不确定图数据库中高效查询处理

张 硕 高 宏 李建中 邹兆年
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 近年来,在多种领域中产生的大量数据都可以自然地建模为图结构,比如蛋白质交互网络、社会网络等.测量手段的不准确性以及数据本身的性质导致不确定性在很多图数据中普遍存在.文中研究不确定图数据库中的高效查询处理方法.首先给出一种数据模型来表示图的不确定性.鉴于对用户提交的查询图通常会产生大量匹配结果,高效得到概率最大的 k 个匹配常常更具有现实意义.因此文中形式化提出概率 top-k 子图匹配查询的问题.为了解决提出的查询问题,以附带概率信息的邻居子图为基础,设计了一种有效的索引结构.另外,提出一种高效的基于索引的查询处理方法.该查询处理方法的核心是一个基于搜索树的匹配算法,其中运用了一种概率剪枝技术来提高性能.实验结果表明,所提出方法具有良好的效率和可扩展性.

关键词 不确定性;不确定图;top-k 查询;查询处理;图索引
中图法分类号 TP311 **DOI** 号: 10.3724/SP.J.1016.2009.02066

Efficient Query Processing on Uncertain Graph Databases

ZHANG Shuo GAO Hong LI Jian-Zhong ZOU Zhao-Nian
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract In recent years, lots of data in various domains have been naturally modeled by graphs, e.g. protein interaction networks, social networks, etc. Uncertainty is inherent in many of these graphs due to the imprecise characteristics of equipments or the nature of data. This paper addresses efficient query processing on uncertain graph databases. A data model is proposed for representing uncertainties in graphs, and a new formulation for probabilistic top-k subgraph matching query is presented. An effective index structure based on neighborhood subgraphs with probabilistic information in uncertain graphs in databases is devised. In addition, based on indexes, an efficient search-tree based algorithm with probabilistic pruning techniques is proposed to search large uncertain graphs. Experimental results show that the proposed algorithms are efficient and scalable.

Keywords uncertainty; uncertain graph; top-k query; query processing; graph indexing

1 引 言

近年来,在多种领域中产生的大量数据都可以自然地建模为图结构,比如蛋白质交互网络、分子化合物、社会网络等.对大量累积的图数据进行查询处理已成为一项重要研究课题.现已提出若干图数据的查询处理方法[1-4].这些方法全部针对确定图数据

收稿日期: 2009-07-15;最终修改稿收到日期: 2009-08-25.本课题得到国家"九七三"重点基础研究发展规划项目基金(2006CB303000)、国家自然科学基金重点项目(60533110)、国家自然科学基金(60773063)和国家自然科学基金委与香港研究资助局联合科研基金(60831160525)资助.张 硕,男,1982 年生,博士研究生,主要研究方向为图数据管理. E-mail: zhangshuocn@hit.edu.cn. 高 宏,女,1966 年生,教授,博士生导师,主要研究领域为并行数据库、数据仓库. 李建中,男,1950 年生,教授,博士生导师,主要研究领域为并行数据库、传感器网络. 邹兆年,男,1979 年生,博士研究生,主要研究方向为图数据挖掘.

(即完整且精确的图数据). 然而, 在许多实际应用中还存在大量的不确定图数据, 例如, 生物信息学中的蛋白质交互(Protein-Protein Interaction, PPI) 网络是一类不确定图, 其顶点表示蛋白质, 边表示蛋白质交互. 由于 PPI 实验检测方法的局限性, 很大一部分检测到的 PPI 是不确定的. 文献[5]提出一种 PPI 可靠性指数来衡量 PPI 真实存在的可能性. 图 1 给出了一个真实的 PPI 网络实例, 其中, 每个矩形表示一个顶点, 矩形内的文字为顶点对应蛋白质的名称, 矩形外附着的文字表示蛋白质的功能, 边上的数值表示 PPI 可靠性指数转换得到的存在可能性.

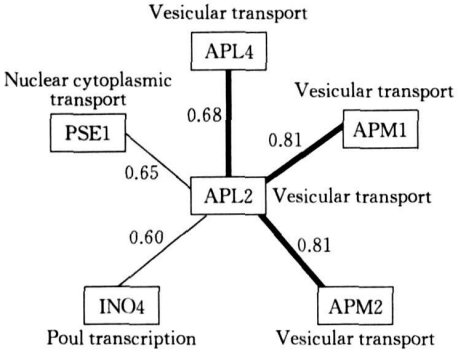


图 1　蛋白质交互网络实例

对不确定图数据库进行查询处理具有十分重要的实际意义. 例如, 生物学工作者通常对蛋白质功能之间联系的结构信息感兴趣. 具体地说, 在已知的一类生物过程中, 生物学工作者总结出了一些重要的蛋白质交互的结构模式, 这些模式表征一类生物过程所特有的性质和特征. 每个结构模式都是蛋白质功能之间联系的图, 即其中顶点表示蛋白质功能, 边表示功能之间的联系模式. 例如用一个结构模式 P 表示一个具有功能"Vesicular transport"的蛋白质交互于一个"Vesicular transport"功能的蛋白质和一个"nuclear cytoplasmic transport"功能的蛋白质. 对于最新获得的一个新的 PPI 网络, 生物学工作者需要在其上找出满足特定蛋白质功能联系的结构模式的蛋白质集合以及其间的交互信息. 例如, 生物学工作者希望在图 1 所示的 PPI 网络中找出满足上述结构模式 P 的蛋白质集合和其间的交互信息, 进而预测该 PPI 网络可能关联的生物过程. 需要注意的是, PPI 网络即图数据本身是不确定的. 不确定性成为评价查询结果的一项重要指标. 只有当某个查询结果出现的可能性较大时, 才被视为有用的结果. 也就是说, 相对于大量的所有可能结果, 用户常常希望找出可能性最大的前 k 个即可. 因此, 不确定图数据库上的概率 top-k 查询处理是一项具有现实意义的工作.

对于在大量累积的图数据(即图数据库) 上进行查询处理的问题, 近年来研究者们已提出若干方法[1-3]. 已有的对图数据库查询处理的研究工作主要包括两大类: 一类是在小尺寸图的大规模集合(简称为集合类图数据库) 上处理查询, 其中包括子图查询处理、超图查询处理和相关子图查询处理等; 一类是在一个或为数不多的大尺寸图(简称为大图类图数据库) 上处理查询, 如子树/子图匹配查询处理、可达查询和最短路径/距离查询等. 然而, 这些方法全部针对确定图数据, 无法应用到不确定图数据库的查询处理中.

最近几年, 关于不确定数据(或称不确定性数据) 的研究工作正在如火如荼地进行[6-8]. 在不确定数据库上, 研究者已提出了一些查询类型及其处理方法, 主要包括处理一般的关系查询[9]、top-k 查询[10]、k 最近邻查询、概率 Skyline 查询等. 然而, 据我们所知, 目前尚无从数据库角度研究不确定图数据库中查询处理的工作. 由于子图同构检测是一个 NP 完全问题[4], 对不确定图数据库中多种查询类型(如子图匹配查询) 的高效处理是一项具有挑战性的工作.

另外需要提及的是, 在不确定图数据挖掘方面, 研究者们现已开展少许工作, 主要包括在不确定图中计算最可靠子图[11-12] 以及在不确定图中挖掘频繁子图模式[13].

综上所述, 对不确定图数据库上查询处理方法的研究具有现实意义和理论价值, 有待探索.

本文研究在不确定的大图类图数据库中进行概率 top-k 子图匹配查询的处理问题. 主要内容如下:

(1) 基于给出的不确定图数据模型, 形式化地提出概率 top-k 子图匹配查询的问题.

(2) 以附带概率信息的邻居子图为基础, 设计一种有效的索引结构. 邻居子图的概念在文中第 3.2 节给出.

(3) 提出一种高效的基于搜索树的算法来进行查询处理. 其中运用了一种概率剪枝技术来提高性能.

(4) 通过实验考察并证实提出方法具有良好的效率和可扩展性.

本文第 2 节介绍相关术语并给出数据模型和问题定义; 第 3 节给出子图匹配概率的快速计算方法、索引结构和其构造方法以及查询处理算法; 第 4 节给出实验结果和分析; 第 5 节总结全文.

2 问题定义

2.1 数据模型

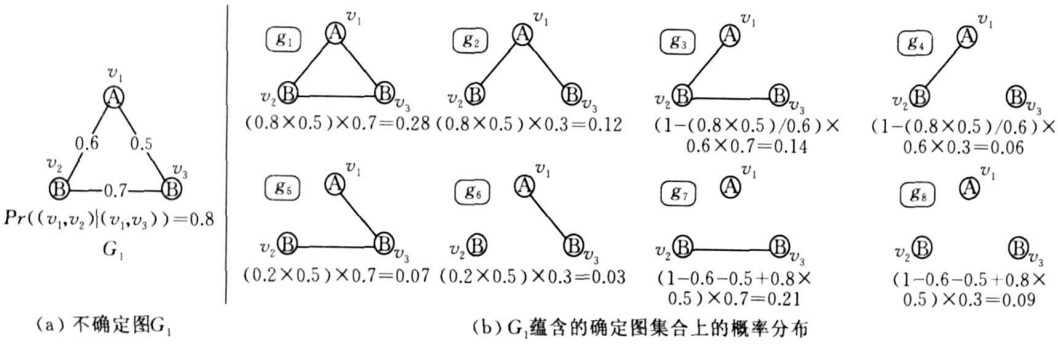
本文着重考虑无向带标签的简单图. 通过简单修改, 提出的查询处理方法也适用于有向、无标签的伪图. 一个(确定) 图 g 定义为一个四元组 $g=(V, E, \Sigma, l)$, 其中 V 是顶点集, $E \subseteq V \times V$ 是边集, Σ 是标签集, $l: V \rightarrow \Sigma$ 是为顶点分配标签的标签函数. 对于图 g 中一个顶点 v, 与 v 之间有边的顶点的集合记为 $Adj_g(v)$. 若无特别说明, 在后文中将确定图简记为图, 将 g 的顶点集、边集、标签集和标签函数分别记作 V_g, E_g, Σ_g 和 l_g. 定义图 g 是图 g' 的子图当且仅当 $V_g \subseteq V_{g'}$, $E_g \subseteq E_{g'}$, $\Sigma_g \subseteq \Sigma_{g'}$ 且 $l_g = l_{g'}|_V$, 其中 $l|_V$ 表示将函数 l 约束在 V 上得到的函数. 给定 $V' \subseteq V$, 由 V' 诱导的 g 的子图是 (V', E', Σ', l'), 记作 $g[V']$, 其中 $E' = E \cap (V' \times V')$, $\Sigma' = \{l(v') | v' \in V'\}$ 及 $l' = l|_{V'}$. 下面给出不确定图的定义.

