基于坐标映射及多重图划分的图相似查询研究

刘哲峰[1,2],梁　平[1,2],顾进广[1,2]
(1. 武汉科技大学 计算机科学与技术学院,湖北 武汉 430065;
2. 智能信息处理与实时工业系统湖北省重点实验室,湖北 武汉 430065)

摘　要:图相似查询是图数据库资源管理最重要的操作之一。目前的相似性查询算法几乎都是采用对整个图数据库进行过滤得到候选集的方式,没有考虑在实际图数据库中各数据图规模之间存在着一定的差距,没有必要对整个图数据库进行计算。因此,提出了一种基于坐标映射的批量处理方式,从规模上对数据图进行剔除,使得后续需要计算的数据图数量大大减少。同时给出了一个参数化的、基于选择性划分的 GED 下界,使得图划分方式具有约束性,而不是随机的,并在此基础上给出了一个多层索引结构,用于 GED 下限交叉检查。模拟实验结果表明,所提出的处理方法在通过坐标映射来尽量缩减计算时间的同时,较好地提升了过滤精度,甚至能在过滤阶段就得到相似查询的结果。

关键词:图数据库;图相似查询;坐标映射;选择性图划分;多层索引结构
中图分类号:TP311　　**文献标识码**:A　　**文章编号**:1673-629X(2023)12-0058-07
doi:10.3969/j.issn.1673-629X.2023.12.008

Research on Graph Similarity Query Based on Coordinate Mapping and Multigraph Partition

LIU Zhe-feng[1,2],LIANG Ping[1,2],GU Jin-guang[1,2]
(1. School of Computer Science and Technology,Wuhan University of Science and Technology,
Wuhan 430065,China;
2. Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,
Wuhan 430065,China)

Abstract:Graph similarity search is one of the most important operations in graph database resource management. Currently,most similarity search algorithms filter the entire graph database to obtain a candidate set,without considering the significant differences in the size of the data graphs of the actual graph database,so it is not necessary to calculate the entire graph database. A batch processing method based on coordinate mapping is proposed to remove data graphs from the graph database,which greatly reduces the number of data graphs that need to be calculated subsequently. Moreover,a parameterized and selective partition-based GED lower bound is given to make the graph partitioning method constrained rather than random. Based on this,a multi-level index structure is provided for GED lower bound cross-checking. Simulation results show that the proposed processing method not only minimizes the calculation time through coordinate mapping but also improves filtering accuracy. Furthermore,it can even obtain the results of similarity queries in the filtering stage.

Key words:graph database;graph similarity search;coordinate mapping;selective map partitioning;multilayer index structure

0　引　言

随着信息技术的迅速发展,所获得的数据呈爆炸式增加。根据国际数据公司(IDC)的统计,互联网上的数据每年将增长 50% 以上,而且世界上 90% 以上的数据是最近几年产生的[1]。这些数据中,容量巨大、结构复杂、种类繁多而且语义多变的数据占据了很大比例[2-3]。

总体而言,在数据及数据间的关系飞速增长的时代,图数据库被应用到越来越多的领域,如机电电力[4-5]、医学信息[6]、信用卡欺诈检测[7]等。而图形的

收稿日期:2023-02-15　　**修回日期**:2023-06-15
基金项目:国家社会科学基金重大项目(11&ZD189)
作者简介:刘哲峰(1999-),男,硕士研究生,研究方向为图数据库相似搜索和子图匹配;通信作者:梁　平(1975-),女,讲师,博士,研究方向为数据库技术、数据库恢复和数据挖掘;顾进广(1974-),男,教授,博士,CCF 杰出会员(05460D),研究方向为语义网与知识图谱、分布式计算等。

激增也激发了人们对在大型图数据库中实现高效访问功能和灵活的结构感知查询的兴趣[8]。关于图数据库的查询方式,有检索条件约束[9]以及查询结果约束两大类,而基于查询结果约束,已经出现了许多相关的研究,它们可以大致分为两类:图精确搜索[10-11]和图相似性搜索[12-13]。与精确搜索相比,相似性搜索可以提供一种鲁棒的解决方案,允许容错并支持搜索未精确定义的模式。

