签, 终止阈值 λ 。

输出: 查询结果分类树 T 。

创建一个根节点 r

if R 中所有对象有同样的类标签, 终止算法

for 每个属性 $A_j \in A$

if A_j 是一个文本属性 then

for A_j 的每个属性值 v do

在当前根节点下创建一个分支, 将满足条件 “ $A_j = v$ ” 的那些对象加入到该分支中

计算 $g(A_j, T_j)$, T_j 是由划分属性 A_j 而产生的子树

else

for A_j 的每个范围 $[low_i, up_i]$ do

在当前根节点下创建一个分支, 将满足条件 “ $low_i \leq A_j < up_i$ ” 的那些对象加入到该分支中

计算 $g(A_j, T_j)$, T_j 是由划分属性 A_j 的值域产生的子树

end if

end for

选择具有最大 $g(A_j, T_j)$ 的属性 A_j , 从 A 中移除 A_j

if $g(A_j, T_j) > \lambda$ then

用子树 T_j 替代 r , 对于 T_j (用 R_k 表示 $leaf_k$ 中的对象集合) 中每一个叶节点 $leaf_k$, 递归执行 $\text{BuildTree}(A, R_k, C, \lambda)$

end if

利用算法3, 可在查询结果集上动态地生成一个查询结果分类树, 用户可通过检查标签的方式自左至右或自上而下访问该树, 并最终定位到其感兴趣的叶子节点。

算法3的复杂度分析 假设结果集中有 n 个空间对象, m 是分类属性个数, k 为类别数; 在计算某个属性的信息增益时, 可以采用一些优化算法。本文先按该属性的属性值对结果中的所有对象进行排序, 然后对该属性上的所有不同属性值进行一次性的信息增益计算, 这样该属性上的信息增益计算时间复杂度为 $O(k)$ 。对于一个树节点, 由于每次计算信息增益的复杂度为 $O(k)$, 并且最多有 m 个候选分类属性以及 n 种可能的划分(最坏情况就是该属性下有 n 个不同的属性值), 因此对一个节点的所有可能划分进行信息增益计算的复杂度为 $O(mnk)$ 。根据文献[11]对决策树的分析, 生成的查询结果分类树的深度为 $\log n$, 则算法3总的时间复杂度为 $O(mnk \log n)$ 。

5 实验

5.1 实验设置

所有实验在Windows 2007操作系统, Intel P4 3.2 GHz CPU和8 GB内存的PC机上运行, 使用下列真实数据来评估算法效果和性能。

数据集 测试数据是从Yahoo房地产网站提取的包含100 000个房屋信息的房地产数据集。每个房屋都包含位置信息和描述信息, 位置信息用经度和纬度表示, 描述信息由 {Price, City, SqFt, Bedrooms, Livingarea, Schooldistrict, View, Neighborhood, Pool, Garage, Boatdock, Buildyear} 属性来描述,



其中 Price, SqFt, Bedrooms 和 Buildyear 是数值属性, 剩余是文本属性。

查询历史 邀请 50 名本科生作为测试者, 他们作为不同类型的用户向房地产数据集提交空间查询, 利用这种方式收集了 500 条查询作为查询历史, 用于数值属性值域的划分和用户对空间对象聚类感兴趣概率的估计。

所有算法采用 Java 编写, 使用 R-tree^[21]建立空间对象索引, 空间对象数据表增加一列用于存放每条房屋记录的类标签, 查询结果分类树构建算法(算法 3)的终止阈值 λ 设为 0.005。

对比方法 不同聚类方法得到的聚类结果不同, 进而影响到查询结果分类树的搜索代价, 本文选取了两种聚类方法: 最远距离优先的聚类方法和基于淘汰的聚类方法(本文方法), 然后对基于不同聚类方法构建的查询结果分类树的实际搜索代价进行对比。

最远距离优先聚类方法的基本思想是: 首先从空间对象集合中随机抽出 1 个对象, 然后从剩余对象中选取与该对象耦合关系距离最大的对象, 并将其从集合中移除, 下一步再从剩余对象中选出与上一个选出对象耦合关系距离最大的对象, 重复上述步骤, 直到 k 个对象被选出为止。最后, 把剩余对象按其与这 k 个对象的耦合关系距离分别划归到相应类别中。

5.2 属性值相关度评估方法的合理性测试

该实验目标是评估空间对象文本值语义相关度评估方法的合理性。表 1 分别给出了在不同大小的数据集下与给定文本值最相关的前三个文本值。注意, 在该实验中式(9)的参数 β 设为 0.5, 也就是说在评估文本值的语义相关度时内耦合度和间耦合度起到同等重要的作用。

