ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, Supplement, December 2009, pp.144−153

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于最小生成树的图数据库索引算法*

李 楠, 高 宏+, 李建中

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Minimal Spanning Tree Based Graph Indexing Algorithm

LI Nan, GAO Hong+, LI Jian-Zhong

(Department of Computer Science and Technology, Harbin Institute of Technology, 150001, China)

+ Corresponding author: E-mail: honggao@hit.edu.cn

Li N, Gao H, Li JZ. Minimal spanning tree based graph indexing algorithm. *Journal of Software*, 2009, 20(Suppl.):144−153. http://www.jos.org.cn/1000-9825/09018.htm

Abstract: Graphs have become popular for modeling structured data. As a result, graph indexing technique has come to play an essential role in query processing. This paper investigates the issues of indexing graphs and propose an approximation solution. The proposed approach, called MSTA, makes use of minimal spanning tree as basic indexing feature. By containment relation of edge lists and maximal common subgraph based graph distance, those minimal spanning trees are organized into an indexing structure named MST tree. MST tree can support many kinds of queries efficiently, such as subgraph queries. The performance study shows that index size and constructing time of traditional methods are tens or even a hundred times larger than MSTA.
Key words: graph database; graph index; approximation algorithm; subgraph query

摘 要: 对复杂数据进行图模式建模近几年越来越流行,因此,在查询执行的优化过程中图索引技术变得至关重要.研究了图模式的索引问题,并且提出了一种近似的索引方法,称为 MSTA 方法.MSTA 方法利用最小生成树结构作为索引特征,依据最小生成树边序列的包含关系和基于最大公共子图的图距离度量,将最小生成树组织到一个称为 MST 树的索引结构中.MST 树索引结构可以高效地支持多种查询,例如子图查询.MSTA 方法具备高效的索引性能.在索引大小和索引建立时间方面,传统方法是 MSTA 方法的数十倍,甚至上百倍.MSTA 方法虽然不能返回完整结果,但是可以返回经图距离度量排序最好的部分结果.
关键词: 图数据库;图索引;近似算法;子图查询

近年来,用图对复杂数据进行建模的应用渐渐兴起.图广泛应用于生物领域[1,2]、化学领域[3]、社会科学领域[4]、WEB领域[5]、XML领域[6],甚至是基于内容的图片检索和视频检索领域[7,8].在上述领域中,一般用无向图将结构化的数据建模成图数据,用图数据库管理及查询图数据.

在用图数据库管理及查询图数据时,随着数据量的增大,返回的结果数目也随之增加.但是在实际应用中,用户可能并不需要得到全部的结果,只需要依据某种度量返回最好的部分结果.这与搜索引擎的原理十分相似.

* Supported by the National Natural Science Foundation of China under Grant Nos.60933001, 60773063 (国家自然科学基金); the National Basic Research Program of China under Grant No.2006CB303000 (国家重点基础研究发展计划(973))

Received 2009-05-01; Accepted 2009-07-20

例如在图片检索时,用户可能只需要更贴近查询要求的几个图片;在蛋白质研究中,研究人员可能只需要返回与结构基因组同源性最相近的部分化合物;又如在 WEB 领域中,用户可能只需要系统返回最贴近查询要求的 1 个或几个网页.在上述这些应用情景中,用户更希望数据管理系统能够快速的返回满足要求最好的部分结果,而不是等待很长时间返回全部结果.数据库中的数据量越大,这种趋势就越明显.

例如,我们的实验表明,我们的方法可以用 8 秒对一个拥有 32 000 个图的数据库建立索引,查询时返回了最好的绝大部分结果,而GraphGrep[9]则需要 28 分钟建立索引,查询时返回全部结果.若假设此时我们的图数据库是应用在图片检索领域中的,在信息高速发展的当代,我们有理由相信用户更可能会花费 8 秒获得最满足要求的绝大部分图片,而不是等待 28 分钟获得全部满足要求的图片.

图距离度量广泛应用于图的相似性查询领域.研究中我们发现,在子图查询等多种查询应用中,距离更小的图是更加满足要求的图.例如,在药物化合物检测中,如果检测出的化合物与查询化合物的距离更小,那么检测出的化合物具备该功能的可能性就更强.

本文提出了一种索引方法:MSTA(minimal spanning tree based approximation)方法.MSTA 方法采用最小生成树作为索引特征,根据边序列的包含关系和基于最大公共子图的图距离度量,将所有的最小生成树组织到 MST 树索引结构中,并在该索引结构上支持了子图查询,查询返回最好的部分结果.正是由于这种近似的查询算法,使得索引性能和查询性能大为提高.传统的索引方法在索引大小和索引建立时间上是 MSTA 方法的几倍甚至几十倍.MSTA 方法的查询执行时间也要比传统方法小得多.在建立 MST 树索引结构时,MSTA 方法的主要原理是将图之间的子图关系先映射为最小生成树之间的子树关系,再映射为 MST 树中的祖先后代节点关系.