两个标记图之间的相似性计算是图相似性搜索的核心操作,目前最主要的相似性度量为:图编辑距离[13-14]、最大公共子图[15-16]、图对齐[17]和图核函数[18-19]。该文考虑基于图编辑距离(GED)约束定义的相似性搜索问题,主要是因为其通用性和广泛的适用性[20],GED 是一种几乎适用于任何类型的图形的度量。直观的图编辑操作可以精确捕捉与图形结构和内容相关的任何细粒度差异[21]。典型的编辑操作[22]是插入和删除顶点或边以及重新标记顶点或边。

该文研究了以下图相似性搜索问题:给定一个图数据库 $G=\{g_1,g_2,\cdots,g_n\}$ 和一个查询图 q,需要找到图 G 中的所有图 g,使其相对于查询图 q 的图编辑距离 GED(g,q)在用户指定的阈值 τ 内。

然而,GED 的计算是 NP-hard,因此计算所有查询图 q 和 $g\in G$ 的 GED 可能会导致较差的计算效率。现有的解决方案大多采用过滤验证框架来加快图形的相似性搜索。在过滤阶段,通过种种算法相对快速地评估出一个 GED 下界,它被用于从图数据库中修剪尽可能多的假阳性图,剩下的图构成候选集 C。在验证阶段,必须在 q 和每个候选图之间执行 GED 计算来验证。

到目前为止,已经提出了不同的 GED 下界[13,21,23-24],这些技术具有一些缺点:(1)对于图数据库 G 中的所有数据图进行 GED 下界的评估,没有考虑到图规模的差距,会导致很多无意义的 GED 下界计算;(2)过于宽松的 GED 下界并没有保证其过滤能力,导致很多假阳性图无法识别;(3)GED 下界评估耗时,会带来较高的验证成本。

该文考虑了一种基于坐标映射的多重图划分方法。首先,将图数据库以及查询图进行坐标系的映射,并以此为基础构建坐标系上的查询矩形对图数据库中不符合条件的图进行粗粒度过滤,以降低后续图划分的计算成本。这种过滤方式十分快速,但是并不精密;然后,使用基于参数的图划分的选择性计算方法有效地控制图划分的随机性,并基于该方法得到有效的 GED 下界,来提升过滤精度;最后,通过不同的划分方式以及参数控制构建不同的索引集,以实现多层索引过滤,更进一步提升过滤的精度。

1　问题定义

1.1　图与相似查询

该文采用的图均为简单的、无向的标记图。图 g 定义为一个四元组 (V_g,E_g,l_g,Σ),其中 V_g 是顶点集;$E_g\subseteq V_g\times V_g$ 是边集;$l_g:V_g\cup E_g\to\Sigma$ 是一个标签函数,其中 Σ 是顶点和边的标签集。

图 g 可以被执行的图编辑操作为:(1)删除一条边;(2)在两个顶点间插入一条边;(3)修改一条边的标签;(4)删除一个孤立的顶点;(5)插入一个孤立的顶点;(6)修改一个顶点的标签。

定义 1(相似查询):给定一个图数据库 $G=\{g_1,g_2,\cdots,g_n\}$,一个查询图 q 和一个 GED 阈值 τ,相似查询的问题是找到一个数据图 $g_i\in G$,使得 GED(g_i,q) $\leqslant\tau$。

1.2　半边图与图划分

定义 2(半边图):半边图 $g=(V_g,E_g,l_g,\Sigma)$,其中,$E_g\subseteq V_g\times(V_g\cup\{*\})$ 是一个可能存在半边 $(u,*)\in E_g$ 的图,其中一个关联顶点 $u\in V_g$ 是一个确定的顶点,但该半边的另一个顶点(及其标号)没有明确的指定,表示为 $*$。

定义 3(图划分):图 g 可以划分为集合穷举、互斥和非空半边图的集合 p,如下所示:

$$P(g)=\{p_i\mid \cup_i V_{p_i}=V_g,\cup_i E_{p_i}\subseteq E_g\cup V_g\times\{*\},\ p_i\cap p_j=\varnothing,\forall i,j\neq j\}$$

