计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
48(10): 1871-1878, 2011

面向不确定图的 k 最近邻查询

张 旭 何向南 金澈清 周傲英
(华东师范大学软件学院上海市高可信计算重点实验室 上海 200062)
(cqjin@sei.ecnu.edu.cn)

Processing k-Nearest Neighbors Query over Uncertain Graphs

Zhang Xu, He Xiangnan, Jin Cheqing, and Zhou Aoying
(*Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai* 200062)

Abstract Complex networks, such as biological networks, social networks, and communication networks, have been widely studied, and the data extracted from those applications is inherently uncertain due to noise, incompleteness and inaccuracy, so these applications can be modeled as uncertain graphs. The k-nearest neighbors (kNN) is a fundamental query for uncertain graphs, which is to compute the k nearest nodes to some specific node in a graph. In this paper, we design a framework for processing kNN query in uncertain graphs. We firstly propose a new kNN query over uncertain graphs, following which a novel algorithm is proposed to solve the kNN query. Then we optimize this algorithm which greatly improves the efficiency of the kNN query. Theoretical analysis and experimental results show that the proposed algorithm can efficiently retrieve the answer of a kNN query for an uncertain graph.

Key words biological network; social network; uncertain graph; k-nearest neighbors; possible worlds

摘 要 生物网络、社会网络、交际网络等复杂的网络被广泛的研究,由于数据抽出时引入的噪声和错误使这些数据具有不确定性,因此可以对这些应用使用不确定图模型建模,k 最近邻查询问题是查询一个图上的距离某个特定点最近的 k 个邻居节点的问题,它是不确定图上的一个基础问题.设计了一个解决不确定图上最近邻问题的框架,首先定义了一种新颖的不确定图上的 k 最近邻查询,然后提出了针对该查询的一般处理算法,同时对该算法进行了优化,使算法效率得到极大提高.理论分析和实验结果表明提出的算法能够高效地处理不确定图上的 k 最近邻查询.

关键词 生物网络;社会网络;不确定图; k 最近邻查询;可能世界

中图法分类号 TP391

在生物网络、社交网络和交通网络等应用中所产生的数据往往可以用图来建模,因为这些数据中存在大量节点,而且部分节点之间还拥有某种关联.例如,在铁路交通图中,各个城市对应图中的节点,而相邻城市之间的铁路线则可以用边来描述.

在实际应用中,图数据往往还具有不确定性,因而需要使用不确定图模型来管理这些数据.例如,在生物网络图中,节点代表基因或蛋白质,边代表它们

收稿日期:2011-06-23;修回日期:2011-08-26
基金项目:国家自然科学基金项目(60803020,60933001,60925008,61021004);高等学校博士学科点新教师基金项目(200802511010)
通信作者:金澈清(cqjin@sei.ecnu.edu.cn)

之间的相互关系. 由于这些相互关系是通过含有噪声的实验得出,所以每一条边都有一个不确定的权值[1]. 在蛋白质交互网络[2]中,通过对有限数量的蛋白质进行实验,可以获得它们之间相互关系的强弱信息,并可以用概率值来进行标注. 在大型社交网络中,例如 Facebook 和 Twitter 等,每个用户都有一个人际关系网,节点代表用户,边的概率值描述两个用户之间关系的密切程度.

kNN(k-nearest neighbors)查询是一种重要查询类型,它返回离查询点最近的 k 个节点. kNN 查询在不确定图管理领域也很重要. 例如,在蛋白质交互网络中,kNN 查询可以用于计算和某一蛋白质关系较为密切的蛋白质集合;社交网络中的好友推荐功能就是通过 kNN 查询来查找与目标用户关系最近的若干用户.

近年来,不确定 kNN 问题得到了较多关注,并出现了多种不确定 kNN 模型[2-8]. 针对不确定图的研究工作主要包括计算可靠子图挖掘[9-10]、Top-k 查询[11]、频繁子图挖掘[12-14]等. 面向不确定图的 kNN 查询要比传统的 kNN 问题复杂得多,其主要原因在于不确定 kNN 查询处理问题涉及到可能世界模型[15-16]. 在该模型中,可能世界实例的数量随着元组个数的增长呈指数级增长. 不确定图自身的数据结构已经比较复杂,所蕴含的数据量很大,因而会衍生出大量可能世界实例,加剧了计算难度.