本文的主要贡献如下:(1) 提出了 MSTA 索引方法,子图查询时高效地返回最好的部分结果.MSTA 方法的索引性能大大的优于传统方法.(2) 利用最小生成树作为图数据库的索引特征,并对这种索引方式的可行性进行了理论分析和大量的实验辅证.

本文第 1 节介绍问题定义、MSTA 索引原理.第 2 节介绍 MSTA 索引方法.第 3 节介绍 MST 树上的子图查询算法.实验结果在第 4 节介绍.第 5 节给出本文的结论.

1 MSTA 索引原理

1.1 问题定义

定义 1(图). 图G是一个四元组(V,E,Σ,l),其中非空集合V代表顶点集合,$E\subseteq V\times V$代表边的集合,Σ代表标号的集合,$l:V\cup E\to\Sigma$代表为顶点和边分配标号的映射函数.

定义 2(子图). 给定图$G=(V,E,\Sigma,l)$和图$G'=(V',E',\Sigma',l')$,图G'是图G的子图,如果满足以下条件,则有$l(v)=l'(v),l(e)=l'(e)$:

(1) $V'\subseteq V$;

(2) $E'\subseteq E$;

(3) $\Sigma=\Sigma',\forall v\in V',e\in E'$.

定义 3(同构). 对于图$G=(V,E,\Sigma,l)$与图$G'=(V',E',\Sigma',l')$,若存在一个双射$f:V\leftrightarrow V'$,且f 满足下列条件时,则称G与G'是同构的:

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

(2) $\forall u,v\in V,((u,v)\in E)\Leftrightarrow((f(u),f(v))\in E')$;

(3) $\forall(u,v)\in E,l(u,v)=l'(f(u),f(v))$.

定义 4(子图同构). 给定两个图$G=(V,E,\Sigma,l)$和图$G'=(V',E',\Sigma',l')$,一个从G到G'的子图同构是一个单射函数$f:V\to V'$,其满足:

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

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

如果存在一个从 G 到 G'的子图同构,则称 G 为 G'的子图,记为 $G\subseteq G'$.

定义 5(子图查询). 给定一个图数据集合$D=\{G_1,G_2,\ldots,G_n\}$和查询图Q,一个子图查询是要从D中查找图集合S,使得$S=\{G_i|Q\subseteq G_i,G_i\in D\}$.

子图查询返回 D 中所有以查询图为子图的图.

1.2 原理分析

MSTA的索引原理可以用图 1 形象地表示出来,首先由图G计算最小生成树T_G,然后根据图Q的顶点由T_G导出一个子图T_Q,那么T_Q一定是Q的最小生成树.所以,如果我们知道了G和Q的最小生成树T_G和T_Q,且T_G包含T_Q,那么G就很可能包含Q.所以,MSTA索引方法是为T_G构造MST索引树.当需要进行子图查询时,首先计算查询图Q的最小生成树T_Q.然后搜索MST索引树,若索引中存在某一T_G,使得T_Q是T_G的子树,那么T_G所对应的G可能包含Q,将G加入到候选集中.

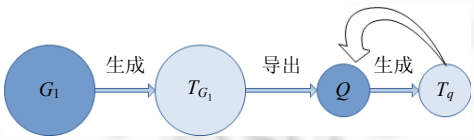

Fig.1 MSTA graph indexing theory

图 1 MSTA 图索引原理

为了更加快速地判断两棵树之间的子树关系,我们在MST树中用一种特殊的方法来组织最小生成树.在MST树中,若T_Q是T_G的子树,那么T_Q一定是T_G的祖先节点.但是出于效率方面的考虑,我们不将所有的子树关系都映射为这种祖先后代关系,而是只映射图距离最小的部分.

总的来说,MSTA 将图之间的子图关系映射为最小生成树之间的子树关系.查询时,将所有最小生成树与查询图的最小生成树具有子树关系的图加入到候选图集中.为了高效地判断生成树间的这种子树关系,我们在组织 MST 树索引结构时,尽可能地将部分距离最小的树组织到一起.

