

# P2P 环境下面向不确定数据的 Top- $k$ 查询

孙永佼 袁 野 王国仁

(医学影像计算教育部重点实验室(东北大学) 沈阳 110819)

(东北大学信息科学与工程学院 沈阳 110819)

**摘 要** 分布式环境中的 top- $k$  查询已经有了广泛的研究. 由于仪器不精确和网络延时等原因, 大多数分布式数据都存在不确定性. 文中基于水平分布在 P2P 网络中的不确定数据提出了一个有效的 top- $k$  查询处理方法. 首先利用 Quad-tree 构建一个分布式的 uncertain 数据的索引, 并基于索引提出了一个空间剪枝算法. 然后, 根据局部 top- $k$  概率与全局 top- $k$  概率之间的关系提出 uncertain 数据成为 top- $k$  概率的上界, 根据 top- $k$  概率与 skyline 概率之间的关系提出 uncertain 数据成为 top- $k$  概率的下界, 通过两种概率剪枝算法来减少 top- $k$  查询在网络中的传输和计算代价, 并且进一步减少候选集大小. 最后文中采用采样的方法来计算候选集的 top- $k$  概率以确定最终的 top- $k$  查询结果. 大量的实验验证了算法的有效性.

**关键词** top- $k$  查询; skyline 概率; P2P; Quad-tree; uncertain 数据

**中图法分类号** TP311 **DOI 号**: 10.3724/SP.J.1016.2011.02155

## Answering Probabilistic Top- $k$ Queries over P2P Networks

SUN Yong-Jiao YUAN Ye WANG Guo-Ren

(Key Laboratory of Medical Image Computing of Ministry of Education (Northeastern University), Shenyang 110819)

(College of Information Science and Engineering, Northeastern University, Shenyang 110819)

**Abstract** Top- $k$  queries in distributed databases have been studied widely in recent years. There exists an inherent uncertainty on the data objects due to imprecise measurements and network delays. In this paper, based on horizontally distributed data among peers, we propose an efficient approach of processing uncertain top- $k$  queries in P2P networks. Firstly, we construct a distributed index using Quad-tree, and based on the index, propose a spatial pruning algorithm. Secondly, we propose the upper bound of top- $k$  probabilistic according to the relationship between local top- $k$  probabilities and global top- $k$  probabilities. We also propose the lower bound of top- $k$  probabilities according to the relationship between skyline probabilities and top- $k$  probabilities. Using the two probabilistic pruning algorithms, we can further reduce computation costs and network overhead of top- $k$  queries, and further reduce the number of candidate sets. Finally, we develop a sampling algorithm to estimate top- $k$  probabilities of candidates. Extensive experiments are conducted to verify the effectiveness and efficiency of the proposed methods.

**Keywords** top- $k$  query; skyline probability; P2P; Quad-tree; uncertain data

## 1 引言

P2P 技术致力于高效合理地组织和利用 Internet 边缘大量分布的计算、存储、通信以及信息等资源, 已经成为有效实现分布式网络环境下信息资源共享和数据交换, 实现资源整合利用的重要手段. P2P 网络已成为当前网络中资源共享的重要应用之一<sup>[1-4]</sup>. 在数据共享和通信的过程中, P2P 网络中数据的传输代价通常是非常重要的. 但是, 在许多应用中, 查询往往不需要传输每个网络节点的所有数据, 而是传输和用户需要最匹配的一小部分数据就可以了, 比如 top- $k$  查询通过用户指定的打分函数返回和用户需要最匹配的  $k$  个数据对象. 在 P2P 环境中的 top- $k$  查询已经有了广泛的研究. 但是随着计算机网络和计算机应用技术的迅猛发展, 特别是 Web 服务、传感器网络、RFID 以及基于位置的服务等技术的广泛应用, 使得不确定数据无处不在, 并大量存在于这些先进的应用系统中. 因此在 P2P 环境中不确定数据的 top- $k$  查询问题仍旧很具挑战性.

不确定数据的表现形式多种多样, 根据应用特点和数据形式差异, 研究者已经提出了多种针对不确定数据的数据模型. 这些不确定数据模型的核心都来自于可能世界模型<sup>[5-6]</sup>. 可能世界模型从一个或多个不确定的数据源演化出诸多确定的数据库实例, 称为可能世界实例, 并且所有实例的概率之和为 1. 虽然可以首先分别为各个实例计算查询结果, 然后合并中间结果以生成最终的查询结果, 但是由于可能世界实例的数量远大于不确定数据的规模, 这种方法是不可行的. 在 P2P 环境中, 获得所有的可能世界实例需要昂贵的网络代价, 即使获得了所有的实例, 计算也是不现实的. 因此在 P2P 环境下对不确定数据进行 top- $k$  查询处理更是一个具有挑战的问题.

为了解决这个问题, 本文对 P2P 环境中返回不确定数据的 top- $k$  查询提出了有效的处理方法. 本文假设数据是水平分布在超级节点构建的 P2P 网络中的, 并且用户指定的打分函数是单调的. 在本文提出的方法中, 第 1 步, 采用 Quad-tree 在 P2P 网络中对不确定数据建立全局索引, 然后基于不确定数据的得分范围提出了 top- $k$  结果在空间中的上界与下界, 提出了空间剪枝算法和基本的 top- $k$  计算方法. 第 2 步, 本文通过有力的证明得出本地 top- $k$  概率和全局 top- $k$  概率之间的关系, 提出了 top- $k$  概率

的上界. 更重要的是通过不确定数据之间的支配关系, 证明出 skyline 概率和 top- $k$  概率存在一定关系, 提出了 top- $k$  概率的紧凑下界. 通过这两步过滤, 可以大大减少候选集的大小, 进而减少了处理 top- $k$  查询时的网络传输代价. 最后通过分布式采样算法来计算候选集的 top- $k$  概率以确定最终的 top- $k$  查询结果.

本文贡献:

- (1) 本文对 P2P 环境下不确定数据的 top- $k$  查询做了形式化的定义.
- (2) 本文基于 Quad-tree 提出了一个对 P2P 网络中不确定数据的全局索引.
- (3) 本文提出了有效的空间剪枝方法和概率剪枝方法来减少 top- $k$  查询时网络的传输代价.
- (4) 本文采用蒙特卡洛采样方法在 P2P 环境中采样计算最终的 top- $k$  结果.

## 2 问题定义

在这节中, 首先介绍不确定数据的模型, 然后定义 P2P 环境中不确定数据的 top- $k$  查询.