然而,面向不确定图的 kNN 查询的研究工作仍然比较少. 文献[8]研究了不确定图上的 kNN 查询问题,该文献定义了 3 种不确定 kNN 查询,分别采用中位距离(median-distance)、多数距离(majority-distance)和期望可信距离(expected-reliable-distance)衡量两个节点之间的距离. 其中,中位距离返回在所有可能世界中居中的最短路径;多数距离返回最有可能的最短距离;期望可信距离返回在两个节点均连通状态下的期望距离. 这些距离定义在特定应用场景之下各有意义. 本文的目的则在于考虑另外一种最短距离(minimum-distance)定义,并且针对这种距离定义给出解决方案. 所谓的最短距离是指在所有可能世界之中两个节点间的最短距离.

鉴于可能世界的数量会随着边的增加而呈现指数级增长,不确定图上的 kNN 查询面临着巨大的挑战. 本文提出了基于子图扩展的精确算法:首先扩展查询点,获得最小的 kNN 候选子集,再计算候选子集中每个节点与查询点连通的概率,最后输出与查询点距离最近的 k 个节点(若多个节点的最短距离相同,则输出连通概率较高的节点).

本文的贡献主要有以下几点:

1) 提出了一种不确定图上的 kNN 查询,即 kMinDist 查询;

2) 提出了一种针对 kMinDist 查询的算法,该算法具有较高的执行效率,优化方案进一步提高了效率;

3) 执行了一系列对比实验来验证新方法的有效性和高效性.

1 查询定义

1.1 距离定义

定义 1. 不确定图 G 可被表示为一个三元组 $G=(V,E,P)$,其中 V 为无向图的顶点集,$E\subseteq V\times V$ 是无向图中边的集合,P 是边的存在性概率函数,$\forall e\in E, P(e)$ 表示边 e 存在的概率.

在可能世界模型下,一个不确定图 $G=(V,E,P)$ 可以派生出一组确定图 $G'=(V',E')$,每个 G' 为一个图实例,满足 $V'=V, E'\subseteq E$. 令 $P_{G'}$ 表示实例 G' 出现概率,$P_{G'}$ 可被计算为

$$P_{G'}=\prod_{e\in E'}P(e)\times\prod_{e\in E\setminus E'}(1-P(e)),$$

这样的图实例共有 $2^{|E|}$ 个,且对于任意 G',$P_{G'}\in[0,1]$. 此外,所有图实例的概率和等于 1,即:$\sum_{G'}P_{G'}=1$.

图 1 就是一个不确定图,共有 6 个节点和 6 条边;各条边上附加一个概率值,表示该条边的存在性概率;各条边均相互独立. 图 2 是图 1 的一个不确定图实例,这个实例的存在概率为 0.0144(=0.8×0.4×0.5×0.6×0.5×(1−0.7)).

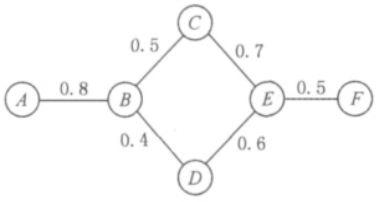


Fig. 1 An example of uncertain graph.

图 1 一个不确定图

定义 2. 最短距离. 不确定图 $G=(V,E,P)$ 上的任意一对节点 s 和 t 之间的最短距离即为在所有可能的图实例中 s 和 t 之间的最短距离,标记为 $d_L(s,t)$.

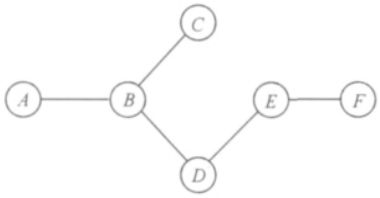


Fig. 2　An possible world instance of uncertain graph.

图 2　一个不确定图的图实例

例如,$d_L(A,F)=4$,$d_L(A,C)=2$.

定义 3. 最短距离概率. 给定不确定图 $G=(V,E,P)$上的任意一对节点 s 和 t 之间的最短距离概率即为在所有可能的图实例中,s 和 t 的距离等于 $d_L(s,t)$的总概率,标记为 $P_L(s,t)$.