1.3 可信性分析

对于大小为n的图数据库D,查询图为q,$D_q=\{g|g\in D,g\supseteq q\}$,且不存在图$g^1$,$g^1\in D$,$g^1\notin D_q$,$g^1\supseteq q$,则$D_q$为最优化解.假设MSTA算法对于一个查询图为$q$的查询返回的结果为$D_q^1=\{g|g\in D,g\supseteq q\}$,且$D_q^1\subseteq D_q$,那么$D_q^1$为近似解.根据前面的定义,子图查询时一个求最大解的问题,即 $0\le D_q^1\le D_q$,其近似比$\rho(n)=|D_q|/|D_q^1|$.

$D_q=\{g_1,g_2,\ldots,g_n\}$,$D_q$中哪些图会被包含到$D_q^1$中,这取决于两点.首先,这与插入的顺序有关.若图$g_i\subseteq g_j$,对应的最小生成树$T_i\subseteq T_j$,对应的最小生成树的边集合$E_i\subseteq E_j$.此时,若$g_i$在$g_j$之前插入,则$g_i$会以较大的概率被包含到$D_q^1$中,否则,很有可能不被包含到$D_q^1$中.这与MST树索引的建立算法有关.其次,这与图数据集合中图的标号的分布有关.

图数据集合标号通常具有不可预测性.虽然大多数文献生成模拟数据时标号的分布符合泊松分布,但是我们也很难计算准确的近似比的量值.

1.4 补充方法

MSTA 方法是近似方法,所以不能返回全部结果.为了使子图查询能够返回更多的结果,我们可以在建立索引时对所有的图数据库中的图,按照图的大小排序,按照非递减的顺序将图插入到 MST 树结构中建立索引.这样会加大 MSTA 方法的近似度.

2 MSTA 的索引方法

在本节,我们首先介绍建立 MST 树索引结构所需要的最小生成树产生算法;然后介绍最小生成树的子树测试算法和最大公共子图距离度量的相关知识;最后详细介绍 MST 树索引结构.

2.1 最小生成树产生算法

如图 2 所示的图数据库,假设我们要以最小生成树为特征对该数据库建立索引,那么我们首先需要对每一个图求得最小生成树.由于我们所处理的图都是标号图,所以我们对Kruskal算法进行了如下改进:(1) 图$G=(V,E,\Sigma,l)$,$\forall e=(v_1,v_2)\in E$,我们将边e的权重定义为一个三元组$(l(e),l(v_1),l(v_2))$,其中$l(e)$是边e的标号,$l(v_1)$和$l(v_2)$分别是e关联的两个顶点v_1和v_2的标号;(2) 两个边e_i和e_j权重的大小是通过三元组$(l(e_i),l(v_{1i}),l(v_{2i}))$和$(l(e_j),l(v_{1j}),l(v_{2j}))$按照顺序依次比较来决定;(3) 用二元对$(v,n)$记录节点所在集合,并将其组织成平衡二叉树的形式.

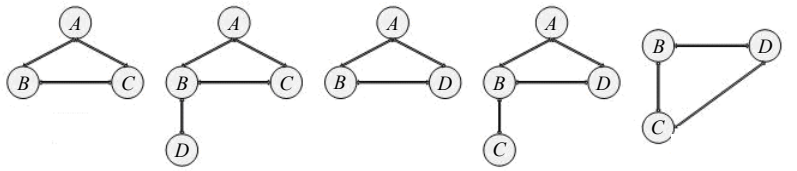

Fig.2 A sample database

图 2 示例数据库

由于我们用边及相关的两个顶点共 3 个标号构成的三元组来表示这条边,这种表示与边或顶点的 ID 无关.若一个图有多个相同标号的边,则可能会得到多个最小生成树.但这些最小生成树所对应的边的三元组序列是一样的,所以从三元组边序列角度,每个图都有唯一的最小生成树.

算法 1. GetMinimalSpanningTree.

输入:图$G=(V,E,\Sigma,l)$.

输出:最小生成树的边集合 A.

1. $A\leftarrow\varnothing$.
2. for $\forall v\in V$ do:
3. 初始化其点集合.
4. for $\forall e\in E$ do:
5. 计算其权重.
6. 建立包含所有的点集合的平衡二叉树.
7. 按照权重排序所有边,得到非递减的边集合E_{sort}.
8. for $\forall e=(v_1,v_2)\in E_{sort}$ do:
9. 查找v_1和v_2分别对应的二元组(v_1,n_1),(v_2,n_2).
10. if $n_1\neq n_2$
11. $n_2\leftarrow n_1$.
12. $A\leftarrow A\cup\{e\}$.
13. return A.

2.2 最小生成树子树测试算法

由算法 1 得到的最小生成树是一棵无根树,实际上就是一个边的子集.每条边e通过三元组$(l(e),l(v_1),l(v_2))$表示,与边及顶点的ID无关.对于图G_1,G_2的最小生成树T_1,T_2,这两棵树所对应的边集合为E_1和E_2,若T_1是T_2的子树,则$E_1\subseteq E_2$.所以,如果我们想测试T_1和T_2之间的子树关系,只要测试E_1和E_2之间的子集关系即可.