定义 **1**. 一个不确定图定义为一个六元组 $G=(V, E, \Sigma, l, P, R)$, 其中 V 是顶点集, $E \subseteq V \times V$ 是边集, Σ 是标签集, $l: V \rightarrow \Sigma$ 是为顶点分配标签的标签函数, $P: E \rightarrow (0, 1]$ 是边的概率函数(称为成员概率函数), R 是一组条件概率的集合(称为生成规则集), 其满足

(1) R 中的所有条件概率具有形式: $Pr(e'_1 \wedge e'_2 \wedge \ldots \wedge e'_m | e'_{m+1})$, 其中 $Pr(A)$ 表示事件 A 的概率值(下同), $1 \le m \le |E| - 1$, 且对于 $\forall j$ $(1 \le j \le m+1)$ 有 $e'_j \in E$;

(2) 令E 是边集 E 的一个划分(划分的每个元素即 E 的子集 E_i 称为一个划分块), 其满足如果在 R 中存在条件概率 $Pr(e'_1 \wedge e'_2 \wedge \ldots \wedge e'_m | e'_{m+1})$, 那么 $e'_1, e'_2, \ldots, e'_m, e'_{m+1}$ 在一个划分块中. 则对于E 中每一个划分块 E_i, 在 R 中有且仅有 $2^{|E_i|} - |E_i| - 1$ 个关于 E_i 中边的条件概率; 且在这些条件概率中, 对于 $\forall j$ $(1 \le j \le |E_i| - 1)$ 有且仅有 $C^{j+1}_{|E_i|}$ 个 $Pr(e''_1 \wedge e''_2 \wedge \ldots \wedge e''_j | e''_{j+1})$ 形式的条件概率, 而 R 中任意两个条件概率 $Pr(e''_1 \wedge e''_2 \wedge \ldots \wedge e''_j | e''_{j+1})$ 和 $Pr(e_{\ominus} \wedge e_{\ominus} \wedge \ldots \wedge e_{\ominus} | e_{\ominus})$ 满足 $\{e''_1, e''_2, \ldots, e''_j, e''_{j+1}\} \neq \{e_{\ominus}, e_{\ominus}, \ldots, e_{\ominus}, e_{\ominus}\}$.

例 **1**. 图 2(a) 示例了一个不确定图 G_1. 圆圈表示顶点, 圆圈间的连线表示边. 顶点旁边的文字(如"v_1")表示顶点 ID, 顶点内部的文字(如"A")表示顶点的标签. 边上的文字(如"0.6")表示对应边的成员概率. "$Pr((v_1, v_2) | (v_1, v_3)) = 0.8$"是此例的生成规则中唯一的条件概率. 该条件概率显然满足定义 1 中条件(1). 参考定义 1 条件(2), 边集 E_{G_1} 的划分E= $\{\{(v_1, v_2), (v_1, v_3)\}, \{(v_2, v_3)\}\}$. 对于划分块 $\{(v_1, v_2), (v_1, v_3)\}$, 有且仅有 $2^2 - 2 - 1 = 1$ 个条件概率, 且对于 $j=1$ 有且仅有 $C^{j+1}_{|E_i|} = C^{1+1}_2 = 1$ 个 $Pr(e''_1, e''_2, \ldots, e''_j | e''_{j+1})$ 形式的条件概率即"$Pr((v_1, v_2) | (v_1, v_3))$". 因此满足定义 1 条件(2).



(a) 不确定图 G_1 (b) G_1 蕴含的确定图集合上的概率分布

图 2 不确定图 G_1 及其蕴含的确定图

(不) 确定图的每一个顶点都有一个唯一的编号, 即顶点的 ID, 同时这些顶点 ID 间有一个序. 这样每一个顶点可由其 ID 唯一地标识. 类似地, 为了便于处理, 每个(不) 确定图的标签集上有一个序. 边的成员概率表示边在其两个端点之间实际存在的可能性, 生成规则中的条件概率表示多个边存在的可能性之间的相关性. 我们将不关联于生成规则中任何条件概率的边称为独立边. 类似于不确定关系数据库模型中的可能世界语义[9-10], 一个不确定图 G 的状态是不确定的, 即其具有多种可能的存在形式(即可能世界), 每个存在形式(可能世界) 是一个确定图, 同时每个形式关联着一定的概率. 我们将一个不确定图 G 的所有概率非零的存在形式(确定图) 的集合称为 G 所蕴含的确定图集合, 记为 $Imp(G)$. 换言之, 如果一个确定图 g 是不确定图 G 的一个概率非零的存在形式, 即 $g \in Imp(G)$, 则称 G 蕴含 g.

记作 $G \Rightarrow g$. 对于 $\forall g \in Imp(G)$, 有 $V_g \subseteq V_G$, $E_g \subseteq E_G$, $\Sigma_g \subseteq \Sigma_G$ 和 $l_g = l_G|_{V_g}$. G 蕴含 g 的概率记为 $\underline{P}(G \Rightarrow g)$. 显然, 类似于不确定关系数据库模型, $\underline{P}(G \Rightarrow g)$ 定义了样本空间 $Imp(G)$ 上的一个概率分布. 于是, 不确定图 G 蕴含确定图 g 的概率 $\underline{P}(G \Rightarrow g)$ 等于

$$\underline{P}(G \Rightarrow g) = Pr\left(\left(\bigwedge_{e \in E_g}(e)\right) \wedge \left(\bigwedge_{e' \in E_G - E_g}(\emptyset e')\right)\right) \quad (1)$$

不确定图 G 蕴含确定图 g 的概率 $\underline{P}(G \Rightarrow g)$ 由 G 中边的成员概率和生成规则中的条件概率共同约束.

例 **2**.　图 2(b)给出了图 2(a)中不确定图 G_1 蕴含的全部确定图 $g_1, g_2, \cdots, g_8$ 及其被 G_1 蕴含的概率值. 以 g_2 为例, 因为生成规则中的条件概率 "$Pr((v_1, v_2)|(v_1, v_3)) = 0.8$" 以及边 (v_1, v_3) 的成员概率为 0.5, 故 $Pr((v_1, v_2) \wedge (v_1, v_3)) = 0.8 \times 0.5$. 独立边 (v_2, v_3) 的成员概率为 0.7, 故边 (v_2, v_3) 不出现的概率为 $1 - 0.7 = 0.3$. 因此 G_1 蕴含 g_2 的概率 $\underline{P}(G_1 \Rightarrow g_2)$ 为 $Pr((v_1, v_2) \wedge (v_1, v_3) \wedge \emptyset(v_2, v_3)) = Pr((v_1, v_2) \wedge (v_1, v_3)) \times Pr(\emptyset(v_2, v_3)) = (0.8 \times 0.5) \times 0.3 = 0.12$. 再以 g_7 为例, 生成规则中条件概率 "$Pr((v_1, v_2)|(v_1, v_3)) = 0.8$" 及边 (v_1, v_3) 成员概率 0.5 可得 $Pr((v_1, v_2) \wedge (v_1, v_3)) = 0.8 \times 0.5$. 于是, $Pr(\emptyset(v_1, v_2) \wedge \emptyset(v_1, v_3)) = 1 - (Pr((v_1, v_2)) + Pr((v_1, v_3)) - Pr((v_1, v_2) \wedge (v_1, v_3))) = 1 - (0.6 + 0.5 - 0.8 \times 0.5)$. 独立边 (v_2, v_3) 的成员概率为 0.7. 因此 G_1 蕴含 g_7 的概率 $\underline{P}(G_1 \Rightarrow g_7)$ 为 $Pr(\emptyset(v_1, v_2) \wedge \emptyset(v_1, v_3) \wedge (v_2, v_3)) = Pr(\emptyset(v_1, v_2) \wedge \emptyset(v_1, v_3)) \times Pr((v_2, v_3)) = (1 - (0.6 + 0.5 - 0.8 \times 0.5)) \times 0.7 = 0.21$. G_1 蕴含的全部确定图的概率之和为 $0.28 + 0.12 + 0.14 + 0.06 + 0.07 + 0.03 + 0.21 + 0.09 = 1$.

定义 **2**.　对于两个确定图 $g = (V, E, \Sigma, l)$ 和 $g' = (V', E', \Sigma', l')$, 一个从 g 到 g' 的子图同构定义为一个单射函数 $f: V \to V'$, 满足: (1) $\forall v \in V$, $l(v) = l'(f(v))$; (2) $\forall (u, v) \in E$, $(f(u), f(v)) \in E'$. 如果存在一个从 g 到 g' 的子图同构 f, 则称 g (相对于 f) 子图同构于 g', 记为 $g \subseteq^f g'$, 或简记为 $g \subseteq g'$, 也称 g' 包含 g. 如果 $g \subseteq^f g'$, 则在子图同构 f 下 g 在 g' 中的子图匹配定义为 g' 的子图 $(V'', E'', \Sigma'', l'')$, 记为 $M_{g(f)g'}$, 其中 $V'' = \{f(v) | v \in V\}$, $E'' = \{(f(u), f(v)) | (u, v) \in E\}$, $\Sigma'' = \{l'(f(v)) | v \in V\}$, $l'' = l'|_{V''}$. 顶点 $f(v)$ 称为在子图同构 f 下 v 的匹配顶点, 或称在子图同构 f 下顶点 v 匹配于顶点 $f(v)$.

如果两个确定图 g 和 g' 满足 $g \subseteq g'$ 且 $|V_g| \neq |V_{g'}|$, 则称 g 真子图同构于 g', 记为 $g \subset g'$; 如果 g 和 g' 满足 $g \subseteq g'$ 且 $|V_g| = |V_{g'}|$, 则称 g 同构于 g'. 对于一个确定图集合 $SET = \{g_1, g_2, \cdots, g_n\}$ 和一个确定图 g, g 在图集 SET 中的支持集, 记为 $sup_{SET}(g)$, 定义为所有包含 g 的 SET 中图的集合, 即 $sup_{SET}(g) = \{g_i | g_i \in SET, g \subseteq g_i\}$. $|sup_{SET}(g)|$ 称为 g 在 SET 中的支持度. 对于一个用户给定的最小支持度阈值 σ_{min} ($1 \leq \sigma_{min} \leq |SET|$), 定义 g 在 SET 中是频繁的, 若 $|sup_{SET}(g)| \geq \sigma_{min}$. 将只含有一条边的确定图称为一边(确定)图. 设该边两个顶点的标签为 L_1 和 L_2, 则该一边图表示为 L_1—L_2, 其满足按照在该图的标签集上的序 L_1 在 L_2 之前. 两个一边图 L_1—L_2 和 L'_1—L'_2 同构, 当且仅当 $L_1 = L'_1$ 且 $L_2 = L'_2$.

定义 **3**.　对于一个确定图 $q = (V_q, E_q, \Sigma_q, l_q)$ 和一个不确定图 $G = (V, E, \Sigma, l, P, R)$, 在子图同构 f 下 q 在 G 中的子图匹配, 记为 $M_{q(f)G}$, 定义为在子图同构 f 下 q 在 g 中的子图匹配 $M_{q(f)g}$, 满足 $\exists g \in Imp(G)$ 使得 $q \subseteq^f g$. 顶点 $f(v)$ 称为在子图同构 f 下 v 的匹配顶点, 或称在子图同构 f 下顶点 v 匹配于顶点 $f(v)$.