例如,$P_L(A,B)=0.8$,$P_L(B,E)=0.4\times0.8+0.5\times0.7-0.4\times0.8\times0.5\times0.7=0.558$.

定义 4. kMinDist 查询. 给定不确定图 $G=(V,E,P)$,源点 s,找到一个含有 k 个节点的集合 $T_k(s)=\{t_1\ldots t_k\}$,$\forall t\notin T_k(s)$,$t_i\in T_k(s)$,若 $d_L(s,t)=d_L(s,t_i)$,则 $P_L(s,t)<P_L(s,t_i)$;否则恒有 $d_L(s,t)>d_L(s,t_i)$.

以图 1 为例,假设源点为 A,$k=2$. 显然,节点 B 与 A 的距离最小,$d_L(A,B)=1$,因此节点 B 被输出. 此外,节点 C 和 D 到 A 的距离相等,$d_L(A,C)=d_L(A,D)=2$. 但是,$P_L(A,C)=0.4$,$P_L(A,D)=0.32$,因而 C 比 D 的优先级要高. 最终,该查询返回$\{B,C\}$.

2　kNN 查询算法

本节开始描述针对 kMinDist 查询的解决方案,如算法 1 所示.

算法 1. $kMinDist(G,s,k)$.

输入:$G=(V,E,P)$为不确定图;

　　s 为查询源点;

　　k 为返回的节点个数;

输出:$Result$ 为返回查询结果集合.

① $d:=1$;

② 初始化 T 为空集;

③ 从点 s 开始对图 G 进行广度优先遍历;

④ while G 中的节点没有遍历完 do

⑤　将所有到点 s 距离为 d 的节点加入到 T 中;

⑥　if $|T|<k$ then

⑦　　$d:=d+1$;

⑧　else

⑨　　break;

⑩　end if

⑪ end while

⑫ for each v in T do

⑬　调用函数 $getPL(G,s,v,d)$以计算 $P_L(s,v)$;

⑭ end for

⑮ 返回 k 个结果元组;

算法 1 的输入参数包括不确定图 $G=(V,E,P)$、查询源点 s 和查询参数 k. 集合 T 表示包含所有候选节点的集合,$|T|$ 表示集合 T 中包含的节点个数. 算法首先生成集合 T,使得其所包含的节点个数恰巧不少于 k 个(行③～⑪). 然后,调用函数 $getPL$ 计算各个节点的最短距离概率,并生成最终的查询结果(行⑫～⑮).

2.1　计算最短距离概率

算法 1 通过层次遍历得到 kMinDist 查询的最小候选子集,然后计算候选子集中所有节点至源点的最短距离概率. 事实上,在计算最短距离概率的时候并不需要扩展整个不确定图,而是仅需要扩展不确定图的一部分节点即可,该不确定子图仅包括从源点到目标节点的最短路径上的所有节点以及相关边. 这样即可显著减少计算时间.

例如,图 3 包含 9 个节点. 现在,要求节点 A 与节点 H 之间的最短距离概率. 显然,$d_L(A,H)=3$. 总共有 4 条最短路径:$ABFH$,$ABEH$,$ACEH$ 和 $ADGH$. 图 4 则是上述最短路径所构成的子图,未在最短路径之中的节点和边均被舍去. 与图 3 相比,图 4 总共减少了 4 条边,因此显著减少了子图扩展的计算开销.

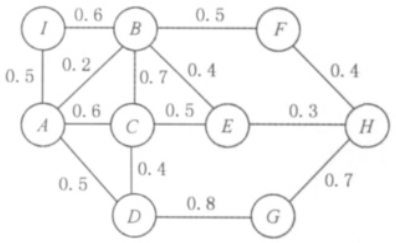


Fig. 3　An example of uncertain graph.

图 3　不确定子图的例子

依据可能世界模型,一个不确定图 $G=(V,E,P)$可以衍生出 $2^{|E|}$ 个可能世界实例. 令 $n=|E|$,可以用向量 $\boldsymbol{X}=\{X_1,\cdots,X_n\}$来描述不确定图 G 的一个实例,其中:

$$X_i=\begin{cases}1, & \text{表示第 } i \text{ 条边存在;}\\ 0, & \text{表示第 } i \text{ 条边不存在.}\end{cases}$$

算法 2($getPL$)描述了计算最短距离概率的步

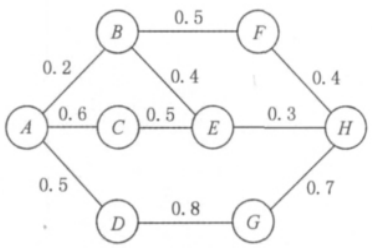


Fig. 4 The generated subgraph based on Fig. 3.