其中,P 被称为 g 的一个划分。

将图划分为半边子图对 GED 有一个明显的优势,就是在给定任何图编辑操作的情况下,它最多只能影响一个半边图的划分。

2　基于坐标映射的多重图划分

2.1　坐标映射

在现实的图数据库中,图的规模均有较大的差异,而在图编辑操作中,对于顶点和边的删除或增加均是从图规模上进行操作。基于这一点,设计了一个坐标映射的批量过滤方法。

对于任一数据图 g,将其以坐标 $(|V_g|,|E_g|)$,也就是基于顶点-边的方式,映射于二维坐标系中,其中,x 坐标和 y 坐标分别代表图 g 的顶点数和边数。因此,对于图数据库 G,可以获得点集 $\{(|V_g|,|E_g|):g\in G\}$,而这些点可以组成一个矩形区域 $\boldsymbol{A}=[x_{\min},x_{\max}]\times[y_{\min},y_{\max}]$,其中 $x_{\min}/x_{\max}$ 和 $y_{\min}/y_{\max}$ 分别是节点和边的最小数量和最大数量。

定义 4(查询矩形):给定一个查询图 q 和阈值 τ,查询矩形 $\boldsymbol{A}_q$ 被定义为由点集 $\{(x,y):|x-|V_q||+|y-|E_q||\leqslant\tau\}$ 组成的矩形。

对于一个数据图 g,如果 GED(g,q) $\leqslant\tau$,那么必

定会有：$\| V_g | - | V_q \| + \| E_g | - | E_q \| \leqslant \tau$。由此可以在矩形区域 $\boldsymbol{A}$ 中进行图规模上的批量过滤，圈定出如图 1 所示的查询矩形 $\boldsymbol{A}_q$。

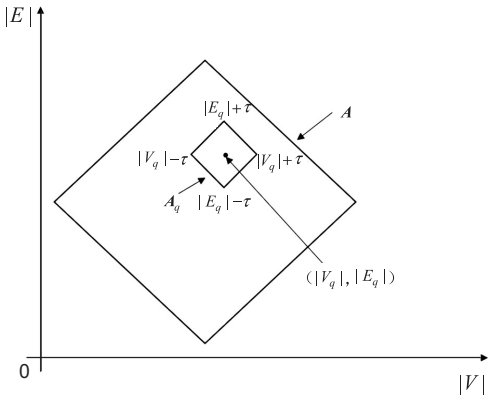

图 1 图数据库的坐标矩形 $\boldsymbol{A}$ 和查询矩形 $\boldsymbol{A}_q$

坐标映射的着眼点在于图的大小，考虑的是一种边界情况，这有助于从图规模上剔除了那些不符合要求的图，是一种粗粒度的过滤手段，以减少后续精细过滤的时间开销。因此，经过这坐标映射这一层的粗粒度过滤之后，所需要参与后续多重图划分的细粒度过滤的数据图数量明显减少，能显著加快后续细粒度过滤方法的索引构建时间。

算法 1 给出了坐标映射方法的算法描述。首先，初始化坐标矩形 $\boldsymbol{A}$ 的最大值和最小值(行 1)；然后，对于图数据库中的每个图，计算其顶点数和边数并更新 $\boldsymbol{A}$ 的最大值和最小值(行 2 ~ 6)，并将其坐标存储到坐标矩形 $\boldsymbol{A}$ 中(行 7)；接着，使用查询图的顶点数和边数计算查询矩形 $\boldsymbol{A}_q$，并找到与查询图相似的图的坐标(行 9 ~ 11)；最后，返回坐标矩形 $\boldsymbol{A}$ 和查询矩形 $\boldsymbol{A}_q$(行 12)。

算法 1：坐标映射方法

输入：图数据库 G；阈值 τ；查询图 q；

输出：坐标矩形 $\boldsymbol{A}$ 和查询矩形 $\boldsymbol{A}_q$

1→ $x_{\min}$ = infinity, $x_{\max}$ = −infinity, $y_{\min}$ = infinity, $y_{\max}$ = −infinity;

2→for each g in G do

3→ · · · · if $|V_g| < x_{\min}$ then $x_{\min} = |V_g|$;

4→ · · · · if $|V_g| > x_{\max}$ then $x_{\max} = |V_g|$;

5→ · · · · if $|E_g| < y_{\min}$ then $y_{\min} = |E_g|$;

6→ · · · · if $|E_g| > y_{\max}$ then $y_{\max} = |E_g|$;

7→ · · · · $\boldsymbol{A}$.add(($|V_g|$, $|E_g|$));

8→ $\boldsymbol{A}_q$ = set();

