ERSearch:一种高效的子图查询算法

黄 云[1,2],洪佳明[3],覃遵跃[1,2],钟 键[2],李梦婷[1],印 鉴[1]
(1. 中山大学信息科学与技术学院,广东广州 510006;2. 吉首大学软件服务外包学院,湖南张家界 427000;
3. 广州中医药大学医学信息工程学院,广东广州 510006)

摘 要: 子图查询是图数据库研究中的一个重要问题,许多方法基于"过滤-验证"策略进行子图查询,算法研究的重点为快速找到有效的特征集.通过对特征模式在数据图集中的嵌入信息进行分析,离线建立基于重叠关系、邻接关系和近邻关系的嵌入关系索引,提出基于嵌入关系的子图查询算法 ERSearch.在给定查询图后,利用特征共现关系与特征嵌入关系联合进行过滤操作,并将过滤阶段的嵌入关系比对结果用于验证过程,提高验证效率.在真实及模拟数据上的实验表明,通过与 PathIndex 等方法的对比,ERSearch 算法有效缩减了候选集的规模,能有效提高过滤与验证阶段的执行效率.

关键词: 子图查询;特征模式;嵌入关系;图索引;图数据库
中图分类号: TP311.131 **文献标识码:** A **文章编号:** 0372-2112 (2017)02-0368-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.02.015

ERSearch:An Efficient Subgraph Query Algorithm

HUANG Yun[1,2], HONG Jia-ming[3], QIN Zun-yue[1,2], ZHONG Jian[2], LI Meng-ting[1], YIN Jian[1]
(1. *School of Information Science and Technology, Sun Yat-sen University, Guangzhou, Guangdong* 510006, *China*;
2. *School of Software and Service Outsourcing, Jishou University, Zhangjiajie, Hunan* 427000, *China*;
3. *School of Medical Information Engineering, Guangzhou University of Chinese Medicine, Guangzhou, Guangdong* 510006, *China*)

Abstract: Subgraph query is an important problem in the research of graph databases, and many methods about subgraph query are based on "filtering-verification strategy", which key target is to find effective feature patterns. Through the analysis of the embedding information of feature patterns in the data graphs, we propose to construct embedding relation indexing in the offline stage, and propose a new feature pattern embedding based subgraph query algorithm ERSearch. When query graph is given, we will use the co-occurrence relations and embedding relations combined to prune the unmatched data graphs, and the comparing results of embedded relationship in filtering phase can be used in the verification process, improving the efficiency of the verification. Via the experiment in the real and synthetic datasets, compared with PathIndex and other methods, we show that our algorithm can effectively reduce the size of candidate set, and effectively improve the efficiency of filtering and verification stages.

Key words: subgraph query; feature pattern; embedding relationship; graph index; graph database

1 引言

图被广泛用于复杂关系结构的表示,例如蛋白质-蛋白质交互(PPI)网络[1]、社交网络[2]、通信网络[3]、交通网络[4]等.子图查询是图数据管理[5]中的核心功能之一,根据数据源和查询目标的不同,子图查询可分为两类:一类的数据源为图集 $D=\{G_1,G_2,\cdots,G_n\}$,给定查询图 q 之后,需输出 D 中包含 q 的数据图集合;另一类子图查询是在某个大型图结构 G 中,找出与查询图 q 匹配的部分[6,7].本文研究为前一类工作.

子图同构[8]检测是子图查询中的关键操作,这已被证明是一个 NP 完全[9]问题,为了提高子图查询的效率,许多算法使用"过滤-验证"策略:首先挖掘出数据图集所包含的特征模式并构建索引;然后提取出查询图 q 包含的特征模式集合,若查询图 q 包含某特征模式 f,则所有不包含 f 的数据图 G 均不包含 q,这些图被排

收稿日期:2015-09-02;修回日期:2015-12-31;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61033010,No.61272065,No.61472453);广东省自然科学基金(No.S2011020001182,No.2014A030309013);广东省科技计划基金(No.2009B090200450,No.2010A040303004,No.2011B040200007);广东省医学科研基金(No.B2014174)

除在候选集之外;最后,对候选集进行子图同构检测.

上述策略基于特征在数据图与查询图中的共现关系实现过滤,减少了子图同构测试的次数,但由于忽略了特征在不同图中的相对位置关系,在许多实例中的过滤效果不明显. 例如,图 1 列出了 NCI(http://cactus.nci.nih.gov/ncidb2/download.html)数据库中的六个数据图 G_1,G_2,G_3,G_4,G_5,G_6 和所选特征模式 f,对于给定查询图 q_1,q_2,由于数据图和查询图均包含了 f,因此无法使用"过滤-验证"策略进行过滤.

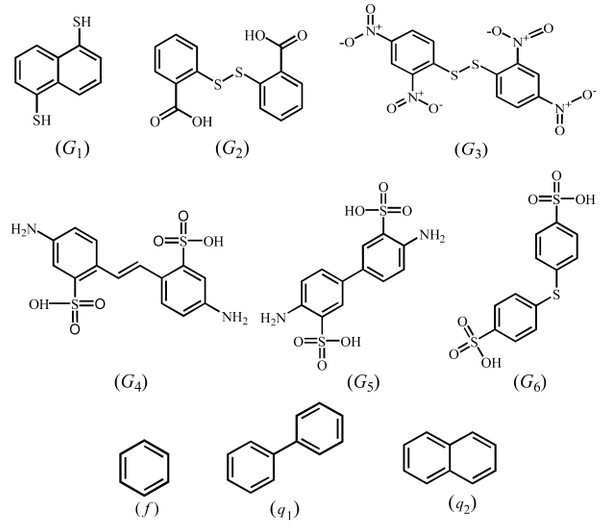

图1　子图查询示例

在图 1 中,虽然 f 在各数据图与查询图中均有两个嵌入,然而通过分析两个嵌入的相对位置关系,可发现各图存在差异. 在 G_1 和 q_2 中,两个嵌入之间有两个共同顶点;G_5 和 q_1 中的两个嵌入则具有邻接关系;G_6 中的两个嵌入含有共同的邻接点(顶点标号为 S);而 G_2,G_3,G_4 中的两个嵌入之间的距离大于 2. 由此,当给定查询图 q_1 后,可以快速过滤图 G_1,G_2,G_3,G_4 和 G_6;同理,对于查询图 q_2,仅保留 G_1 作为候选图.