我们可以通过构造一个二分图,计算二分图的最大匹配来测试这种子集关系.首先,我们将每个用三元组表示的边看作一个二分图的顶点.若两个顶点的三元组相同,则在这两个顶点之间建立一条边.最后,在这个二分图上求最大匹配.最大匹配的大小即为两个边集合交集的大小.这个方法可以进一步通过定义边的三元组的匹配程度得到两个边集合的匹配程度,将布尔的交集程度转换为带权重的交集程度.计算最大匹配的复杂度为

$O(n^{5/2})$.

由于边的三元组是有序的,所以可以用一种简单的方法求得交集的大小.首先将两个边集合排序成有序的序列,然后顺序匹配求出交集大小.这个简单方法的复杂度为$O(n^2)$.算法过程如下:

算法 2. GetEdgeIntersect.

输入:边集合E_1,E_2,假设$|E_1|\le|E_2|$.

输出:交集大小ω.

1. 列表$L_1\leftarrow sort(E_1)$.
2. 列表$L_2\leftarrow sort(E_2)$.
3. 交集大小$\omega\leftarrow 0$.
4. 下一个匹配开始位置 $loc\leftarrow 0$.
5. for i: $0\rightarrow L_1.size$ do:
6. for j:$loc\rightarrow L_2.size$ do:
7. if 边$L_1[i]$与边$L_2[j]$匹配, then
8. $loc\leftarrow i+1$; $\omega\leftarrow\omega+1$; break.
9. 返回ω.

2.3 最大公共子图的图距离度量

在许多研究工作中阐述了最大公共子图的图距离度量,两个图之间的距离用下面的公式计算:

$$d(G_1,G_2)=1-\frac{|mcs(G_1,G_2)|}{|\max(|G_1|,|G_2|)|} \tag{1}$$

其中,$|mcs(G_1,G_2)|$是G_1和G_2的最大公共子图大小.假设G_1和G_2的最小生成树是T_1和T_2.由于图的大小是由图的顶点数目度量的,则有$|G_1|=|T_1|$,$|G_2|=|T_2|$,可得$|mcs(G_1,G_2)|\le|mcs(T_1,T_2)|=|E_1\cap E_2|$.故图距离度量可以用如下公式计算:

$$d(G_1,G_2)\ge d(T_1,T_2)=1-\frac{|E_1\cap E_2|}{\max(|E_1|,|E_2|)} \tag{2}$$

2.4 MST树结构

若要对一个图数据库建立MST树索引结构,则首先需要求出所有图的最小生成树,即用算法 1 求得一个边集合E_t.将边集合E_t排序得到有序序列L_t,L_t即为三元组序列$\{(\alpha_1,\beta_1,\gamma_1),(\alpha_2,\beta_2,\gamma_2),\ldots\}$.我们将这些边序列组织到MST树中.

在MST树中,子节点对应的最小生成树比父节点所对应的最小生成树多一条边,且子节点所对应的最小生成树包含父节点所对应的最小生成树的所有边.若父节点与最小生成树T_p相关联,子节点与最小生成树T_q相关联,T_p,T_q分别对应边集合E_p,E_q,那么T_p是T_q的子树,即$E_p\subseteq E_q$.在MST树中,有些节点与最小生成树相关联,并且与导出这棵最小生成树的图相关联;而其他一些节点是在MST树的插入中加入的,不与具体的图相关联.这些为了保证MST树每层增加一条边而填充的节点,称为哑元节点.处于MST树的第n层的节点所关联的最小生成树具有n条边.

若需要向MST树中插入某个图的索引,则首先需要对该图最小生成树的边序列排序,然后确定其插入位置.若在预计的插入位置上已经存在一个非哑元节点,则我们关联该图;若预计的插入位置上已经存在一个哑元节点,则我们同时关联该图和该图的最小生成树;若预计的插入位置上无节点存在,则建立新的节点以及到达该节点的路径上的所有节点,关联该图和该图的最小生成树.在确定插入位置的过程中,我们从根节点出发,不断地选择可以作为该最小生成树的子树的子节点前进,若有多个子节点满足要求,则利用公式(2)选择图距离最小的一个子节点.若该最小生成树有n条边,那么这个选择过程到达第n层为止.所到达的位置即为预计的插入位置.例如一个节点T_p,T_p的两个子节点为T_{c_1},T_{c_2},待插入的新节点为T_i.若T_i包含T_{c_1}和T_{c_2},那么选择$d(T_i,T_j)$最小的子节

点前进;若两个节点都不被包含,那么T_i作为T_p的第 3 个子节点插入.