9→for each g in G do

10→ · · if abs($|V_g| - |V_q|$) + abs($|E_g| - |E_q|$) $\leqslant \tau$ then

11→ · · · · · · · $\boldsymbol{A}_q$.add(($|V_g|$, $|E_g|$))

12→return $\boldsymbol{A}$, $\boldsymbol{A}_q$

2.2 参数化的 GED 下界

由 1.2 中的定义 3，可以推导出如下定理：

定理 1：给定一个图 g，该图被划分为($\tau + k$)个半边图，该半边图的集合为 $P(g)$，其中 τ 为阈值，k($k \geqslant 1$)是一个整数参数。对于给定的查询图 q，如果 $GED(g,q) \leqslant \tau$，则在半边图集合 $P(g)$ 中至少存在 k 个半边图：$p_{i_1}, p_{i_2}, \cdots, p_{i_k} \in P(g)$，满足 $p_{i_l} \subseteq q (1 \leqslant l \leqslant k)$。

对于任意数据图 $g \in G$，其分区集合为 $P(g) = \{p_1, p_2, \cdots, p_{\tau+k}\}$，如果 $p_i \subseteq q$，则称 p_i 为匹配分区，否则称其为不匹配分区。根据定理 1，如果分区集合 P 中的匹配分区数小于 k，则图编辑距离 GED(g,q)必然是大于 τ 的，由此可判断出 g 是一个假阳性图，大大节约了时间成本。同时，为了更好地了解图划分方法的使用，给出了例子 1 进行说明。

例子 1 对于图 2 中的数据图 g_1 以及查询图 q，假设阈值 $\tau = 2$。令整数参数 $k = 2$，那么 g_1 需要划分出四个半边图。对于图 g_1 的划分 $P(g_1)$ 如图 2 所示。因为 $P(g_1)$ 相对于图 q 来说有三个不匹配分区：p_1，p_2 和 p_4。因此 g_1 是一个假阳性图。

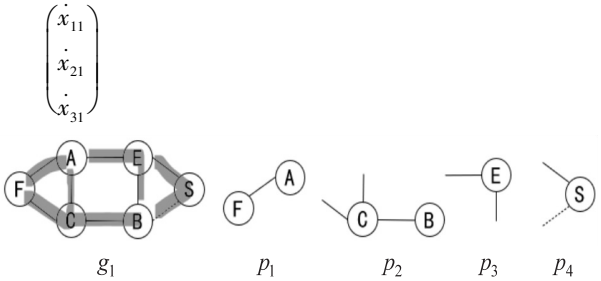

图 2 g_1 的四个半边图：$P(g_1) = \{p_1, p_2, p_3, p_4\}$

定理 1 提供了一个参数化的，基于分区的 GED 下界，通过设置不同的 k 值，就可以生成一系列新的 GED 下界。同时当 k 取不同的值时，新生成的 GED 下界的过滤能力也具有了一定的差异。当 $k = 1$ 时，实例化的 GED 下界归结为一个特殊的降级情况。对于给定的一个假阳性图 $g \in G$，它的($\tau + 1$)个分区，可以很容易地找到 1 个($k = 1$)匹配分区，一旦找到，g 将被错误的纳入候选图中，然而当 $k > 1$ 时，g 作为假阳性图将更有可能被识别和过滤，因为从 g 中检测 $k > 1$ 个匹配分区的可能性要远小于 $k = 1$ 时。

因此，$k > 1$ 时，它比 $k = 1$ 这种退化情况更有可能识别出假阳性图。当 $k > 1$ 时，GED 下界可以实例化为一系列更紧密的下界，对相似性搜索具有更好的过滤能力。

2.3 选择性图划分

对于给定的任意一个数据图来说，可以有很多种划分方法，能将其划分为($\tau + k$)个不同大小和结构分区。因此如果不加以限制，图划分方法会充满随机

性,针对这种情况同样给出了例子 2 进行说明。

例子 2　同样是图 2 中的数据图 g_1,如果采用图 3 的划分方式, $P_1(g_1)$ 中, p_2 和 p_3 均为匹配分区,此时数据图 g_1 会通过筛选。