通过对 NCI 与 AIDS(http://dtp.nci.nih.gov)等数据库的进一步分析表明,数据图中存在大量具有重叠、近邻等关系的特征嵌入. 将传统的子图查询算法应用到这些图集上时,存在较大的局限性:一方面因为单个数据图中嵌入了多个特征模式,导致过滤效果不佳;另一方面,重叠或者邻接的特征嵌入不仅导致多个数据图之间具有相似结构(例如图 1 中的 G_2 和 G_3 之间),同时单个数据图内也具有相似结构(例如图 1 中的 G_1,G_3),增加了同构验证难度.

有鉴于此,根据特征嵌入关系的特点,结合查询图的结构分析,提出了基于特征嵌入关系的子图查询算法 ERSearch:首先,通过对查询结果和查询图的结构进行分析,提出适用于三种不同情况的查询策略;其次,基于特征共现关系和特征嵌入关系实现快速过滤;第三,利用重叠嵌入和邻接嵌入的匹配结果提升同构验证效率.

2　相关工作

GraphGrep 算法[10]在子图查询中首先采用"过滤-验证"策略,它将图中包含的一定长度范围内的路径作为特征,基于以下规则进行过滤:包含查询图 q 的数据图 G 必然包含 q 中所有的路径. 由于路径所含结构信息较少,过滤性能较差,因此一些研究关注如何快速得到具有较高过滤性能的特征集. 文献[11]提出了基于频繁子图的索引方法 gIndex,选择具有辨别力的频繁子图作为特征索引项. FG-Index 算法[12]采用两层索引策略,硬盘上维护外层索引,使用全部频繁子图作为索引项;内层索引为 δ 容忍频繁闭图,它们被维护在内存之中. 由于频繁子树既包含了结构信息,复杂性又远小于频繁子图,文献[13]提出了基于频繁子树的图索引方法 TreePi. 文献[14]使用频繁特征子树和少量有判别力的小图作为索引项. 一方面,使用特征子树和较小的特征子图建立索引相对比较迅速,另一方面,特征子树与特征子图的结合使得算法具有较强的过滤能力.

此外,文献[15]将特征模式与其顶点的度序列、顶点近邻信息等进行联合压缩编码. 文献[16]提出了交叉过滤框架,首先使用简单特征快速过滤掉部分数据图,然后使用图特征进行深度过滤.

3　相关定义及问题描述

在本文中,图 G 定义为 $G(V,E,\Sigma,l)$,其中 $V=\{v_1,v_2,\cdots,v_m\}$ 表示顶点集合;$E=\{(v_i,v_j)\mid v_i,v_j\in V\}$ 表示边集,边数 $|E|$ 表示图的大小,文章研究无向简单图,即 $i\neq j$,且 $(v_i,v_j)=(v_j,v_i)$;Σ 为顶点标号集,$l:V\to\Sigma$ 为顶点到标号的单值匹配函数. 若无特殊说明,下文中的图均指顶点具有标号的无向连通简单图. 通过简单处理,本文方法也可用于其他简单图的子图查询.

下面依次介绍图同构、子图同构及子图查询等概念.

定义 1(图同构)　对于给定的图 $G(V,E,\Sigma,l)$ 和 $G'(V',E',\Sigma',l')$,若 G 与 G' 是同构的,则存在双射函数 $f:V\to V'$,满足:

(1) $\forall u\in V,l(u)=l'(f(u))$;

(2) $\forall u,v\in V,((u,v)\in E)\Leftrightarrow((f(u),f(v))\in E')$ 且 $\forall(u,v)\in E,l(u,v)=l'(f(u),f(v))$.

为了明确子图同构的概念,下面给出子图和超图的定义.

定义 2(子图/超图)　给定图 $G(V,E,\Sigma,l)$ 和 $G'(V',E',\Sigma',l')$,若 $V'\subseteq V,E'\subseteq E,\Sigma'\subseteq\Sigma$,且对于 $\forall v\in$

V',$l'(v)=l(v)$,则称 G'为 G 的子图,记为 $G'\subseteq G$;称 G 为 G'的超图.

定义 3(子图同构) 给定图 G 和 H,若 G 中存在子图 G'与 H 同构,则称 H 为 G 的同构子图,记为 $H\subseteq G$. 称 G'为 H 在 G 中的一个嵌入,记为 $G'\in Em(H,G)$.

在 ERSeasrch 算法中,为了简化求解,将特征 f 对应的顶点集相同的多种嵌入方式(相同标号的顶点匹配顺序不同)作为同一个嵌入进行处理.

定义 4(子图查询) 给定图集 $D=\{G_1,G_2,\cdots,G_n\}$ 及查询图 q,子图查询的目标是找到 D 中的最大子集 D_q,满足:q 为 D_q 中任一元素的同构子图,即 $D_q=\{G\mid G\in D\wedge q\subseteq G\}$.

4 基于特征嵌入关系的索引

4.1 特征嵌入关系

"过滤-验证"策略考虑了特征在不同图之间的共现关系,但是忽略了各图中的多个特征嵌入之间存在的位置差异,本文针对特征嵌入的重叠、邻接及近邻关系进行分析并建立索引. 下面首先定义特征的重叠嵌入关系.