不确定数据模型: 设  $D$  是一个不确定数据集, 并且  $D$  由  $N$  个多维元组  $t_i$  组成,  $t_i \in D (1 \leq i \leq N)$ . 每一个元组  $t_i$  都由一组不确定的实例  $a_i (a_{i1}, a_{i2}, \dots, a_{in})$  组成. 每一个实例都有一个存在概率  $Pr(a)$ , 并且  $Pr(a_1) + Pr(a_2) + \dots + Pr(a_n) = 1$ . 表 1 描述的是不确定数据模型.

表 1 不确定数据模型

元组	实例	概率
$t_1$	$a_{11}, a_{12}$	$Pr(a_{11}) = 0.2, Pr(a_{12}) = 0.8$
$t_2$	$a_{21}, a_{22}$	$Pr(a_{21}) = 0.3, Pr(a_{22}) = 0.7$
$t_3$	$a_{31}, a_{32}, a_{33}$	$Pr(a_{31}) = 0.1, Pr(a_{32}) = 0.5, Pr(a_{33}) = 0.4$
$t_4$	$a_{41}$	$Pr(a_{41}) = 1$

给定一个含有  $n$  个数据的数据集  $D, t_i \in D (1 \leq i \leq n)$ , 对于传统的 top- $k$  查询, 通常用户可以指定一个单调的打分函数  $s(t_i) = \sum_{j=1}^d \omega_j \times s_j(t_i) (1 \leq j \leq d)$ , 其中  $s_j(t_i)$  表示  $t_i$  的  $d$  维特征,  $\omega_j$  是数据在第  $j$  维上的权重.  $s(t_i)$  是用户针对数据的喜好程度的得分.  $s(t_i)$  最终将数据对象在各个维上的特征值聚合成一个值, 通过每个数据的得分情况, 返回给用户最满意的  $k$  个结果. 在本文中, 假设分值越小, 排位越靠前.

在不确定数据的背景下, top- $k$  查询的定义则大不相同. 不再是通过判断  $s(t_i)$  的大小来决定数据

对象是否是 top-k 的最终结果,而是通过判断一个元组  $t_i$  能够成为 top-k 的概率  $Pr_{\text{top}}(t_i)$ .

**定义 1.** 不确定数据的 top-k 查询. 假设有一个不确定数据库  $D(t_1, t_2, t_3, \dots, t_N)$ , 一个用户指定的打分函数  $s$  和一个整数  $k$ . 那么, 对不确定数据  $t_i$  排在第  $m$  位的概率  $Pr_m(t_i)$  定义如下:

$$Pr_m(t_i) = \sum_{s_1}^{s_n} \left( Pr\{s(t_i) = f\} \cdot \sum_{\forall (P_1, P_2, \dots, P_{m-1}) \in D \setminus \{t_i\}} \left( \prod_{l=1}^{m-1} Pr\{s(t_l) \geq f\} \cdot \prod_{\forall P_j \in D \setminus \{t_i, P_1, \dots, P_{m-1}\}} Pr\{s(P_j) \leq f\} \right) \right) \quad (1)$$

在这里,  $s_n$  是不确定数据  $t_i$  的每个实例的得分. 在  $k$  个不确定数据  $(t_1, t_2, \dots, t_m, \dots, t_k) \in D$  中, 通过计算它们的  $Pr_m(t_i)$ , 会得到每个数据排在第  $m$  位的概率, 而 top-k 查询返回每个位置上概率最大的数据. 根据定义 1, 在 P2P 环境中下面定义 2.

**定义 2.** P2P 环境中不确定数据的 top-k 查询. 假设 P2P 网络中的不确定数据库为  $D = \{D^1, D^2, \dots, D^n\}$ ,  $D^i$  为第  $i$  个网络节点上的不确定数据集. 对于  $D$  中的任意一个数据对象  $t \in D$  来说, 排在位置  $m$  上概率表示为  $Pr_m(t)$ ,  $Pr_m(t) \in [0, 1]$ . 最大的概率表示为  $MP_m = \max\{Pr_m(t_1), \dots, Pr_m(t_N)\}$ , 其中  $N$  为  $D$  中数据对象的个数. 在 P2P 网络中不确定数据的 top-k 查询返回的结果集合  $t^*$  定义为每个位置上概率最大的数据对象集合, 即

$$t^* = \{\arg MP_1, \arg MP_2, \dots, \arg MP_k\}, \\ MP_1 \geq MP_2 \geq \dots \geq MP_k \quad (2)$$

**定义 3.** 概率阈值的 top-k 查询. 给定一个概率阈值  $p$  ( $0 < p \leq 1$ ), PT-top-k 找到一组数据对象满足  $\{t | t \in T, P_{\text{top-k}}(t) \geq p\}$ .

### 3 Top-k 查询处理框架

给定一组 P2P 环境下的不确定数据库, 那么在 P2P 环境中的不确定数据的 top-k 查询可以由以下几个步骤完成:

(1) 空间剪枝. 基于 Quad-tree, 在网络中每一个节点上构建一个索引. 超级节点的索引结构与普通节点的索引结构相似, 但是超级节点的索引是通过搜集它所在的子网络中普通结点的一些概要信息而建立的. 在超级结点之间通过广播这些概要信息建立一个全局的索引. 基于全局索引, 每个超级节点都拥有 P2P 网络中不确定数据的概要信息, 因此可

以在 top-k 查询到达时在超级节点上对全局数据进行空间剪枝.

(2) Top-k 概率计算. 这个部分介绍了 top-k 概率计算的整个过程, 其中主要分为三个方面. 首先根据局部 top-k 概率与全局 top-k 概率之间的关系给出不确定数据成为 top-k 概率的上界, 然后根据 top-k 概率与 skyline 概率之间的关系给出不确定数据成为 top-k 概率的下界, 最后给出了分布式 Skyline 概率的计算方法.

(3) 采样计算. 在最终候选集中采用蒙特卡洛采样方法对分布式不确定数据进行采样计算来获得最终结果.

### 4 基本方法

不确定数据模型在空间中的表现形式如图 1, 每个不确定元组都是以离散的形式分布在一定范围的最小边界矩形中的.

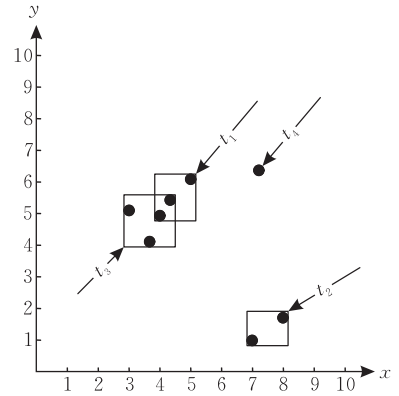


图 1 不确定数据模型在空间中的表示

显而易见, 每个不确定的元组都有一个得分的最大值  $s_U(t)$  和最小值  $s_L(t)$ . 因此, 有下面定理.

