基于阈值的概率图可达查询

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 110004)

(东北大学信息科学与工程学院 沈阳 110004)

摘 要 图的可达性查询被广泛应用于生物网络、社会网络、本体网络、RDF 网络等.由于对数据操作时引入的噪声和错误使这些图数据具有不确定性,而确定图的可达查询不能有效地处理不确定性,因此该文研究用概率语义描述的图可达性查询.具体的,该文使用可能世界概率模型定义不确定图(称为概率图),基于该模型,研究了基于阈值的概率可达查询(T-PR).首先为避免枚举所有可能世界,给出一个基本算法可精确求解 T-PR 查询.其次为进一步加速基本算法,给出 3 种改进方法,它们是不确定事件界、同构图的缩减、基于不相交路径和割集的界.通过合理的组合给出 3 种方法的合并算法.最后基于真实概率图数据的大量实验验证了该文的设计.

关键词 概率图;可能世界;不确定事件;同构图缩减;路径集;割集

中图法分类号 TP391 **DOI 号**: 10.3724/SP.J.1016.2010.02219

Answering Threshold-Based Reachability Queries Over Probabilistic Graphs

YUAN Ye WANG Guo-Ren

(*Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, Shenyang* 110004)

(*College of Information Science and Engineering, Northeastern University, Shenyang* 110004)

Abstract Graph reachability queries are widely used in biological networks, social networks, ontology networks and RDF networks. Meanwhile, data extracted from those applications is inherently uncertain due to noise, incompleteness and inaccuracy, and traditional certain reachability queries cannot effectively express semantics of such uncertain graph data. Therefore, in this paper, the authors study the reachability queries over uncertain graphs under the probabilistic semantics. Specifically, they study a threshold-based probabilistic reachability (T-PR) query over an uncertain graph using the possible world semantics (called probabilistic graph). Firstly, to avoid enumerating all possible worlds, the authors propose a basic algorithm that can exactly compute T-PR query. To further speed up the basic algorithm, they develop three improved approaches, that is, u-event bounds, isomorphic graph reduction, and disjoint path/cut set bounds. Moreover, the authors combine the three improved algorithms into one entire algorithm. Finally, they have verified the effectiveness of the proposed solutions for T-PR queries through extensive experiments on real probabilistic graph datasets.

Keywords probabilistic graph; possible world; uncertain event; isomorphic graph reduction; path set; cut set

---

收稿日期:2010-08-22.本课题得到国家自然科学基金重点项目(60933001)、国家自然科学基金面上项目(60773221)及国家"八六三"高技术研究发展计划项目基金(2009AA01Z150)资助.**袁 野**,男,1981 年生,博士研究生,研究方向为图数据库、概率数据库、P2P 数据管理和数据隐私.E-mail: linuxyy@gmail.com. **王国仁**,男,1966 年生,教授,博士生导师,研究领域为 XML 数据管理技术、查询处理与优化、概率数据库和生物信息学.

1 引 言

本文研究一个概率图上任意两点的可达性问题.具体地,给出一概率图,其任意两点 s 和 d,一用户查询阈值 $\varepsilon(0\leq\varepsilon<1)$,返回 s 和 d 的连通概率是否不小于 ε.由于图的不确定性存在于许多应用中[1-7],研究概率图中的可达查询十分必要,下面给出两个具体应用的实例.

应用实例 1. 已有很多关于概率 XML 数据库的研究[1-2].在这些文献中,一个 XML 文档被建模成一颗概率 XML 树.树的结点被分成"ordinary 结点"和"distribution 结点",这些结点定义了父结点和孩子结点之间概率的依赖关系.在 XML 数据库中,可达性查询是最基本的路径查询.其形式为 $//P1//P2$,表示 $P1$ 是 $P2$ 的祖先.在概率 XML 树中,该查询不仅返回从 $P1\sim P2$ 的一条路径,而且还包括 $P1\sim P2$ 的可达概率.这样的查询也可被看作有向概率图的可达查询.

应用实例 2. 对本体的查询日趋重要,因为很多领域都在建立自己的本体知识库,其中"基因本体(GO)"是最著名的本体知识库(http://www.geneontology.org).GO 被构建成有向无环图,其中图结点表示概念,边表示概念之间的从属关系,而概念是对有机物中不同种类 RNA 的抽象.对于 GO 网络,生物专家会关心某两个概念之间的关系在特定的生物过程中起到的作用.对该问题,可用两概念间的可达查询来回答.而由于不准确的 RNA 检测方法,使概念之间的联系不再确定[4-6],因此使用概率图表示 GO 网络是合适的[3],其中边的权值表示概念之间存在相互作用的可能性.

对应用实例 2,文献[7]给出一个真实概率网络,其中每条边被赋予一个概率.此概率被定义为两种 RNA 之间的可靠性,该值越大,两种 RNA 之间越可靠.图 1 给出该网络的一部分,在图中,边 e_3 上的概率表示概念 B 对概念 A 影响的概率是 0.75.对图 1,会提出查询"概念 s 影响概念 d 的概率是多少?".传统的 s 和 d 之间的可达性查询仅返回 s 可以影响 d.但是,上述查询不仅返回"可以影响",而且还返回影响的程度.

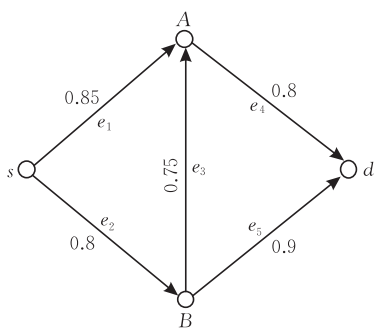


图 1 概率图

为解决概率有向图的 T-PR 查询问题,本文采用可能世界模型[8-9](一种被用来描述概率数据库的模型)定义概率图.具体地,给出一个概率有向图,赋予每边一个存在概率,每个顶点是确定的,即其概率为 1.一个概率图的可能世界是一个确定图,称为可能世界图(简称可能图),它是概率图中所有边(取决于它们的存在性)组合的一个实例.一个可能图的概率是其所有存在边概率和其不存在边的不存在概率的乘积.给定概率图两顶点 s 和 d,从 s 到达 d 的可达概率是部分可能图概率的和,在这些可能图中,s 和 d 必须是连通的.