表 1 不同大小数据集下对于给定属性值的 top-3 个相关文本值
Tab. 1 Top-3 similar values to given text value under different size of datasets

给定的文本值	top-3	房屋数		
		相关文本值	10 000	20 000
(Attribute: View)	Territorial	0.7558	0.7571	0.7671
	Street Greenbelt	0.7475	0.7580	0.7631
	City	0.7193	0.7256	0.7421
(Attribute: Schooldistrict)	Kent Highline	0.5855	0.5982	0.6077
	Seattle	0.5848	0.5913	0.6043
	Shoreline	0.5736	0.5892	0.5988
(Attribute: Livingarea)	Burien	0.6982	0.6962	0.7127
	West Seattle	0.6641	0.6629	0.6885
	Des Moines	0.6504	0.6611	0.6749

从表 1 中可以看出, 与给定值最为相关的前三个文本值之间的语义相关度是合理的。还可以看出, 从 10 000, 20 000 个房屋数据集中获取的相关度值低于从 50 000 个房屋数据集中获取的相关度值, 但相关度值的变化程度不大, 并且相关度值大小的相对顺序没有发生改变。由此可见, 本文提出的文本值之间的语义相关度评估方法是合理稳定的。

5.3 搜索代价实验

该实验目的是测试本文提出的查询结果分类方法的实际搜索代价, 同时也对基于不同聚类方法构建的查询结果分类树的实际搜索代价进行对比。实际搜索代价定义如下:

$$rCost(T) = \sum_{\forall v \in Leaf(T) \text{ visited by a subject}} (K_1 N(v) +$$

$$K_2 \sum_{v_i \in Anc(v)} |Sib(v_i)|) \quad (12)$$

与定义 2 中的预计搜索代价不同, 实际搜索代价是用户通过使用查询结果分类树找到感兴趣对象而真正访问过的中间节点数和叶节点中的对象数。假设访问中间节点标签和访问叶节点中对象的代价是相等的, 即 $K_1 = K_2 = 1$, 实际搜索代价越低, 表明查询结果分类方法的效果就越好。

为了评估实际搜索代价, 首先邀请 10 名测试者, 每一名测试者提出 1 个测试查询条件(比如, 包含区域位置和价格区间), 然后在对应的结果集中标出他们认为最满足其需求和偏好的 5 个对象。最后, 利用查询结果分类树构建算法(算法 3)根据不同聚类标准在查询结果集上生成分类树, 用户通过使用分类树找出他们标注的对象。在这一过程中, 记录用户通过使用分类树上找出所有标注对象的实际搜索代价。表 2 分别给出了不同空间查询条件下基于不同聚类方法生成的查询结果分类树的实际搜索代价。

表 2 不同查询条件的查询结果分类树实际搜索代价对比

Tab. 2 Real search cost comparison of query results categorization tree under different query condition