**定理 1.** 分值剪枝. 给定一个不确定数据库  $D$  和一个用户指定的打分函数  $s$ ,  $t_1, t_2, \dots, t_N$  是  $D$  中的不确定数据. 如果  $s_L(t_i) \geq s_U(t_j)$ , 则  $s(t_i) \geq s(t_j)$ , 其中  $1 \leq i \leq N, 1 \leq j \leq N$ . 那么,  $t_i$  排在  $t_j$  之前. 反之亦成立.

证明. 因为  $s_L(t_i) \geq s_U(t_j) \cup s_U(t_j) \geq s(t_j) \Rightarrow s_L(t_i) \geq s(t_j)$ , 所以  $s(t_i) \geq s_L(t_i) \cup s_L(t_i) \geq s(t_j) \Rightarrow s(t_i) \geq s(t_j)$ . 证毕.

**推论 1.** 空间剪枝. 给定一个不确定数据库  $D$ , 一个用户指定的打分函数  $s$  和一个整数  $k$ ,  $T$  是  $D$  中的任意一个集合,  $o$  是空间中的任意一点. 如果  $s(o) \geq s_U(T) \cap |T| \leq k$ , 那么,  $T$  一定是 top-k 中的

结果; 如果  $s(o) \geq s_U(T) \cap |T| \geq k \cap s(o) \leq s_L(\{D-T\})$ , 那么,  $\{D-T\}$  一定不是 top- $k$  中的结果.

通过以上结论, 可以排除掉一部分不属于结果集中的数据并且可以返回一部分结果集中的数据. 在下文中主要处理的是不能被确定是否是结果集中的数据. 因此首先利用 Quad-tree 在 P2P 网络上建立全局索引, 然后根据空间剪枝方法找出不能够被确定是否是结果集中的数据, 最后在分布式环境中利用式(1)计算出最终结果<sup>[7]</sup>.

## 5 Top- $k$ 概率计算

首先介绍一下 top- $k$  概率计算的整个过程:

(1) 构建全局索引.

(2) 在全局索引上利用基本方法中的空间剪枝策略进行过滤.

(3) 计算剩余不确定数据的 top- $k$  概率的上界和下界(定理 3 和定理 5).

(4) 执行概率剪枝(定理 2).

根据定义 1, 可以在本地计算出任意一个不确定数据排在第  $m$  位置的概率, 从而得到每个位置概率最大的数据作为本地数据的 top- $k$ . 但是定义 1 的计算代价比较大, 并且在分布式计算时还需要多次这样的计算, 所以这种方法效率不高. 如果能够找出一个概率区间  $[Pr_L(t), Pr_U(t)]$  约束  $Pr_m(t)$ , 并利用定义 3 的阈值进一步缩小候选集的范围, 那么就会提高查询效率.

**定理 2.** 概率剪枝. 给定一个不确定数据库  $D$ , 一个用户指定的打分函数  $s$ , 一个整数  $k$  和一个阈值  $p$ .  $t$  是  $D$  中的一个不确定元组,  $Pr_m(t)$  是在第  $m$  位置的最大概率并且  $Pr_{mL}(t) \leq Pr_m(t) \leq Pr_{mU}(t)$ . 如果  $Pr_{mU}(t) \leq p$ , 那么  $Pr_m(t)$  被剪枝掉; 如果  $Pr_{mL}(t) \geq p$ , 那么  $Pr_m(t)$  则是 top- $k$  中的结果.

证明. 因为  $Pr_{mU}(t) \leq p \cup Pr_m(t) \leq Pr_{mU}(t) \Rightarrow Pr_m(t) \leq p$ , 所以根据定义 3,  $Pr_m(t)$  被剪枝掉. 同理, 因为  $Pr_{mL}(t) \geq p \cup Pr_{mL}(t) \leq Pr_m(t) \Rightarrow Pr_m(t) \geq p$ , 所以根据定义 3,  $Pr_m(t)$  则是 top- $k$  中的结果.

证毕.

根据定理 2, 可以估计  $Pr_m(t)$  的上界与下界, 然后对候选集进行筛选, 得到最精炼的候选集, 减少最终的计算代价和网络传输代价.

### 5.1 概率上界的估计

假设  $D^i$  是 P2P 网络中某一节点的不确定数据

库  $D^i \in D$ ,  $t_1, t_2, \dots, t_k$  是排在前  $k$  个的不确定数据,  $Pr_{r_1}(t_1), Pr_{r_2}(t_2), \dots, Pr_{r_k}(t_k)$  分别是其排在第  $m$  位置的概率, 其中  $s(t_1) > s(t_2) > \dots > s(t_k)$ . 如果  $t_1, t_2, \dots, t_k \in D^i$ , 则称  $Pr_{r_1}(t_1), Pr_{r_2}(t_2), \dots, Pr_{r_k}(t_k)$  为本地 top- $k$  概率  $Pr_{L\text{top-}k}(t)$ ; 如果  $t_1, t_2, \dots, t_k \in D$ , 则称  $Pr_{r_1}(t_1), Pr_{r_2}(t_2), \dots, Pr_{r_k}(t_k)$  为全局 top- $k$  概率  $Pr_{G\text{top-}k}(t)$ .

**定理 3.** 概率上界. 给定一个不确定元组  $t$ ,  $t \in T^i$ , 有

$$Pr_{G\text{top-}k}(t) \leq Pr_{L\text{top-}k}(t).$$

证明. 假设  $t_{\text{new}}$  是新到  $T^i$  中的一个元组. 如果  $t_{\text{new}}$  的得分排在  $t$  的后面, 那么  $t_{\text{new}}$  对整个 top- $k$  结果没有影响. 如果  $t_{\text{new}}$  的得分排在  $t$  的前面, 那么  $t$  的 top- $k$  概率为

$$Pr'_{\text{top-}k}(t) = \sum_{i=1}^m Pr(a_i) Pr_{\text{top-}(k-1)}(t) + \sum_{i=m}^n Pr(a_i) Pr_{\text{top-}k}(t),$$

其中,  $a_i$  为元组  $t_{\text{new}}$  的实例. 因为,  $Pr_{\text{top-}(k-1)}(t) \leq Pr_{\text{top-}k}(t)$ , 所以  $Pr'_{\text{top-}k}(t) \leq Pr_{\text{top-}k}(t)$ . 证毕.

通过定理 2 和定理 3, 可以计算出任意一个不确定数据在本地的 top- $k$  概率  $Pr_{L\text{top-}k}(t)$ , 然后判断  $Pr_{L\text{top-}k}(t)$  与阈值  $p$  的大小. 如果  $Pr_{L\text{top-}k}(t) \leq p$ , 那么  $Pr_{G\text{top-}k}(t) \leq p$ . 因此可以在本地剪枝掉不属于结果集中的数据, 并且不需要在网络中传输数据, 大大减少了网络传输代价.