一种直接求解 T-PR 查询的方法是枚举概率图所有的可能图,并且对每个可能图做传统的可达查询处理.找出所有给定两点连通的可能图,对其概率求和,所得结果即是可达概率,再与查询阈值比较返回结果.称该方法为 Naive 算法.但是,此方法的效率非常低,因为需枚举指数级的可能图.因此,为避免 Naive 算法,本文首先设计一基本算法可精确计算出可达概率.但处理 T-PR 查询是 $\sharp P$ 完全问题,因此在最坏情况下,基本算法的计算代价会很高.为此本文对基本算法开发出 3 种改进方法,它们分别是"不确定事件界"、"同构图的缩减"、"基于不相交路径/割集界".3 种方法都是充分利用图的结构特点和阈值界加速查询,尽可能早地使基本算法收敛.最后给出 3 种方法的合并算法.正如用真实概率图数据的实验结果所示,合并算法具有非常快的运行速度.

2 问题定义

定义 1. 一个概率图是一集合 $G=((V,E),Pr)$,其中 (V,E) 是有向无环图,$Pr:E\rightarrow(0,1]$ 是定义边集中每条边存在的概率函数.

从定义 1 易知确定图是一个特殊的边概率为 1 的概率图,可表示为 $G=((V,E),1)$.正如第 1 节提到的,本文使用可能世界模型来定义概率图.在可能世界模型下,一概率图可派生出一组确定图 $G'=(V',E')$,此确定图称为可能世界图,简称可能图,它满足 $V'=V,E\subseteq E'$.

根据文献[1-7]应用的规定,本文假设概率图不

同边的概率分布是相互独立的，因此可能图的概率为

$$Pr(G')=\prod_{e\in E'}Pr(e)\cdot\prod_{e\in E\setminus E'}(1-Pr(e)) \quad (1)$$

定义 2. 给一概率图 G，其两个顶点 s 和 d，一查询阈值 $\varepsilon(0\leq\varepsilon<1)$，基于阈值的概率可达查询(T-PR)返回如下信息：(1)“两顶点能够以概率 ε 连通”，如果可达概率 $q_{pr}\geq\varepsilon$；(2)“两顶点不能以概率 ε 连通”如果可达概率 $q_{pr}<\varepsilon$. 其中可达概率计算如下

$$q_{pr}=\sum_{G'\in RA(G)}Pr(G') \quad (2)$$

其中 $RA(G)$ 是 s 可达 d 的可能图的集合.

例如对图 1 中的概率图，枚举其 $2^5=32$ 个可能图，然后判断 s 和 d 连通的可能图，对它们的概率求和即是可达概率，其值是 0.9176. 如果用户的查询阈值是 0.5，则系统返回“两顶点能够以 0.5 的概率连通”.

注意到 s 和 d 在 G 中可能不连通，本文采用“Path-Tree Cover”[10]方法首先测试 s 是否可到达 d. 如果它们连通，然后计算可达性概率 q_{pr}. 否则，直接返回不可达信息. 接下来，本文都假设在 G 中 s 和 d 是连通的.

对计算可达概率的复杂度，引用文献[11]中的结论.

引理 1. 计算概率图中的可达概率是一个 $\sharp P$ 完全问题.

3　基本算法

定义 3. 给定一个概率图 G，其两个顶点 s 和 d，一个 s—d 路径是连通从 s 到 d 的连续边的集合. 如果一个 s—d 路径中不包含任何的 s—d 子路径，则它被称为最小 s—d 路径(表示为 $I_{\min}(s—d)$).

根据定义 1 知，图 G 是无环图，因此易知一 s—d 路径即是 $I_{\min}(s—d)$.

定义 4. 对任意一边 $e\in G$，定义一布尔变量 x_e，其中事件 $x_e=1$(称为成功事件)的概率是 e 的存在概率，事件 $x_e=0$(称为失败事件)的概率是 e 不存在的概率. 接下来用 $Event(e)$ 表示事件 $x_e=1$，用 $Event(\bar{e})$ 表示事件 $x_e=0$.

这样，有了基本事件便可定义复合事件. 例如事件 $I=Event(\bar{i}jk)$ 表示边 j 是成功的，边 i 和 k 是失败的. 其它不在 I 中的边是不确定的(即不是成功的也不是失败的)，每个 I 把图 G 的边集 E 分成 3 个不相交的子集 $E_s(I)$、$E_f(I)$ 和 $E_u(I)$，分别用来记录成功的边集、失败的边集和不确定的边集. 例如对于 $Event(\bar{i}jk)$，有 $E_f(I)=\{i,k\}$，$E_s(I)=\{j\}$，$E_u(I)=E\setminus\{i,j,k\}$. 易知 $E_s(I)\cup E_f(I)\cup E_u(I)=E$. 由于边的分布是独立的，事件 I 的概率为 $Pr(I)=\prod_{e\in E_s}Pr(e)\prod_{e\in E_f}(1-Pr(e))$. 于是我们可以定义 3 个重要事件.

定义 5. 给定一个概率图 G，其两顶点 s 和 d. 如果 $E_s(I)$ 中的边能使 s 和 d 连通，复合事件 I 是成功的(记为 s-event). 如果 $E_f(I)$ 中的边不能使 s 和 d 连通，I 是失败的(记为 f-event)，如果 I 既不是成功的也不是失败的，则 I 是一个不确定的事件(记为 u-event).

例如在图 1 中，$Event(e_1e_4)$ 是 s-event，由于其成功边集 $\{e_1,e_4\}$ 的存在使 s 和 d 连通. Event$(\overline{e_1e_2}\ e_3)$ 是 f-event，因为当其失败边集 e_1，e_2 不存在时，必使 s 和 d 不连通. $Event(\overline{e_2}e_3)$ 是 u-event，因为不能通过判断其成功边集 e_3 的存在和其失败边集 e_2 的不存在，来确定 s 和 d 是连通还是断开.

定理 1. 对概率图 G 中任意一 u-event I，I 中至少有一条 $I_{\min}(s—d)$ 路径.

证明. 设 I 的 $E_u(I)$ 为 $e_1e_2\cdots e_\alpha$，那么有 $(e_1e_2\cdots e_\alpha)\cup E_s(I)=E\setminus E_f(I)$. 因为 I 是不确定事件，故可分两种情况考虑：(1) 当 I 不失败时，由于 I 的边集 $E(I)=E_s(I)\cup E_f(I)$，所以 $I\wedge Event(e_1e_2\cdots e_\alpha)$ 必是 s-event；(2) 当 I 不成功时，边集 $(e_1e_2\cdots e_\alpha)$ 是非空的. 根据(1)、(2)，易得 $(e_1e_2\cdots e_\alpha)$ 是 $I_{\min}(s—d)$ 路径. 证毕.