定义 5(重叠嵌入关系) 给定图 G 及特征集合 $F=\{f_i\mid i=1,\cdots,n\}$,若存在 $f_i,f_j\in F(i,j=1,\cdots,n)$,$f_i$ 与 f_j 在 G 中的嵌入 G_i' 和 G_j' 满足:(1) $G_i'\not\subset G_j'$,$G_j'\not\subset G_i'$,且 $G_j'\neq G_i'$;(2)$R(G_i',G_j')=V(G_i')\cap V(G_j')\neq\varnothing$. 则称 G_i' 和 G_j' 之间具有重叠嵌入关系,称 $R(G_i',G_j')$ 为 G_i' 和 G_j' 的重叠嵌入顶点集,称 $R(G_i',G_j')$ 中顶点对应标号的集合(注:集合中同一标号允许重复出现)$LR(G_i',G_j')$ 为重叠顶点标号集.

在图 2 中,特征 f_1 在数据图 G 的嵌入为 $Em(f_1,G)=(\{v_1,v_2,v_3\},\{(v_1,v_2),(v_1,v_3),(v_2,v_3)\})$;$f_2$ 在数据图 G 有两个对应嵌入,分别为 $Em(f_2,G)^1=(\{v_1,v_2,v_3,v_4\},\{(v_1,v_4),(v_2,v_4),(v_3,v_4)\})$,$Em(f_2,G)^2=(\{v_1,v_5,v_3,v_4\},\{(v_1,v_4),(v_5,v_4),(v_3,v_4)\})$;$f_3$ 在数据图 G 的嵌入为 $Em(f_3,G)=(\{v_6,v_7\},\{(v_6,v_7)\})$.

其中,$Em(f_1,G)$ 与 $Em(f_2,G)^1$ 存在重叠嵌入顶点集$\{v_1,v_2,v_3\}$,对应的重叠顶点标号集为$\{a,b,c\}$;$Em(f_2,G)^1$ 与 $Em(f_2,G)^2$ 的重叠顶点集$\{v_1,v_3,v_4\}$,重叠顶点标号集为$\{a,c,e\}$.

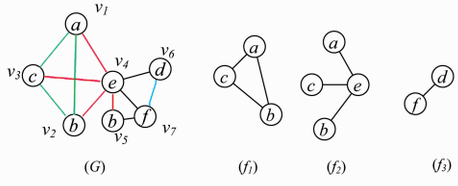

图2 嵌入关系示例图

为了定义邻接嵌入关系,首先定义邻接顶点集.

定义 6(邻接顶点集) 若 V'为图 G 的顶点集 V 的子集,V'在图 G 中的邻接顶点集为:$N_G(V')=\{v\mid(v,u)\in E(G),u\in V',v\in V,\wedge v\notin V'\}$.

例如:在图 2 的 G 图中,顶点集 $V'=\{v_1,v_2,v_3\}$ 的邻接顶点集为 $N_G(V')=\{v_4\}$.

定义 7(邻接嵌入关系) 给定图 G 及特征集合 $F=\{f_i\mid i=1,\cdots,n\}$,若存在 $f_i,f_j\in F(i,j=1,\cdots,n)$,$f_i$ 与 f_j 在 G 中的嵌入 G_i' 和 G_j' 满足:(1)$V(G_i')\cap V(G_j')=\varnothing$;(2)$A(G_i',G_j')=N_G(V(G_i'))\cap V(G_j')\neq\varnothing$或 $A(G_j',G_i')=N_G(V(G_j'))\cap V(G_i')\neq\varnothing$. 则称 G_i' 和 G_j' 之间存在邻接嵌入关系,称 $A(G_i',G_j')$ 为 G_i' 在 G_j' 中的邻接嵌入顶点集,称其顶点对应的标号集 $LA(G_i',G_j')$ 为 G_i' 在 G_j' 中的邻接顶点标号集.

在图 2 的 G 图中,$Em(f_2,G)^1$ 与 $Em(f_3,G)$ 之间存在邻接嵌入关系,$Em(f_2,G)^1$ 在 $Em(f_3,G)$ 中的邻接嵌入顶点集为$\{v_6,v_7\}$,邻接顶点标号集为$\{d,f\}$,$Em(f_3,G)$ 在 $Em(f_2,G)^1$ 中的邻接嵌入顶点集为$\{v_4\}$,邻接顶点标号集为$\{e\}$.

定义 8(2-近邻嵌入关系) 给定图 G 及特征集合 $F=\{f_i\mid i=1,\cdots,n\}$,若存在 $f_i,f_j\in F(i,j=1,\cdots,n)$,$f_i$ 与 f_j 在 G 中的嵌入 G_i' 和 G_j' 满足:(1)$V(G_i')\cap V(G_j')=\varnothing$;(2)$N_G(V(G_i'))\cap V(G_j')=\varnothing$且 $N_G(V(G_j'))\cap V(G_i')=\varnothing$;(3) $N_2(G_i',G_j')=N_G(V(G_i'))\cap N_G(V(G_j'))\neq\varnothing$. 则称 G_i' 和 G_j' 之间存在 2-近邻嵌入关系,称 $N_2(G_i',G_j')$ 为 G_i' 与 G_j' 的 2-近邻嵌入顶点集,称其对应的顶点标号集合 $LN_2(G_i',G_j')$ 为 2-近邻顶点标号集.

在图 2 的 G 图中,$Em(f_1,G)$ 与 $Em(f_3,G)$ 之间存在 2-近邻嵌入关系,其中 2-近邻嵌入顶点集为$\{v_4\}$,2-近邻顶点标号集为$\{e\}$.

4.2 特征嵌入关系索引的建立

文章选用图集中的频繁闭图作为特征模式建立特征索引树,首先给出频繁闭图的相关定义.

定义 9(频繁子图) 对于给定图集 $D=\{G_1,G_2,\cdots,G_n\}$ 与阈值 min_sup,若有子图 f,满足:$|D_f|=|\{G\mid G\in D\wedge f\subseteq G\}|\geqslant$min_sup,则称 f 为图集 D 的频繁子图,其中 $|D_f|/|D|$ 称为 f 在 D 中的支持度.