虽然概率上界能够有效地剪枝不属于结果集中的数据, 并且不需要网络传输代价, 但是利用概率上界剪枝是在用户发送查询之后才能进行的. 本地 top- $k$  概率计算时间仍然包括在用户查询的响应时间内的. 如果能够找到一种方法, 在查询到来时就能立即判断出该不确定数据是否属于 top- $k$  查询的结果集中, 那么会大幅度地减少查询响应时间.

### 5.2 概率下界的估计

**定理 4.** 支配剪枝. 给定一个不确定数据库  $D$ , 不确定数据  $(t_1, \dots, t_i, \dots, t_j, \dots, t_N) \in D, 1 \leq i \leq N$ , “ $>$ ”为支配关系. 假设用户指定的打分函数是单调的, 那么有

如果  $t_i > t_j$ , 那么,  $s(t_i) > s(t_j)$ .

证明. 假设  $t$  是  $d$  维数据,  $s(t)$  的大小是  $t$  在各维得分的和. 因为  $t_i > t_j$ , 所以  $s(t_i)$  在各维的得分的和小于  $s(t_j)$  在各维的得分的和, 因此  $s(t_i) > s(t_j)$ . 证毕.

给定  $D$  是一个不确定数据库,  $t$  是  $D$  中的不确

定数据,  $t_i \in D$ , 整数  $k$  代表 top- $k$  查询的结果个数, 整数  $k_1$  代表 skyline 上数据的个数. 那么, 有以下定理.

**定理 5.** 概率下界. 给定一个不确定元组  $t$ ,  $t \in T$ , 当  $k \geq k_1$  时有

$$Pr_{\text{sky}}(t) \leq Pr_{\text{top-}k}(t).$$

证明. 假设  $D$  是一个不确定数据库,  $t = \{t_1, t_2, t_3, \dots, t_N\}$  是  $D$  中的不确定数据,  $t_i \in D, 1 \leq i \leq N$ . 如果不确定数据  $t_i$  能够成为 top- $k$ ,  $t_i$  的得分  $s(t_i)$  大于剩余  $N-k$  个数据得分的事件的可能性用“ $\geq$ ”表示, 那么, 可以给出不确定数据  $t$  的 top- $k$  概率的表示形式:

$$Pr_{\text{top-}k}(t) = \sum_{i=1}^{\binom{N-k}{N-k}} Pr\{s(t) \geq s(t_1) \cap s(t) \geq s(t_2) \cap \dots \cap s(t) \geq s(t_{N-k})\} \quad (3)$$

类似的, 假设  $k_1$  为 skyline 上数据的个数. 如果  $t_i$  能够成为 skyline 中的点,  $t_i$  支配剩余  $N-k_1$  个数据的事件的可能性用“ $>$ ”表示, 那么, 可以给出不确定数据  $t$  的 skyline 概率的表示形式:

$$Pr_{\text{sky}}(t) = \sum_{i=1}^{\binom{N}{N-k_1}} Pr\{t > t_1 \cap t > t_2 \cap \dots \cap t > t_{N-k_1}\} \quad (4)$$

根据定理 4, 因为  $t_i > t_j \Rightarrow s(t_i) > s(t_j)$ , 所以事件  $s(t_i) > s(t_j)$  包含事件  $t_i > t_j$ . 因此, 根据式 (3) 和 (4), 可以得到, 当  $k \geq k_1$  时,  $Pr_{\text{sky}}(t) \leq Pr_{\text{top-}k}(t)$ .

证毕.

通过定理 5, 得到了 top- $k$  概率的下界,  $Pr_{\text{sky}}(t) \geq p$  时, 可以判断哪些数据是 top- $k$  中的数据. 但是, 需要在 P2P 网络中预先计算出  $Pr_{\text{sky}}(t)$ . 下面就介绍如何在 P2P 网络中计算 skyline 概率.

### 5.3 分布式 skyline 概率计算

**定义 4.** 不确定数据的分布式 skyline. 给定  $h$  个网络节点  $C = \{c_1, c_2, \dots, c_h\}$ , 每一个节点都拥有一个不确定数据库  $D^I (1 \leq I \leq h)$ .  $t = \{t_1, t_2, t_3, \dots, t_N\}$  是  $D$  中的不确定数据,  $t_i \in D (1 \leq i \leq N)$ ,  $a = \{a_1, a_2, \dots, a_n\}$  是元组  $t$  的实例. 那么元组  $t$  的全局 skyline 概率  $Pr_{\text{Gsky}}(t)$  为

$$Pr_{\text{Gsky}}(t) = Pr(t) \times \prod_{i' \in D^1, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \times \prod_{i' \in D^2, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \times \dots \times \prod_{i' \in D^h, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \quad (5)$$

如果将所有分布式的数据库统一到一个数据库  $D$  中, 那么式 (5) 可以被改写成

$$Pr_{\text{Gsky}}(t) = Pr(t) \times \prod_{i' \in D, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \quad (6)$$

根据定义, 可以计算出不确定数据在分布式中的 skyline 概率. 但是这个计算复杂度仍然是指数级的. 因此, 给出以下几个过滤方法减少计算代价.

假设不确定元组  $t_i$  是  $D^I$  中的元组, 那么  $t_i$  在  $D^I$  中的 skyline 概率为

$$Pr_{\text{sky}}(t_i, D^I) = \prod_{i' \in D^I, a_{j'} < a_j, j'=1}^{n'} Pr(a_{j'}) \quad (7)$$

**引理 1.** 给定  $h$  个网络节点  $C = \{c_1, c_2, \dots, c_h\}$ , 每一个节点都拥有一个不确定数据库  $D^I (1 \leq I \leq h)$ .  $t_i$  在  $D^I$  中的全局 skyline 概率为

$$Pr_{\text{Gsky}}(t_i) = \prod_{j=1}^h Pr_{\text{sky}}(t_i, D^j) \quad (8)$$

证明.

$$\begin{aligned} Pr_{\text{Gsky}}(t) &= Pr(t) \times \prod_{i' \in D^1, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \times \\ &\quad \prod_{i' \in D^2, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \times \dots \times \\ &\quad \prod_{i' \in D^h, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \\ &= Pr(t) \times \prod_{i' \in D^1, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \times \\ &\quad \prod_{R=1, R \neq I}^h \prod_{i' \in D^R, a_{j'} < a_j, j'=1}^{n'} \sum_{j=1}^n Pr(a_{j'}) Pr(a_j) \\ &= Pr_{\text{sky}}(t_i, D^I) \times \prod_{R=1, R \neq I}^h Pr_{\text{sky}}(t_i, D^R) \\ &= \prod_{I=1}^h Pr_{\text{sky}}(t_i, D^I). \end{aligned} \quad \text{证毕.}$$

根据引理 1, 有以下推论.