例如在图 1 中，在 u-event $Event(e_2\overline{e_3})$ 下，可找到 $I_{\min}(s—d)=\{e_1e_4\}$，并且 $Event(e_1\overline{e_3})\wedge Event(e_1e_4)$ 是一个 s-event.

这个定理是基本算法的基础. 概率图 G 的整个事件空间 Ω 是 u-event，定理 1 保证在 Ω 下能够找到一个 $I_{\min}(s—d)=(e_1e_2\cdots e_n)$，也就等于找到了成功事件 $Event(e_1e_2\cdots e_n)$. 而对于任何一 u-event I，根据定理 1 都可以找到成功事件 $Event(I_{\min}(s—d))$. 下面的公式生成不相交的事件.

对于一个 u-event I，由概率分解定理得到

$$\begin{aligned}I&=I\wedge\Omega=I\wedge(Event(\overline{e_1e_2\cdots e_n})\vee Event(e_1e_2\cdots e_n))\\&=(I\wedge Event(\overline{e_1}))\vee(I\wedge Event(e_1\overline{e_2}))\vee\cdots\vee\\&\quad(I\wedge Event(e_1e_2\cdots\overline{e_n}))\vee(I\wedge Event(e_1e_2\cdots e_n))\end{aligned} \quad (3)$$

起初在 G 的整个事件空间 Ω 下找到 $I_{\min}(s—d)=(e_1\cdots e_n)$，后用式(3)产生的事件 $I\wedge Event(e_1e_2\cdots e_n)$ 是一个 s-event，而事件 $I_i=I\wedge Event(e_1,e_2,\cdots,$

$\overline{e_i}$)($1\leq i\leq n$)可能是成功、失败或者不确定的.此时可以继续用式(3)对不确定事件分解产生成功事件.重复此过程直到没有不确定事件为止.最后把所有成功事件的概率加到一起①,即可得到可达概率 q_{pr}.算法1给出基本算法的具体步骤.

算法1. 基本算法.

输入:概率图 G,s 和 d,阈值 ε

输出:有效或无效的查询结果

算法描述:

1. $S=\{\Omega,\varnothing,0,0\}$; //为整个 G 的事件空间 Ω 初始化一集合
2. $q_{pr}=0$; //初始化可达概率
3. while ($S\neq\varnothing$){
4. 根据集合 S 中的索引 i,获得一事件 I^i; //起始的 I^i 即是 Ω
5. if ($i<n_i$)
6. $i=i+1$;
7. else 集合 S 被舍弃;
8. 在事件 I^i 下找到一 s—d 路径 $I^i_{\min}(s—d)$;
9. if ($Event(I^i_{\min}(s—d))==\varnothing$)
10. 删除 I^i; //此时事件 I^i 是失败的.
11. else if (I^i 是成功的)
12. $q_{pr}=q_{pr}+Pr(I^i)$;
13. else $q_{pr}=q_{pr}+Pr(I^i\wedge Event(I^i_{\min}(s—d)))$; //$I^i\wedge Event(I^i_{\min}(s—d))$ 是成功事件
14. if ($q_{pr}\geq\varepsilon$)
15. return "两顶点能够以概率 ε 连通";
16. 将一个新的集合$\{I^i,I^i_{\min}(s—d),n_i,i=1\}$放入 S;
17. }//while 结束
18. return"两顶点不能以概率 ε 连通".

此算法的过程可用一棵解决树表示,其定义如下.

定义6. 一棵解决树是一树形结构,其根节点表示 G 的整个事件空间 Ω,其它节点表示由式(3)产生的事件 $I\wedge Event(e_1e_2\cdots\overline{e_i})$($1\leq i\leq n$)或 $I\wedge Event(e_1e_2\cdots e_n)$.特别地,每个叶子节点表示一个 s-event 或者 f-event,但中间节点仅能表示 u-event.

根据解决树,算法1的工作原理如下:开始解决树仅包含根节点表示 G 的整个事件 Ω(Ω 是一 u-event),接下来用式(3)找到一 s—d 路径 $e_1e_2\cdots e_n$,并将 Ω 分成不相交的事件.这些事件构成了解决树的第一层结点②,其中 $Event(e_1e_2\cdots e_n)$ 是成功的,事件 $Event(e_1e_2\cdots\overline{e_i})$($1\leq i\leq n$)可能是失败、成功或者不确定的.同样地,对每个 u-event 进一步递归划分为不相交的事件,直到没有 u-event 为止.解决树不断由 u-event 结点进行扩展直到只剩下 s-event 和 f-event 结点为止,并且这些结点构成解决树的叶子节点.q_{pr} 的值为所有 s-event 概率的和.最后将 q_{pr} 和阈值 ε 比较,并返回查询结果.

例如,欲计算图1中 s 到 d 的 q_{pr},图2给出其基本算法的解决树.如图所示,树根是 Ω,在 Ω 下找到 s—d 路径 e_1e_4,产生事件 $Event(\overline{e_i})$(u-enent)、$Event(e_1\overline{e_4})$(u-enent)和 $Event(e_1e_4)$(s-event).每个 u-event 继续被分解,直到解决树中只包含 s/f-event,如图2中叶子节点所示.所有的 s/f/u-event 都在图中列出,并且给出每个 s-event 的概率,它们的和 0.68+0.1224+0.072+0.0162+0.027=0.9176 即为 q_{pr}.

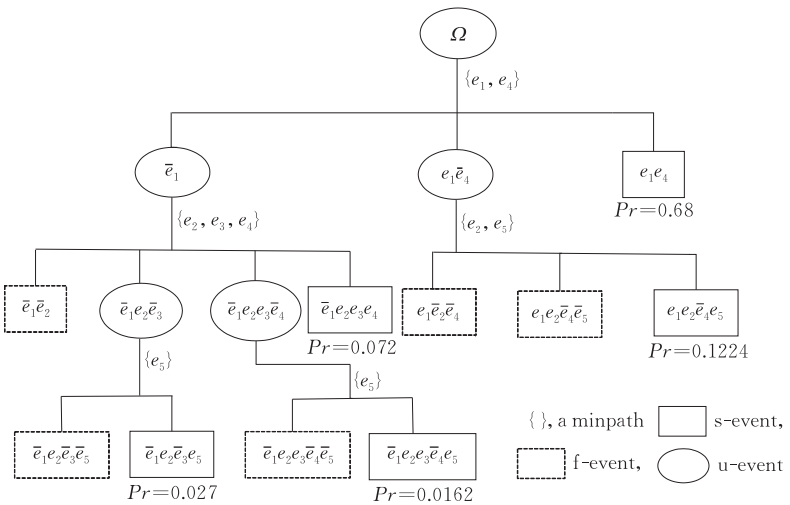


图2 图1中概率图的解决树

① 由于式(3)产生的是不相交事件,故其概率具有可加性.