定义 10(频繁闭图) 给定图集 D 及其频繁子图集 F,频繁闭图集 CF 为 F 的一个子集,CF 仅包含 F 中相同频率的最大子图. 即:

$$CF=\{f\mid f\in F\wedge(\nexists f'\in F \text{ such that } f'\subseteq f\wedge|D_{f'}|=|D_f|)\}$$

文章采用 CloseGraph 算法[17]挖掘频繁闭图作索引项建立特征索引树(见图 3);在建立特征索引树的同时,建立特征模式到数据图嵌入的临时索引表,每条索引记录包括图号,特征模式及嵌入顶点集合三部分

内容.

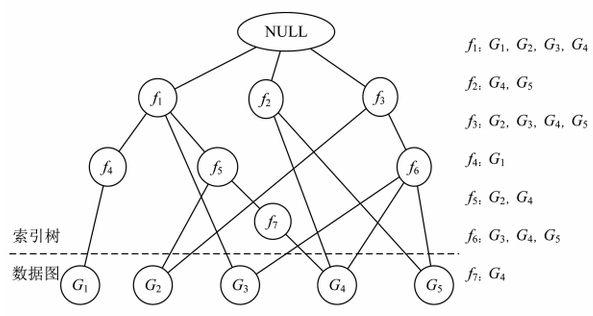

图3　特征索引树CFT

基于特征模式到数据图嵌入的临时索引表,建立特征嵌入统计表 Tab_EStat,记录单个数据图中嵌入数量大于 1 的特征模式. Tab_EStat 表中的每条记录由三部分组成,即 $\boldsymbol{Rec\_EStat} = (G, f, G\_f\_num)$,分别表示数据图,特征模式,以及特征模式在该图中的嵌入数量($\geqslant 2$).

然后建立特征模式在数据图中的三种嵌入关系的索引表.

定义 11(特征模式重叠嵌入关系索引项)　特征模式重叠嵌入关系索引项为 6 元组 $\boldsymbol{ERInd} = (G, f_1, f_2, o_1, o_2, \mathrm{ELSet})$. 其中 G 表示数据图,f_1, f_2 为两个特征,o_1o_2 分别为 f_1f_2 在 G 中的嵌入. ELSet 为 o_1o_2 中的重叠顶点标号集.

在建立特征模式重叠嵌入关系索引时,由于特征模式间存在相似结构,导致索引项的数量巨大,因此仅选用重叠部分相对较小的重叠嵌入,即 $\dfrac{|V(o_1)\cap V(o_2)|}{\min(|V(o_1)|, |V(o_2)|)}$ 小于给定阈值 ε.

定义 12(特征模式邻接嵌入关系索引项)　特征模式邻接嵌入关系索引项为 7 元组 $\boldsymbol{EAInd} = (G, f_1, f_2, o_1, o_2, \mathrm{ELSet}_{12}, \mathrm{ELSet}_{21})$. 其中 G 表示数据图,f_1f_2 为两个特征模式,o_1o_2 分别为 f_1f_2 在 G 中的嵌入,ELSet_{12} 为 o_1 在 o_2 中的邻接顶点标号集,ELSet_{21} 为 o_2 在 o_1 中的邻接顶点标号集.

定义 13(特征模式 2-近邻嵌入关系索引项)　特征模式 2-近邻嵌入关系索引项为 6 元组 $\boldsymbol{ENInd} = (G, f_1, f_2, o_1, o_2, \mathrm{ELSet})$. 其中 G 为数据图,f_1f_2 为两个特征模式,o_1o_2 分别为 f_1f_2 在 G 中的嵌入,o_1o_2 具有 2-近邻嵌入关系,ELSet 为 o_1o_2 的 2-近邻顶点标号集.

4.3　基于特征嵌入关系的过滤规则

规则 1　假设查询图 q 中存在一条特征嵌入统计记录$(q, f_i, q_f_i_num)$,若在数据图 G 中找不到记录$(G, f_i, g_f_i_num)$,使得 $q\_f_i\_num \leqslant g\_f_i\_num$,则 $q \not\subseteq G$.

规则 1 表述的思想是:若 q 为 G 子图,则任意特征模式在 q 中的嵌入数量必然不大于其在 G 中的嵌入数量.

规则 2　假设特征 f_1f_2 在图 q 中的嵌入存在重叠嵌入关系,且其重叠顶点标号集为 A;若在 G 中找不到 f_1f_2 的一组具有重叠嵌入关系,且重叠顶点标号集等于 A 的嵌入,则 $q \not\subseteq G$.

证:设 f_1 和 f_2 在图 q 中有一组具有重叠关系的嵌入 o_1o_2,其重叠顶点标号集 $LR(o_1, o_2) = A$,另 o_1o_2 合并所得图为 $o(o \subseteq q)$. 任选一组 f_1f_2 在图 G 中的重叠嵌入 t_1 和 t_2(若存在这样的嵌入,否则直接可得结论),令其合并得到图 t,设其重叠顶点标号集 $LR(t_1, t_2) = B, A \neq B$. 由已知,因为 o_1 与 t_1 同构(均与 f_1 同构),o_2 与 t_2 同构,且 $A \neq B$,故 o 与 t 不同构,故 o 不是 G 的同构子图,故 $q \not\subseteq G$.

需要说明的是,规则 2 中的 f_1f_2 可以是指同一特征,但两个嵌入必须不同.

规则 3　对于 q 中的任一特征模式邻接嵌入关系索引项$(q, f_1, f_2, o_1, o_2, \mathrm{ELSet}_{12}, \mathrm{ELSet}_{21})$,在数据图 G 中若不存在对应的索引项$(G, f_1, f_2, o_1', o_2', \mathrm{ELSet}_{12}', \mathrm{ELSet}_{21}')$,使得 $\mathrm{ELSet}_{12} \subseteq \mathrm{ELSet}_{12}'$,且 $\mathrm{ELSet}_{21} \subseteq \mathrm{ELSet}_{21}'$,则 $q \not\subseteq G$.