图 4 由图 3 产生的不确定子图

骤,具体如下.

1) 求出目标节点 v 与源点 s 之间的所有最短路径的集合 $pathSet$,以及仅包含 $pathSet$ 的子图 G'(行①②);

2) 以向量形式描述所有最短路径(行③④);

3) 遍历子图 G' 生成的所有可能实例,并判断各个实例是否满足使目标点与源点以最短路径相连. 判断方法为:将实例的边向量与所有最短路径进行逐位与操作,如果结果向量中位 1 的数目等于最短路径长度 d,则表明该实例中存在源点和目标节点的最短路径(行⑥~⑪);

4) 找到所有使目标节点与源点能够以最短路径相连的子图实例,调用 $pVector$ 函数(算法 3)来计算该实例的概率 p,并最终计算出最短距离概率 $sumP$(行⑫~⑰);

5) 返回 $sumP$(行⑱).

算法 2. $getPL(G,s,v,d)$.

输入:$G=(V,E,P)$为不确定图;

s 为查询源点;

v 为目标节点;

d 为节点 v 到源点 s 的最短距离;

输出:p_L 为节点 v 到源点 s 的最短距离概率.

① 创建集合 $pathSet$,包括所有从 v 到 s 且长度为 d 的路径;

② 基于 $pathSet$ 创建新子图 G',令 $edgeSet$ 代表 G' 的边集;

③ 对 $edgeSet$ 中的边编号,序号从 1 开始到 $|edgeSet|$;

④ $pathSet$ 中的所有路径均可用长度为 $|edgeSet|$ 的 0/1 向量表示,记为 $pathVecSet$;

⑤ $sumP := 0$;

⑥ for each 由 G' 生成的可能世界实例 do

⑦ 令 ***iVector*** 表示该实例中所有边组成的 0/1 向量;

⑧ for each ***pathVec*** in $pathVecSet$ do

⑨ ***iTemp*** = ***iVector*** & ***pathVec***;

⑩ 令 $count$ 表示向量 ***iTemp*** 中为 1 的个数;

⑪ if($count=d$) then

⑫ 调用 $pVector(edgeSet, iVector)$ 函数,计算每个向量 ***iVector*** 的概率 p_v;

⑬ $sumP = sumP + p_v$;

⑭ break;

⑮ end if

⑯ end for

⑰ end for

⑱ return $sumP$;

图 1 是一个不确定子图,源点为 A,计算节点 F 到节点 A 的概率.

首先,计算节点 F 到源点 A 所有距离为最短距离的路径为:$A\rightarrow B\rightarrow C\rightarrow E\rightarrow F$ 和 $A\rightarrow B\rightarrow D\rightarrow E\rightarrow F$;

其次,得到所有最短路径组成的子图(AB,BC,BD,CE,DE,EF),并对边进行编号;

然后,把所有路径用向量表示,分别为(110101),(101011),然后遍历子图的所有可能实例(从(000000)到(111111)),对每个子图实例进行判断:分别与(110101),(101011)进行逐位与操作,如果两次与的结果中 1 的个数均等于最短路径长度 4 时,则将该向量保存到候选向量数组中,因为该子图共有 6 条边,所以要遍历 $2^6=64$ 种情况;

最后,计算所有候选向量数组中向量的概率值并求和,得到概率和 $sumP$,$sumP$ 就是节点 A 与源点 F 以最短距离连通的概率. 由于情况太多我们就不一一列举遍历过程. 通过计算,我们可以求得 $sumP$ 为 0.202 4,即节点 I 到源点 G 的概率为 0.202 4.

算法 3. $pVector(edgeSet, \textbf{vector})$.

输入:$edgeSet$ 为向量中所有边的集合;

$vector$ 为需要计算概率的向量;

输出:p_v 返回向量 ***vector*** 的概率.