② 解决树的根定义为第0层.

4　改进算法

由引理 1 知，计算 q_{pr} 的精确值是很困难的. 因此对于一个很大的图，基本算法仍需要枚举大量的 s—d 路径. 本节将给出 3 种优化算法使基本算法尽可能快地收敛.

4.1　不确定事件界

T-PR 查询包括阈值 ε，因此可以利用这个阈值来加速查询. 一种方法是计算 q_{pr} 的界，然后将此界与 ε 比较给出查询结果.

对固定的 $\alpha(0\leq\alpha<1)$，如果一个 u-event I 的概率不小于 $\alpha(Pr(I)\geq\alpha)$，则称 I 为有 α 资格的 u-event，否则，I 为非 α 资格的 u-event，如果 $Pr(I)<\alpha$. 对给定 α，设 X_α，Y_α，Z_α 分别为解决树中的 s-event 集合、f-event 集合和非 α 资格的 u-event 集合，则定理 2 给出 q_{pr} 的上下界.

定理 2.　对 $0\leq\alpha\leq\beta<1$，有 $\sum_{I\in X_\beta}Pr(I)\leq\sum_{I\in X_\alpha}Pr(I)\leq q_{pr}\leq 1-\sum_{I\in Y_\alpha}Pr(I)\leq 1-\sum_{I\in Y_\beta}Pr(I)$.

证明.　设 $Tree_\alpha$ 和 $Tree_\beta$ 为对应于 α 和 β 的子解决树. 如果 $\alpha\leq\beta$，那么每个有 β 资格的 u-event 也是有 α 资格的 u-event. 因此每个有 β 资格的 u-event 可在 $Tree_\alpha$ 和 $Tree_\beta$ 被分成若干子事件. 然而，一个有 α 资格非 β 资格的 u-event 只能在 $Tree_\alpha$ 被划分而不能在 $Tree_\beta$ 被划分. 因而有 $X_\beta\subseteq X_\alpha$，$Y_\beta\subseteq Y_\alpha$，从而推出结果.　　证毕.

该定理表明，一个较小的 α 会使 q_{pr} 的上界和下界非常接近，也就会有非常好的过滤能力. 如果选择一系列的 α 满足 $1>\alpha_1>\alpha_2>\alpha_3>\cdots>0$，便可获得一系列递增的下界和递减的上界以保证精确的 q_{pr}. 得界以后再与 ε 比较给出查询结果. 这种方法避免了在解决树上对 u-event 事件的深度分解，并能快速地逼近准确值. 但欲应用定理 2，必须预先确定 α 的值，此种方法不是最佳的选择. 但我们可以用解决树中 s-event 和 f-event 的叠加概率作为界限，它的本质即是定理 2. 设 $Pr(I_s)$ 和 $Pr(I_f)$ 是当前从解决树第一层开始累加的 s-event 和 f-event 的概率. 根据定理 2 有 $Pr(I_s)\leq q_{pr}\leq 1-Pr(I_f)$. 根据查询结果的条件 $q_{pr}\geq\varepsilon$ 做如下测试：(1) 如果 $Pr(I_s)\geq\varepsilon$，则停止基本算法返回有效结果；(2) 如果 $1-Pr(I_f)\leq\varepsilon$，则也停止基本算法，并返回无效结果. 随着累加值的不断递增，就像选择一系列的 α 一样，可得越来越紧的界限，尽可能早地停止计算.

4.2　同构图缩减

如式(3)所示，任何事件 $I\wedge Event(e_1,e_2,\cdots,\overline{e_i})$ $(1\leq i\leq n)$ 可能是一个 u-event. 因此在基本算法中，如果在解决树的低层产生较多的 u-event，那么会产生对 u-event 的深度分解. 这样会导致整个解决树的结点数量以指数增长，从而产生巨大的计算开销. 幸运的是，我们发现在一系列相邻的 u-event 中存在一种结构关系，通过这种关系，能够大幅度地缩减结点的数量，以减少计算代价. 本小节给出该方法的具体实现.

对一 u-event I，设 $E(I)=E_s(I)\cup E_f(I)$，$G\backslash E(I)$ 为 G 删除边集 $E(I)$ 后得到的子图，$Pr(G\backslash E(I))|_{con}$ 为 $G\backslash E(I)$ 中 s 到 d 可达概率. 从基本算法可得，事件 $Event(E(I))$ 对 q_{pr} 的贡献概率为 $Pr(E(I))\cdot Pr(G\backslash E(I))|_{con}$，其中 $Pr(E(I))=Pr(E_s(I))\cdot Pr(E_f(I))$. 例如在图 2，不确定事件 $Event(\overline{e_1})$ 的贡献概率是 $Pr(\overline{e_1})\cdot Pr(G\backslash e_1)|_{con}$，并且以 $(\overline{e_1})$ 为根的子树给出了 $Pr(G\backslash e_1)|_{con}$ 的计算过程.

设 $S=G\backslash E(I)$，$S\backslash e$ 表示从 S 中删除边 e. 又设 $S\lhd e$ 表示从 S 缩 e，即从 S 中移除 e，并将关联 e 两顶点的边关联到一个顶点上. 定理 3 给出关于图 $S=G\backslash E(I)$ 的一重要性质.

定理 3.

$$Pr(S)|_{con}=Pr(e)Pr(S\lhd e)|_{con}+Pr(\bar{e})Pr(S\backslash e)|_{con} \tag{4}$$

证明.　设 S_{con} 表示 S 中 s 到 d 连通的事件，由全概率公式可得

$Pr(S)|_{con}=Pr(e)Pr(S_{con}|e)|_{con}+Pr(\bar{e})Pr(S_{con}|e)$，其中 $Pr(\cdot|\cdot)$ 表示条件概率.

易知 $Pr(S_{con}|\bar{e})$ 等于 $Pr(S\backslash e)|_{con}$，而 $Pr(S_{con}|e)$ 表示边 e 已存在时对 S_{con} 的影响，因此可以缩减 e，并且有 $Pr(S_{con}|\bar{e})$ 等于 $Pr(S\lhd e)|_{con}$.　　证毕.

