基于 DAG解构的图近似包含查询算法

李先通,李建中

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001, lxt@hit.edu.cn)

摘 要: 为解决图集近似包含查询, 提出一种基于图解构的 GCSS算法. 该算法通过对图集中的目标图进行解构,得到图集中子图分布情况,并利用该子图分布建立索引. 在此索引基础上的查询算法对图集进行近似包含查询,不但可以避免确定候选集的过程中产生过多子图同构测试,而且形成较小候选集,提高验证阶段效率. 实验结果表明, GCSS算法能高效完成近似包含查询任务,得到图集中被查询图近似包含的正确结果.

关键词: 图查询; 图挖掘; 近似包含

中图分类号: TP311.132 文献标识码: A 文章编号: 0367-6234(2009)06-0113-05

DAG decomposition based algorithm for graph similarity containment query

LI Xian-tong, LI Jian-zhong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, lxt@hit.edu.cn)

Abstract: An algorithm based on DAG decomposition, namely GCSS, is proposed to implement the problem of graph similarity containment query. The index is built on the subgraph distribution drawn from the decomposition of target graph in graph dataset. Deployed on such index structure, the algorithm can not only avoid extra subgraph isomorphism tests in finding out candidate answer set, but also output a smaller set to increase the efficiency of verification stage. Experimental result shows that GCSS performs an efficient graph similarity containment query and gets the right results.

Key words: graph query; graph mining; similarity containment

图具有广泛的应用性,针对图数据的研究也越来越引起人们的重视. 然而,随着图数据数量级的不断增加,支持大数据集的图查询算法的需求也日益迫切. 图查询算法的输入为一个图数据集 $D=\{g_1, g_2, \cdots, g_n\}$和一个查询图 q,输出是查询得到的结果集 $Q=\{q_1, q_2, \cdots, q_m\}$, 其中, $q_i \in D$. 按结果集 Q中元素性质不同,可将图查询算法分为两类: 1)子图查询算法; 2)包含查询算法. 其中,子图查询算法为: $\forall q_i \in Q$是查询图 q 的超图 (或表示为 $q \subseteq q_i$), 而包含查询算法为: $\forall q_i \in Q$是查询图 q的子图 (或表示为 $q \supseteq q_i$).

子图查询算法的研究取得了一定成果[1~5]. 在该类查询中, 对近似查询的研究也已经展开[6,7]. 文献[8]提出了一种解决包含查询的算法 cIndex,用于解决包含查询. cIndex构建于相对子图集 (contrast subgraph set), 并从中选择出具有判断性且无冗余的索引集, 其目的是排除索引集中结构信息的重复, 简化索引结构. 这个简化过程通过两个矩阵 (Feature-Graph Matrix和 Contrast Graph Matrix)合作完成. 而且, 在索引构建过程中,还需要查询日志的参与. cIndex算法用于解决图集包含查询问题.

子图查询与包含查询都遵循过滤与验证的方法,但由于它们使用的剪枝方法的不同,使得这两个问题对于索引项的识别上有明显的区别. 本文提出一种基于图解构的 GCSS(Graph Containment Similarity Search)算法,用于处理图集中近似包含查询问题. 实验结果表明, GCSS算法不但能高效的解决该查询问题,并且能得到正确的结果.

1 定义

作为通用数据结构,标号图能描述具有复杂

收稿日期: 2008-09-03
基金项目: 国家自然科学基金资助项目 (60773063).
作者简介: 李选通 (1973-), 男, 博士研究生;
李建中 (1950-), 男, 教授, 博士生导师.

结构的数据. 在标号图中, 结点与边分别描述实体与实体之间的关系. 与结点或边相关联的属性值被定义为标号.

定义 **1**(标号图) 一个标号图是一个四元组 $G=\{V, E, \Sigma, l\}$. 其中, V为图中节点的集合, E为图中边的集合, Σ为节点与边标号的集合, l为标号函数, 用于完成标号向节点或边的映射, $l: V \to \Sigma$或 $l: E \to \Sigma$.

给定标号图 g, 其节点的集合记为 $V(g)$, 边的集合记为 $E(g)$, 图的尺寸为该图所包含的节点数, 即 $|V(g)|$, 记为 $S(g)$.

定义 **2**(子图同构) 给定标号图 g与 g', 若存在一个单射 $f: V(g) \to V(g')$, 且满足条件:

1) $\forall v \in V(g)$, then $l(v) = l'(f(v))$.