**推论 2.** skyline 概率上界.

$$Pr_{\text{Gsky}}(t) \leq Pr_{\text{sky}}(t_i, D^I).$$

Skyline 概率是可以在查询到来之前计算完成的, 为了能够在线计算的时候更加有效, 根据定理 4 可得引理 2.

**引理 2.** 给定一个不确定数据  $t$ ,  $t$  的 skyline 概率为  $Pr_{\text{sky}}$ , 有

$$Pr_{\text{sky}}(t) \geq Pr_{\text{top-1}}(t).$$

证明. 因为 top-1 一定在 skyline 上, 所以当  $t$  是 top-1 时,  $t$  一定是 skyline 上的点; 反之当  $t$  是 skyline 上的点时,  $t$  不一定是 top-1, 所以,  $Pr_{\text{sky}}(t) \geq$

$Pr_{top-1}(t)$ . 证毕.

通过以上结论和定义 4, 可以计算出任意不确定元组  $t$  的全局 skyline 概率. 因此, 可以满足定理 5 的计算条件.

## 6 采样计算

通过剪枝之后, 剩余的数据采用采样计算的方式计算出最后的结果. 从问题定义中可以看出, 根据定义 2 处理 top- $k$  查询的计算代价很大. 因此本节给出一个简单而有效的抽样方法近似计算  $Pr_{top-k}$ .

设  $X_i$  是一个随机变量, 表示对元组  $t$  中实例的一次采样. 如果成功采样,  $X_i = 1$ , 否则,  $X_i = 0$ . 那么对整个数据库的采样就构成随机向量  $\mathbf{X} = (X_1, X_2, \dots, X_N)$ .  $\mathbf{X}$  表示对整个数据库的采样.

定义  $\phi(\mathbf{X})$  为

$$\phi(\mathbf{X}) = \begin{cases} 1, & \text{如果元组 } t \text{ 在 } X \text{ 中是 top-}k \text{ 结果} \\ 0, & \text{否则} \end{cases}$$

根据定义 2, 可得  $Pr_{top-k}$  的值是  $E(\phi(\mathbf{X}))$ , 算法 1 给出了近似求解  $E(\phi(\mathbf{X}))$  的步骤.

**算法 1.** *LocalApproxCompute()*.

输入: 不确定数据

1.  $L=0$ ;
2. Repeat;
3. 对元组实例的概率  $Pr(X_i=a)$  的采样;
4. if ( $\phi(\mathbf{X}) = 1$ )
5.  $L=L+1$ ;
6. Until  $N$  times;
7. Return  $L/N$ .

当  $N$  的值足够大, Chernoff-Hoeffding 界给出近似解的精确保证<sup>[8]</sup>.

**引理 3.** 对任意的  $\delta(0 < \delta < 1)$  和  $\tau(\tau > 0)$ , 如果  $N \geq \left(3 \ln \frac{2}{\delta}\right) / \tau^2$ , 则

$$Pr |L/N - E(\phi(\mathbf{X}))| \leq \tau E(\phi(\mathbf{X})) \leq \delta.$$

引理 3 保证随机算法 1 的近似解有很高的精度.

算法 1 的第 3 步骤需要在 P2P 网络的各个节点上进行采样. 而在 P2P 网络的各个节点上采样的关键是如何能够减少网络代价. 如果能够在尽量少的节点上进行采样并且使样本能够尽量覆盖整个数据空间, 那么在采样的过程中需要对网路节点进行有效的选择. 因为通过全局索引, 超级节点已经对整个网络数据的分布情况有了大致的了解, 并且全局索引中的每个结点都记录了数据来自于那些网络节点. 因此可以在超级节点上找出最少的节点并且这

些节点的数据能够尽量覆盖整个数据空间.

选择网络节点主要分为两个步骤:

第 1 步是在全局索引中找出只含有一个网络节点的结点, 如果这些网络节点也包含于其他的索引结点中, 那么就不考虑这些索引结点中的网络节点.

第 2 步是在全局索引中找出包含两个直到  $N$  (索引结点中包含的最大的网络节点个数) 个网络节点的结点, 最终找出能够覆盖整个数据空间的网络节点. 算法 2 给出了在 P2P 环境中近似求解  $E(\phi(\mathbf{X}))$  的步骤.

**算法 2.** *ApproxCompute()*.

输入: Quad-tree 索引 QT

Quad-tree 的结点中包含的最大的网络节点数  $N$

返回 Quad-tree 中某个结点的网络节点集合  $P()$

输出: 覆盖所有结点的网络节点集合  $S$ ;

1. For(int  $i=1$ ;  $i < N+1$ ;  $i++$ )
2. For(each node  $n$  in QT) ( $n$  为当前结点)
3.  $R=P(n)$ ; (得到当前结点里的网络节点集合)
4. If( $|R| \neq i$ )
5. Break;
6. If( $S \cap R = \emptyset$ )
7.  $S=S \cup R$ ;
8. Return  $S$
9. *LocalApproxCompute()*.

## 7 性能分析

本节采用合成的不确定数据验证本文的算法, 算法代码用 Visual C++ 6.0 编写, 运行环境是 AMD Athlon 双核 2.6 GHz CPU, 2 GB 内存和 320 GB 硬盘. 对本文给出的算法, BPT 表示“基本算法”, IPT 表示带有概率剪枝和分布式采样的 top- $k$  查询算法. 在一个  $d$  维数据空间  $S=[0, 1]^d$  上生成 2~5 维数据. 首先在每个网络节点上随机生成确定数据, 然后以确定数据为中心  $r$  为半径的区域内随机生成  $\beta=[0, 3]$  个实例, 其中  $r$  是一个尽可能小的值. 数据集的大小是 1.2 MB~2 MB. P2P 网络是由超级节点连接起来的, 超级节点之间可以用任意的拓扑结构相连接. 实验中, 网络规模的大小由普通节点中的数量 ( $N_p$ ) 来衡量,  $N_p$  的范围是 40~120, 其中网络超级节点的个数  $N_{sp}$  是普通节点个数  $N_p$  的 10%. 每次实验时, 随机产生 20 个查询, 记录网络平均传输代价. 本文同时也对采样计算时本地计算精度和整个网络的计算精度进行了比较. 表 2 给出具体的实验参数, 黑体为默认值.