在子图同构 f 下 q 在 G 中的子图匹配 $M_{q(f)G}$ 的存在可能由任意一个 G 的蕴含图 g, 满足 $g \in Imp(G)$ 且 $q \subseteq^f g$, 导致. 因此, 子图匹配 $M_{q(f)G}$ 的概率等于所有满足 $g \in Imp(G)$ 且 $q \subseteq^f g$ 的确定 g 的概率之和, 即

$$\sum_{g \in Imp(G), q \subseteq^f g} \underline{P}(G \Rightarrow g) \quad (2)$$

因为不确定图蕴含其自身任意一个蕴含图的概率大于零, 因此任意一个子图匹配的概率均大于零.

例 **3**.　图 3 给出了一个查询图的例子 q_1. 考察图 2 中的不确定图 G_1 及其蕴含图集合. 用 $(u_1 \to v_1, u_2 \to v_2, u_3 \to v_3)$ 表示从 q_1 到 g_1 的一个子图同构函数 f. 在子图同构 f 下 q_1 在 g_1 中的子图匹配 $M_{q_1(f)g_1}$ 是 g_1 的子图 $(\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_3)\}, \{A, B\}, l_{g_1})$. 在 G_1 蕴含的全部确定图 g_i ($1 \leq i \leq 8$) 中, 只有 g_1 和 g_3 满足 $q_1 \subseteq^f g_i$. 因此, 在子图同构 f 下 q 在 G_1 中的子图匹配 $M_{q_1(f)G_1}$ 的概率等于 $\underline{P}(G_1 \Rightarrow g_1) + \underline{P}(G_1 \Rightarrow g_3) = 0.28 + 0.14 = 0.42$.

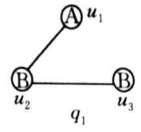


图 3　查询图 q_1

2.2 问题描述

本文研究在不确定的大图类图数据库中进行概率 top-k 子图匹配查询的处理问题, 具体定义为:

输入: 一个由不确定图组成的大图类图数据库 $D=\{G_1, G_2, \cdots, G_n\}$, 一个查询图即确定图 q, 正整数 k.

输出: 对于 $\forall i(1\le i\le n)$, q 在 G_i 中概率最大的前 k 个子图匹配(若只存在 $k'(<k)$ 个子图匹配则输出所有匹配).

3 不确定图数据库中高效查询处理

为了解决提出的不确定图数据库的查询处理问题, 一个直接方法是逐个考察数据库中的不确定图, 枚举每个不确定图的所有蕴含确定图, 在蕴含确定图上进行子图匹配查询的处理并聚集得出每个匹配的概率值. 由于枚举每个不确定图的所有蕴含图是非常低效的, 甚至所有蕴含图的数量可能非常巨大, 因此直接方法非常低效.

为了高效地处理提出查询, 首先, 通过分析给出不确定图上子图匹配概率的计算方法, 该方法无需枚举不确定图的所有蕴含确定图. 其次, 以附带概率信息的邻居子图为基础, 提出一种有效的索引结构. 再次, 提出一种高效的基于搜索树的算法, 并结合给出的概率剪枝技术, 对在线提交的查询进行处理. 下面分别详述这 3 项内容.

3.1 在不确定图上计算子图匹配概率

令 G 是一个不确定图, q 是一个查询图(确定图). 假设 $\exists g\in Imp(G)$ 使得 $q\subseteq^f g$, 其中 f 是 q 到确定图 g 的一个子图同构, 进行下面分析. 令满足 $g\in Imp(G)$ 且 $q\subseteq^f g$ 的确定图 g 的集合为 $Imp_{(q,f)}(G)=\{g_1, g_2, \cdots, g_m\}$, 简记为 $Imp_{(q,f)}$, 其中 $m\le|Imp(G)|$. 根据第 2.1 节中式(1)和式(2), 子图匹配 $M_{q(f)G}$ 的概率等于 $\sum_{g\in Imp_{(q,f)}}\underline{P}(G\Rightarrow g)=\sum_{g\in Imp_{(q,f)}}Pr((\bigwedge_{e\in E_g}(e))\wedge(\bigwedge_{e'\in E_G-E_g}(\emptyset e')))$.

根据定义 2 和定义 3 可知, 对于 $\forall g_i\in Imp_{(q,f)}(G)$, 在子图匹配 $M_{q(f)g_i}=(V_i, E_i, \Sigma_i, l_i)$ 中有 $V_i\subseteq V_G$, $E_i\subseteq E_G$, $\Sigma_i\subseteq\Sigma_G$ 及 $l_i=l_G|_{V_i}$; 另外, 对于 $\forall g_i, g_j\in Imp_{(q,f)}(G)$, 子图匹配 $M_{q(f)g_i}$ 与 $M_{q(f)g_j}$ 相等, 即 $V_i=V_j$, $E_i=E_j$, $\Sigma_i=\Sigma_j$, $l_i=l_j$, 其中 $M_{q(f)g_i}=(V_i, E_i, \Sigma_i, l_i)$, $M_{q(f)g_j}=(V_j, E_j, \Sigma_j, l_j)$. 因此, 结合上一段的结果, $M_{q(f)G}$ 的概率等于 $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))$, 其中 $E_{M_{q(f)G}}$ 是 $M_{q(f)G}$ 的边集.

例 4. 考察图 2 中的不确定图 G_1 和图 3 中的查询图 q_1. 令 f 是一个从 q_1 到 G_1 的蕴含图 g_1 的子图同构函数, 其表示为 $(u_1\to v_1, u_2\to v_2, u_3\to v_3)$. 那么, 满足 $g\in Imp(G_1)$ 且 $q_1\subseteq^f g$ 的确定图 g 的集合为 $Imp_{(q_1,f)}(G_1)=\{g_1, g_3\}$. 子图匹配 $M_{q_1(f)G_1}$ 的概率等于 $\underline{P}(G\Rightarrow I_1)+\underline{P}(G\Rightarrow I_3)=Pr((v_1, v_2)\wedge(v_1, v_3)\wedge(v_2, v_3))+Pr((v_1, v_2)\wedge(v_1, v_3)\wedge\emptyset(v_2, v_3))$.

在实际应用中, 通常由规则引擎[10]如智能商业过程或其它复杂模型(如贝叶斯网络等)来管理生成规则集, $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))$ 的值可以直接由规则引擎计算得出. 规则引擎的计算细节超出本文讨论的范围. 但为了保持本文完整性, 下面仍然给出一种子图匹配 $M_{q(f)G}$ 概率即 $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))$ 的简单计算方法. 基于生成规则集 R_G, 令 $E_M=\{E_{M_1}, E_{M_2}, \cdots, E_{M_r}\}$ 是子图匹配的边集 $E_{M_{q(f)G}}$ 的一个划分, 其满足两条边 e 和 e' 在一个划分块中, 当且仅当在 R_G 中存在一个条件概率同时关联 e 和 e'. 对于 $\forall e\in E_{M_{q(f)G}}$, 如果在 R_G 中不存在关联 $E_{M_{q(f)G}}-\{e\}$ 中任何边的条件概率, 则在边集划分 E_M 中边 e 位于一个仅包含自身的划分块中. 于是, $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))=\prod_{j=1}^{r}Pr(\bigwedge_{e\in E_{M_j}}(e))$. 根据定义 1, 对于 $\forall E_{M_j}\in E_M$ 满足 $|E_{M_j}|\ge 2$, 在 R_G 中均存在条件概率 $Pr(e'_{j1}, e'_{j2}, \cdots, e'_{jm}|e'_{j(m+1)})$, 其中 $(m+1)=|E_{M_j}|$, 且 E_{M_j} 被表示为 $E_{M_j}=\{e'_{j1}, e'_{j2}, \cdots, e'_{jm}, e'_{j(m+1)}\}$. 因此对于 $\forall E_{M_j}\in E_M$ 满足 $|E_{M_j}|\ge 2$, $Pr(\bigwedge_{e\in E_{M_j}}(e))=Pr(e'_{j1}, e'_{j2}, \cdots, e'_{jm}|e'_{j(m+1)})\times Pr(e'_{j(m+1)})$. 综上, 子图匹配 $M_{q(f)G}$ 的概率即 $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))$ 的值得以计算而无需枚举不确定图 G 的所有蕴含确定图.

例 5. 继续考察图 2 中的不确定图 G_1 和图 3 中的查询图 q_1. 令 f 是一个从 q_1 到 G_1 的蕴含图 g_1 的子图同构函数, 其表示为 $(u_1\to v_1, u_2\to v_2, u_3\to v_3)$. 那么, 基于生成规则 R_{G_1} 边集 E_{G_1} 的划分 $E=\{\{(v_1, v_2), (v_1, v_3)\}, \{(v_2, v_3)\}\}$. 由定义 3, 在 f 下 q_1 在 G_1 中的子图匹配 $M_{q_1(f)G_1}$ 为图 $(\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_3)\}, \{A, B\}, l_{G_1})$. 基于 E, 边集 $E_{M_{q_1(f)G_1}}$ 的划分 $E_M=\{E_{M_1}=\{(v_1, v_2)\}, E_{M_2}=\{(v_2, v_3)\}\}$. 这里, 因为对于 $\forall e\in E_{M_{q_1(f)G_1}}=\{(v_1, v_2), (v_2, v_3)\}$, 在 R_{G_1} 中不存在关联 $E_{M_{q_1(f)G_1}}-\{e\}$ 中任何边的条件概率, 故在 E_M 中边 e 位于一个仅包含自身的划分块中. 因此, 子图匹配 $M_{q_1(f)G_1}$ 的概率等

于 $Pr(\bigwedge_{e\in E_{M_{q_1(f)G_1}}}(e)) = Pr((v_1, v_2)) \times Pr((v_2, v_3)) = 0.6 \times 0.7 = 4.2$. 与例 3 的结果一致.

下面对上述给出的子图匹配 $M_{q(f)G}$ 概率的简单计算方法的时间复杂度进行分析. 计算不确定图 G 的边集 E_G 的划分 E 的时间复杂度为 $O(|E_G|^2)$, 该计算对一个不确定图只进行一次. 对于每一个子图匹配 $M_{q(f)G}$, 基于得到的划分 E, 计算 E_M 的时间复杂度为 $O(|E_{M_{q(f)G}}|)$; 对于 $\forall E_{M_j} \in E_M$, 计算 $Pr(\bigwedge_{e\in E_{M_j}}(e))$ 的时间复杂度为 $O(1)$; 因此, 计算子图匹配 $M_{q(f)G}$ 概率即 $Pr(\bigwedge_{e\in E_{M_{q(f)G}}}(e))$ 的值的时间复杂度为 $O(|E_{M_{q(f)G}}|)$.

3.2　索引结构和构造方法

为了高效处理子图匹配查询, 给定查询图 q 的一个顶点 u, 要快速计算出数据库中的图的所有可能与之匹配的顶点集合 $MV(u)$. 同时, 为了支持查询处理算法(在第 3.3 节给出), 提出方法以附带概率信息的邻居子图为基础, 构造一种有效的索引结构, 称为 NG-Index. 首先, 回顾两个顶点间的最短距离, 并给出邻居子图的概念.

在一个确定图中, 两个顶点间的一条路径的长度是指路径上边的个数. 两个顶点间的最短距离是指两点间所有路径中最短路径的长度.

定义 **4**.　令 ρ 是一个由用户给定的最大层数阈值的常数. 令不确定图 $G=(V, E, \Sigma, l, P, R)$, 确定图 $(V, E, \Sigma, l)=g$. 对于 G 中的一个顶点 $v \in V$, v 在 G 中的邻居子图是一个确定图, 记为 $NG_G(v)$, 定义为 g 中与 v 的最短距离不大于 ρ 的所有顶点及其间的所有边构成的 g 的子图.