条件编号	查询条件	实际搜索代价			条件编号	查询条件	实际搜索代价		
		查询总数	标注对象 ID	最远距离			查询总数	标注对象 ID	最远距离
1	区域: Seattle; 价格: 300 001 – 500 000	652	997	11	8	1	1961	8	6
	North	3620	9	7		3620	9	7	
	4971	13	10		4971	13	10		
	4726	6	5		4726	6	5		
	合计	47	36		1576	16	11		
2	区域: Seattle; 价格: 1200 001 – 1995 000	549	3478	15	9	2	3940	12	7
	4502	13	8		3478	15	9		
	1568	12	7		4502	13	8		
	合计	68	42		1568	12	7		
	4449	16	7		4449	16	7		
3	区域: Skyway Area	1256	1262	16	7	3	3678	12	9
	2377	13	7		2377	13	7		
	2338	11	7		2338	11	7		
	合计	68	37		1262	16	7		
	546	18	4		3678	12	9		
4	区域: Burien Park	831	396	8	7	4	101	15	7
	3270	12	8		3270	12	8		
	4383	17	10		4383	17	10		
	合计	70	36		396	8	7		
	591	16	7		591	16	7		
5	区域: Bainbridge Island	1043	787	8	7	5	94	17	8
	105	10	8		105	10	8		
	1711	11	9		1711	11	9		
	合计	62	39		787	8	7		
	462	14	11		1050	33	23		
10	区域: Bainbridge Island	500 001 – 800 000	690	2859	30	22	3193	17	12
	300 000	3146	21	18	3146	22	20		
	3742	16	15		3742	16	15		
	1200 000	3742	16	15	3742	16	15		
	合计	75	60		1974	32	23		
9	区域: Forest; 价格: 800 001 – 1200 000	149 991 – 300 000	690	2859	30	22	3904	30	20
	300 000	3146	21	18	3146	22	20		
	3742	16	15		3742	16	15		
	1200 000	3742	16	15	3742	16	15		
	合计	147	108		1050	33	23		
8	区域: Lake Forest; 价格: 800 001 – 1200 000	1120	2973	13	10	8	2941	17	11
	2973	13	10		2973	13	10		
	3146	21	18		3146	21	18		
	3742	16	15		3742	16	15		
	1200 000	3742	16	15	3742	16	15		
7	区域: Des Moines; 价格: 500 001 – 800 000	757	3079	7	5	7	2941	17	11
	3079	7	5		3079	7	5		
	3528	10	7		3528	10	7		
	720	13	7		720	13	7		
	合计	37	25		720	13	7		
6	区域: Queen Anne	988	3368	21	7	6	532	8	6
	3368	21	7		3368	21	7		
	3329	19	7		3329	19	7		
	1085	16	13		1085	16	13		
	合计	104	44		1085	16	13		
1	区域: North Seattle	3620	9	7		1	1147	3	2
	9	25	7		9	25	7		
	3329	19	7		3329	19	7		
	1085	16	13		1085	16	13		
	合计	104	44		1085	16	13		



从表2可以看出,对于测试者提出的10条测试查询(指定了位置范围和/或价格区间),每个查询返回的查询结果总数都在500个以上,如果逐个检查结果的相关性是非常耗时的。通过构建查询结果分类树,用户可以根据分类树上的分支标签检查各分支的相关性,进而快速定位到感兴趣的信息。从表2可以看出,与基于最远距离优先的聚类方法相比,在基于淘汰思想的聚类算法上生成的查询结果分类树具有明显较低的搜索代价。由此可见,本文的空间聚类方法具有较高准确性。图2(a)和图2(b)分别给出了10条测试查询在不同聚类标准下的实际搜索代价和平均搜索代价对比。平均搜索代价是指找到每个相关对象的平均实际搜索代价。

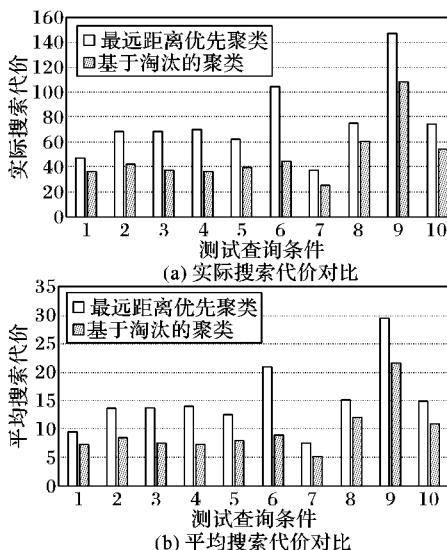


图2 10条测试查询下基于不同聚类方法生成的分类树的搜索代价对比

Fig. 2 Search cost comparison of category tree generated by 10 test queries based on different clustering methods

5.4 分类树的生成时间测试

对于一个给定的空间查询,通常会返回多个查询结果,而查询结果分类树又是在在线阶段动态生成的,因此需要能够快速生成。该实验目的是测试在不同规模的查询结果下分类树的生成时间。图3给出了房地产数据集上不同查询结果个数下的分类树生成时间。

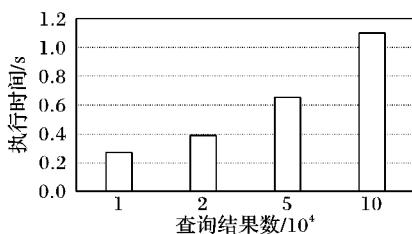


图3 不同查询结果数下的分类树生成时间

Fig. 3 Time of generating category tree on different numbers of query results

从图3可以看出,在查询结果数为100 000条的情况下,分类树的构建时间也不超过1.1 s,因此能够适用于大规模数据情况下的查询分类。

6 结语

本文提出了一种用于处理空间数据库查询信息过载问题的分类方法,该方法在离线阶段评估空间对象之间的耦合关系,使用基于概率密度估计方法对空间对象进行聚类。在在线阶段,对于一个空间查询条件,采用改进决策树算法在查询