表 2 实验参数

参数	值
$Np$ : 普通节点个数	40, 60, <b>80</b> , 100, 120
$R$ : 数据集大小	1.2 MB, 1.4 MB, <b>1.6 MB</b> , 1.8 MB, 2 MB
$p$ : 阈值	0.1, 0.3, <b>0.5</b> , 0.7, 0.9
$k$ : 返回结果个数	100, 300, <b>500</b> , 700, 900

首先检验当普通节点个数  $Np$  变化时 BPT 和 IPT 算法的网络传输代价. 图 2 给出了实验结果, 其中横坐标是普通节点个数, 纵坐标是网络中传递元组的个数. 从结果看出, 网络传输代价在两种算法中都随着  $Np$  的增加而增加. 这个结果跟预期的一样, 因为当网络中节点数量增加时, 待采样的数据并没有发生改变, 采样过程增加了网络通信, 因此网络传输代价随之增加. 从图 2 中可以明显地看出, IPT 算法所造成的网络代价很低, 大量的数据在概率剪枝过程中被剪掉了, 减少了最终采样计算时的数据量. 比较这两种算法, IPT 算法比 BPT 算法平均多减掉 90% 以上的网络代价.

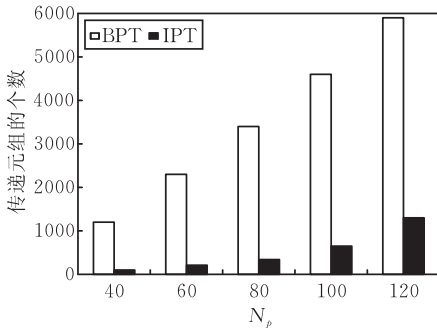


图 2 不同节点个数时网络传输代价

其次测试当数据集大小  $R$  变化时 BPT 和 IPT 算法的网络传输代价. 图 3 给出了实验结果, 其中横坐标是  $R$ , 纵坐标是网络中传递元组的个数. 从实验结果可以看出, 网络传输代价在 BPT 和 IPT 算法中随着  $R$  的增加而增加. 而且通过观察可以发现, 柱状图的变化趋势并不大. 这说明了当数据集变化时, 最终进行采样计算的数据个数变化不大. 因为概率剪枝不会因为数据量的增加而降低剪枝效力. 从图 3 中可以明显地看出, IPT 算法所造成的网络代

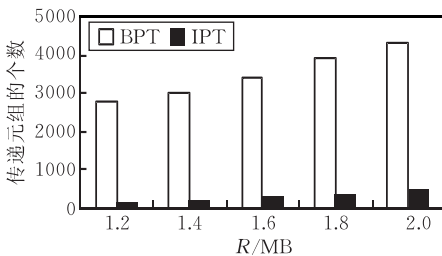
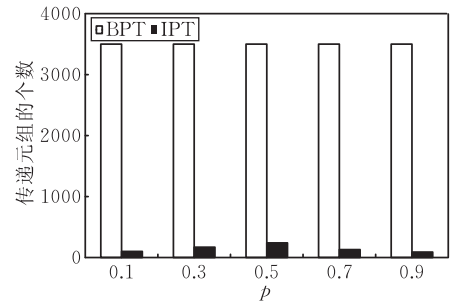


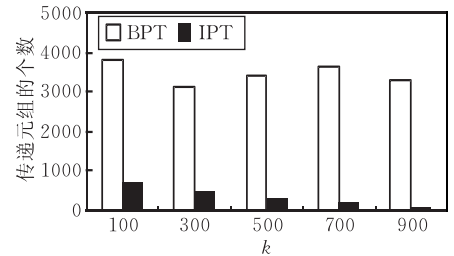
图 3 不同数据集大小时网络传输代价

价仍然很低, 大量的数据在概率剪枝过程中被剪掉了, 减少了最终采样计算时的数据量. 比较这两种算法, IPT 算法比 BPT 算法平均多减掉 94% 以上的网络代价.

然后测试当阈值  $p$  变化时 BPT 和 IPT 算法的网络传输代价. 图 4 给出了实验结果, 其中横坐标是阈值  $p$ , 纵坐标是网络中传递元组的个数. 从实验结果可以看出, 当阈值  $p$  变化时 BPT 算法的网络传输代价没有发生改变, 因为 BPT 算法没有进行概率剪枝. IPT 算法当  $p$  为 0.1 和 0.7 时网络传输代价很小, 而当  $p$  为 0.3 和 0.5 时网络代价要比 0.1 和 0.7 时大很多. 这说明本文采用的两种概率剪枝的方法都很有效, 在阈值很小的时候概率下界保证了剪枝效率, 而当阈值很大时概率上界保证了剪枝效率. 当  $p$  为 0.5 时, IPT 算法仍然比 BPT 算法平均多减掉 85% 以上的网络代价.

图 4 阈值  $p$  变化时网络传输代价

接下来测试当返回结果个数  $k$  变化时 BPT 和 IPT 算法的网络传输代价. 图 5 给出了实验结果, 其中横坐标是返回结果个数  $k$ , 纵坐标是网络中传递元组的个数. 从实验结果可以看出, 当返回结果个数  $k$  变化时 BPT 算法的网络传输代价并不稳定, 因为空间剪枝的效率是随着数据分布的变化而变化的. 而网络传输代价在 IPT 算法中随着  $k$  的增加而减少. 因为概率下界的前提是  $k \geq k_1$ , 所以, 当  $k$  值越大能够被定理 5 确认的结果的可能性就越大, 而概率上界并不因为  $k$  值的变化而变化, 因此 IPT 算法的网络传输代价呈减少趋势. 当  $k$  为 100 时, IPT 算

图 5 结果个数  $k$  变化时网络传输代价

法比 BPT 算法平均多减掉 88% 以上的网络代价。

然后测试当阈值  $p$  变化时概率上界、概率下界和空间剪枝这 3 种剪枝方法的剪枝效率. 图 6 给出了实验结果, 其中横坐标是阈值  $p$ , 纵坐标是剪枝掉数据的比例. 3 种剪枝方法分别在一个数据集中进行测试. 从实验结果可以看出, 当阈值  $p$  变化时空间剪枝算法所剪掉的数据基本没有发生改变, 因为空间剪枝算法与阈值之间没有相关联系. 概率上界的剪枝效率会随着  $p$  的变大而提高, 因为根据定理 3,  $p$  值越大概率上界越有效. 相反的, 概率下界的剪枝效率会随着  $p$  的变大而减小, 因为根据定理 5,  $p$  值越大概率下界越有效.