令 $I\wedge Event(e_1e_2\cdots\overline{e_i})=I_i$ 和 $I\wedge Event(e_1e_2\cdots\overline{e_{i+1}})=I_{i+1}$ 是解决树中两个相邻的 u-event，并令 F_i 和 F_{i+1} 分别表示 $G\backslash E(I_i)$ 和 $G\backslash E(I_{i+1})$ 得到的子图. 则可得

$$\begin{aligned}q_{pr}&=Pr(I_i)Pr(F_i)|_{con}+Pr(I_{i+1})Pr(F_{i+1})|_{con}\\&=Pr(I_i)[Pr(e_{i+1})Pr(F_i\lhd e_{i+1})|_{con}+\\&\quad Pr(\overline{e_{i+1}})Pr(F_i\backslash e_{i+1})|_{con}]+\\&\quad Pr(I_{i+1})Pr(F_{i+1})|_{con}.\end{aligned}$$

对于两个相邻的 u-event，F_i 和 F_{i+1} 有相同的顶点，并且 F_i 仅比 F_{i+1} 多边 e_{i+1}，因此，$F_{i+1}\backslash e_{i+1}$ 和

F_{i+1}是相同(同构)的图①,式(4)可重写为

$$q_{pr}=Pr(I_i)Pr(F_i)|_{con}+Pr(I_{i+1})Pr(F_{i+1})|_{con}$$
$$=Pr(I_i)Pr(e_{i+1})Pr(F_i\lhd e_{i+1})|_{con}+$$
$$Pr(F_{i+1})|_{con}[Pr(I_i)Pr(\overline{e_{i+1}})+Pr(I_{i+1})] \quad (5)$$

式(5)给出的结果是合并同构的图,并产生两个新事件,而其中一个事件对应于更小的图.对于解决树中一系列的 u-event,用式(5)分解它们,这样会产生大量同构的且规模小的图.合并同构图,直到没有相邻 u-event 为止.最后得到的新事件个数与被分解的 u-event 的个数相同,而基本算法会产生指数级的新事件.这说明通过缩减同构图后,树节点的数量会大幅度地减少,从而大大地降低了计算代价.

图 3 给出图 2 解决树中两个 u-event $Event(\overline{e_1})$和 $Event(e_1\overline{e_4})$分解和合并的过程.其中把两个同构图(标注在矩形中)缩减成一个图.从图中可见,两个 u-event 产生对应于小图的两个新事件,且这些事件都是成功的,因此可直接计算出结果.给出的结果 0.2376 和图 2 中基本算法求得的结果 0.027+0.0162+0.072+0.1224=0.2376 是一样的.然而基本算法又产生两个 u-event,还得继续分解下去.这个例子充分地说明了缩减同构图对降低计算代价的重要性.

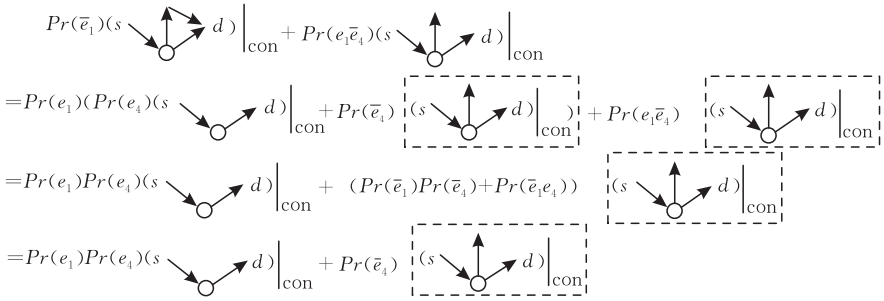


图 3 对图 2 中不确定事件 $Event(\bar{e}_1)$和 $Event(e_1\bar{e}_4)$的缩减

4.3 不相交路径/割集界

4.1 节给出的 u-event 界是在解决树中不断对 s-event和 f-event 概率的累加.但只是"加"的操作,而"乘"的操作会快得多,会使界限收敛得更快.本小结便给出通过"乘"操作获得的上下界.

设图 G 中 s 到 d 共有 m 条 s—d 路径,又设 I_1,I_2,…,I_m为这 m 条路径连通的事件.那么易知

$$q_{pr}=Pr(I_1\vee I_2\vee\cdots\vee I_m)=1-Pr(\overline{I_1}\wedge\overline{I_2}\wedge\cdots\wedge\overline{I_m}) \quad (6)$$

由于 m 条路径中有很多路径相互交叠,因此事件 $I_1,I_2,\cdots,I_m$互相不独立,那么 $Pr(\overline{I_1}\wedge\overline{I_2}\wedge\cdots\wedge\overline{I_m})\neq\prod_{i=1}^{m}[1-Pr(I_i)]$.其实该不等式也就是计算 q_{pr} 是困难的本质.但如果能找到不相交路径的集合,那么这些路径对应的事件便是独立的.设有 l 条不相交的路径,那么式(6)便可写成

$$\begin{aligned}q_{pr}&=1-Pr(\overline{I_1}\wedge\overline{I_2}\wedge\cdots\wedge\overline{I_m})\\&\geq 1-Pr(\overline{I_1}\wedge\overline{I_2}\wedge\cdots\wedge\overline{I_f})\\&=1-\prod_{i=1}^{l}\Big(1-\prod_{e\in P_i}Pr(e)\Big)\\&=LowerB\end{aligned} \quad (7)$$

上式给出一 q_{pr} 的下界 $LowerB$.如果此下界不小于阈值 ε,则即可返回有效结果.类似地,可得 q_{pr} 的上界 $UpperB$:

$$q_{pr}\leq 1-\prod_{i=1}^{k}\Big(1-\prod_{e\in C_i}Pr(e)\Big)=UpperB \quad (8)$$

其中 $C_i(1\leq i\leq k)$为 G 的 k 个不相交的割.如果此上界小于 ε,那么即可返回无效结果.

从式(7)和式(8)可见,如果 l 和 k 越大,界限越紧.因此我们欲求最大的 l 和 k 值,即求出图 G 中 s 到 d 的最大不相交路径数和割数.这里应用网络流理论中的两个定理[12].

引理 2. 最大不相交 s—d 路径数等于最小 s—d 割的基数.

引理 3. 最大不相交 s—d 割数等于最短 s—d 路径的长度.

这里使用文献[12]中的算法求解最大的 l 和 k(设为 l_{max}和 k_{max}),该算法可在多项时间内完成,因此可快速地求出紧的解.例如在图 1 中,最大的不相交路径集和割集分别为{{e_1,e_4},{e_2,e_5}}和{{e_1,e_2},

① 这里不需要做同构测试,通过分析显然会产生同构的图.