当给定查询图 q 的一个顶点 u, 要计算数据库中的图的所有可能与 u 匹配的顶点集 $MV(u)$ 时, u 的邻居子图与 $MV(u)$ 中顶点 v 的邻居子图必须满足一定的关系. 下面通过邻居子图的关系, 给出两个顶点匹配的必要条件.

定理 **1**(顶点匹配的必要条件).　对于一个查询图 q 和图数据库中的一个不确定图 G, 若 q 中顶点 u(即 $u\in V_q$)匹配于 G 中顶点 v(即 $v\in V_G$), 那么一定有邻居子图 $NG_q(u)$ 子图同构于 $NG_G(v)$.

证明.　令不确定图 $G=(V, E, \Sigma, l, P, R)$, 确定图 $(V, E, \Sigma, l)=g$. 根据定义 2, 如果 q 中顶点 u 匹配于 G 中顶点 v, 则说明存在从确定图 q 到确定图 g 的一个子图同构函数 f, 即 $q\subseteq^f g$, 且 $f(u)=v$.

考虑对于邻居子图 $NG_q(u)$ 中的任意一个顶点 u', 其满足在 q 中 u' 与 u 的距离不大于 ρ, 即 $dist_q(u', u)\le\rho$, 其中 $dist_q(u', u)$ 表示在 q 中 u' 与 u 的距离. 我们有, 在 g 中 $f(u')$ 与 v 的距离一定不大于 $dist_q(u', u)$, 即 $dist_g(f(u'), v)\le dist_q(u', u)$, 原因如下: 令 $u\to u_1\to u_2\to\cdots\to u'$ 为在 q 中 u 到 u' 的一条最短路径, 则 $v\to f(u_1)\to f(u_2)\to\cdots\to f(u')$ 是在 g 中 v 到 $f(u')$ 的一条路径. 于是, 顶点 $f(u')$ 一定存在于 $NG_G(v)$ 中.

由上一段也可知, 对于 $NG_q(u)$ 中的任意一条边 (u', u''), 顶点 $f(u')$ 和 $f(u'')$ 均存在于 $NG_G(v)$ 中, 再根据定义 4 可知, 边 $(f(u'), f(u''))$ 也存在于 $NG_G(v)$ 中.

综合上面两段, 结合定义 2 可知, 函数 f 也是从确定图 $NG_q(u)$ 到 $NG_G(v)$ 的一个子图同构.　证毕.

由于子图同构检测是一个 NP 完全问题, 对查询图的每个顶点和数据库中图的每个顶点嵌套循环来检测对应邻居子图间的子图同构是非常耗时和低效的. 因此, 下面给出对于查询图的每个顶点 u, 求得数据库中图的可能匹配顶点集合 $MV(u)$ 的高效方法.

本文将数据库中不确定图上的多个邻居子图中特定的子图, 即邻居子图的子图, 作为索引单元. 因为多个图构成的图集合中频繁子图表征了图集合内在特征, 并已被证实可作为良好的索引信息[3], 因此, 本文提出的方法是从多个邻居子图的图集合的所有频繁子图中挑选一部分作为索引单元. 下面先给出关于频繁子图的分析, 再给出挑选准则.

令不确定图 G 中标签为 L 的顶点集合为 $V_G^L=\{v\mid v\in V_G, l_G(v)=L\}$, G 的所有标签为 L 的顶点的邻居子图集合为 $NG_G^L=\{NG_G(v)\mid v\in V_G^L\}$. 对于一个确定图 S, 令 G 中相应的邻居子图包含 S 的顶点集合为 $V_G^L(S)=\{v\mid v\in V_G^L, S\subseteq NG_G(v)\}$. 考虑不确定图 G 的一个顶点 v 和一个确定图 S, 其满足 $S\cong NG_G(v)$, 当给定查询(确定)图 q 的一个顶点 u 时, 如果 $S\subseteq NG_q(u)$ 则根据子图同构关系的传递性可以得出 $NG_q(u)\cong NG_G(v)$. 于是, 此时 $MV(u)=V_G^L(S)$. 因此, 对于一组确定图 $SGS=\{S_1, S_2, \cdots, S_t\}$, 其满足对于 $\forall i(1\le i\le t)$ 顶点集合 $V_G^L(S_i)$ 已知, 则当给定查询(确定)图 q 的一个顶点 u 时, $MV(u)=\bigcap_{S_i\in SGS, S_i\subseteq NG_q(u)} V_G^L(S_i)$.

假设有一组确定图 $\underline{SGS}=\{\underline{S}_1, \underline{S}_2, \cdots, \underline{S}_m\}$ 已经作为索引单元, 其满足对于 $\forall i(1\le i\le m)$, $V_G^L(\underline{S}_i)$ 已被索引. 考虑另外一个确定图 S, 其满足对于 $\forall i(1\le i\le m)$ 有 $\underline{S}_i\subseteq S$ 且 $V_G^L(S)=\bigcap_{i=1}^{m} V_G^L(\underline{S}_i)$, S 无需被选

择作为索引单元,因为对于任意 q 的任意顶点 u,无论 S 是否子图同构于 $NG_q(u)$,S 都不能进一步缩小顶点匹配集合 $MV(u)$. 具体说:若 $S \subseteq NG_q(u)$,则对于 $\forall i(1 \le i \le m)$ 有 $\underline{S}_i \subseteq S$,那么

$$\bigcap_{\underline{S}_i \in \underline{SGS},\, \underline{S}_i \subseteq NG_q(u)} V_G^L(\underline{S}_i) = \bigcap_{\underline{S}_i \in \underline{SGS}} V_G^L(\underline{S}_i) =$$

$$\left(\bigcap_{\underline{S}_i \in \underline{SGS}} V_G^L(\underline{S}_i)\right) \cap (V_G^L(S)) = \bigcap_{\underline{S}_i \in \underline{SGS} \cup \{S\}} V_G^L(\underline{S}_i) =$$

$$\bigcap_{\underline{S}_i \in \underline{SGS} \cup \{S\},\, \underline{S}_i \subseteq NG_q(u)} V_G^L(\underline{S}_i);$$

若 $S \not\cong NG_q(u)$,那么

$$\bigcap_{\underline{S}_i \in \underline{SGS},\, \underline{S}_i \subseteq NG_q(u)} V_G^L(\underline{S}_i) = \bigcap_{\underline{S}_i \in \underline{SGS} \cup \{S\},\, \underline{S}_i \subseteq NG_q(u)} V_G^L(\underline{S}_i).$$

因此,这样的确定图 S(产生于频繁子图挖掘过程中)不被挑选作为索引单元. 一个确定图 S 的判别度,记为 $discr(S)$,定义为:(当 $\underline{SGS} \not\cong$ 时) $discr(S) = \dfrac{|\bigcap_{\underline{S}_i \in \underline{SGS}} V_G^L(\underline{S}_i)|}{|V_G^L(S)|}$,(当 $\underline{SGS} \cong$ 时), $discr(S) = +\infty$,其中 $\underline{SGS} = \{\underline{S}_1, \underline{S}_2, \ldots, \underline{S}_m\}$ 是所有满足 $\underline{S}_i \subseteq S$ 的已经索引的确定图的集合. 判别度大于一个用户设定的最小判别度阈值 $\delta_{\min}$(其为不小于 1 的正实数)的图定义为可判别的图. 为了选择索引单元,即索引子图,将图数据库 $D = \{G_1, G_2, \ldots, G_n\}$ 中所有不确定图的所有顶点按其标签分组:对于一个标签 L,在 D 中所有不确定图的所有标签为 L 的顶点的邻居子图集合中,即 $\bigcup_{G_i \in D} NG_{G_i}^L$ 中,挑选可判别的频繁子图作为索引单元,具体请参见第 3.2.2 节. 同时,对于 D 中每个不确定图 G_i,在构造索引时还要考察每个 G_i 邻居子图 NG 在 G_i 中相关的概率信息,即 $P_{G_i|E_{NG}}$ 和 R_{G_i} 的涉及 E_{NG} 中边的子集,这些作为邻居子图 NG 的附带概率信息.

对于一个标签 L,对于其对应的每个索引子图 S,如下的信息也被索引记录:

(1) 集合 $V_G^L(S)$. 注:图数据库中所有不确定图的所有顶点具有可唯一识别的顶点 ID(请回忆第 2 节).

(2) 对于 $\forall v \in V_G^L(S)$,正实数值 $matchmaxprs(v)$. 其中 $matchmaxprs(v)$ 是 S 在 $NG_G(v)$ 中所有子图匹配的概率的最大值,S 在 $NG_G(v)$ 中一个子图匹配的概率的计算请参见第 3.1 节.

(3) 如果 $|E_S| = 1$,对于 $\forall v \in V_G^L(S)$,正整数值 $nums(v)$. 其中 $nums(v)$ 等于一边图 S 在 $NG_G(v)$ 中子图匹配的个数.

3.2.1 索引结构

本文提出的索引结构称为 NG-Index,其是一个包含两层的混合索引结构,即高层索引和低层索引. 高层索引是一个三元组的列表,每个三元组的第 1 项是标签 L;第 2 项是 L 对应的索引子图 S 的编号;第 3 项是 S 自身. 低层索引是一个五元组的列表,每个五元组的第 1 项是索引子图 S 的编号;第 2 项是图数据库中不确定图 G 的编号 $GID(G)$;第 3 项是不确定图 G 的顶点 v 的编号 $VID(v)$,满足 $S \subseteq NG_G(v)$;第 4 项是 $matchmaxprs(v)$,即 S 在 $NG_G(v)$ 中所有子图匹配的概率的最大值;第 5 项或者是 $nums(v)$ 当 $|E_S| = 1$ 时,或者是 0 当 $|E_S| > 1$ 时,其中 $nums(v)$ 是一边图 S 在 $NG_G(v)$ 中匹配的个数.

例 6. 图 4 示例了一个简单的 NG-Index 的索引结构,其中对应于标签 L_1,有 3 个索引子图 S_1,S_2 和 S_3. 对于索引子图 S_1,在图数据库的不确定图 G_1 中有两个顶点 v_1 和 v_2,满足 $S_1 \subseteq NG_{G_1}(v_1)$ 和 $S_1 \subseteq NG_{G_1}(v_2)$,同时 S_1 在 $NG_{G_1}(v_1)$ 中所有子图匹配的概率的最大值为 0.21,S_1 在 $NG_{G_1}(v_2)$ 中所有子图匹配的概率的最大值为 0.40. 因为 S_1 是一边图(即 $|E_{S_1}| = 1$),因此在低层索引列表中,第 5 项记录着一边图 S_1 在相应邻居子图中子图匹配的个数,即是说,在图 $NG_{G_1}(v_1)$ 中有 2 个 S_1 的子图匹配,在 $NG_{G_1}(v_2)$ 中有 4 个 S_1 的子图匹配.