2) $\forall (u, v) \in E(g)$, $(f(u), f(v)) \in E(g')$, then $l(u, v) = l'(f(u), f(v))$.

则称 g与 g'是子图同构的. 其中, l与 l'分别为 g与 g'的标号函数, f为 g在 g'中关于节点标号与边标号的一个映射.

给定标号图 g与 g', 若 g是 g'的子图, 则 g与 g'之间存在子图同构, 记为 $g \subseteq g'$. 此时, g'称为 g的超图, 记为 $g' \supseteq g$.

给定标号图 g与 g', 且 $S(g) \leqslant S(g')$. 若 f是此时 g在 g'中关于节点标号与边标号的一个映射, 则 g与 g'的相似度可表示为

$$d_f(g, g') = \sum [l(u) \neq l'(f(u))] + \sum [l(u, v) \neq l'(f(u), f(v))].$$

其中, u, v为图中节点, 而 (u, v)为图中边.

设 p为 g与 g'的公共子图, 且 f为 p在 g'中的一个映射, 则当 p为 g在 g'的最大公共子图时, $d_f(p, g')$的值最小. 此时, 称 $d_f(p, g')$为标号图 g与 g'之间的距离, 记为 $d(g, g')$, 即 $d(g, g') = \min(d_f(p, g'))$.

定义 **3**(图的相似度) 给定标号图 g与 g', 并给定相似度阈值 δ_{min}, 当 $S(g) \leqslant S(g')$时, g与 g'的相似度表示为

$$\delta = 1 - \frac{d(g, g')}{S(g)}.$$

当 $\delta \geqslant \delta_{min}$时, 称图 g与 g'是相似的. 此时, 称 g被 g'近似包含, 或 g为 g'的近似子图, 记为 $g \subseteq g'$.

图近似包含查询的目的是找到图集中所有与查询图 q的相似度不小于给定阈值的数据库图. 在不会产生混淆的前提下, 称数据库图为目标图.

定义 **4**(近似包含查询) 给定图数据库 $D=\{g_1, g_2, \ldots, g_n\}$和一个查询图 q, 图近似包含查询为查找所有被 q近似包含的数据库图, 即

$$A = \{g_i \mid g_i \in D \wedge g_i \subseteq q\}.$$

本文提出的算法基于无向图, 但经过简单调整, 可应用于其它标号图.

2 近似包含查询算法

图查询算法广泛采用过滤与验证方法, 即通过过滤手段缩小候选集尺寸, 再通过对候选集的验证得到准确结果. 由于验证阶段是通过子图同构测试进行的, 而子图同构测试是 NP—完全问题[9], 因此, 候选集的大小将直接影响整个算法的效率. 本文提出算法 GCSS的目的是尽量缩减候选集尺寸, 从而降低子图同构测试次数, 进而提高查询效率.

2.1 图的标准代码

图可被描述为邻接表与邻接矩阵两种形式. 然而, 无论采用邻接表或邻接矩阵, 都可以将图转化为序列进行描述. 而且, 可以选择最大或者最小的序列来与此图建立一一对应. 总而言之, 函数 $f: g \to s$, 如果 g与 g'是同构的, 则有 $f(g) = f(g')$. 否则, 有 $f(g) \neq f(g')$. 其中, g为图, s为与之对应的序列.

在本文中, 采用图的邻接表来描述图, 并使用最小邻接表作为图的标准代码.

2.2 图的解构

通过一个有向无环图(Directed Acyclic Graph, DAG)来记录图 G分解的全过程. DAG中记录了下述信息(设 p与 g为 DAG中的两个节点):

1)每个节点均是图 G的子图.

2)若 p与 g之间存在一条边, 且这条边由 p指向 g, 则 $p \subset g$.

3)若 p与 g之间存在一条路径, 且 p为路径的起始节点, g为路径的终止节点, 则 $p \subset g$. 设该路径上存在 n个节点, p, g_1, g_2, …, g_{n-1}, g, 则 $p \subset g_1 \subset g_2 \subset \ldots \subset g_n \subset g$. 称这种图的分解方式为图的解构, 该 DAG称为解构 DAG.