$\{e_4,e_5\}\}$. 因此下界是 $LowerB=1-(1-P_1P_4)(1-P_2P_5)=1-(1-0.85\times0.8)(1-0.8\times0.9)=0.9104$，上界是 $UpperB=1-(1-P_2P_5)(1-P_1P_4)=0.9506$. 即 $(LowerB=0.9104)\leq(q_{pr}=0.9176)\leq(UpperB=0.9506)$.

用 $l_{\max}$ 和 $k_{\max}$ 得到相当客观的界限，但每个路径和割都有一个权值，即其存在概率，因此权值也会影响界限的大小. 欲产生更好的界限，在保持求得 $l_{\max}$ 和 $k_{\max}$ 的同时，尽可能地选择权值大的集合. 首先给出选取权值最大的 $l_{\max}$ 路径集：给每边 $e\in G$ 一代价 $-\log_2 p_e$，其中 p_e 为 e 的存在概率，因此一条代价为 v 的路径的存在概率为 2^{-v}. 易得，如果代价越小，此路径的存在概率越大. 从而选择最大权值的路径集问题可转化成求解具有最大代价的 $l_{\max}$ 路径集. 本文通过简单地修改网络流算法[12]来求解此问题，即原来的算法每步都是寻找一“增广路径”，这里变成寻找“代价最小的增广路径”，而代价最小的路径可通过最短路径算法求得[12]. 类似地也可求得存在概率最大的 $k_{\max}$ 割集.

欲在基本算法中应用“不相交路径/割集界”，这里对基本算法做如下改动：因为解决树中每个 u-event的结点都对应于 G 的一个子图，便可对该子图应用“不相交路径/割集界”，然后和 ε 比较，如果满足判断条件，便可停止基本算法返回查询结果.

5　合并算法

上节给出的 3 种改进算法是互相独立的，即分别是对基本算法的优化. 如果将 3 种方法同时作用于基本算法，那么将对计算的加速有“质”的提高. 本节给出基本算法和 3 种改进方法的合并算法.

算法 2 给出合并算法的详细步骤. 此算法主要分为 3 个步骤. 第 1 步(6～10 行)：在基本算法每产生一个 u-event 后，便对此 u-event 应用“不相交路径/割集界”，如果满足条件，就返回结果. 第 2 步(11～12 行)：如果第 1 步不能停止计算，就开始分解连续的 u-event，并合并同构图. 第 3 步(13～31 行)：对第 2 步输出的 u-event 继续用式(3)分解，但此时开始不断地积累“不确定事件界”尽早地结束迭代.

算法 2.　合并算法.

输入：概率图 G，s 和 d，阈值 ε

输出：有效或无效的查询结果

算法描述：

1. $S=\{\Omega,\Phi,0,0\}$; //为整个 G 的事件空间 Ω 初始化一集合
2. $LBU=0$, $UBU=1$; //初始化 u-event 的下界和上界
3. $q_{pr}=0$; //初始化可达概率
4. while($S\neq\varnothing$){
5. 根据集合 S 中的索引 i，获得一事件 I^i；
   //起始的 I^i 即是 Ω.
6. 获取不相交路径集下界 $LBPath$ 和割集上界 $UBCut$；
7. if ($LBPath+q_{pr}\geq\varepsilon$)
8. return “两顶点能够以概率 ε 连通”；
9. if ($UBCut+q_{pr}<\varepsilon$)
10. return “两顶点不能以概率 ε 连通”；
11. if (从 I^i 开始有连续的 u-event)
12. 用式(5)分解这些事件并合并同构图，直到没有连续的 u-event；
13. if ($i<n_i$)
14. $i=i+1$；
15. else 集合 S 被舍弃；
16. 在事件 I^i 下找到一 s—d 路径 $I^i_{\min}(s—d)$；
17. if (Event($I^i_{\min}(s—d)$)==$\varnothing$){
18. $UBU=UBU-Pr(I^i)$；
19. 删除 I^i；//此时事件 I^i 是失败的.}
20. else if (I^i 是成功的){
21. $q_{pr}=q_{pr}+Pr(I^i)$；
22. $LBU=LBU+Pr(I^i)$;}
23. else{
24. $q_{pr}=q_{pr}+Pr(I^i\wedge Event(I^i_{\min}(s—d)))$;
   //$I^i\wedge Event(I^i_{\min}(s—d))$是成功事件.
25. $LBU=LBU+Pr(I^i\wedge Event(I^i_{\min}(s—d)))$;}
26. 将一个新的集合$\{I^i,I^i_{\min}(s—d),n_i,i=1\}$放入 S；
27. if ($LBU\geq\varepsilon$)
28. return “两顶点能够以概率 ε 连通”；
29. if ($UBU<\varepsilon$)
30. return “两顶点不能以概率 ε 连通”；
31. } //while 结束.

通过第 4 节的分析，“不相交路径/割集界”具有最强的过滤能力(实验也给出验证)，因此把此过滤步骤放在第 1 步尽可能地缩减计算空间. “不确定事件界”是依赖于解决树结点数目的，因此如果对第 1 步后剩余的计算空间(其实该剩余空间已很小)直接用“不确定事件界”，很有可能由于有很多的树节点使过滤效果不好. 但算法 2 中先用“同构图缩减”，这会使树的宽度不再增加，导致计算空间已是节点数目线性的阶. 之后在对宽度很小的树应用“不确定事件界”，这样会最大限度地发挥“不确定事件界”的过滤能力. 实验说明这样组合的“合并算法”有相当强的过滤能力.

6　性能分析

本节用真实和合成的概率图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

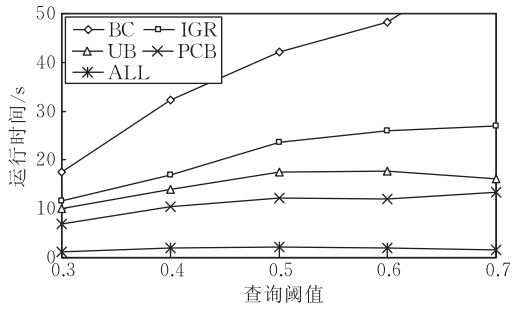
奔腾 4 3.0GHz CPU,2GHz 内存和 160GB 硬盘.对本文给出的算法,BC 表示“基本算法”,IGR 表示“同构图缩减”算法,UB 表示“不确定事件界”算法,PCB 表示“不相交路径/割集界”算法,ALL 表示“合并算法”.真实概率图数据采用 Entrez gene 6091-OMIM 127700 数据库①.它是描述人类基因网络的数据库,其中网络节点代表“显性基因”,边被赋予小数权值以表示基因相互作用的大小.使用该数据库的 5 个基因网络作为概率图数据,表 1 给出它们的参数.