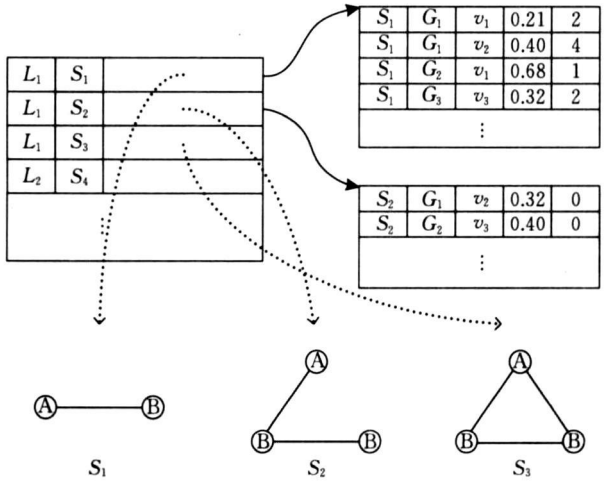


图 4 混合索引结构

值得说明的是,NG-Index 的索引结构可以很容易地实现于成熟的关系数据库中,高层索引可以实现为具有 3 个属性的表,即模式为(Label, SID, S),其中第 3 个属性的值可以是索引子图 S 对应的邻接表或邻接矩阵;低层索引可以实现为具有 5 个属性的表,即模式为(SID, GID, VID, MATCHMAXPr, NUM). 该实现不但具有强健壮性,而且保证了 NG-Index 是一种基于磁盘的索引结构.

3.2.2　索引构造方法

本文选取所有顶点数不大于 $maxV$ 的可判别的频繁子图作为索引子图, 其中 $maxV$ 是一个由用户给定的阈值常数. 在索引构造过程中, 可判别的频繁子图集合的获得是重要步骤. 下面, 先分析索引子图的性质, 然后给出索引子图产生算法, 进而总结出索引构造方法.

对于任意给定的最小判别度阈值 $\delta_{min}(\geq 1)$, 考虑这样的一个频繁子图 S, 其满足存在另一个频繁子图 S' 使得 $S' \subset S$ 且 $V_G^L(S') = V_G^L(S)$. 如果 S' 已被选择作为索引子图, 则由判别度定义得: $discr(S) \leq \frac{|V_G^L(S')|}{|V_G^L(S)|} = 1 \leq \delta_{min}$, 因此根据可判别图的定义(即判别度大于 δ_{min})可知 S 将不被挑选作为索引子图; 如果 S' 未被选择作为索引子图(注: 当考察 S 时, 所有被 S 包含的频繁子图都已经考察完毕), 则一定存在一个索引子图集合 $\{\underline{S}'_1, \underline{S}'_2, \cdots, \underline{S}'_t\}$ 满足对于 $\forall i(1 \leq i \leq t)$ 有 $\underline{S}'_i \subseteq S'$ 且

$$\frac{|\bigcap_{i=1}^{t} V_G^L(\underline{S}'_i)|}{|V_G^L(S')|} \leq \delta_{min},$$

于是对于 $\forall i(1 \leq i \leq t)$ 有 $\underline{S}'_i \subseteq S$, 进而

$$discr(S) \leq \frac{|\bigcap_{i=1}^{t} V_G^L(\underline{S}'_i)|}{|V_G^L(S)|} = \frac{|\bigcap_{i=1}^{t} V_G^L(\underline{S}'_i)|}{|V_G^L(S')|} \leq \delta_{min},$$

因此 S 仍将不被挑选作为索引子图. 综合上面两种互补情况可知, 这样的频繁子图 S 一定不会被挑选作为索引子图. 其逆否命题为, 任何一个索引子图 S 一定满足: 不存在一个频繁子图 S' 使得 $S' \subset S$ 且 $V_G^L(S') = V_G^L(S)$.

最近的一项工作[14]提出了频繁的"生成子子图"概念及其产生方法. 一个图集合的频繁生成子子图集合是其频繁子图集合的一个子集, 其为 $GEN = \{S | S \in FS$ 且 $(\nexists S' \in FS$ 满足 $S' \subset S$ 且 $sup(S') = sup(S))\}$, 其中 FS 表示所有频繁子图的集合, $sup(S)$ 表示频繁子图模式 S 的支持集. 结合上一段的分析可知, 任何一个索引子图 S 都是一个频繁生成子子图, 即是说, 本文的索引子图集合是邻居子图集合上的频繁生成子子图集合的一个子集. 因此, 索引子图的产生可以借助于现有的频繁生成子子图产生方法来高效地完成. 算法 1 给出了索引子图产生算法.

算法 **1**.　索引子图产生.

输入: 不确定图数据库 $D = \{G_1, G_2, \cdots, G_n\}$

输出: 索引子图集族 $idxS = \{SGS^L | L \in \bigcup_{G_i \in D} \Sigma_{G_i}\}$, 其中 SGS^L 是对应于标签为 L 的顶点的邻居子图集合的索引子图集合

1. 初始化 $idxS$ 为空集族;
2. for $\bigcup_{G_i \in D} \Sigma_{G_i}$ 中每个标签 L do
3. 初始化 SGS^L 为空集; 初始化变量 l 为 1;
4. 产生图集合 $\{NG_{G_i}(v) | G_i \in D, l_{G_i}(v) = L\}$ 中所有频繁生成子子图的集合 GEN^L;
5. while $l \leq maxV$ do
6. for GEN^L 中每个子图 S, 其满足 $|E_S| = l$ do
7. if $discr(S) > \delta_{min}$ then $SGS^L \leftarrow SGS^L \cup \{S\}$;
8. $l \leftarrow l + 1$;
9. $idxS \leftarrow idxS \cup \{SGS^L\}$;
10. return $idxS$;

需要说明的是, 产生所有频繁生成子子图(第 4 行)需要最小支持度阈值 α_{min} 参数. 为了产生所有的一边子图, 当(生成子)子图模式是一个一边图时 $\alpha_{min} = 1$. 另外, 因为用户给定阈值 ρ, δ_{min} 和 $maxV$ 也都可以视为系统变量, 因此它们都没有显式作为算法 1 的输入项.

算法 1 的工作过程如下: 初始时, 待输出结果数据结构 $idxS$ 置空(第 1 行). 在数据库 D 中所有不确定图的标签集的并集中, 算法逐个考察每个标签 L, 从 D 中所有不确定图的 L 标签顶点对应的邻居子图集合并集中, 产生频繁生成子子图集合 GEN^L(第 4 行); 按顶点数由小(1)到大($maxV$)的顺序逐层产生对应于标签 L 的可判别子图即索引子图, 并记录于数据结构 SGS^L 中(第 5~8 行), 再将 SGS^L 记录于 $idxS$ 中(第 9 行). 最后, 算法输出结果 $idxS$.

本算法第 4 行采用文献[14]的频繁生成子子图产生方法(即目前文献中唯一的相关方法)来产生集合 GEN^L. 该方法基于 DFS 编码树枚举框架[15], 对于每一个子图模式 S, 其要求找出 S 在其支持集的每个图中的所有子图匹配. 因此, 对每个子图模式 S 在每个 $NG_G(v)$(其中 $v \in V_G^L(S)$)中的所有匹配的概率最大值 $matchmaxprs(v)$ 的计算可以直接嵌入于该方法中; 同时, 对每个一边子图模式 S(即 $|E_S| = 1$)在 $NG_G(v)$(其中 $v \in V_G^L(S)$)中匹配的个数 $nums(v)$ 也可以直接嵌入. 为了表述简洁, 嵌入 $matchmaxprs(v)$ 和 $nums(v)$ 计算步骤的频繁生成子子图产生算法在此不予详述. 因此, 算法 1 得出构造 NG-Index 所需的全部内容.

3.3　查询处理算法

当输入一个查询图 q 时, 提出方法逐个考察 q 的每个顶点. 当考察查询图 q 的顶点 u 时, 提出方法通过 NG-Index 索引得出数据库中的图的所有可能

与之匹配的顶点集合 $MV(u)$, 以及对于 $MV(u)$ 中的每个顶点 v, 得出一个概率上界 $upbndpr(u, v)$. 然后, 运用概率剪枝技术, 运行一个基于搜索树的算法产生最后的子图匹配结果集. 下面分别给出对使用 NG-Index 索引以得出 $MV(u)$ 集合和概率上界 $upbndpr(u, v)$ 的方法以及基于搜索树的算法.

3.3.1 使用 NG-Index 索引

首先, 给出使用 NG-Index 索引的方法, 即对于查询图 q 的每个顶点 u, 计算出图数据库 D 中的图的所有可能与之匹配的顶点集合 $MV(u)$; 以及对于 $\forall v \in MV(u)$, 得出 q 在 G_i 中所有满足 u 对应于 v 的子图匹配的概率的上界, 记为 $upbndpr(u, v)$, 其中 G_i 是顶点 v 所属的不确定图. 首先给出下面的引理, 然后给出对于 $\forall v \in MV(u)$ 的 $upbndpr(u, v)$ 计算方法.

引理 **1**(概率非升). 令 G 是一个不确定图. 对于 G 上的两个子图匹配 M_1 和 M_2, 其满足 M_1 是 M_2 的子图, 有 M_1 的概率不小于 M_2 的概率.

证明. M_1 是 M_2 的子图即 $E_{M_1} \subseteq E_{M_2}$. 根据第 3.1 节可知, M_1 的概率为 $Pr(\bigwedge_{e \in E_{M_1}}(e))$, M_2 的概率为 $Pr(\bigwedge_{e \in E_{M_2}}(e))$. 类似于第 3.1 节的讨论, 基于生成规则集 R_G, 令 $E_{M_1} = \{E_{M_1-1}, E_{M_1-2}, \cdots, E_{M_1-t}\}$ 是边集 E_{M_1} 的一个划分, 其满足两条边 e 和 e' 在一个划分块中, 当且仅当在 R_G 中存在一个条件概率同时关联 e 和 e'. 同理, 基于 R_G, 令 $E_{M_2} = \{E_{M_2-1}, E_{M_2-2}, \cdots, E_{M_2-t}\}$ 是边集 E_{M_2} 的一个划分, 其满足两条边在一个划分块中, 当且仅当在 R_G 中存在一个条件概率同时关联这两条边. 对于 $\forall E_{M_1-i} \in E_{M_1}$, 一定 $\exists E_{M_2-j} \in E_{M_2}$ 使得 $E_{M_1-i} \subseteq E_{M_2-j}$, 因为如若不然, 或者(当 $|E_{M_1-i}| = 1$ 时, 令 $E_{M_1-i} = \{e\}$, 则 $e \notin E_{M_2}$ 进而 $E_{M_1-i} \not\subseteq E_{M_2}$)得出 $E_{M_1} \not\subseteq E_{M_2}$ 矛盾, 或者(当 $|E_{M_1-i}| \geq 2$ 时, 令 $e, e' \in E_{M_1-i}$ 即 R_G 中存在一个条件概率同时关联 e 和 e', 则 e, e' 一定也在 E_{M_2} 的一个划分块里)得出 E_{M_2} 中存在这样的划分块 E_{M_2-j} 使得 $E_{M_1-i} \subseteq E_{M_2-j}$ 矛盾. 因此, 重新表示 E_{M_2} 为 $E_{M_2} = \{E'_{M_2-1}, E'_{M_2-2}, \cdots, E'_{M_2-t}\}$, 其满足 $r \leq t$ 且对于 $\forall i(1 \leq i \leq r)$ 有 $E_{M_1-i} \subseteq E'_{M_2-i}$.