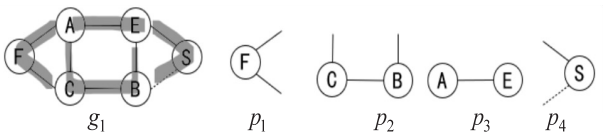

图 3　g_1 的四个半边图: $P_1(g_1) = \{p_1, p_2, p_3, p_4\}$

为了对这种随机划分方式进行限制,为每个分区赋予了一个如式 1 所示的选择性增益 $s(p_i)$,它表明如果 g 是假阳性图, p_i 作为不匹配分区 $p_i \not\subset q$ 的选择性有多高,其中 $f(\,)$ 是图数据库 G 中的顶点/边的标签频率。为此,对于数据图的最优划分的目标是将数据图划分为具有最高总体选择值的($\tau + k$)个分区。

$$s(p_i) = \frac{|V_{p_i}| + |E_{p_i}|}{\sum_{v \in V_{p_i}} f(l_v)/|V_{p_i}| + \sum_{e \in E_{p_i}} f(l_e)/|E_{p_i}|} \quad (1)$$

式中, $s(p_i)$ 的值受两个因素的影响:

(1)分区大小:较大的分区更有可能受图编辑距离的影响,从而使得该分区成为不匹配分区的概率更大;

(2)顶点/边标签的频率:在查询图 q 中,标签频率在 G 中较小的分区可能按比例很少出现。因此,包含低频率顶点/边的分区更有可能是不匹配分区。

选择性划分的算法如算法 2 所示。对于输入的数据图 g,首先创建一个布尔数组 M,它表示是否将顶点 v 分配给某个分区,并将 M 初始化为 false(行 1);同时初始化另一个集合 Una,它保存即将处理的数据图 g 的未分配顶点,初始化为空(行 2);接下来选择 $\tau + k$ 个顶点,以此作为 $\tau + k$ 个分区的初始顶点,同时这些顶点的相邻且未被分配的顶点 $N(\cdot)$ 添加到 Una 中,这些存放在 Una 中的顶点将在下一步中考虑用于分区分配(行 3 ~ 8);接着通过评估将每个顶点 v 分配给每个现有分区 p_i 的选择性增益(由式 1 计算)来检查每个未分配的顶点,使得每个顶点分配到该分区后具有最大的选择性增益,并将顶点 v 的相邻且未被分配的顶点添加到 Una 中(行 9 ~ 15);在所有顶点都分配完成后,就需要开始考虑跨区的边,并将它们作为半边分配给其中一个参与分区,同样需要计算所分配半边的选择性增益(行 16 ~ 22)。这样通过贪心算法,将每一个除了初始化顶点以外的顶点和跨分区的边进行选择性增益的计算,最终返回相对最优的分区集合。

算法 2:选择性图划分

输入:数据图 g,阈值 τ,参数 k,查询图 q

输出:图划分 $P(g) = \{p_1, p_2, \cdots, p_{\tau+k}\}$

1→for each $v \in V_g$ do $M[v]$ = false;

2→Una = $\emptyset^*$, / * Una 为未分配顶点集合 * /

3→for i = 1 to $\tau + k$ do

4→· · · · for one $v \in V_g$ and $M[v]$ = = false

5→· · · · · · · · $p_i = \{v\}$;

6→· · · · · · · · $M[v]$ = true;

7→· · · · · · · · for each $u \in N(v)$, $M[u]$ = = false and $u \notin$ Una

8→· · · · · · · · · · · · · Una = Una$\cup \{u\}$;

9→for each $v \in$ Una

10→· · for i = 1 to $\tau + k$ do

11→· · · · · · · $e_i = s(p_i \cup \{v\}) - s(p_i)$;

12→· · $p_i^* = p_i^* \cup \{v\}$ where $i^* = = i$ in max e_i;

13→· · $M[v]$ = true;

14→· · for each $u \in N(v)$, $M[u]$ = = false and $u \notin$ Una

15→· · · · · · · Una = Una$\cup \{u\}$;

16→for each $(u,v) \in E_g$, u and $v \in p_i$, $i \neq j$

17→· · $e_i = s(p_i \cup (u, *)) - s(p_i)$;

18→· · $e_j = s(p_i \cup (v, *)) - s(p_i)$;

19→· · if $e_i \geqslant e_j$ then

20→· · · · · · · $p_i = p_i \cup \{(u, *)\}$;

21→· · else

22→· · · · · · · $p_i = p_i \cup \{(v, *)\}$;