简证:设 q 中的嵌入 o_1, o_2 及其邻接的边组成的子图为 o,o 中 o_1, o_2 的邻接点组成的子图为 o_{12};设 G 中的嵌入 o_1', o_2' 及其邻接的边组成的子图为 o',o' 中 o_1', o_2' 的邻接点组成的子图为 o_{12}'. 由已知条件,o_1 与 o_1' 同构,o_2 与 o_2' 同构,由于 $\mathrm{ELSet}_{12} \not\subseteq \mathrm{ELSet}_{12}'$,且 $\mathrm{ELSet}_{21} \not\subseteq \mathrm{ELSet}_{21}'$,则 $o_{12} \not\subseteq o_{12}'$. 故 $q \not\subseteq G$.

规则 4　对于 q 中的任一特征模式邻接嵌入关系索引项$(q, f_1, f_2, o_1, o_2, \mathrm{ELSet})$,在数据图 G 中若不存在对应的索引项$(G, f_1, f_2, o_1', o_2', \mathrm{ELSet}')$,使得 $\mathrm{ELSet} \subseteq \mathrm{ELSet}'$,则 $q \not\subseteq G$.

简证:设 q 中的嵌入 o_1, o_2 及其 2-近邻点组成的子图为 o,设 G 中的嵌入 o_1', o_2' 及其 2-近邻点组成的子图为 o'. 由已知条件,o_1 与 o_1' 同构,o_2 与 o_2' 同构,由于 $\mathrm{ELSet} \not\subseteq \mathrm{ELSet}'$,则 $o \not\subseteq o'$. 故 $q \not\subseteq G$.

5　基于嵌入关系的子图查询算法

5.1　基于查询图结构特征的查询策略分析

子图查询的效率除了受算法本身的时空复杂度影响之外,查询图的结构特征也是影响查询效率的关键因素. 接下来讨论不同结构的查询图对查询效率的影响,并提出相应的查询策略.

对于某个给定的查询图 q,如果查询结果为频繁的,则存在特征模式 f,使得 $q \subseteq f$. 另一方面,若 q 为某个特征模式 f 的子图,则必有 $D_f \subseteq D_q$.

考虑到大部分特征模式 f 远小于数据图的大小,为

了提高验证 $q\subseteq f$ 的效率,对特征索引树进行深度优先遍历选取特征. 在搜索 q 包含的特征子集的同时,选用图的大小、图的顶点标号统计信息等对所选特征初步筛选,然后与查询图进行子图同构验证,检测包含 q 的特征. 若找到满足 $q\subseteq f$ 的特征 f,将停止遍历. 我们选择 q 的一个较大的特征子图 f',必然满足 $f'\subseteq q\subseteq f$,进一步可推出 $D_f\subseteq D_q\subseteq D_{f'}$. 因此,$D_f$ 直接作为 q 的结果集的一部分,然后仅需对 $D_{f'}-D_f$ 中的数据图进行验证检测.

对于查询图 q,若查询结果为非频繁的,我们首先挖掘 q 中包含的特征模式集,确定特征模式在 q 中的嵌入. 若 q 中存在重叠、邻接和 2-近邻嵌入关系,则可建立对应的关系索引项,并可基于规则 2-4 进行过滤. 否则从某个特征嵌入的邻接点出发,向其他特征嵌入进行扩展得到非频繁子图,然后在特征共现过滤的基础上从非频繁子图开始对候选图集进行子图同构测试,得到结果集.

下一小节将介绍基于嵌入关系的子图查询算法 ERSeasrch.

5.2 子图查询算法

对于给定图集,采用离线方式建立索引结构,见 CreateInd 算法. 第 1 行对算法使用的索引结构进行初始化. 第 2 行挖掘频繁闭图集 CF,构建特征索引树 CFT,并建立特征嵌入索引表. 第 3-11 行为各数据图建立特征嵌入索引结构.

算法 1 CreateInd **算法** //离线构建索引结构算法

输入:图集 $D=\{G_1,G_2,\cdots,G_n\}$,支持度阈值 min_sup,重叠度阈值 ε.
输出:特征索引树 CFT,特征嵌入统计表 Tab_EStat,特征嵌入关系索引表 $ERInd$,$EAInd$,$ENInd$.
步骤:
1: initialized(CFT,$EmTab$,Tab_Estat,$ERInd$,$EAInd$,$ENInd$);
2: createCFT(D) $\rightarrow$ <CFT,$EmTab$>;
3: for each $G\in D$
4: for each f in CFT
5: add(G,f,$EmTab$) $\rightarrow Tab\_Estat$;
6: if $\exists o_1\in G,o_2\in G$ and $0<\dfrac{|V(o_1)\cap V(o_2)|}{\min(|V(o_1)|,|V(o_2)|)}<\varepsilon$
7: add(G,o_1,o_2) $\rightarrow ERInd$;
8: if $\exists o_1\in G,o_2\in G$ and $isEmbAdj(G,o_1,o_2)$
9: add(G,o_1,o_2)$\rightarrow EAInd$;
10: if $\exists o_1\in G,o_2\in G$ and $isEmbN2(G,o_1,o_2)$
11: add(G,o_1,o_2)$\rightarrow ENInd$;

当给定查询图 q 之后,ERSearch 算法首先递归调用 QFMining 算法,获取 q 的特征及嵌入索引等;然后基于特征共现关系及嵌入关系过滤生成候选集,并在嵌入过滤的基础上对候选集中的数据图进行验证,得到结果集.