① $p_v := 1$; $i := 0$;

② while($i<|\textbf{vector}|$) do

③ $p_i = edgeSet.getP(i)$; /* 获取该边的发生概率 */

④ if(***vector***[i]=1) then $p_v = p_v * p_i$;

⑤ else $p_v = p_v * (1-p_i)$;

⑥ end if

⑦ end while

⑧ return p_v;

3　算法优化

在 3.1 节描述的节点概率计算中,我们采用逐位与的方法求得目标点到源点的概率,下面我们对节点概率计算进行优化.

在一个节点到源点所有最短路径组成的子图中,所有路径可能会有一条或多条共用边(共用边为该子图中所有路径都必须经过的边),这种情况在实际应用中是真实存在的:如在一个社交网络的子网中,A,B 都与 C 有联系,C 与 D 有联系,D 与 E,F 有联系,则 A,B 和 E,F 就由一个共同的关系(C,D)连接,(C,D)就是它们的共用边.

在计算目标节点的概率时,如果目标节点到源点的子图存在共用边,则在计算节点概率时,我们可以暂时不考虑共用边,在子图中除去共同边,计算节点的概率,得到的结果再乘以共用边的概率,即为节点的实际概率. 经过剪枝优化后,每剪掉一条共同边,子图所有的实例数会减少一半,计算效率至少提高一倍.

例 1. 图 1 就是一个含有共用边的例子,在计节点 F 的概率时,共有两条到源点 A 的最短路径 $[AB,BC,CE,EF]$和$[AB,BD,DE,EF]$,其中(A,B)和(E,F)是两条路径的共用边. 按照算法 2,我们需要遍历整个子图的所有可能实例,共有 $2^6=64$ 种可能情况,每个实例都要与 2 个代表路径的 6 维向量进行逐位与操作. 而当我们使用优化算法计算概率时,不需要考虑 AB 和 EF 两条共同边,只需要考虑剩下的 4 条边,因此在遍历所有可能世界的实例时,只有 $2^4=16$ 种情况,与未剪枝相比,少遍历 48 种情况,并且在与操作时,只需要与 2 个 4 维向量进行与操作即可,极大地减少了计算开销.

4　实　　验

4.1　实验准备

为了验证算法的有效性和高效性,我们采用了真实的和合成的不确定图数据进行实验,算法代码用 Java 语言编写,实验均在 Windows 7 的 32 位系统下编译运行,机器配置为 2.66 GHz Intel CPU,2 GB 物理内存.

真实数据集来自美国安然公司电子邮件网络的近 50 万封电子邮件的电子通信信息[17]. 节点代表用户,如果用户 A 和用户 B 之间有过邮件联系,则用户 A 和用户 B 之间就有一条边. 我们对数据进行了预处理,得到的结果为有 10 000 个节点、107 044 条边的不确定图.

模拟数据集通过如下方式合成得到:首先产生 V 个顶点,然后通过随机选择两个端点的方式生成 E 条边,并对每条边生成一个随机的概率值. 在上述合成方法中,V 和 E 为实验中设定的参数,可以合成不同特征的数据.

4.2　实验结果与分析

我们进行了多组实验,主要设置图参数 N 和查询参数 k,并对优化前和优化后的算法效率进行对比.

实验 1. 当不确定图节点数 N 不变、k 变化时,查询执行时间的变化.

本组实验是固定 N 不变,通过改变 k 的值来观察算法的执行时间与 k 的关系. 图 5 为生成数据集下的实验效果,模拟数据集中 $N=2\,000$,图 6 为真实数据集下的实验效果,真实数据集中 $N=10\,000$. 从图中的对比结果易知,无论是真实数据集还是模拟数据集优化后的算法执行时间远小于优化前的执行时间,因此我们的优化策略极大地提高了查询效率. 同时从图中可以看出,当 k 值在某一个区间内变化时,算法的执行时间相差不大. 这是因为算法 1 中,我们通过层次遍历计算 kNN 的最小候选子集,当 k 变化,但候选集并未扩大新的一层时,候选集中节点个数并没有变化,因此对算法执行时间影响也就不大.