图 1所示为一个标号图和它的解构 DAG. 为了清晰的显示出解构 DAG与原图之间的关系, 用节点序列来描述子图. 解构 DAG中, 最长路径的起点是空图, 终点是该图本身. 可以看出, 距根节点距离相同的节点具有相同的尺寸. 而且, 不同层之间的节点具有子图与超图关系, 如节点'a'与节点'ab', 'a'位于第一层, 'ab'位于第二层, 且有一条边由'a'出发指向'ab', 因此, 'a'是'ab'的子图, 而'ab'是'a'的超图.

同时, 可以对解构 DAG进行扩展, 将其应用于整

个图集. 在对图集解构的过程中, 相同节点仅保留一个, 从而得到整个图集的解构 DAG, 用于索引的构建.

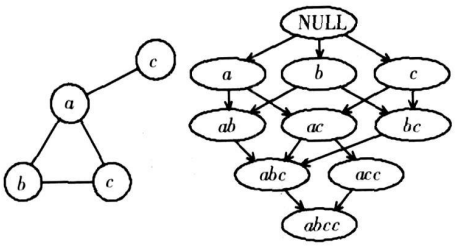

图 1　标号图与 DAG

2.3　DAG索引

DAG索引分为两部分:

1)图集 DAG;

2)图标准代码 Hash表. Hash表中的每一行数据由图的标准代码与指针组成, 该指针指向图集 DAG中与此代码相对应的节点. 查询的时候通过图的标准代码匹配 Hash表中数据, 并将匹配结果通过指针指向图集 DAG中代表图集中该相应子图的位置. 否则, 图集 DAG中不包含该子图.

DAG索引的构建, 通过图 2所示算法完成. 算法逐个检查目标图是否已经加入到 DAG中, 是, 则继续检查另一个目标图; 否, 则将这个图分解并加入到图集 DAG中. 此时, 按照生成时的关系, 添加有向边, 形成 DAG; 同时, 更新 Hash表的标准代码与指针. 在将该图分解的同时, 利用函数 Decomposition按照递归的方式对该图进一步解构, 直至子图的尺寸为 1时, 解构结束. 在这个过程中, 不但将每次解构得到的子图与原 DAG中相同的子图合并, 并对新生成的子图按其谱系关系增加有向图 DAG的边.

```
输入: 图集 D
输出: 图集 DAG, 图标准代码 Hash表
Construct(D)
H=Φ /*标准代码 Hash表 */
    DAG=Φ /*图集 DAG*/
    for图集 D中每个图
        if g不在 DAG中 then
          DAG= DAG∪ g
          H(g) = g /*将 g加入 Hash表 */
          Decomposition( g, DAG, H)
        end if
    end for
    return (DAG, H)

Decomposition( g, DAG, H)
  for g中每个节点 v
    g = g- v;
    if g' 不在 DAG中 then
      DAG= DAG∪ g',
      E_DAG= E_DAG+{ g, g'} /*将 g' 指向 g的边加入
      DAG边集中 */
      H( g' ) = g'
      Decomposition( g', DAG, H)
      end if
    end for
```

图 2　DAG索引构建算法描述

2.4　近似包含查询算法 GCSS

图近似包含算法 GCSS在图 3中给出. 算法的输入为图集 DAG与查询图 DAG, 输出则为查询候选集. 在进行查询之前, 需要先将查询图 q分解为 DAG, 记为 DAG_q, 作为输入项之一. 原因是, 在查询的过程中, 需要找到目标图与查询图的公共子图作为起点, 并逐步找到符合相似度要求的目标图, 这也是 CheckParents函数的主要工作之一.

算法是以倒序开始的, 即从 DAG_q中的 q节点开始, 逐级检查各节点的父亲节点, 以此类推. 一旦发现 DAG_q中某节点存在于图集 DAG中, 则在相似度允许的范围内, 通过 CheckParents查找目标图 DAG节点, 并将找到的目标图节点加入候选集 C中, 作为输出结果的一部分.

```
输入: 图集 DAG, 查询图 DAG_q (为查询图 q的 DAG)
输出: 候选集

GCSS( DAG, DAG_q )
  C=Φ /*候选集 */
  visited=Φ
  h= H( q)
  if h存在于 DAG中 then
    if h是目标图 then
      C= C∪ h
    end if
  end if
  for q在 DAG_q中的每个父亲节点 q'
    h= H( q' )
    CheckParents( h, DAG, DAG_q, C)
  end for
  return C

CheckParents( h, DAG, DAG_q, C)
  if h存在于 DAG中 then
      if h是目标图 then
        C= C∪ h
end if
      if h在 DAG后代中有目标图 h'满足相似性并且 h'∉ visi-
ted then
      if S( h')≤ S( q) then
        C= C∪ h'
      end if
    end if
  end if
visited= visited∪ h /*在 DAG_q中的 h */
for h在 DAG_q中的所有父亲节点 h'
    CheckParents( h', DAG, DAG_q, C)
end for
```