算法 2 ERSearch **算法** //基于嵌入关系的子图查询算法

输入:查询图 q,图集 $D=\{G_1,G_2,\cdots,G_n\}$,特征索引树 CFT, 特征嵌入统计表 Tab_EStat,嵌入关系索引表 $ERInd$,$EAInd$,$ENInd$.
输出:结果集 $C=\{G|q\subseteq G,且\ G\in D\}$
步骤:
1: $C=D$; //C 初始化为整个图集
2: $F\leftarrow\Phi$; $ER\text{-}q\leftarrow\Phi$;$Tab\_EStat_q\leftarrow\Phi$;
3: if QFMining(q,CFT,$NULL$,&F,&Tab_EStat$_q$,&ER-q)
4: find $f'\subseteq q\subseteq f$ from F;
5: $C=\{G|f\subseteq G,且\ G\in D,f\in F\}$;$C'=\{G|f'\subseteq G,且\ G\in D,f'\in F\}$;
6: foreach $c\in C'$-C
7: if verification(c,ER-q)
8: $C=C\cup\{c\}$;
9: return C;
10: else foreach $c\in C$
11: if $\exists f\in F$ and $f\not\subseteq c$
12: $C=C$-c;continue;
13: if $\exists$ 符合规则 1 ~ 4 中的过滤条件
14: $C=C$-c;continue;
15: if not verification(c,ER-q)
16: $C=C$-c;
17: return C.

算法 2 中的第 3 行调用 QFMining 算法,若 q 为某个特征 f 的子图,则根据 5.1 节的分析求出结果集(4 ~ 9 行);否则执行 10 ~ 17 行对每个数据图进行过滤-验证操作. 其中第 11 ~ 12 行基于特征共现关系对数据集进行过滤;第 13 ~ 14 行基于嵌入关系过滤规则进行过滤;15 ~ 16 行为验证过程.

算法 3 实现对查询图的挖掘,第 2 ~ 3 行搜索当前特征在 q 中的嵌入及嵌入统计信息;若当前特征在 CFT 树中的孩子结点包含了 q,则停止继续调用,向上返回其孩子结点及包含状态(5 ~ 7 行);若 q 包含了当前特征在的孩子结点,则递归调用算法 3(8 ~ 12 行);对于属于当前特征的某个嵌入 em,若其孩子结点的所有嵌入均不包含 em,则在 q 的嵌入集中加入 em(13 ~ 15 行).

算法 3 QFMining **算法**

输入:查询图 q,频繁闭图特征索引树 CFT 的当前结点 $cnode$,$cnode$ 在 q 中的嵌入.
输出:q 包含的特征模式集合 F,q 包含的特征模式的嵌入集合 ER-q,q 的特征嵌入统计表 Tab_EStat$_q$,特征与 q 的包含关系 sign.
步骤:
1: sign = 0;
2: emb(q,$cnode$)$\rightarrow ER\text{-}temp$;
3: stat(ER-$temp$,$cnode$)$\rightarrow Tab\_EStat_q$;

```
4:  foreach node = cnode. child
5:      if node⊇q
6:          F = F∪{node};
7:          return 1;
8:      if node⊂q
9:          F = F∪{node};
10:         sign = QFMining (q,node,emb(q,node));
11:         if sign = =1
12:             return sign;
13:foreach   em∈ER-temp
14:     if   em ⊈ ∀ er∈ER-q
15:          ER-q = ER-q∪{em};
16:return sign
```

5.3　算法效率分析

离线构建索引的代价主要包括特征索引树和嵌入关系索引的构建时间,构建特征索引树的时间复杂度为 $O(k|CF|\cdot|D|)$,其中 k 为特征与数据图间子图同构验证的平均代价,$|CF|$ 为闭图特征集的大小,$|D|$ 为数据图的个数. 构建嵌入关系索引的时间复杂度为 $O(|V_f|\cdot|EmSet|^2\cdot|D|)$,其中 $|V_f|$ 为特征模式所含顶点的平均数,$|EmSet|$ 为数据图所含嵌入的平均数. 为了记录每条特征嵌入关系信息,其空间复杂度为 $O(|V_f|\cdot|EmSet|^2\cdot|D|)$.

当给定 q 之后,ERSearch 算法的执行时间主要包括:(1)为 q 生成特征索引的时间 $T_{QFMining}$,主要是深度优先检验每个特征是否为 q 的子图,其时间复杂度为 $O(k|CF|)$. (2)过滤时间 T_{filter} + 验证时间 T_{ver},其中 T_{filter} 主要包括基于特征共现关系和嵌入关系进行过滤的时间,时间复杂度为 $O(|CF|\cdot|D|+|EmSet|^2\cdot|D|)$;$T_{ver}$ 为候选集与 q 之间进行子图同构验证的时间. 若查询图 q 含有特征嵌入关系的子图 g_{fe},对于数据图 g_i,其验证时间为 $T_{ver}(q-g_{fe},g_i-g_{fe})$,即只需对特征嵌入关系之外的部分进行同构测试.

6　实验结果与分析

我们采用 Java 语言编程实现算法,程序运行在配置 I7 3.4GHz,8G 内存的 Win 7(64 位)系统上.

实验选择了 AIDS,NCI 及 PubChem(ftp://ftp.ncbi.nlm.nih.gov/pubchem/)等三个真实数据集以及文献[12]中的图生成器 Graphgen 生成的模拟数据集. 对于 AIDS 数据集,采用文献[11]所选的 1 万个图作为数据图集,其平均顶点数为 25.4,平均边数为 27.4,顶点标号数为 51 个. 利用文献[11]中的 6 组大小为 n 的子图 Q_n 作为查询图集,分别为 $Q_4,Q_8,Q_{12},Q_{16},Q_{20},Q_{24}$,每个图集各 1000 个子图. 我们从 NCI 数据库选择 4 万个图作为数据图集,平均边数为 19.63,并从 NCI 的支持度大于 0.001 的子图中随机选择 5000 个作为查询图集,子图的平均大小为 8.84. 我们从 PubChem 数据集中选择 100 万个图进行大规模图集下的子图查询实验,其平均边数为 25.88,平均顶点数为 23.68,顶点标号数为 81 个.