对于图 2 中的图数据库,我们用算法 1 得到如图 3 所示的最小生成树边序列.G_1,G_2,G_3,G_4,G_5所对应的边序列分别为T_1,T_2,T_3,T_4,T_5.由于所有顶点的标号都相同,我们用 0 表示.于是,示例图数据库的索引结构如图 4 所示.

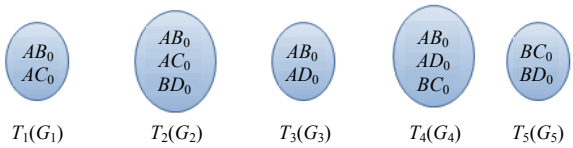

Fig.3 Edge sets of minimal spanning trees

图 3 示例图数据库的最小生成树边集合

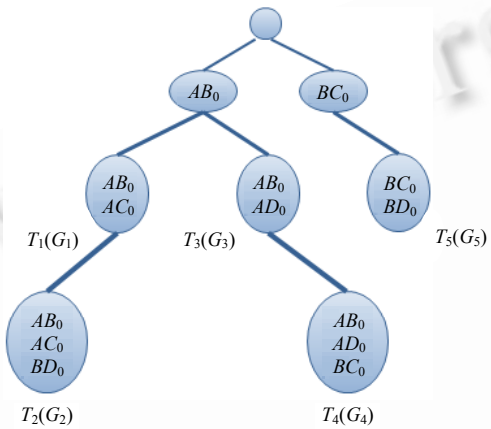

Fig.4 Structure of MST tree

图 4 MST 树结构

3 MSTA 的子图查询算法

MSTA 索引方法利用最小生成树作为索引结构;利用子图、子树和子树、父子节点两个映射关系来近似查询结果;利用插入时选择图距离最小的节点来保证返回的查询结果是距离最小的部分结果.

假设子图查询的查询图为 Q,那么查询需要返回所有包含 Q 的图.我们可以先计算 Q 的最小生成树 T,然后利用 T 的有序边序列在 MST 树中定位到一个节点.在以该节点为根的 MST 子树中,将所包含的所有图加入到候选图集合中.最后,我们利用精确的子图同构算法进行验证.算法具体描述如下:

算法 3. SubGraphQuery.

输入:查询图 Q,MST 树索引结构 *IndexTree*.

输出:结果图集合 *Ans*.

1. E_q←*GetMinimalSpanningTree*(Q).
2. L_q←*sort*(E_q).
3. *loc*←*locate*(L_q).
4. 候选图集合 *CS*←*IndexTree* 以 *loc* 为根的子树中所包含的所有图.
5. for each $g \in CS$ do:
6. if q 是 g 的子图, *Ans*←g.
7. 返回 *Ans*.

4 实验结果

本节我们将分别使用真实数据集合和模拟数据集合来评估MSTA索引方法的索引性能和查询性能.真实数据集采用NCI-HIV化合物的CI类别真实数据集作为实验用数据集.NCI-HIV化合物可在http://dtp.nic.nih.gov/下载,CI类别大约有 42 000 个化合物.模拟数据集用GIndex[10]中所叙述的方法生成.所有实验都在内存为512MB,CPU为Pentium 4,2.93GHz,操作系统为Windows XP的PC机上进行.所有算法都在STL库支持下用C++实现,用DEV内嵌的G++编译.

我们主要与GraphGrep算法[9]、GIndex算法[10]在索引大小、索引建立时间和子图查询执行时间这 3 个方面进行了对比.同时,我们计算了召回率以说明近似算法的可信性.

4.1 真实数据实验结果

对于索引性能的可扩展性分析,我们从 CI 类中随机抽取大小为 2 000,4 000,8 000,16 000,32 000 共 5 个数据集,并在这 5 个数据集上分别建立索引,然后测量索引的大小、索引的建立时间.对于子图查询性能,我们从 CI 类别随机抽取大小为 10 000 的数据集,查询 Query 大小分别为 5,10,15,20,25(根据查询图中的节点个数计算).对于每种大小的查询,我们从 CI 类别里各随机抽取 1 000 个符合大小要求的图.对这 5 组各 1 000 个查询测量查询时间等,然后取平均值.

对于GraphGrep[9]算法,我们采用路径长度lp=4,对于GIndex[10]算法,我们采用GSpan[11]挖掘频率为 10%的频繁子图作为索引特征.