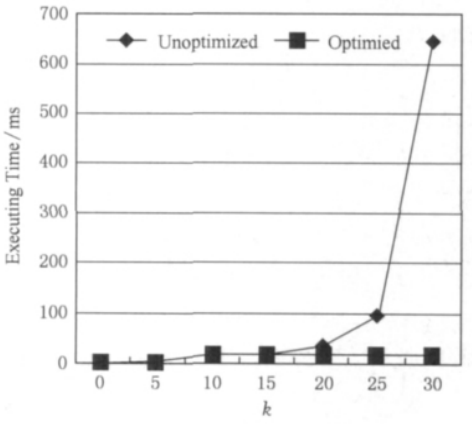


Fig. 5　Executing time vs. k upon the synthetic dataset.

图 5　生成数据集下执行时间随着 k 值变化的情况

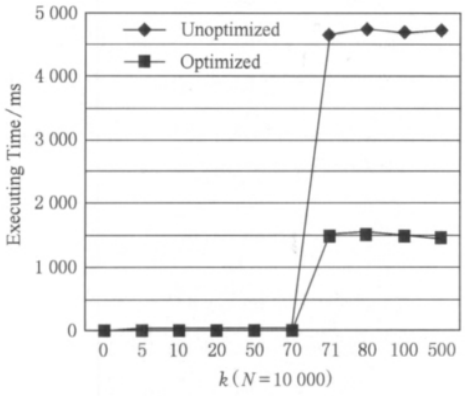
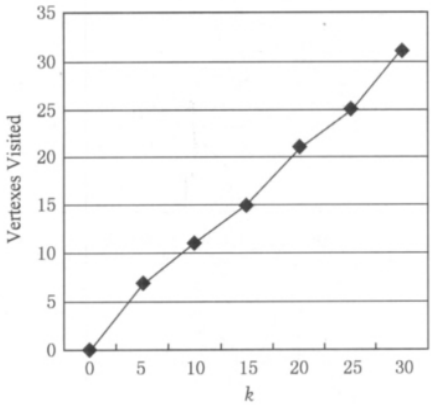


Fig. 6 Executing time vs. k upon the real dataset.

图 6 真实数据集下执行时间随着 k 值变化的情况

实验 2. 当 N 不变、k 变化,遍历图时访问节点数量的变化.

本组实验是验证 k 变化对算法遍历图时访问节点数的影响.其中图 7 为模拟数据集下的实验效果,图 8 为真实数据集下的实验效果.可以看出在模拟数据集中,随着 k 的增加访问节点的数量明显增加,因为模拟数据集是一个稀疏图,所以 k 增加时访问节点数增长较快.而在真实数据集中,访问节点数成阶梯状变化,当 k 到某个点时,访问节点数会突然增加.这是因为在生成的图中,每一层节点的数目较少,而真实数据集中的图比较复杂,每一层节点的数目较多.根据算法 1,每次遍历不确定图时都是遍历完整个层,因此 k 在某一层的范围内变化时,访问节点数目不变.

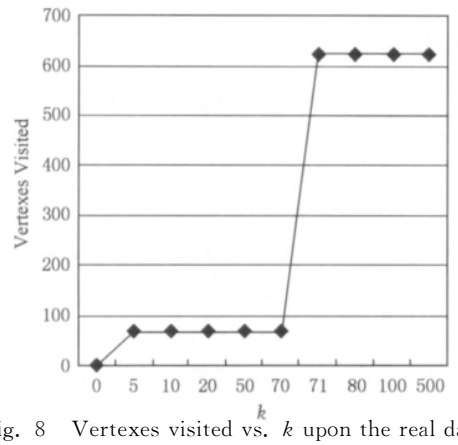


g. 7 Vertexes visited vs. k upon the synthetic dataset.

图 7 生成数据集下访问节点数随着 k 值变化的情况

实验 3. 当 k 不变时随着图中节点数 N 变化算法的执行效率对比.

图 9 是在模拟数据集中,$k=20$,随着不确定图的节点总数 N 变化,对算法执行时间的影响.可以看出随着图上节点总数的增加,而算法的执行时间



ig. 8 Vertexes visited vs. k upon the real dataset.

图 8 真实数据集下访问节点数随着 k 变化的情况

并没有变化,这是因为算法 1 在执行图的层次遍历,以得到最小的候选节点子集.只要保持 k 不变,子图中节点的数目就不变,在计算概率时并不需要考虑其他节点.所以在 k 不变、N 增加的情况下,算法的执行时间不会变化,同理算法的访问节点数也不会变化,如图 10 所示.