图 3　GCSS算法描述

3 试验结果

试验的硬件平台为: CPU采用 P4, 主频为 3.2 GHz,主存为 512 MB,操作系统采用 Windows XP Professional SP3 Chinese.试验数据采用 DTP (Developmental Therapeutics Program)提供的化合物集合,可通过网址 http://dtp.nci.nih.gov/docs/aids/aids_screen.html免费得到.

3.1 试验数据

首先,从 DTP集合中随机选取 20000个图(DTP中存在图数量 > 40000),并将这 20000个图平均分成两部分.一部分作为查询集合,而在另一部分之上进行图挖掘算法,将支持度设为 0.5% ~ 10%,得到的图集为 D_{init}.之后从 D_{init}中随机选择 10000个图作为图数据库.在查询的过程中,将查询结果集按数量划分为 8个区间: [0, 10), [10, 20), [20, 30), [30, 40), [40, 100), [100, 200), [200, 500), [500, ∞).由于算法 GCSS是第一个用于近似包含查询的算法,因此,在相同的区间里,将执行效率与候选集尺寸与 SCAN算法(将查询与图集中目标图逐个进行比较的算法)进行比较,从而得出试验结果.

3.2 试验结果

图 4所示为 GCSS算法与 SCAN算法的平均执行时间的比较.从试验结果可以看出,当结果集尺寸较小时, GCSS算法的运行时间是 SCAN算法的 1/5或 1/6,而当结果集尺寸较大时,是 SCAN

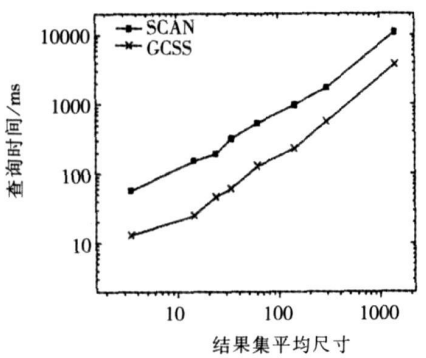

图 4 算法的查询时间

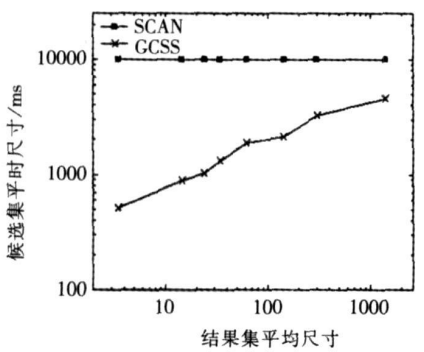

图 5 算法的候选集尺寸

算法的 1/3左右.结合图 5所示的候选集数量比较可以知道, GCSS效率的提高是由于在查询过程中得到较小的候选集,从而避免了大量的子图同构测试.然而,随着结果集尺寸的增加, GCSS的候选集与 SCAN的候选集差距逐渐缩小,导致查询时间差异的降低.

图 6所示为相似度对 GCSS算法的影响. δ_{min}增加时,算法的查询时间也相应地增加.由于图相似尺度 δ的放松,导致查询过程中待考察后代的深度增加,是引起算法查询时间增长的主要原因.

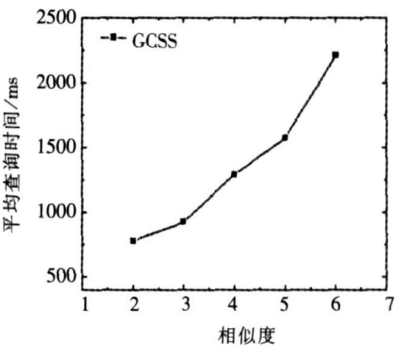

图 6 相似度对效率的影响

4 结论

1)提出了图近似包含查询算法,用于解决无精确匹配情况下或范围包含查询问题.