实验将 FG-Index[12],PathIndex[15],CF-Framework[16] 与 ERSearch 算法进行对比. ERSearch 算法中的重叠度阈值 ε 取为 0.5,FG-Index 中的 δ 为 0.1,PathIndex 算法中采用 1-边路径作为特征索引,在构建特征模式索引时,各算法的最小支持度均设定为 0.1.

6.1　真实数据实验

我们首先在 AIDS 数据集上进行算法的执行效率和过滤效果的对比实验. 实验选择了查询图集 $Q_4,Q_8,Q_{12},Q_{16},Q_{20},Q_{24}$ 中的各 200 个图,通过计算其平均值进行对比. 图 4 为不同查询集中各算法的查询时间,图 5 是查询图大小变化时不同算法所确定的候选集大小. 此外,不同大小的查询图结果集的大小平均值分别为(2305.4,210.3,26.4,10.8,6.5,4.1).

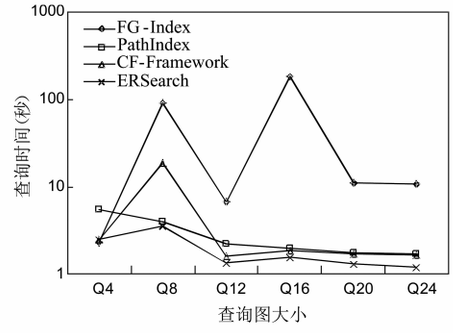

图4　不同查询图大小下的查询时间变化(AIDS)

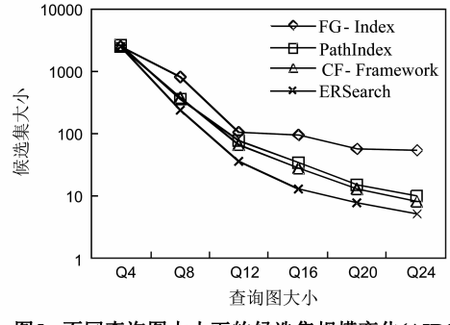

图5　不同查询图大小下的候选集规模变化(AIDS)

图 4 表明,除了 FG-Index 算法,其他三种算法在查询图大小增加时,查询时间呈下降趋势. CF-Framework 与 ERSearch 算法在 Q_8 查询集有极大值,而 FG-Index 算法有 Q_8 和 Q_{16} 两个极大值. 由图 5 可知,所有算法的候选集大小都随查询图的增大而迅速减小. 结合查询时间、候选集大小和结果集大小,可以发现 ERSearch 在不同大小的查询图集下均有较好的过滤性能,查询时间最短.

针对图4中各个算法查询时间的变化情况,原因分析如下:总体分析,随着查询图大小的增加,候选集和结果集的规模在减小,需要进行验证的数据图数量在下降,但对于单个数据图的验证时间在增加;对于 Q_4,由于PathIndex算法在利用顶点的邻接信息进行过滤处理时间较长,且候选集规模较大,过滤效果不明显,导致查询效率较其他算法略差;对于 Q_8,虽然候选集规模减小,但由于单个数据图的验证时间增加,导致FG-Index,CF-Framework与ERSearch算法均出现了极值,而PathIndex算法由于在 Q_4 中处理时间较大,此时并未出现极值;对于FG-Index算法,由于在 Q_{12} 与 Q_{16} 中的候选集规模相当,在 Q_{20} 后候选集规模有明显下降,由于单个数据图验证的时间增加,导致其在 Q_{16} 出现了另一个极值.

我们利用PubChem数据集测试大规模图集下的子图查询性能. 由于FG-Index算法的查询性能较差,我们仅对比了PathIndex,CF-Framework与ERSearch算法在PubChem数据下的执行效率,结果如图6所示. 结果显示,在大规模图集下,ERSearch算法的查询效率也有一定优势.

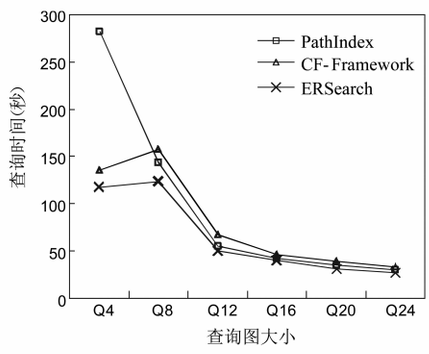

图6 不同查询图大小下的查询时间变化(PubChem_1M)

我们在NCI图集上对比不同规模的数据图集下查询时间的变化情况. 我们选择规模为1万,1.5万,2万,2.5万和3万等5组数据图集进行测试,选择了大小为8,12和16的三组各200个查询图进行测试,平均查询时间如图7~9所示. 实验结果显示,随着数据图集规模增大,四种算法的查询时间都随之增加,ERSearch略优于其他算法.

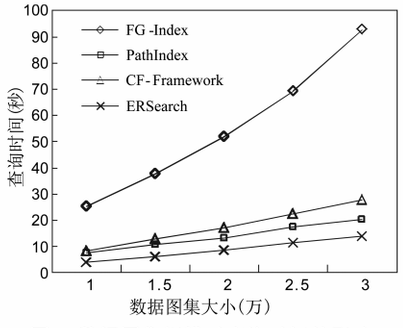

图7 数据图集规模对查询时间的影响(查询图大小为8)

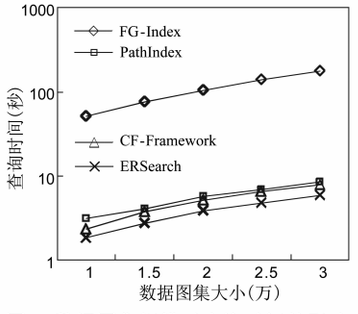

图8 数据图集规模对查询时间的影响(查询图大小为12)

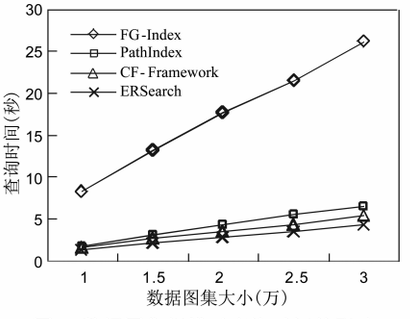

图9 数据图集规模对查询时间的影响(查询图大小为16)

6.2 模拟数据实验

利用生成的模拟数据,测试不同图密度的数据图对子图查询性能的影响. 由Graphgen生成顶点标号集大小为50,图顶点数为20,平均图密度分别为0.15,0.3,0.45和0.6的数据图各1万个,同时生成500个大小为12的图作为查询图集进行测试,图10为各算法查询时间随图集的平均图密度变化情况. 实验显示,对于顶点数一定的数据图,随着图密度的增加,图的大小呈线性增长,查询时间增速大于密度增长的速度,ERSearch算法的增速小于其他三种算法.