结果上构建一棵分类树,用户通过检查分类树中间节点标签的方式逐步定位到其感兴趣的叶节点信息。

本文方法与现有方法有两方面不同:一是空间聚类方法同时考虑了空间对象之间的位置相近性和语义相关度;二是查询结果分类树构建算法同时考虑了访问中间节点的代价和访问叶节点中对象的代价。如何将将查询结果排序方法融入到分类方法之中是进一步的研究工作。

参考文献 (References)

- [1] 刘喜平,万常选,刘德喜,等.空间关键词搜索研究综述[J].软件学报,2016,27(2):329–347.(LIU X P, WAN C X, LIU D X, et al. Survey on spatial keyword search [J]. Journal of Software, 2016, 27(2): 329 – 347.)
- [2] 张金增,孟小峰.移动Web搜索研究[J].软件学报,2012,23(1):46–64.(ZHANG J Z, MENG X F. Research on mobile Web search [J]. Journal of Software, 2012, 23(1): 46 – 64.)
- [3] ZHENG K, SU H, ZHENG B L. Interactive top- k spatial keyword queries [C]// Proceedings of the 2015 IEEE 31st International Conference on Data Engineering. Piscataway, NJ: IEEE, 2015: 423 – 434.
- [4] CARY A, WOLFSON O, RISHE N. Efficient and scalable method for processing top- k spatial boolean queries [C]// Proceedings of the 2010 International Conference on Scientific and Statistical Database Management. Berlin: Springer, 2010: 87 – 95.
- [5] LU Y, LU J H, SHAHABI C. Efficient algorithms and cost models for reverse spatial-keyword k -nearest neighbor search [J]. ACM Transactions on Database Systems, 2014, 39(2): 573 – 598.
- [6] LU J H, LU Y, CONG G. Reverse spatial and textual k nearest neighbor search [C]// Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2011: 349 – 360.
- [7] HU H Q, LI G L. Top- k spatio-textual similarity join [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(2): 551 – 565.
- [8] 周傲英,杨彬,金澈清,等.基于位置的服务:架构与进展[J].计算机学报,2011,34(7):1155–1171.(ZHOU A Y, YANG B, JIN C Q, et al. Location-based services: architecture and progress [J]. Chinese Journal of Computers, 2011, 34(7): 1155 – 1171.)
- [9] CHEN L, CONG G, JENSEN C S, et al. Spatial keyword query processing: an experimental evaluation [J]. Proceedings of the VLDB Endowment, 2013, 6(3): 217 – 228.
- [10] CHAKRABARTI K, CHAUDHURI S, HWANG S. Automatic categorization of query results [C]// Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2004: 755 – 766
- [11] CHEN Z Y, LI T. Addressing diverse user preferences in SQL-query-result navigation [C]// Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2007: 641 – 652.
- [12] AL-MUBAID H, UMAIR S A. A new text categorization technique using distributional clustering and learning logic [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(9): 1156 – 1165.
- [13] ROUSSEAU F, KIAGIAS E, VAZIRGIANNIS M. Text categorization as a graph classification problem [C]// Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics. Beijing: [s. n.], 2015: 1702 – 1712.

(下转第187页)



- for Advanced Applications. Berlin: Springer, 2012: 351–366.
- [6] CAI Z F, YANG H X, SHUANH W, et al. A clustering-based privacy-preserving method for uncertain trajectory data [C]// Proceedings of the 2014 International Conference on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2014: 1–8.
- [7] ABUL O, BONCHI F, NANNI M. Anonymization of moving objects databases by clustering and perturbation [J]. Information Systems, 2010, 35(8): 884–910.
- [8] 霍峰, 孟小峰, 黄毅. PrivateCheckIn: 一种移动社交网络中的轨迹隐私保护方法[J]. 计算机学报, 2013, 36(4): 716–726. (HUO Z, MENG X F, HUANG Y. PrivateCheckIn: trajectory privacy-preserving for check-in services in MSNS [J]. Chinese Journal of Computers, 2013, 36(4): 716–726.)
- [9] HUA J, GAO Y, ZHONG S. Differentially private publication of general time-serial trajectory data [C]// Proceedings of the 2015 IEEE Conference on Computer Communications. Piscataway, NJ: IEEE, 2015: 549–557.
- [10] PAN X, XU J, MENG X. Protecting location privacy against location-dependent attacks in mobile services [J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(8): 1506–1519.
- [11] CHOI T Y. A linear-time heuristic algorithm for k -way network partitioning [J]. Journal of the Korea Safety Management and Science, 2004, 7(8): 1183–1194.
- [12] YAROVY R, BONCHI F, LAKSHMANAN L, et al. Anonymizing moving objects: how to hide a MOB in a crowd? [C] // Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. New York: ACM, 2009: 72–83.
- [13] CHEN R, LI H, QIN K A, et al. Private spatial data aggregation in local setting [C]// Proceedings of the 32nd IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE, 2016: 289–300.
- [14] SU S, TANG P, CHENG X, et al. Differentially private multi-party high-dimensional data publishing [C]// Proceedings of the 2016 International Conference on Data Engineering. Piscataway, NJ: IEEE, 2016: 205–216.
- [15] QIN Z, YANG Y, YU T, et al. Heavy hitter estimation over set-valued data with local differential privacy [C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2016: 192–203.
- [16] 孟小峰, 张啸剑. 大数据隐私管理[J]. 计算机研究与发展, 2016, 52(2): 265–281. (MENG X F, ZHANG X J. Big data privacy management [J]. Journal of Computer Research and Development, 2016, 52(2): 265–281.)