2)采用类似枚举的方法构建查询索引,从而减少了查询过程中可能的子图同构测试,从而提高了算法的效率.

3)试验结果表明,算法 GCSS不但可用于解决近似包含查询问题,而且能得到正确的查询结果.

参考文献:

[1] YAN X F, YU P S, HAN J W. Graph indexing: a frequent structure-based approach[C]//Proceedings of the 2004 ACM SIGMOD international conference on Management of data. New York: ACM, 2004: 335-346.

[2] CHENG J, KE Y P, NG W, et al. FG-Index: towards verification-free query processing on graph databases[C]//Proceedings of the 2007 ACM SIGMOD international conference on Management of data. New York: ACM, 2007: 857-872.

[3] ZHAO P X, YU J X, YU P S. Graph indexing: tree+ delta>= graph[C]//Proceedings of the 33rd international conference on Very large data bases. Austria: VLDB Endowment, 2007: 938-949.

[4] ZHANG S J, HU M, YANG J. TreePi: a novel graph in-

dexing method[C]//IEEE 23rd International Conference on In Data Engineering[s.l.]: ICDE, 2007: 966—975.

[5] HE H H, SINGH A K. Closure-Tree: an index structure for graph queries[C]//Proceedings of the 22nd International Conference on Data Engineering. Washington: IEEE Computer Society, 2006: 38—49.

[6] YAN X F, YU P S, HAN J W. Substructure similarity search in graph databases[C]//Proceedings of the 2005 ACM SIGMOD international conference on Management of data. New York: ACM, 2005: 766—777.

[7] YAN X F, ZHU F D, HAN J W, et al. Searching substructures with superimposed distance[C]//Proceedings of the 22nd International Conference on Data Engineering. Washington: IEEE Computer Society, 2006: 88—99.

[8] CHEN C, YAN X F, YU P S, et al. Towards graph containment search and indexing[C]//Proceedings of the 33rd international conference on Very large data bases. Austria: VLDB Endowment, 2007: 926—937.

[9] GAREY M R, JOHNSON D S. Computers and intractability: a guide to the theory of NP-completeness[M]. New York: W. H. Freeman and Company, 1979.

(编辑　张　红)

(上接第 21页)

2)简单模拟施工的分析方法可以有效地降低传统分析方法造成的误差, 其对主次施工法尤为有效.

3)在主次施工法和巨梁层采用自承重工艺的逐层施工法中, 构件在施工中的峰值内力与结构最终的内力基本相同, 构件不必因施工而进行额外的考虑.

4)在主次施工法中, 在某个巨层内进行次框架的施工时, 对其它巨层已经施工完成的结构影响非常小, 可以分别考虑而不引起太大的误差.

参考文献:

[1] 叶耀先. 中国建筑结构倒塌事故分析[J]. 建筑结构, 1990(5): 54—56.

[2] CHEN W F, LIU X L. Study of concrete framed structures during construction: special session on "Safety Consideration During Construction"[C], ASCE convention. New Orleans, Oct., 1982(10): 25—29.

[3] LIU X L, CHEN W F, BOWMAN M. Construction load analysis for concrete structures[J]. Journal of Structural Engineering, ASCE, 1985(5): 1019—1036.

[4] LIU X L. Sore-Slab interaction in concrete buildings[J]. ASCE, Journal of Construction Engineering and Management, 1986(2): 227—244.

[5] DUAN M Z, CHEN W F. Improved simplified method for slab and shore load analysis during construction[R]. Project Report: CE—STR—95—24. Indiana, USA: Purdue University, 1995.

[6] 方东平, 耿川东, 祝宏毅, 等. 施工期钢筋混凝土结构特性的计算研究[J]. 土木工程学报, 2000(6): 57—62.

[7] 李瑞礼, 曹志远. 高层建筑结构施工力学分析[J]. 计算力学学报, 1999(2): 157—161.

[8] 赵挺生, 方东平, 顾祥林, 等. 施工期现浇钢筋混凝土结构的受力特性[J]. 工程力学. 2004(2): 62—68.

[9] Comite Euro-International du Beton. CEB-FIP Model Code 1990[S]. Thomas Telford House, Switzerland, 1991.

[10] 郑文忠, 谭军, 刘铁, 等. 无支撑自承重现浇混凝土楼盖: 中国, 200410013554.1[P]. 2007—02—07.

(编辑　姚向红)