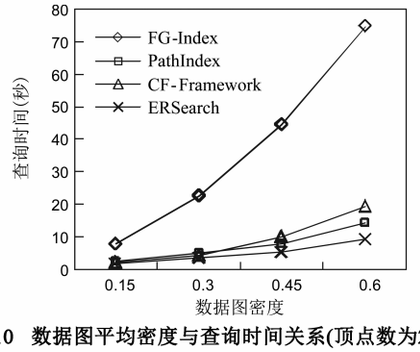

图10 数据图平均密度与查询时间关系(顶点数为20)

7 结论

实验证明,基于特征嵌入关系的ERSearch算法比FG-Index,PathIndex,CF-Framework等算法具有更优的过滤效果和查询性能. 此外,本算法在过滤阶段使用了特征共现关系与特征嵌入关系共同作用,因此本文算法易于结合现有的大部分基于特征选择的算法进行子图查询,提高查询效率.

参考文献

[1] Li Z C,Lai Y H,Chen L L,et al. Identification of human

protein complexes from local subgraphs of protein-protein interaction network based on random forest with topological structure features[J]. Analytica Chimica Acta, 2012, 718: 32 – 41.

[2] 潘磊,金杰,王崇骏,等. 社会网络中基于局部信息的边社区挖掘[J]. 电子学报,2012,40(11):2255 – 2263.
Pan Lei, Jin Jie, Wang Chongjun, et al. Detecting link communities based on local information in social networks[J]. Acta Electronica Sinica, 2012, 40(11): 2255 – 2263. (in Chinese)

[3] Stefanovic´ Č, Vukobratovic´ D, Stankovic´ V, et al. Packet-centric approach to distributed sparse-graph coding in wireless ad hoc networks[J]. Ad Hoc Networks, 2013, 11(1): 167 – 181.

[4] Wu P, Tan Y, Zheng J, et al. A hybrid compression framework for large scale trajectory data in road networks[J]. Chinese Journal of Electronics, 2015, 24(4): 730 – 739.

[5] Wang H. Managing and Mining Graph Data[M]. New York: Springer, 2010.

[6] Hung H H, Bhowmick S S, Truong B Q, et al. QUBLE: towards blending interactive visual subgraph search queries on large networks[J]. The VLDB Journal, 2014, 23(3): 401 – 426.

[7] Zheng W, Zou L, Lian X, et al. SQBC: An efficient subgraph matching method over large and dense graphs[J]. Information Sciences, 2014, 261: 116 – 131.

[8] McKay B D, Piperno A. Practical graph isomorphism, II[J]. Journal of Symbolic Computation, 2014, 60: 94 – 112.

[9] Garey M R, Johnson D S, Stockmeyer L. Some simplified NP complete graph problems[J]. Theoretical Computer Science, 1976, 1(3): 237 – 267.

[10] Shasha D, Wang J, and Giugno R. Algorithmics and applications of tree and graph searching[A]. Proc 21th ACM Symp on Principles of Database Systems[C]. New York: ACM Press, 2002. 39 – 52.

[11] Yan X, Yu P. S, and Han J. Graph indexing: A frequent structure based approach[A]. Proc of SIGMOD Conference[C]. New York: ACM Press, 2004. 335 – 34.

[12] Cheng J, Ke Y, Ng W, et al. Fg-index: Towards verification free query processing on graphdatabases[A]. Proc of SIGMOD Conference[C]. New York: ACM Press, 2007. 857 – 872.

[13] Zhang S, Hu M, Yang J. TreePi: A novel graph indexing method[A]. Proc of the 23rd ICDE Conference[C]. Istanbul, Turkey: IEEE Computer Society Press, 2007. 966 – 975.

[14] Zhao P, Yu J X, and Yu P S. Graph Indexing: Tree + Delta > = Graph[A]. Proc of the 33rd VLDB Conference[C]. New York: ACM Press, 2007. 938 – 949.

[15] Gouda K, Hassaan M. Compressed feature-based filtering and verification approach for subgraph search[A]. Proc of the 16th EDBT Conference[C]. New York: ACM Press, 2013. 287 – 298.

[16] Lee C H, Chung C W. Efficient search in graph databases using cross filtering[J]. Information Sciences, 2014, 286: 1 – 18.

[17] Yan X, Han J. Closegraph: Mining closed frequent graph patterns[A]. Proc of the Ninth ACM SIGKDD Conference[C]. New York: ACM Press, 2003. 286 – 295.

作者简介

黄　云　男,1976 年生,副教授,中山大学信息科学与技术学院博士研究生. 研究方向为数据挖掘、智能信息系统.
E-mail: huangyun109@sina.com

洪佳明　男,1984 年生,博士,讲师,研究方向为数据挖掘、医学信息处理.

覃遵跃　男,1974 年生,副教授,中山大学信息科学与技术学院博士研究生. 研究方向为数据挖掘、XML 处理.

钟　键　男,1983 年生,硕士. 研究方向为数据挖掘、社交网络分析.

李梦婷　女,1988 年生,博士. 研究方向为交通网络挖掘.

印　鉴　男,1968 年生,教授,博士生导师. 研究方向为大数据分析、机器学习等.