4.1.1 MSTA 索引性能分析

图 5 展示了 GraphGrep,MSTA 和 GIndex 的索引大小.MSTA 的索引大小随着数据库大小呈线性增长;GIndex 的索引大小约是 MSTA 的索引大小的 2 倍左右;GraphGrep 的索引大小约是 MSTA 的索引大小的 10 倍左右.

图 6 展示了 GraphGrep,MSTA 和 GIndex 的索引建立时间.MSTA 的索引建立时间随着数据库大小呈线性增长;而 GraphGrep 的索引建立时间的增长较为显著,约是 MSTA 的 40 倍~200 倍;GIndex 约是 MSTA 的 40 倍~60 倍.

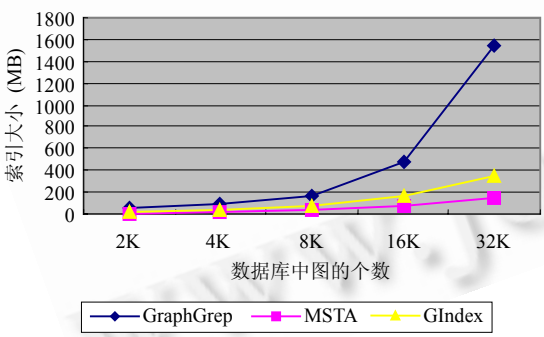

Fig.5 Comparison of index size

图 5 索引大小比较

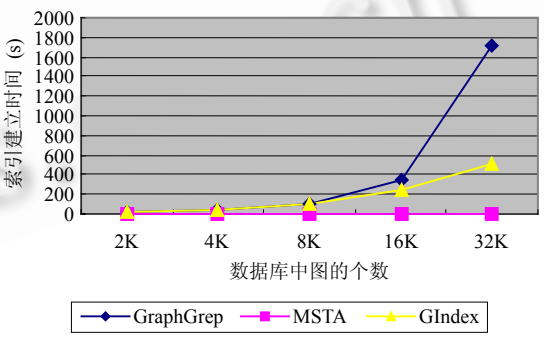

Fig.6 Comparison of construction time

图 6 索引建立时间比较

4.1.2 MSTA 查询性能分析

图 7 展示了 GraphGrep,MSTA 和 GIndex 索引下的子图查询执行时间.我们可以看到,GraphGrep 与 MSTA 的查询执行时间随着查询的增大而减小;而 GIndex 的查询执行时间随着查询图的增大而增加.GraphGrep 的查询执行时间约是 MSTA 的 2 倍~3 倍,GIndex 相对于 MSTA 的查询执行时间的倍数在 3 倍~21 倍之间增长.

图 8 是 GraphGrep,MSTA 与 GIndex 索引下的子图查询的候选集与结果集的比率.MSTA 的比率在 GraphGrep 和 GIndex 之间,约在 130~230 之间变动.

图 9 是 MSTA 索引结构子图查询执行的访问率.我们可以看出,随着查询图的增大,访问率基本成降低趋势,

都在 15%以下.图 10 展示了子图查询的召回率约在 80%~90%的结果,也就是说,MSTA 平均可以返回全部结果的 80%~90%.