由概率基本性质可知, 对于任意两个集合 A 和 A' 其满足 $A \subseteq A'$, 有 $Pr(A) \leq Pr(A')$. 因此, 对于 $\forall i(1 \leq i \leq r)$ 有 $Pr(\bigwedge_{e \in E_{M_1-i}}(e)) \geq Pr(\bigwedge_{e \in E'_{M_2-i}}(e))$. 于是,

$$Pr(\bigwedge_{e \in E_{M_1}}(e)) = \prod_{i=1}^{r} Pr(\bigwedge_{e \in E_{M_1-i}}(e)) \geq \prod_{i=1}^{r} Pr(\bigwedge_{e \in E'_{M_2-i}}(e)) \geq \prod_{i=1}^{t} Pr(\bigwedge_{e \in E_{M_2-i}}(e)) = Pr(\bigwedge_{e \in E_{M_2}}(e)).$$

即 M_1 的概率不小于 M_2 的概率. 证毕.

根据引理 1, 对于任意一个 $S \in SGS^L$ 满足 $S \subseteq NG_q(u)$, 图 $NG_q(u)$ 在 G_i 中所有满足 u 对应于 v 的子图匹配的概率不大于 $matchmaxprs(v)$, 其中 v 为 $V^L_{G_i}(S)$ 中任意一个顶点. 而 $NG_q(u)$ 是 q 的子图, 进而 q 在 G_i 中所有满足 u 对应于 v 的子图匹配的概率不大于 $matchmaxprs(v)$. 综合上面分析, q 在 G_i 中所有满足 u 对应于 v 的子图匹配的概率的一个上界是所有满足 $S \in SGS^L$ 且 $S \subseteq NG_q(u)$ 的 $matchmaxprs(v)$ 的最小值, 即 $upbndpr(u, v) = \min_{S \in SGS^L, S \subseteq NG_q(u)}\{matchmaxprs(v)\}$. 下面的算法 2 给出索引使用算法.

算法 **2**. 索引使用.

输入: 在不确定图数据库 D 上的 NG-Index 索引 $IndexS = \{SGS^L\}$, 查询图 q(确定图)

输出: q 的每个顶点 u 的可能匹配顶点集 $MV(u)$, 以及对于 $\forall v \in MV(u)$ 子图匹配的概率的上界 $upbndpr(u, v)$

1. 初始化 MV 为空集族;
2. for V_q 中的每个顶点 u do
3. 计算出在图 $NG_q(u)$ 中的所有一边图的集合 ${}^eSGS(u)$ 及 $\forall T \in {}^eSGS(u)$ 在 $NG_q(u)$ 中子图匹配个数 $freq(T)$;
4. if $\exists T \in {}^eSGS(u)$ 满足 $T \notin SGS^L$, 其中 $L = l_q(u)$, SGS^L 为关联于 L 的索引子图集合 then return $\cong$;
5. $MV(u) \leftarrow \bigcap_{T \in {}^eSGS(u)}\{v \mid v \in V^L_G(T), freq(T) \leq num_T(v)\}$;
6. if $MV(u) = \cong$ then return 空集族;
7. $MV(u) \leftarrow MV(u) \cap (\bigcap_{S \in SGS^L, S \notin {}^eSGS(u), S \subseteq NG_q(u)} V^L_G(S))$;
8. for $MV(u)$ 中的每个顶点 v do
9. $upbndpr(u, v) \leftarrow \min_{S \in SGS^L, S \subseteq NG_q(u)}\{matchmaxprs(v)\}$;
10. $MV \leftarrow MV \cup \{MV(u)\}$;
11. return MV;

在此算法中用户给定阈值 ρ 与算法 1 中所使用的 ρ 值相等, 其视为系统变量而非显式作为算法的输入项. 算法的第 4, 5, 7, 9 行中分别涉及的 SGS^L; $V^L_G(T)$, $num_T(v)$; SGS^L, $V^L_G(S)$; SGS^L, $matchmaxprs(v)$ 的量均为索引信息.

算法 2 如下工作: 初始时, 待输出结果数据结构 MV 置空(第 1 行). 然后算法逐个考察查询图 q 中

的每个顶点u(第 2~ 10 行), 具体如下. 首先, 计算出在邻居子图 $NG_q(u)$ 中的所有一边图的集合 $^eSGS(u)$及$^eSGS(u)$中每个一边图 T 在 $NG_q(u)$ 中子图匹配的个数, 记为$freq(T)$(第 3 行). 因为在索引构造时, 图数据库中所有不确定图上的邻居子图中的所有一边图都被挑选作为索引子图(请回忆当模式是一边图时有 $\alpha_{\min}=1$ 且一边图的判别度总为 $+\infty$), 所以如果在 $NG_q(u)$中的某个一边图 T 不是索引子图, 那么 $NG_q(u)$ 无法子图同构于图数据库中不确定图上的任何邻居子图, 进而 u 一定无法匹配图数据库中的图的任何顶点(定理 1), 于是查询处理结果是空集(定义 2 和定义 3). 算法判断如果是此情况, 则输出空集(第 4 行). 类似于上面的道理, 如果对于图数据库中图的顶点 v, 在 $NG_q(u)$中的某个一边图 T 在 $NG_q(u)$ 中子图匹配的个数 $freq(T)$大于 T 在v 相应的邻居子图的子图匹配个数(即索引信息 $num_T(v)$), 则 u 无法匹配v. 算法从可能匹配集合 $MV(u)$ 中去除满足上面情况的顶点(第 5 行). 然后, 算法得出 $MV(u)$(第 7 行), 其中条件($S\in SGS^L$, $S\notin {}^eSGS(u)$, $S\subseteq NG_q(u)$)实际上等价于条件($S\in SGS^L$, $|E_S|\neq 1$, $S\subseteq NG_q(u)$). 再次, 对于 $\forall v\in MV(u)$, 算法得出 $upbndpr(u, v)$(第 8~ 9 行). 最后, 算法输出结果MV.

3.3.2　基于搜索树的匹配算法

经过对索引的访问和使用, 对于查询图 q 的每个顶点u, 得到了其可能匹配的顶点集合 $MV(u)$ 以及对于 $MV(u)$ 中的每个顶点 v, 得到了一个概率上界$upbndpr(u, v)$. 本节给出一个以这些中间结果为输入的基于搜索树的匹配算法来输出最后的子图匹配结果集. 该方法先深搜索一个包含所有 q 的子图匹配的搜索空间(搜索树), 并使用一个关于概率界限信息的剪枝技术来提高搜索效率. 下面, 首先介绍搜索树.

令 q 是查询(确定)图, G 是一个不确定图(数据库中的图), $\prec_{V_q}=\{u'_1, u'_2, \ldots, u'_{|V_q|}\}$ 是顶点集 V_q 上的一个序. 搜索树中的每个节点是一个搜索状态(简称状态), 一个搜索状态是一个子图同构. 根(节点)状态是一个空集(在实现时用一个空向量表示), 其层数定义为 0. 一个 t 层($t\geq 1$)状态是一个从 $q[(t)]$ 到 G 的子图同构, 其中 $q[(t)]$ 是 q 的诱导子图即 $q[\{u'_1, u'_2, \ldots, u'_t\}]$的简写. 对于一个 t 层状态 s, 其相应的子图同构记为 f^s, f^s 用一个向量$\langle f^s(u'_1), f^s(u'_2), \ldots, f^s(u'_t)\rangle$来实现, 其中 $\forall i(1\leq i\leq t)$ 有 $f^s(u'_i)\in V_G$, 在子图同构 f^s下 $q[(t)]$在 G 中的子图匹配记为$M_{q[(t)](f^s)G}$(简记为 $M_{(f^s)}$). 在搜索树中, 状态 s 的每个孩子状态是一个从$q[(t+1)]$到 G 的子图同构. s 的孩子状态集合是子图同构集合$\{f\mid q[(t+1)]\subseteq {}^fG, f=f^s\oplus\langle v\rangle, v\in V_G-\{f^s(u'_1), \ldots, f^s(u'_t)\}\}$, 其中"$\oplus$"表示两个向量的串联连接. 由此可知, 对于 s 的每个孩子 cs, $M_{(f^s)}$ 是 $M_{(f^{cs})}$ 的子图. 在搜索树中, 将$|V_q|$层的状态(子图同构)定义为相对于查询图 q 的完全状态(完全子图同构), 其它状态(小于$|V_q|$层)定义为相对于 q 的部分状态(部分子图同构). 显然, 上述描述的搜索树容纳了所有相对于 q 的完全状态即完全子图同构, 因此, 容纳了从 q 到 G 的所有子图同构. 另外, 对于搜索树中每个 t 层状态s, 对应着一个概率值 $Pr(s)$, 称为状态 s 的概率, 其等于匹配 $M_{(f^s)}$ 在不确定图 G 中的概率.

不失一般性, $\prec_{V_q}$为顶点集 V_q 上任一给定的序. 该序亦可根据具体情况给定. 尽管如此, 这里给出一种确定该序的方法, 即将 V_q 中的顶点($u\in V_q$)按照$\{upbndpr(u, v)\mid v\in MV(u)\}$ 平均值的降序作为序$\prec_{V_q}$.

提出的匹配算法逐个考察图数据库中的每个不确定图 G_i, 先深度搜索(Depth-First Search)查询图 q 和 G_i对应的搜索树, 即起始于搜索树的根状态, 结束于 q 在 G_i 中概率最大的前 k 个子图匹配(如果有)被计算并输出. 为了高效地搜索, 下面首先给出一个定理, 然后给出一个用概率界限信息进行剪枝的技术.

定理 **2**(孩子状态概率非升).　令 q 是一个查询(确定)图, G 是一个不确定图(数据库中的图). 在 q 和 G 对应的搜索树中, 对于一个状态 s, 其任意一个孩子状态 cs 满足不等式 $Pr(cs)\leq Pr(s)$.

证明.　状态 s 对应的子图匹配 $M_{(f^s)}$ 是状态 cs 对应子图匹配 $M_{(f^{cs})}$ 的子图, 根据引理 1 可知, $M_{(f^s)}$ 的概率不小于 $M_{(f^{cs})}$ 的概率, 即 $Pr(s)\geq Pr(cs)$.

证毕.

请回忆第 3.3.1 节的讨论, 对于 $\forall v\in MV(u)$ 有 $upbndpr(u, v)$是 q 在 G_i 中所有满足 u 对应于 v 的子图匹配的概率的上界, 其中 G_i 是v 所属的不确定图. 结合定理 2, 可以得到如下推论 1.