23→return $P(g) = \{p_1, p_2, \cdots, p_{\tau+k}\}$

顶点和边标签的频率可以在构建坐标矩形阶段进行一次预计算,因此,该算法的时间复杂度为 $O((\tau + k)|V_g| + |E_g|)$。

2.4　标签频率表与倒排索引

上述的选择性图划分方式给原本的随机划分提供了一个约束,使得不稳定的随机划分变的更加可控。然而半边子图的匹配算法同样具有很高的代价,因此为了进一步降低半边子图匹配的验证成本,可以构建以下两个表进行进一步过滤:

(1)标签频率表:为分区 p 构造一个标签频率表 $R(p)$,其中存储分区 p 的顶点和边标签的频率。在执行半边子图同构的计算之前,首先比较它们的频率表 $R(p)$ 和 $R(q)$:对于 $R(p)$ 中每个顶点/边标签的频率不应该超过 $R(q)$,记为 $R(p) \leqslant R(q)$,这样可以节省半边子图同构的计算代价;

(2)倒排索引:为图 g 的一个划分 p 构建一个倒排索引 $I(p)$。对于给定的查询图 q,如果 $p \subseteq q$,可以在图数据库 G 中快速找到包含划分 p 的所有数据图。这样,在进行完必要的划分 p 的半边子图同构计算后,可以通过倒排索引找到所有包含划分 p 的数据图,以此减少大量的计算。

通过构建与维护上述两个表单,将子半边子图匹配的过程同样划分为过滤-验证两个部分,以进一步压缩计算成本。

2.5 多层索引

虽然可以基于上诉方法选择一个紧凑的、基于划分的 GED 下界,并利用选择性划分的方法产生相对优秀的半边子图,然而可能仍然有一些假阳性图未被识别。因此选择采用多层索引结构来对图数据库进行过滤。

为了充分利用多个 GED 下界,采用集中过滤策略,考虑了 L 种不同的图划分方法, $P_1,P_2,\cdots,P_L$,其中 P_i 将 $g \in G$ 划分为($\tau + k_i$)个分区。具体来说,在第 i 层,评估由 k_i 参数化的第 i 个 GED 下界,并生成候选图 C_i ,并由该候选图 C_i 继续进行下一层的过滤,当且仅当通过了所有层的 GED 下界评估之后,才能得到最终的候选集 C 。

为了保证索引分区和 GED 下界约束能在多层索引结构的不同层之间显著变化,考虑了以下策略:

(1)在不同层应用选择性分区方法时,从数据图 g 中随机选择初始节点;

(2)在不同层间选择不同的参数 k_i 的值。

这样可以使用相同的选择性分区方法,通过不同的初始节点和不同的参数 k_i ,保证每层索引都能产生不同的分区索引集,以此为 GED 下界提供严密性保证。

3 实验结果及分析

3.1 实验环境

为了检测该算法的效率与准确性,将与 Pars 算法[23]进行对比实验。Pars 算法是采用单个降级 GED 下界($k=1$)和随机划分进行索引生成和相似性搜索的图索引方法,相较于其他的算法具有更好的效率,Pars 算法所采用的半边子图同构算法与文中算法均为 A * 算法。同时为了体现坐标映射所带来的性能提升,将对文中算法以及 Pars 算法进行有无坐标映射的对比实验。

而对于文中算法,为了体现出在 2.5 节中提出的多层索引结构的过滤层数对算法的影响,实验中选取双层索引结构和三层索引结构进行对比。

随机从 NCI/NIH 数据集中选择 10 000 个图作为数据集,并从中随机选择一个图作为查询图。

在实验中主要考虑了以下性能评估指标:

(1)索引构建成本:包括参与索引算法的图数量、索引构建时间;

(2)候选图数量:即在进行过滤之后所产生的候选图数量;

(3)查询执行时间:相似性搜索的真实响应时间,是候选集生成时间及 GED 验证时间之和,同样这里的时间是多次实验后平均的响应时间。

实验运行环境如下:Window11 操作系统、16 GB 内存的 Core i7-11800H 2.30 GHz 8 核电脑;Pars 算法和文中算法均采用 C++编程语言实现,编译环境为 Microsoft Visual Studio 2022。

3.2 实验结果分析

3.2.1 索引构建成本

首先,在图数据库中评估不同方法的索引构建成本。值得注意的是,图索引是离线预构建的,对于图数据库 G ,考虑 GED 阈值 τ 是很小的($\tau \leqslant 4$),其主要原因是用户会更倾向于从图数据库中搜索相似的图。