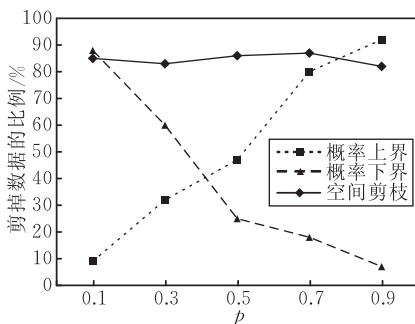


图 6 剪枝效率

最后测试 IPT 算法在 P2P 网络中采样计算的质量. 图 7 给出了实验结果, 其中横坐标是采样数量,  $\text{top-}k$  结果的精度. 两条曲线分别代表了 P2P 网络中的采样计算精度和将所有数据集中化后的采样计算精度. 集中化的数据采样计算精度在 97% 以上, 这说明了引理 3 能够保证采样计算的精度. P2P 数据采样计算的精度在 88% 以上, 虽然精度略比集中化的数据低, 但是前面的几个实验有力地证明了 IPT 算法避免了大量的网络传输代价, 在网络中 IPT 算法能够较好地处理不确定数据的  $\text{top-}k$  查询.

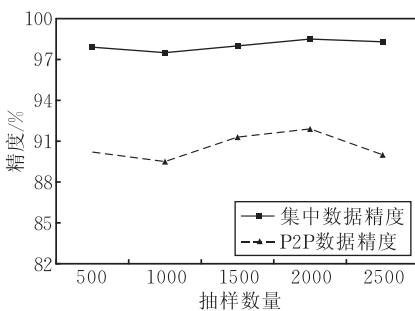


图 7 采样计算的质量

为了考察算法在真实网络中的效力, 以下实验在 1000 个普通节点的情况下分别比较  $R$  值和  $k$  值变化时网络传输代价.

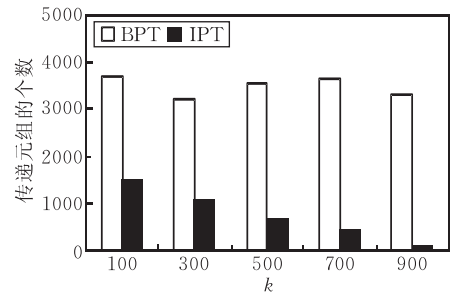


图 8 大规模网络中  $k$  变化时网络传输代价

从图 8 和图 9 中可以看出, 当增大网络规模时, BPT 算法的效果基本保持不变. 因为全局索引保证了空间剪枝的效率不随网络变化而变化. 而 IPT 算法在大规模网络中增加了网络中传递元组的个数. 这是因为数据比较分散, 在采样计算的过程中增加了网络传输代价. 这两个实验说明, 采样计算决定了网络变化时数据在网络中的传输代价.

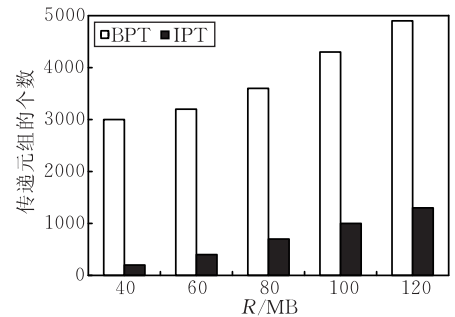


图 9 大规模网络中  $R$  变化时网络传输代价

## 8 相关工作

$\text{Top-}k$  查询只返回最满足用户要求的  $k$  个结果, 从而避免产生大量的无用结果, 因而受到广泛的重视, 在多媒体检索、数字图书馆、Web 搜索和电子商务等领域具有广泛的应用. 然而分布式环境下的  $\text{top-}k$  查询处理比较复杂. TA 算法在集中式网络环境中应用最为广泛. TA 算法的执行步骤是不固定的, 它采用的是随机查找策略, 这种查找方法使 TA 算法的查找代价较大. 基于 TA 算法, 许多有效的方法已经被应用到分布式  $\text{top-}k$  查询中. TPUT<sup>[9]</sup> 算法的目的是剪掉不需要的数据对象并且是一定终止在 3 个步骤之内. KLEE<sup>[10]</sup> 算法得到的  $\text{top-}k$  结果是近似结果, KLEE 算法的执行步骤是固定的, 为 3 步或 4 步, 4 步中多出的一步为优化候选集合. KLEE 的效率和结果质量成反比, 带宽的节省量和通信的步骤成反比.

已有很多面向集中式环境中的不确定数据

top- $k$  查询的工作. U-Top $k$ <sup>[11]</sup> 查询返回一个长度为  $k$  的元组矢量, 其在所有可能世界中的发生概率最大; U- $k$ Ranks<sup>[11]</sup> 返回在各个级别中出现频率最大的元组; PT- $k$ <sup>[12]</sup> 首先定义一个阈值, 返回所有发生概率超过阈值的元组; P $k$ -top $k$  和 Global-top $k$ <sup>[13]</sup> 则返回发生概率最大的  $k$  个元组. Expected rank top- $k$  返回期望排序最低的  $k$  个元组. Soliman 等人提出了基于搜索空间的方法来处理 U-Top $k$  查询与 U- $k$ Ranks 查询. 各元组首先按照其 Rank 从大到小进行排序, 然后不断地构造搜索空间, 缩小空间的范围, 最终获得查询结果. Hua 等人提出了针对 PT- $k$  查询的解决方法. Cheng 等人针对 U-Top $k$  查询提出了一种动态维护的结构, 支持元组的插入与删除. 他们提出了构造 Dominant 集合的方法进行求解. 在其后续工作中, 也提出了近似解决方案<sup>[12]</sup>. 文献[14] 主要研究的是排序查询, 不确定数据假设分布在一定区域中. 文献[15] 是在不确定数据流上研究的 top- $k$  查询. 文献[16] 提出了有效的方法来回答 U-top- $k$  和 U- $k$ Ranks 查询. 其中对于 U- $k$ Ranks 查询利用的是泊松二项式递推<sup>[17]</sup> 的方法. 文献[18] 考虑的是任意 SQL 查询并且排序是根据一个元组满足查询的概率而不是利用一个排序函数. 文献[19] 提出了两种排序理论可以满足离散和连续分布的不确定数据.

面向分布式不确定数据的 top- $k$  查询的研究并不是很多. 文献[20] 提出了在通信和计算方面都比较高效率的算法来计算分布式环境下的 Expected rank top- $k$ . 文献[21] 是在传感器网络中进行 top- $k$  查询, 它返回的是在每个传感器的值在任何时刻的排序之后概率最大的前  $k$  个传感器.