推论 **1**.　令 q 是一个查询(确定)图, G 是一个不确定图(数据库中的图). 在 q 和 G 对应的搜索树中, 令 s 是一个状态, $upbndpr(M_{(f^s)})$是 s 的后代中所有完全状态的概率的一个上界. 对于 s 的一个孩子状态 cs, cs 的后代中所有完全状态的概率的一个

上界, 记为 $upbndpr(M_{(f^{cs})})$, 可以计算如下:

$$upbndpr(M_{(f^{cs})}) = \min\{upbndpr(M_{(f^{s})}), upbndpr(u'(t+1), f^{cs}(u'_{(t+1)})), Pr(M_{(f^{cs})})\}.$$

在推论 1 给出 $upbndpr(M_{(f^{cs})})$ 的计算方法中, 基于第 3.1 节的讨论, 计算 $Pr(M_{(f^{cs})})$ 的时间复杂度是 $O(|E_{q[(t+1)]}|)$. $upbndpr(M_{(f^{s})})$ 的值是在访问状态 s 时已知, $upbndpr(u'(t+1), f^{cs}(u'_{(t+1)}))$ 的值已经在运行算法 2 后得到(从索引中得到).

综上, 下面给出剪枝技术. 在先深搜索查询图 q 和不确定图 G 对应的搜索树过程中, 令当前状态为 s. 倘若此时算法已经找出 q 在 G 中的 k 个子图匹配, 令变量 $lowbndpr$ 记录这 k 个子图匹配的概率的最小值, 那么如果 s 的后代中所有完全状态的概率的上界即 $upbndpr(M_{(f^{s})})$ 小于 $lowbndpr$, 则在搜索树中以 s 为根的子树可以安全地剪枝而无需考察.

另外, 算法在先深搜索搜索树时涉及如下细节. 在搜索过程中, 令当前状态为 s, 当前处于搜索树第 t 层. s 的所有孩子状态计算如下. 在图 $q[(t+1)]$ 中, 将与顶点 $u'_{(t+1)}$ 相连的顶点的集合 $Adj_{q[(t+1)]}(u'_{(t+1)})$ 简记为 Adj. 记录顶点集合 $CHLD_s$: 如果 $Adj_{q[(t+1)]}(u'_{(t+1)}) = \cong$, 则 $CHLD_s = MV(u'_{(t+1)}) - V_{M_{(f^{s})}}$; 如果 $Adj_{q[(t+1)]}(u'_{(t+1)}) \neq \cong$, 则 $CHLD_s = (\bigcap_{u\in Adj} Adj_G(f^s(u)) - V_{M_{(f^{s})}}) \cap MV(u'_{(t+1)})$. 则算法分别从 $CHLD_s$ 中取出一个顶点并添加到向量表示的子图同构 f^s 中从而形成 s 的一个孩子状态(子图同构), 同时集合 $CHLD_s$ 包含了所有可以用来形成 s 孩子状态的顶点.

算法 3 给出匹配产生算法.

算法 3. 匹配产生.

输入: 不确定图数据库 $D = \{G_1, G_2, \cdots, G_n\}$, 查询(确定)图 q, 正整数 k, $M\_V = \{MV(u) \mid u \in V_q\}$, 其中 $MV(u)$ 是 q 的顶点 u 的可能匹配顶点集, 和对于 $\forall v \in MV(u)$ 的子图匹配概率上界 $upbndpr(u, v)$

输出: 对于 $\forall i(1 \le i \le n)$, q 在 G_i 中概率最大的前 k 个子图匹配(若只存在 $k'(<k)$ 个子图匹配则输出所有匹配)

1. for D 中每个不确定图 G_i do
2. for q 中每个顶点 u do $MV_{G_i}(u) \leftarrow MV(u) \cap V_{G_i}$; //$M\_V_{G_i} = \{MV_{G_i}(u) \mid u \in V_q\}$
3. if $\exists u \in V_q$ 满足 $MV_{G_i}(u) = \cong$ then q 在 G_i 中的子图匹配集合为空集; continue;
4. 根据 q 和 $M\_V_{G_i}$ 得到 V_q 上的一个序 $\prec_{V_q} = \langle u'_1, u'_2, \cdots, u'_{|V_q|}\rangle$;
5. 初始化 f 为长度为 $|V_q|$ 的 0 向量(每个分量均为 0);
6. 初始化 $Queue$ 为空优先级队列(按照子图匹配概率非升排序); 初始化 $lowbndpr$ 为 0;
7. for $MV_{G_i}(u'_1)$ 中的按照 $upbndpr(u'_1, v)$ 非升序的每个顶点 v do
8. if $upbndpr(u'_1, v) \ge lowbndpr$ then
9. $f[u'_1] \leftarrow v$;
10. 调用子过程 $DFS(1, f, upbndpr(u'_1, v), Queue, lowbndpr)$;
11. q 在 G_i 中概率最大的前 k 个子图匹配为 $Queue$ 前 k 项(若只有 $k'(<k)$ 项则返回这所有 k' 项);

算法 3 子过程. $DFS(t, f, upbndpr(M_{q(f)G_i}), Queue, lowbndpr)$

// 输入: 当前层数 t, 当前子图同构 f, q 在 G_i 中所有满足 f 的子图匹配的概率上界 $upbndpr(M_{q(f)G_i})$, $Queue$, $lowbndpr$

// 输出: $Queue$, $lowbndpr$

12. if $t = |V_q|$ then
13. $Queue.insert(M_{q(f)G_i})$;
14. if $Queue.size \ge k$ then $lowbndpr \leftarrow Pr(Queue[k])$;
15. 初始化长度为 $|V_q|$ 的向量 f' 为 f;
16. 计算当前子图同构 f 孩子的顶点集合 $CHLD$;
17. for $CHLD$ 中的每个顶点 v' do
18. $f'[u'_{(t+1)}] \leftarrow v'$;
19. 根据推论 1 计算 $upbndpr(M_{q(f')G_i})$, 记为 $upbndpr(M_{q(f\cup(u',v'))G_i})$;
20. for $CHLD$ 中的按照 $upbndpr(M_{q(f\cup(u',v'))G_i})$ 非升序的每个顶点 v' do
21. if $upbndpr(M_{q(f\cup(u',v'))G_i}) \ge lowbndpr$ then
22. $f'[u'_{(t+1)}] \leftarrow v'$;
23. 调用子过程 $DFS(t+1, f', upbndpr(M_{q(f\cup(u',v'))G_i}), Queue, lowbndpr)$;

算法 3 的工作过程如下: 算法逐个考察图数据库中的每个不确定图 G_i. 对于每个不确定图 G_i, 进行如下处理. 如果 V_q 中存在一个顶点 u 满足 u 在 G_i 中的可能匹配顶点集合 $MV_{G_i}(u) = \cong$, 则 q 在 G_i 中没有子图匹配(第 2, 3 行). 否则, 首先, 确定 V_q 上的序 $\prec_{V_q} = \langle u'_1, u'_2, \cdots, u'_{|V_q|}\rangle$, 从而确定 q 和 G_i 对应的搜索树(第 4 行). 子图同构用 $|V_q|$ 长度的向量来实现. 在初始化一个 0 向量 f 和一个空优先级队列 $Queue$(按照子图匹配概率非升排序)及将变量 $lowbndpr$ 置 0 后, 算法先深搜索 q 和 G_i 对应的搜索树, 同时在面临多个孩子节点 cs 时, 优先搜索 $upbndpr(M_{q(f^{cs})G_i})$ 最大的分枝, 以使得算法尽早地考察大概率的 q 的子图匹配(第 7~10, 20~23 行). 在先深搜索过程中, 变量 $lowbndpr$ 记录了当前已经找到的 k 个子图匹配的概率的最小值, 如果一个

状态其后代中所有完全状态的概率的上界都小于 *lowbndpr*, 则以该状态为根的搜索子树可以安全地剪枝(第 8, 21 行). 最后, 算法输出 q 在 G_i 中概率最大的前 k 个子图匹配即 *Queue* 中的前 k 项(若只有 k' ($<k$) 项则返回这所有 k' 项).

综合第 3.3.1 节和 3.3.2 节提出的查询处理方法首先运行算法 2, 然后运行算法 3, 最后得出查询结果.

4　实验结果与分析

实验主要考察提出方法的索引构造性能和查询处理效率以及两者的可扩展性. 实验的运行环境为 PIV 3.0GHz CPU、2GB 内存和运行 RedHat Linux 8.0 操作系统的 PC 机. 本文方法(以下统称为 NG-Index)采用 C 语言实现、GCC 编译(-O2 优化). 据我们所知, 目前尚无解决本文提出查询问题的方法, 因此实验主要集中对本文方法进行性能的检测和分析.

为了检测本文方法在不同特征数据上的性能, 实验数据集的不确定图通过如下方式合成得到: 首先生成 $\underline{V}$ 个顶点, 然后通过随机选择两个端点的方式生成 $\underline{E}$ 条边(简单 Erdos-Renyi 随机图模型); 其次按照 α 为 1 的 Zipf 分布生成 $\underline{V}$ 个标签(共有 $\underline{L}$ 个不同的标签)并为每个顶点分配一个标签; 再次按照均值 $\underline{\mu}$ 和方差 $\underline{\sigma}^2$ 的正态分布生成 $\underline{E}$ 个(0, 1] 概率值并为每条边分配一个概率值. 在上述合成方法中, $\underline{V}$, $\underline{E}$, $\underline{L}$, $\underline{\mu}$ 和 $\underline{\sigma}^2$ 为在实验中设定的参数, 用以合成不同特征的数据. 实验中图数据库只包含一个不确定图. 使用一个而非多个不确定图对大图类图数据库仍然具有代表性. 另外, 为了生成查询图 q, 从图数据库的不确定图中随机提取出特定顶点个数 $\underline{Q}$(或 $|V_q|$)的连通子图(确定图, 即不考虑概率信息).

若无特别说明, 提出方法中涉及的参数在实验中设置如下: 最大层数阈值 ρ 为 2, 最小可判别度阈值 δ_{min} 为 1.5, 最大顶点数阈值 *maxV* 为 8 以及最小支持度阈值 α_{min} 设置为: 若当前(生成子)子图模式是一边图则 $\alpha_{min}=1$, 否则,

$$\alpha_{min}=\sqrt{\frac{curV}{maxV}}\times|SET|\times factor_\alpha,$$

其中 *curV* 是当前(生成子)子图模式的顶点个数, 图集合 $SET=\{NG_{c_i}(v)\mid G_i\in D, v\in G_i, l_{G_i}(v)=L\}$, L 为当前的标签, $factor_\alpha$ 是一个取自(0, 1] 的实数值(无特别说明时 $factor_\alpha=0.1$). 另外, 若无特别说明, 查询图 q 的顶点个数 $\underline{Q}$ 为 6, k 为 10. 所有查询处理时间均为特定尺寸的 100 个查询图处理时间的平均值.

图 5 给出了不同尺寸查询图的 top-k 查询处理执行效率的实验结果. 图数据库的不确定图(记为 G_D)的参数为 $\underline{V}=5000$, $\underline{E}=15000$, $\underline{L}=300$, $\underline{\mu}=0.9$, $\underline{\sigma}=0.1$; 顶点个数 $\underline{Q}$ 分别为 4、5、6、8、10 的五组查询图被生成. 图 5 的横坐标表示查询图的顶点个数 $\underline{Q}$, 纵坐标表示每个查询的平均查询处理时间, 单位为 s. 在图中我们看到, 提出方法的查询处理时间随着查询图顶点的增多而逐渐变长, 最大的查询图(10 个顶点)可在 4s 以内处理完成.