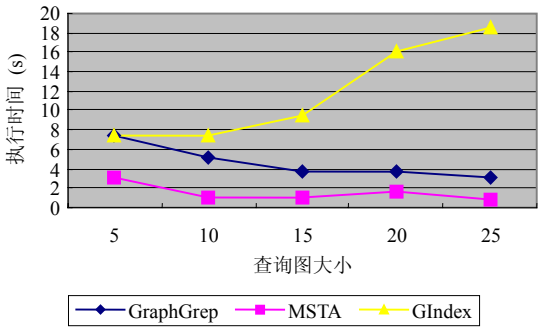

Fig.7 Comparison of query time

图 7 查询执行时间比较

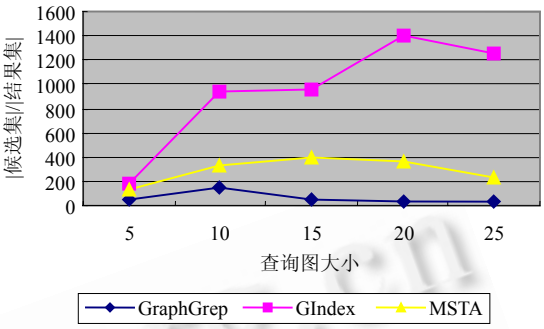

Fig.8 Comparison of accuracy of candidates

图 8 候选集合精确度比较

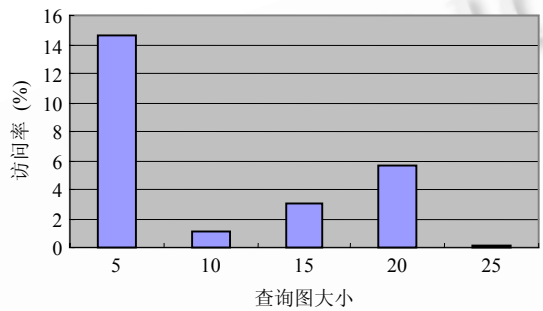

Fig.9 Access ratio

图 9 MSTA 的访问率

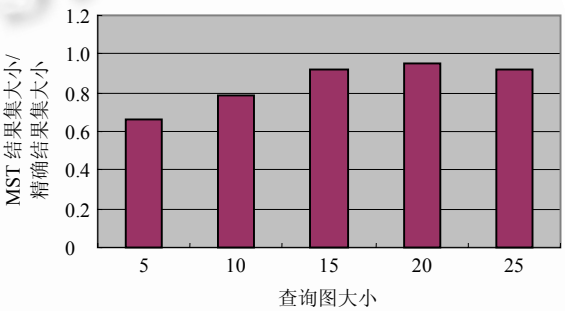

Fig.10 Recall ratio

图 10 MSTA 的召回率

4.2 模拟数据实验结果

我们用GIndex[10]中介绍的方法生成模拟数据.生成时,种子数目与要生成的图的数目的比例是 2%,种子大小的中值为 5,生成图的大小的中值为 50.我们分别生成大小为 2 000,4 000,8 000,16 000,32 000 这 5 组模拟数据,并在这 5 个数据集上分别建立索引,然后测量索引的大小、索引的建立时间.对于子图查询性能,我们生成了大小为 10 000 的模拟数据集,查询Query大小分别为 5,10,15,20,25(根据查询图中的节点个数计算).对于每个大小的查询,我们从CI类别里各随机抽取 1 000 个符合大小要求的图.对这 5 组各 1 000 个查询测量查询时间等,然后取平均值.

对于GraphGrep[9]算法,我们采用路径长度lp=4.由于模拟数据中图的密度较大, GSpan挖掘频繁子图时,即使在 2 000 大小的数据集上,挖掘支持度大于 90%的图的时间也都在 1 个小时以上,这使得GIndex不具有可比性,所以我们只对GraphGrep进行了比较.

4.2.1 MSTA 索引性能分析

图 11 是真实数据集合上的索引大小比较,从图中我们可以看出,MSTA 的索引大小随着数据库大小的增加呈线性增长.而 GraphGrep 的增长较为显著,GraphGrep 的索引大小约是 MSTA 索引大小的 3.5 倍~7.5 倍.

图 12 是真实数据集合上的索引建立时间比较,GraphGrep 约是 MSTA 的 40 倍~300 倍.图 13 和图 14 比较了真实和模拟数据之间的索引大小与索引建立时间两个指标的差异.这说明 MSTA 无论在真实数据还是在模拟数据上都很稳定.