## 9 结束语

本文所述的是一个面向 P2P 网络的不确定数据的 top- $k$  查询. 采用离散的可能世界模型定义不确定数据. 在指出可能世界实例的数量远大于不确定数据的规模之后, 本文首先采用 Quad-tree 索引不确定数据并利用索引结构以及数据之间的支配关系对不确定数据进行空间剪枝, 然后提出了一个在 P2P 环境中的 top- $k$  查询的基本方法. 在分析了基本方法的不足之后, 采用了概率剪枝和分布式采样计算改进基本方法. 其中, 局部 top- $k$  概率和全局 skyline 概率分别是全局 top- $k$  概率的上界和下界. 通过概率剪枝和采样计算可以大大减少网络中数据

的传输量和本地的计算代价. 实验结果验证了本文的设计. 今后将研究在 P2P 环境中基于不确定数据的多种查询.

## 参 考 文 献

- [1] Babcock Brian and Olston Chris. Distributed top- $k$  monitoring//Proceedings of SIGMOD. Madison, USA, 2003; 28-39
- [2] Cormode Graham, Garofalakis Minos N et al. Holistic aggregates in a networked world; Distributed tracking of approximate quantiles//Proceedings of SIGMOD. Baltimore, USA, 2005; 25-36
- [3] Kashyap Srinivas R, Ramamirtham Jeyashankher et al. Efficient constraint monitoring using adaptive thresholds//Proceedings of ICDE. Cancun, Mexico, 2008; 526-535
- [4] Sharfman Izchak, Schuster Assaf, and Keren Daniel. A geometric approach to monitoring threshold functions over distributed data streams//Proceedings of SIGMOD. Chicago, USA, 2006; 301-312
- [5] Dalvi N N, Suciu D. Management of probabilistic data; Foundations and challenges//Proceedings of PODS. Beijing, China, 2007; 1-12
- [6] Khoussainova N, Balazinska M. Towards correcting input data errors probabilistically using integrity constraints//Proceedings of the MobiDE Workshop. Chicago, USA, 2006; 43-50
- [7] Sun Yongjiao, Yuan YeandWang Guoren. Top- $k$  query processing over uncertain data in distributed environments. World Wide Web, Online First<sup>TM</sup>, 2011
- [8] Angluin D, Valiant L G. Fast probabilistic algorithms for hamiltonian circuits and matchings//Proceedings of STOC. Boulder, USA, 1977; 30-41
- [9] Cao Pei, Wang Zhe. Efficient top- $K$  query calculation in distributed networks//Proceedings of the PODC. Newfoundland, Canada, 2004; 206-215
- [10] Michel S, Triantafillou P, Weikum G. KLEE: A framework for distributed Top- $k$  query algorithms//Proceedings of VLDB. Trondheim, Norway, 2005; 637-648
- [11] Soliman M S, Ilyas I F, Chang K C-C. Top- $k$  query processing in uncertain databases//Proceedings of ICDE. Istanbul, Turkey, 2007; 896-905
- [12] Hua Ming, Pei Jian, Zhang Wenjie, Lin Xuemin L. Ranking queries on uncertain data; A probabilistic threshold approach//Proceedings of the SIGMOD. Vancouver, Canada, 2008; 673-686
- [13] Cheng R, Kalashnikov D, Prabhakar S. Evaluating probabilistic queries over imprecise data//Proceedings of the SIGMOD. Madison, USA, 2003; 551-562
- [14] Lian Xiang, Chen Lei. Probabilistic ranked queries in uncertain databases//Proceedings of the EDBT. Nantes, France, 2008; 511-522

- [15] Jin Cheqing, Yi Ke, Chen Lei, Yu J X, Lin Xuemin. Sliding-window Top- $k$  queries on uncertain streams. PVLDB, 2008, 1(1): 301-312
- [16] Yi Ke, Li Feifei, Kollios G, Srivastava D. Efficient processing of top- $k$  queries in uncertain databases//Proceedings of the ICDE. Cancun, Mexico, 2008: 1406-1408
- [17] Lange Kenneth. Numerical analysis for statisticians. Statistics and computing. 2nd Edition. New York, USA; Springer, 2010
- [18] Re C et al. Efficient top- $k$  query evaluation on probabilistic data//Proceedings of the ICDE. Istanbul, Turkey, 2007: 886-895
- [19] Wang C H et al. Top- $k$  ranking for uncertain data//Proceedings of the FSKD. Alberta, Canada, 2010: 363-368
- [20] Li Feifei, Yi Ke, Jests Jeffrey. Ranking distributed probabilistic data//Proceedings of the SIGMOD. Providence, Rhode Island, 2009: 361-374
- [21] Silberstein Adam et al. A sampling-based approach to optimizing top- $k$  queries in sensor networks//Proceedings of the ICDE. Atlanta, USA, 2006: 68



**SUN Yong-Jiao**, born in 1983, Ph.D..

His main research interest is P2P data management.

**YUAN Ye**, born in 1981, Ph. D. , associate professor. His research interests include graph database, probabilistic database, P2P data management, and data piracy.

**WANG Guo-Ren**, born in 1966, Ph. D. , professor. His research interests include XML, data management, query processing and optimization, probabilistic database, and bioinformatics.

## Background

Top- $k$  queries retrieve the objects that best match the user requirements by employing user-specified scoring functions that result in an ordered set of objects containing the best  $k$  objects only. It avoids huge and overwhelming result sets. In many applications, such as multimedia retrieval, digital libraries, web search, and e-commerce, top- $k$  query becomes increased interest research. Due to increasingly distributed data appeared in many applications, an emerging challenge is how to support top- $k$  query processing over a highly distributed network of collaborative computers. TA algorithm is the most widely used in a centralized network environment. TA algorithm steps are not fixed. It uses a random search strategy. But it makes large costs. Three-Phase Uniform Threshold (TPUT) that aims to prune unnecessary data objects and it is guaranteed to be terminated in three round-trips. Later, TPUT was improved by KLEE, and KLEE has two variants. One requires three phases and the other only needs two round-trips. KLEE also provides mechanisms for trading performances with result having a high quality. Thus KLEE can support approximate top- $k$  retrievals. Most of these search methods are used in a star net-

work. When the central node sends top- $k$  query, all other nodes can communicate directly. But in real life, more applications use a tree topology.

Currently, uncertain data query processing technologies are proposed for centralized data management system, not suitable for large-scale distributed system. Similarly, many kinds of complex query processing techniques are proposed to process certain data in P2P environment, such as Range query, Top- $k$  query and Skyline query. Therefore, they are also not suitable for handling uncertain data in distributed systems. Although the research of uncertain data query processing is still in trial stage in P2P environment, it has attracted internationally renowned research team's attentions. Thus, combining the P2P environment and uncertain characteristics of the data, research and development the query processing of uncertain data in P2P environment is imperative. This research was supported by the National Natural Science Funds for Distinguish Young Scholar (grant No. 61025007), the National Science Fund of China Key Program (grant No. 60933001).