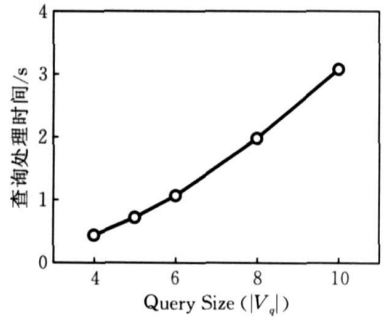


图 5　不同尺寸查询图的查询处理时间

图 6 给出了不同 k 值的 top-k 查询处理执行效率的实验结果. 图数据库的不确定图仍选用 G_D; 分别提交了 5 个 k 值即 5、10、15、20、30, 每个 k 值的处理结果为顶点个数 $\underline{Q}$ 为 6 的 100 个查询图的平均值. 图 6 的横坐标表示提交的 k 值, 纵坐标表示每个查询的平均查询处理时间, 单位为 s. 在图 6 中可以看到, 查询处理时间随着 k 值的增大不断增长, 概率最大的前 30 个匹配可以在 3s 以内全部输出. 以上这些实验结果显示了本文提出的方法具有较强的可用性, 其适用于处理小尺寸查询图在大尺寸的不确定图上的概率子图匹配查询.

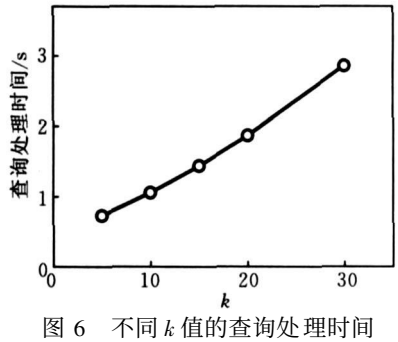


图 6　不同 k 值的查询处理时间

图 7 给出了索引构造性能的实验结果. 最小支持度阈值 α_{min} 对索引子图集合影响很大, 因此本实验考察在不同 $factor_\alpha$ 取值下的索引构造性能. $factor_\alpha$ 分别取值 0.10、0.15、0.20. 图 7 的横坐标表

示 $factor_\theta$ 取值, 纵坐标表示索引构造时间, 单位为 s. 从实验结果中我们看到, 提出方法的索引构造时间随着 $factor_\theta$ 的增大而逐渐变短. 原因是因为随着参数 $factor_\theta$ 的增大, 最小支持度阈值 σ_{min} 也增大, 从而导致频繁生成子子图的数量减少, 也导致索引构造(算法 1)中计算判别度时涉及的子图同构检测数量减少.

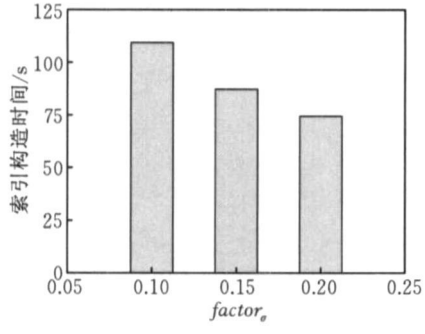


图 7 不同 $factor_\theta$ 值的索引构造时间

图 8、图 9 给出了可扩展性实验结果. 所选用的 5 个不确定图构造如下: 先从不确定图 G_D 中随机提取一个 1000 个顶点的子图, 然后基于该图从 G_D 中再提取 1000 个顶点形成 2000 个顶点的子图, 以此类推提取出 3000, 4000 和 5000 个顶点的共 5 个图. 图 8、图 9 的横坐标均表示图数据库的不确定图(简称数据图)的顶点个数, 图 8 的纵坐标表示每个查询图的平均查询处理时间, 单位为 s; 图 9 的纵坐标为索引构造时间, 单位为 s. 通过这两个图我们看到本文提出的方法具有良好的可扩展性.

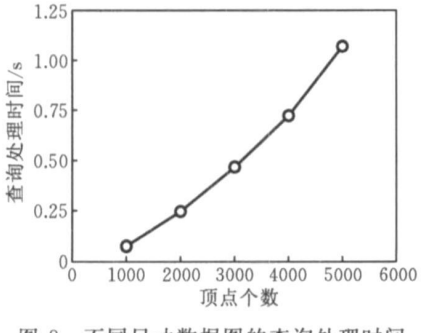


图 8 不同尺寸数据图的查询处理时间

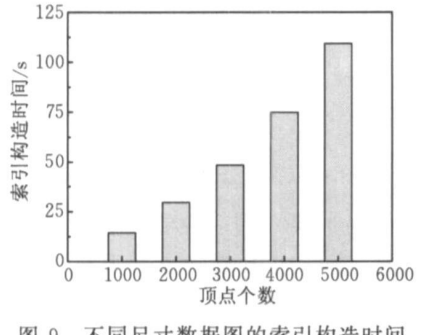


图 9 不同尺寸数据图的索引构造时间

图 10 给出了不同最小判别度阈值 δ_{min} 对查询处理影响的实验结果. 图 10 横坐标表示 δ_{min} 取值, 为 1.5、1.75、2、2.25, 纵坐标表示每个查询图的平均查询处理时间, 单位为 s. 通过该实验可以看出, 随着 δ_{min} 的增大, 查询处理时间逐渐变长. 然而, 索引子图数量会随着 δ_{min} 的增大而逐渐减小, 在实际应用中我们需要在查询处理时间和索引子图数量即索引大小上做出折中.

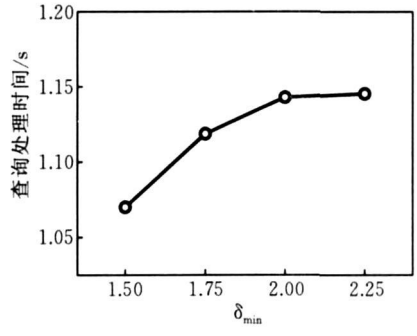


图 10 不同最小判别度阈值 δ_{min} 的查询处理时间

在对其它参数特征的图数据集的实验中得到了与上述实验类似的结果, 限于篇幅在此不予详述.

5 结 论

本文研究如何在不确定图数据库上高效地处理查询. 给出了一种数据模型来表示图的不确定性, 提出了不确定图数据库上的概率 top-k 子图匹配查询的问题. 为了高效地处理提出的查询问题, 以附带概率信息的邻居子图为基础, 设计了一种有效的索引结构 NG-Index; 提出了一种带有概率剪枝的基于搜索树的匹配算法来进行查询处理. 实验结果表明, 所提出方法具有良好的效率和可扩展性.

参 考 文 献

[1] Zhou Shu-Geng, Yu Zhao-Chun, Jiang Hao-Liang. Concepts, issues, and advances of searching in graph-structured data. Communications of the China Computer Federation, 2007, 3(8): 59-65(in Chinese)
(周水庚, 蔚赵春, 蒋豪良. 图结构数据搜索的概念、问题与进展. 中国计算机学会通讯, 2007, 3(8): 59-65)

[2] Shasha D, Wang T L J, Giugno R. Algorithmics and applications of tree and graph searching//Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. Madison, 2002: 39-52

[3] Yan X, Yu P S, Han J. Graph indexing: A frequent structure-based approach//Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris, 2004: 335-346

[4] Zhang Shuo, Li Jian-Zhong, Gao Hong, Zou Zhao-Nian. Approach for efficient subgraph isomorphism testing for multiple graphs. Journal of Software, to appear(in Chinese)
(张硕, 李建中, 高宏, 邹兆年. 一种多到一子图同构检测方法. 软件学报, 待发表)

[5] 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

[6] Zhou Ao-Ying, Jin Che-Qing, Wang Guo-Ren, Li Jian-Zhong. A survey on the management of uncertain data. Chinese Journal of Computers, 2009, 32(1): 1-16(in Chinese)
(周傲英, 金澈清, 王国仁, 李建中. 不确定性数据管理技术研究综述. 计算机学报, 2009, 32(1): 1-16)

[7] Li Jian-Zhong, Yu Ge, Zhou Ao-Ying. Requirements and challenges of the management of uncertain data. Communications of the China Computer Federation, 2009, 5(4): 6-14 (in Chinese)
(李建中, 于戈, 周傲英. 不确定性数据管理的要求与挑战. 中国计算机学会通讯, 2009, 5(4): 6-14)

[8] Gao Hong, Zhang Wei. Research status of the management of uncertain graph data. Communications of the China Computer Federation, 2009, 5(4): 31-36(in Chinese)
(高宏, 张炜. 不确定图数据管理研究现状. 中国计算机学会通讯, 2009, 5(4): 31-36)

[9] Dalvi N N, Suciu D. Efficient query evaluation on probabilistic databases. VLDB Journal, 2007, 16(4): 523-544

[10] Soliman M A, Ilyas I F, Chang K C. Top-*k* query processing in uncertain databases//Proceedings of the 23rd International Conference on Data Engineering. Istanbul, 2007: 896-905

[11] Hintsanen P. The most reliable subgraph problem//Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases. Warsaw, 2007: 471-478

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

[13] Zou Zhao-Nian, Li Jian-Zhong, Gao Hong, Zhang Shuo. Mining frequent subgraph patterns from uncertain graphs. Journal of Software, Published online 2008-12-22(http://www.jos.org.cn/1000-9825/3473.htm)(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, 2008-12-22 在线出版)

[14] Zeng Z, Wang J, Zhang J, Zhou L. FOGGER: An algorithm for graph generator discovery//Proceedings of the 12th International Conference on Extending Database Technology. Saint Petersburg, 2009: 517-528

[15] Yan X, Han J. gSpan: Graph-based substructure pattern mining//Proceedings of the 2002 IEEE International Conference on Data Mining. Maebashi City, 2002: 721-724

ZHANG Shuo, born in 1982, Ph. D. candidate. His research interests focus on graph data management.

GAO Hong, born in 1966, professor, Ph. D. supervisor. Her research interests include parallel database, data warehouse.

LI Jian-Zhong, born in 1950, professor, Ph. D. supervisor. His research interests include parallel database, wireless sensor networks.

ZOU Zhao-Nian, born in 1979, Ph. D. candidate. His research interests focus on graph data mining.

Background

This work is partially supported by the National Grand Fundamental Research 973 Program of China under grant No. 2006CB303000; the Key Program of the National Natural Science Foundation of China under grant No. 60533110; the National Natural Science Foundation of China under grant No. 60773063; NSFC-RGC of China under grant No. 60831160525.

In recent years, lots of data in various domains have been naturally modeled by graphs, e. g. sensor networks, protein interaction networks, etc. Uncertainty is inherent in much of the data due to the imprecise characteristics of equipments or the nature of data. It is important and demanding to efficiently process queries on these databases and sensor networks with uncertainty. The purpose of those projects is to investigate the issues arising from these systems and environments, and provide efficient and effective algorithms and practical solutions. Nowadays, the research on uncertain databases is one of the current research focuses. The existing work in the early stage focused on presenting data models for uncertain relational databases, formulating problems on uncertain relational databases and devising algorithms for these problems. For uncertain graph databases, the existing works have proposed algorithms for finding reliable subgraphs from large probabilistic graphs, and for mining frequent subgraph patterns from uncertain graphs. This paper focuses on processing queries on uncertain graph databases, and proposes an index-based query processing method, which has been verified by experiments to be efficient and scalable.