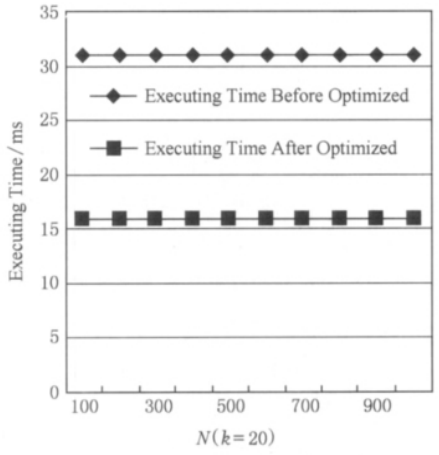


Fig. 9 Executing time vs. Number of vertexes.

图 9 执行时间随着节点总数变化的情况

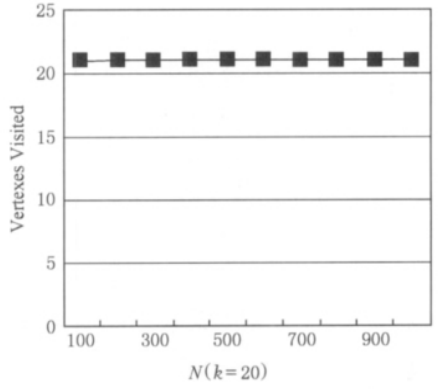


Fig. 10 Vertexes visited vs. number of vertexes.

图 10 访问的节点数随着节点总数变化的情况

5　相关工作

不确定数据管理是近年来数据库领域的研究热点之一,也出现了不少原型系统,如 BayesStore[18], MayBMS[19], MCDB[20], MystiQ[21], ORION[22], PrDB[23]和 Trio[24]等.

kNN 查询是一个基础的数据管理问题,广泛应用在链接预测、聚类、分类、图数据挖掘等应用中.针对不确定数据的 kNN 查询的研究工作已经得到了广泛的关注,出现了一些新的查询定义以及解决方案[2-7].不确定图数据管理也得到了较多关注,当前的研究内容包括可靠子图挖掘[9-10]、top-k 查询[11],频繁子图挖掘[12-14]等.

不确定图上的 kNN 问题比普通的不确定 kNN 问题复杂得多,传统的不确定 kNN 问题不适用于不确定图.

目前关于不确定图上 kNN 问题的文章比较的少,文献[8]提出了一种基于抽样算法来解决不确定图上的 kNN 问题,但是它只能得到近似的结果,不能得到完全准确的结果.

6　结　　论

本文研究了不确定图上的 k 最近邻查询问题,提出了一种不确定图上的 k 最近邻查询定义,同时提出了一种方法来求解这个 k 最近邻问题.本文还进一步提出了优化措施,具有更高的效率.最后,一系列实验验证了算法的有效性和高效性.未来可能的研究方向包括带权值的不确定图上的 k 最近邻查询问题.

参　考　文　献

[1] Asthana S, King O D, Gibbons F D, et al. Predicting protein complex membership using probabilistic network reliability [J]. Genome Research, 2004, 14(6): 1170-1175

[2] Bekales G, Soliman M A, Ilyas I F. Efficient search for the top-k probable nearest neighbors in uncertain databases [J]. Proc of the VLDB Endowment, 2008, 1(1): 326-339

[3] Cheng R, Chen J, Mokbel M, et al. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data [C] //Proc of ICDE. Piscataway, NJ: IEEE, 2008: 973-982

[4] Cheng R, Kalashnikov D V, Prabhakar S. Querying imprecise data in moving object environments [J]. IEEE TKDE, 2004, 16(9): 1112-1127

[5] Kriegel H P, Kunath P, Renz M. Probabilistic nearest-neighbor query on uncertain objects [C] //Proc of DASFAA. Berlin: Springer, 2007: 809-824

[6] Zhang W, Lin X, Cheema M A, et al. Quantile-based kNN over multi-valued objects [C] //Proc of ICDE. Piscataway, NJ: IEEE, 2010: 16-27

[7] Zhang Y, Lin X, Zhu G, et al. Efficient rank based kNN query processing over uncertain data [C] //Proc of ICDE. Piscataway, NJ: IEEE, 2010: 28-29