表 1 真实概率图数据的参数

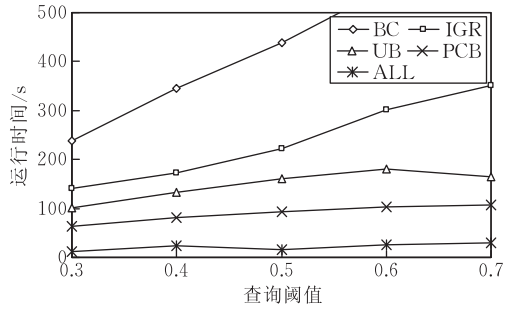
图数据编号	节点数量	边集大小	边的平均概率
*Dys*1	452	1113	0.423
*Dys*2	1775	4163	0.396
*Dys*3	3259	8790	0.212
*Dys*4	6786	14056	0.237
*Dys*5	11368	32754	0.311

同时从 Citeseer② 中抽取图数据($|V|=12140$, $|E|=36874$),并按正态分布 $N(\mu,\delta)$为每边产生概率.每次实验时,随机地产生 100 个查询,记录下查询的平均代价.

首先给出对真实概率图数据的实验结果.这里选取小图(*Dys*1)和大图(*Dys*2)两个具有代表性的数据,给出在不同阈值下的结果.图 4 给出阈值是 0.3～0.7 的运行时间.从图中可见,所有的曲线都呈上升趋势,这是因为越高的阈值需要需要枚举越多的 s—d 路径.很明显,3 种改进算法都比 BC 快得多,并且 PCB 是最高效的改进算法.这一点验证了对 3 种算法的设计与分析.而 ALL 算法的运行效率比其它任何方法都要高出一块.例如,对 *Dys*1 的平均运行时间 ALL 只需 1s 多,即使对边数超过 3 万的 *Dys*5 平均也不超过 15s.



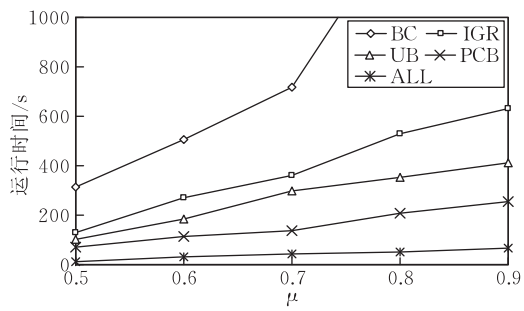
(a) *Dys*1图数据



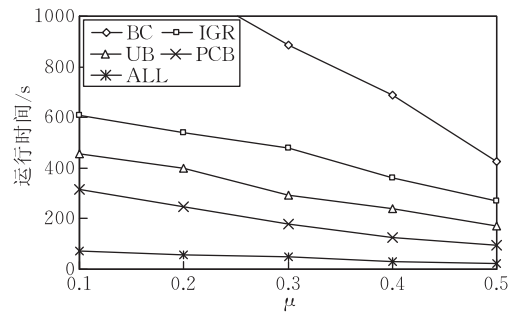
(b) *Dys*5图数据

图 4 真实数据下算法的运行时间

其次给出对合成概率数据的实验结果.因为此图边的存在概率是合成的,故可以测试不同概率分布下的运行效率.这里通过改变 $N(\mu,\sigma)$分布中的 μ 和 σ 来实现.具体地,μ 取 0.5～0.9,默认值是 0.7;σ 取 0.1～0.5,默认值是 0.4.图 5 给出测试结果.如图5(a)所示,随着μ的增加,所有算法的运行时间都在增加.尽管 μ 的增加导致可达概率指数级的增长,但除了 BC,其它算法都有较好的可扩展性.类似的结果也在图 5(b)中给出,其中 3 种改进方法和合并算法的曲线都缓慢地下降.这说明改进算法的有效性.ALL 在图 5 中的两个实验都表现出极高的运行效率.



(a) 不同的均值



(b) 不同的方差

图 5 合成数据下算法的运行时间

最后用真实数据测试算法的可扩展性.图 6 给出测试结果,其中横坐标是 *Dys*1～*Dys*5,纵坐标是运行时间.如图所示,所有曲线都随图规模的增加而增长.其中BC增长的最快,到*Dys*4时就已经超过

① http://www.ncbi.nlm.nih.gov/omim
② http://citeseer.ist.psu.edu/oai.html

300s，其曲线增长的趋势呈近似指数级的增长. 而其它算法都可避免此种爆炸性的增长，具有较好的可扩展性. 尤其 ALL 有非常快的速度，即使对边数量超过 3 万的图 $Dys5$ 求解 NP 难查询也可在 20s 内完成.

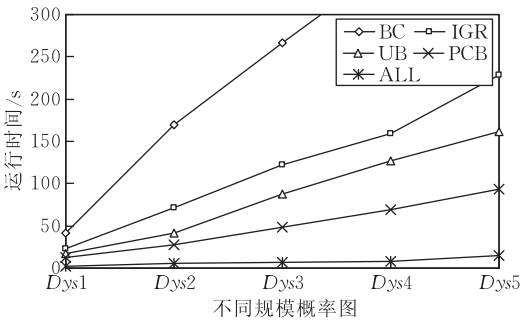


图 6　不同数据规模下算法的可扩展性

7　结　语

已有很多面向确定图的可达查询工作. 文献[15]用图的传递闭包来压缩原始图，并在求解过程中把问题转化为网络流问题. 文献[16]把图分成若干有向链，并每个顶点都记录了与其相邻的链，从而可常数时间求解查询. 文献[17]首先提出树覆盖的方法，并以树为单元对图进行压缩，并证明此种压缩是保证可以求解查询的最优压缩. 文献[18]和文献[19]都是在索引构建时间上改进了树覆盖方法. 文献[10]拓展了有向链的方法，把问题转换成平面图的问题，并只需 2 跳即可完成查询.

面向概率数据库的研究是现在的热点. 早期工作的重点是在概率关系数据库上如文献[8-9,20]，即处理概率 SQL 查询. 之后研究者提出了一些查询类型及其处理方法，主要包括 Top-k 查询[21] 和 Skyline 查询等[22]. 近年来开始关注结构化数据如概率 XML 数据[1-2,13-14]. 就图数据库而言，邹等人[23]研究了不确定图的频繁子图挖掘，而张等人[24]研究了带索引的不确定图 Top-k 查询.