This work is partially supported by the National Natural Science Foundation of China (61502279), the Natural Science Foundation of Hebei Province (F2015207009), the Scientific Research Projects in Colleges and Universities in Hebei Province (BJ2016019, QN2016179), the Soft Science Project of Ningbo City (2016A10066).

HUO Zheng, born in 1982, Ph. D., lecturer. Her research interests include privacy-preserving, mobile object database.

CUI Honglei, born in 1976, Ph. D., lecturer. Her research interests include economic data applications under big data environment.

HE Ping, born in 1982, Ph. D., lecturer. Her research interests include wireless sensor network, graph optimization algorithm.

(上接第 158 页)

- [14] ZHENG W B, TANG H, QIAN Y T. Collaborative work with linear classifier and extreme learning machine for fast text categorization [J]. Journal of World Wide Web, 2013, 18(2): 1–18.
- [15] ZENG H J, HE Q C, CHEN Z. Learning to cluster Web search results [C]// SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference On Research and Development in Information Retrieval. New York: ACM, 2004: 210–217.
- [16] GRBOVIC M, DJURIC N, RADOSAVLJEVIC V. QueryCategorizr: a large-scale semi-supervised system for categorization of Web search queries [C]// WWW 2015: Proceedings of the 24th International Conference on World Wide Web Companion. New York: ACM, 2015: 199–202.
- [17] TWEEDIE L, SPENCE R, WILLIAMS D, et al. The attribute explorer [C]// CHI '94: Proceedings of the 1994 Conference Companion on Human Factors in Computing Systems. New York: ACM, 1994: 435–436.
- [18] WANG C, CAO L B, WANG M C. Coupled nominal similarity in unsupervised learning [C]// CIKM '11: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. New York: ACM, 2011: 973–978.
- [19] 王秀红, 鞠时光. 用于文本相似度计算的新核函数[J]. 通信学报, 2012, 33(12): 43–48. (WANG X H, JU S G. Novel kernel function for computing the similarity of text [J]. Journal of Communications, 2012, 33(12): 43–48.)
- [20] BORIAH S, CHANDOLA V, KUMAR V. Similarity measures for categorical data: a comparative evaluation [C]// Proceedings of the 2008 SIAM International Conference on Data Mining. Atlanta, Georgia: [s. n.], 2008: 243–254.
- [21] GUTTMAN A. R-trees: a dynamic index structure for spatial searching [C]// Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. New York: ACM, 1984: 47–57.
- This work is partially supported by the National Natural Science Foundation of China (61401185), the General Project of Liaoning Province Education Department (LJYL018), the Natural Science Foundation of Liaoning Province (201705 40418).
- BI Chongchun**, born in 1992, M. S. candidate. His research interests include spatial data analysis and query.
- MENG Xiangfu**, born in 1981, Ph. D., associate professor. His research interests include Web database query, spatial data analysis.
- ZHANG Xiaoyan**, born in 1983, Ph. D. candidate, engineer. Her research interests include spatial data query, city calculation.
- TANG Yanhuan**, born in 1992, M. S. candidate. His research interests include spatial data mining, recommender system.
- TANG Xiaoliang**, born in 1980, Ph. D., lecturer. His research interest include machine learning.
- LIANG Haibo**, born in 1995. His research interest include data mining, database query.