[8] Potamias M, Bonchi F, Gionis A, et al. k-Nearest neighbors in uncertain graphs [J]. Proc of the VLDB Endowment, 2010, 3(1): 997-1008

[9] Hintanen P. The most reliable subgraph problem [C] //Proc of the 11th European Conf on Principles and Practice of Knowledge Discovery in Databases. Berlin: Springer, 2007: 471-478

[10] Hintsanen P, Toivonen H. Finding reliable subgraphs from large probabilistic graphs [J]. Data Mining and Knowledge Discovery, 2008, 17(1): 3-23

[11] Zhang Shuo, Gao Hong, Li Jianzhong, et al. Efficient query processing on uncertain graph databases [J]. Chinese Journal of Computers, 2009, 32(10): 2066-2079 (in Chinese)
(张硕, 高宏, 李建中, 等. 不确定图数据库中高效查询处理 [J]. 计算机学报, 2009, 32(10): 2066-2079)

[12] Zou Z, Li J, Gao H, et al. Mining frequent subgraph patterns from uncertain graphdata [J]. IEEE Trans on Knowledge and Data Engineering, 2010, 22(9): 1203-1218

[13] Zou Z, Li J, Gao H, et al. Finding top-k maximal cliques in an uncertain graph [C] //Proc of ICDE 2010. Piscataway, NJ: IEEE, 2010: 649-652

[14] Zou Z, Li J, Gao H. Discovering probabilistic frequent subgraphs over uncertain graph databases [C] //Proc of SIGKDD. New York: ACM, 2010: 633-642

[15] Dalvi N, Suciu D. Management of probabilistic data Foundations and challenges [C] //Proc of PODS. New York: ACM, 2007: 1-12

[16] Khoussainova N, Balazinska M. Towards correcting input data errors probabilistically using integrity constraints [C] //Proc of MobiDE Workshop. New York: ACM, 2006: 43-50

[17] Cohen W. Enron email data. [2011-06-01]. http://snap.stanford.edu/data/email-Enron.html

[18] Wang D Z, Michelakis E, Garofalakis M, et al. Bayesstore: Managing large uncertain data repositories with probabilistic graphical models [J]. Proc of the VLDB Endowment, 2008, 1(1): 340-351

[19] Antova L, Jansen T, Koch C, et al. Fast and simple relational processing of uncertain data [C] //Proc of ICDE. Piscataway, NJ: IEEE, 2008: 983-992

[20] Jampani R, Xu F, Wu M, et al. Mcdb: A monte carlo approach to managing uncertain data [C] //Proc of ACM SIGMOD 2008. New York: ACM, 2008: 687-700

[21] Dalvi N, Suciu D. Efficient query evaluation on probabilistic databases [C] //Proc of VLDB. San Francisco: Morgan Kaufmann, 2004: 523-544

[22] Singh S, Mayfield C, Mittal S, et al. Orion 2.0: Native support for uncertain data [C] //Proc of ACM SIGMOD 2008. New York: ACM, 2008: 1239-1242

[23] Sen P, Deshpande A, Getoor L. PrDB: Managing and exploiting rich correlations in probabilistic databases [J]. The VLDB Journal, 2009, 18(5): 1065-1090

[24] Agrawal P, Benjelloun O, Sarma A D, et al. Trio: A system for data uncertainty and lineage [C] //Proc of VLDB. New York: ACM, 2006: 1151-1154

Zhang Xu, born in 1986. Master candidate at Software Engineering Institute, East Normal University. His research interests include mobile data management and uncertain data management.

He Xiangnan, born in 1992. BSc candidate at Software Engineering Institute, East China Normal University. His research interests include data stream management, uncertain data management and mobile data management.

Jin Cheqing, born in 1977. PhD and associate professor. His research interests mainly include data stream management, uncertain data management and mobile data management.

Zhou Aoying, born in 1965. Professor, PhD supervisor. Senior member of China Computer Federation. His research interests focus on data management and information system, inclusive of Web data management, Chinese Web infrastructure, Web searching and mining, data streaming and mining, complex event processing and real-time business intelligence, uncertain data management and applications, data intensive computing, distributed storage and computing, peer-to-peer computing and management, Web service.