本文是第一个面向概率图研究可达查询的工作. 采用细粒度的可能世界模型定义概率图，从而使可达查询具有丰富的概率语义. 在给出问题是 $\sharp P$ 难后，本文首先设计了一个基本算法可精确地给出查询结果. 在用解决树重新描述基本算法后，用不确定事件界对解决树进行了剪枝. 用同构图的缩减可以使基本算法只对图中相同的元素进行一次计算，从而降低计算空间. 用不相交路径/割集界可使计算快速地收敛. 最后所有算法的合并在实验中显示出相当高的运行效率，说明本文开发的算法具有实际应用的价值.

参　考　文　献

[1] Nierman A, Jagadish H V. ProTDB: Probabilistic data in XML//Proceedings of the VLDB. Hong Kong, China, 2002: 646-657

[2] Senellart P, Abiteboul S. On the complexity of managing probabilistic XML data//Proceedings of the PODS. Beijing, China, 2007: 283-289

[3] Haase P, Volker J. Ontology learning and reasoning-dealing with uncertainty and inconsistency//Proceeding of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW). New York, USA, 2005: 45-55

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

[5] Chui H N, Sung W K, Wong L. Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions. Bioinformatics, 2007, 22(13): 47-58

[6] Jiang R, Tu Z, Chen T et al. Network motif identification in stochastic networks. PNAS, 2006, 103(25): 9404-9409

[7] Saito R, Suzuki H, Hayashizaki Y. Interaction generality: A measurement to assess the reliability of a protein-protein interaction. Nucleic Acids Research, 2002, 30(5): 1163-1168

[8] Dalvi N N, Suciu D. Management of probabilistic data: Foundations and challenges//Proceedings of the PODS. Beijing, China, 2007: 1-12

[9] Khoussainova N, Balazinska M. Towards correcting input data errors probabilistically using integrity constraints//Proceedings of the MobiDE Workshop. Chicago, Illinois, USA, 2006: 43-50

[10] Jin R, Xiang Y, Ruan N. Efficiently answering reachability queries on very large directed graphs//Proceedings of the SIGMOD. Vancouver, Canada, 2008: 595-608

[11] Yuan Ye, Wang Guo-Ren. Answering probabilistic reachability queries over uncertain graphs. Chinese Journal of Computers, 2010, 33(8): 1378-1386(in Chinese)
(袁野，王国仁. 面向不确定图的概率可达查询. 计算机学报，2010，33(8)：1378-1386)

[12] Cormen T H, Leiserson C E, Rivest R L et al. Introduction to Algorithms. New York: The MIT Press/McGraw-Hill Book Company, 2001

[13] Nierman A, Jagadish H V. ProTDB: Probabilistic data in XML//Proceedings of the VLDB. Hong Kong, China, 2002

[14] Senellart P, Abiteboul S. On the complexity of managing probabilistic XML Data//Proceedings of PODS. Beijing, China, 2007: 105-114

[15] Jagadish H V. A compression technique to materialize transitive closure. TODS, 1990, 15(4): 558-598

[16] Cheng J, Yu J, Lin X. Fast computing reachability labelings for large graphs with high compression rate//Proceedings of the EDBT. Nantes, France, 2008: 193-204

[17] Agrawal R, Borgida A, Jagadish H V. Efficient management of transitive relationships in large data and knowledge bases. ACM SIGMOD Record, 1989, 18(2): 253-262

[18] Chen L, Gupta A, Kurul M. Stack-based algorithms for pattern matching on dags//Proceedings of the VLDB. Trondheim, Norway, 2005: 493-504

[19] Tribl S, Leser U. Fast and practical indexing and querying of very large graphs//Proceedings of the SIGMOD. Beijing, China, 2007: 845-856

[20] Benjelloun O, Sarma A D, Hayworth C. An introduction to ULDBs and the Trio system. IEEE Data Engineering Bulletin, 2006, 29(1): 5-16

[21] Soliman M A, Ilyas I F, Chang K C. Top-*k* query processing in uncertain databases//Proceedings of the ICDE. Istanbul, 2007: 896-905

[22] Pei J, Jiang B, Lin X. Probabilistic skylines on uncertain data//Proceedings of the VLDB. Vienna, Austria, 2007: 15-26

[23] Zou Zhao-Nian, Li Jian-Zhong, Gao Hong, Zhang Shuo. Ming frequent subgraph patterns from uncertain graphs. Journal of Software, 2009, 20(11): 2965-2976(in Chinese)
(邹兆年, 李建中, 高宏, 张硕. 从不确定图中挖掘频繁子图模式. 软件学报, 2009, 20(11): 2965-2976)

[24] Zhang Shuo, Gao Hong, Li Jian-Zhong, Zou Zhao-Nian. Efficient query processing on uncertain graph databases. Chinese Journal of Computers, 2009, 32(10): 2066-2079(in Chinese)
(张硕, 高宏, 李建中, 邹兆年. 不确定图数据库中高效查询处理. 计算机学报, 2009, 32(10): 2066-2079)

YUNA Ye, born in 1981, Ph. D. candidate. His research interests include graph database, probabilistic database, P2P data management, and data piracy.

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

Background

Efficiently answering reachability queries against very large graphs is becoming an increasingly important research topic driven by many emerging real world applications, such as biological networks, social networks, ontologies, XML and RDF databases. However, data extracted from those applications is inherently uncertain due to noise, incompleteness and inaccuracy, and many works have been proposed to study uncertain RDF and XML databases. Therefore, it is important and necessary to efficiently process reachability queries over graph data with uncertainty.

The topics on managing uncertain data are very hot nowadays, and there have been huge number of works on this topic. Initial works have focused on how to store and process uncertain data within database systems, and thus how to answer SQL-style queries. Subsequently, there has been a growing realization that in addition to storing and processing uncertain data such as KNN query, range query, top-k query and skyline query. There also exists several advanced algorithms to analyze uncertain data, e. g. , clustering uncertain data and finding frequent items within uncertain data. For uncertain graph databases, the existing works have proposed algorithms for mining frequent subgraphs patterns and finding top-k patterns from uncertain graphs. This paper focuses on processing uncertain reachability queries that are not only common on uncertain graph databases, but also serve as fundamental operations for many other uncertain graph queries.

This research was supported by the National Natural Science Foundation of China (Grant No. 60773221), the 863 Program (Grant No. 2009AA01Z150), and National Basic Research Program of China (Grant No. 60933001).