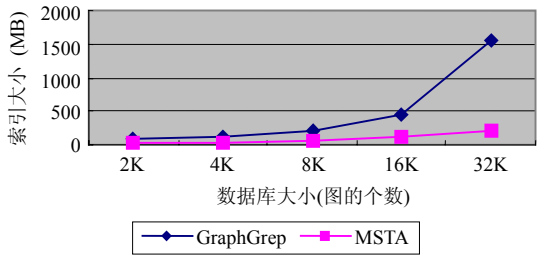

Fig.11 Index size

图 11 模拟数据集上的索引大小

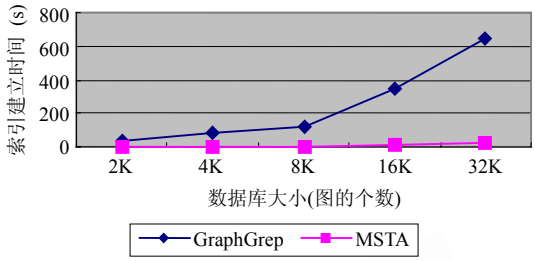

Fig.12 Construction time

图 12 模拟数据集上的索引建立时间

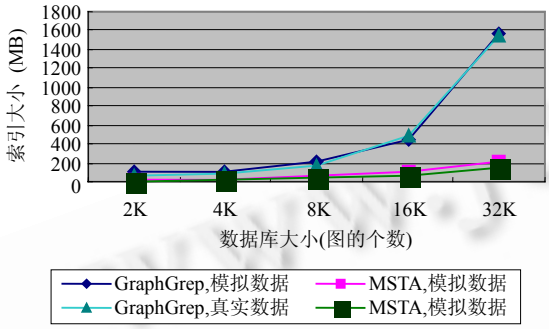

Fig.13 Index size between real and synthetic data

图 13 真实数据和模拟数据的索引大小

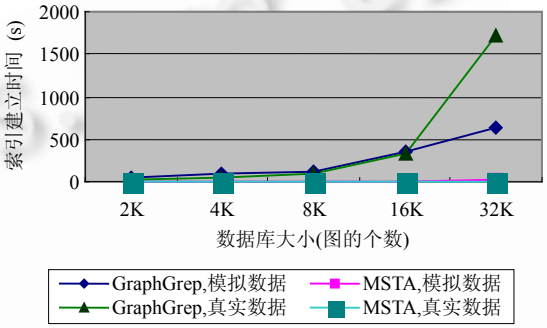

Fig.14 Construction time between real and synthetic data

图 14 真实数据和模拟数据的索引建立时间

4.2.2 MSTA 查询性能分析

图 15 和图 16 是模拟数据上的子图查询时间和召回率.查询执行时间的情况与真实数据相似,但召回率相对于真实数据偏低.

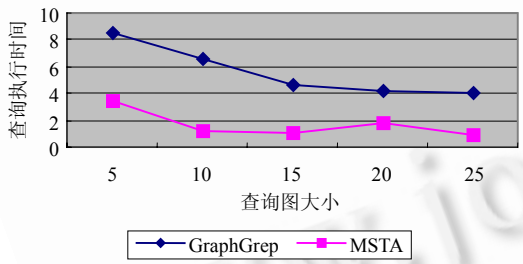

Fig.15 Query time

图 15 查询执行时间

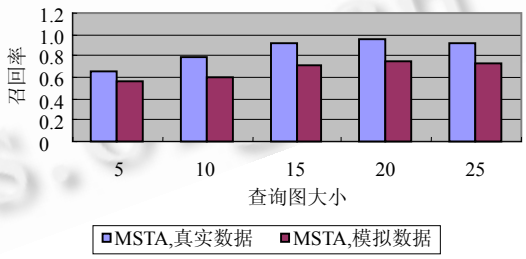

Fig.16 Recall ratio

图 16 召回率

5 结 论

本文提出了 MSTA 图索引方法.MSTA 方法利用最小生成树作为索引特征,根据边序列的包含关系和基于最大公共子图的图距离度量将最小生成树组织成 MST 树结构作为索引结构,并在此结构下高效地支持了子图查询.与传统方法相比,传统方法的索引大小和索引建立时间是 MSTA 方法的几倍甚至几百倍.MSTA 方法查询执行时间也要优于传统方法,并且具有较好的可扩展性.MSTA 方法虽然是近似方法,但是可以获得基于图距离度量最好的绝大部分结果,所以在可应用性方面有较好的表现.

致谢 感谢哈尔滨工业大学数据与知识工程研究中心的老师及同学的大力支持.

References:

[1] Berman HM, Westbrook J, Feng ZK, Gilliand G, Bhat TN, Wessing H, Shindyalov IN, Boume PE. The protein data bank. Nucleic Acids Research, 2000,28(1):235−242.

[2] Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M. KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research, 1999,27(1):29−34.

[3] National library of medicine. http://chem.sis.nlm.nih.gov/chemidplus

[4] The international network for social network analysis. http://www.insna.org/

[5] Raghavan S, Garcia-Molina H. Representing Web graphs. In: Proc. of the ICDE. 2003. 405−416.

[6] DBLP dataset. http://dblp.uni-trier.de/xml/

[7] Berretti S, Bimbo AD, Vicario E. Efficient matching and indexing of graph models in content-based retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2001,23(10):1089−1105.

[8] Lee JK, Oh JH, Hwang S. STRG-Index: Spatio-Temporal region graph indexing for large video databases. In: Proc. of the SIGMOD. 2005. 718−729.

[9] Shasha D, Wang JTL, Giugno R. Algorithmics and applications of tree and graph searching. In: Proc. of the PODS. 2002. 39−52.

[10] Yan XF, Yu PS, Han JW. Graph indexing: A frequent structure-based approach. In: Proc. of the SIGMOD. 2004. 335−346.

[11] Yan XF, Han JW. gSpan: Graph-Based substructure pattern mining. In: Proc. of the ICDM. 2002. 721−724.

李楠(1984－),女,黑龙江牡丹江人,硕士生,主要研究领域为图数据库索引技术.

高宏(1966－),女,博士,教授,博士生导师,主要研究领域为并行数据库,图数据管理挖掘.

李建中(1950－),男,博士,教授,博士生导师,主要研究领域为并行数据库,传感器网络.