表 1 为在不同阈值 τ 下,无坐标映射和有坐标映射时,参与索引构建的数据图数量。从表 1 可知,有坐标映射时,会过滤掉相当一大批从图的规模上就不符合要求的数据图,从而大大减轻了索引的构建负担。同样,当阈值越大时,说明用户对相似度的要求越松,通过坐标映射的数据图数量也会大幅度上升,但是哪怕阈值 τ 已经为 6 了,它仍然能过滤掉近 2/3 的数据图。

表 1 有无坐标映射时参与索引构建的图数量

阈值 τ	无映射	有映射
1	10 000	621
2	10 000	973
3	10 000	1 543
4	10 000	2 381
5	10 000	2 896
6	10 000	3 383

图 4 表示在无坐标映射时 Pars 算法、采用双层索引及采用三层索引的文中算法的时间开销。可以看到可以有效构建多层索引。同样,因为需要构建多层索引,以及需要选择较好的图划分,多层索引在构建时间上略长于 Pars 算法的单层随机划分的索引。而随着阈值 τ 的增加,这种索引构建时间的差距开始变的微不足道,这是因为当 τ 变得更大时,每个数据图会相应地被划分为更大数量的更小分区,从而导致半边子图同构计算加速,而半边子图同构的计算时间通常会占用总索引构建时间的 90% 以上。

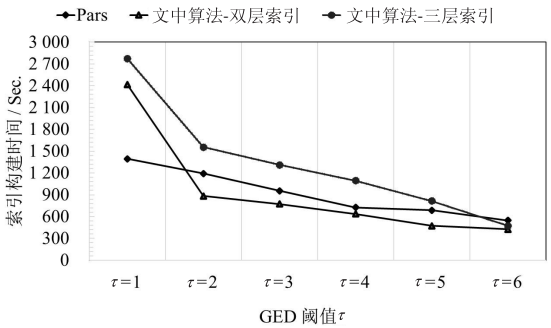

图 4 无坐标映射的索引构建时间

表 2 给出了在 AIDS 中不同阈值、无标签频率表及倒排索引表的情况下,半边子图同构所需要的时间。之所以排除标签频率表和倒排索引表的干扰,是因为这两者很大程度上会受到图划分的影响,若是采用随机划分方法,那么在不排除标签频率表和倒排索引表的情况下,半边子图同构时间会变的相对随机,若是采用文中的选择性划分方法,那么半边子图同构时间的影响因子又会变的过多。从表 2 可以看出,随着阈值的增大,半边子图同构的时间会被加速。

表 2　半边子图同构时间

阈值 τ	时间/s
1	1 104.17
2	978.57
3	831.024
4	740.2
5	624.98
6	475.45

图 5 表示在有坐标映射时,Pars 算法、采用双层索引和采用三层索引的文中算法的索引构建时间对比。从图 5 可知,其与图 4 变化趋势不同:随着阈值 τ 增大,索引构建的时间整体呈上升趋势。结合表 1 可以发现,当阈值 τ 增大时,表明对查询图的相似性要求越来越低,通过坐标映射的图的数量显然会越来越多,这就导致了参与索引构建的图的数量上升,因此在阈值变大的情况下,哪怕有着半边子图同构的计算加速,参与计算的图数量显然影响更大。

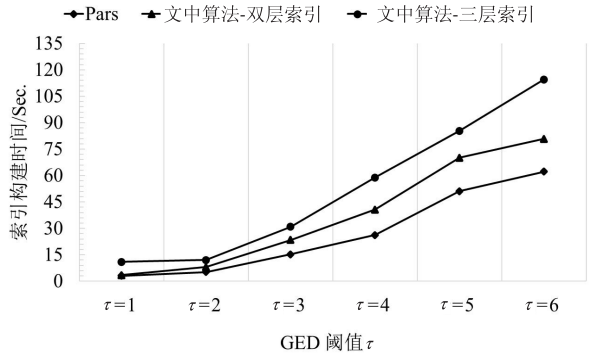

图 5　有坐标映射的索引构建时间

3.2.2　候选图数量

其次,比较了不同方法所产生的候选图数量。图 6 表示 Pars 算法、采用双层索引及采用三层索引的文中算法所产生的候选图数量,其中 Real 表示实际的相似图数量。从图中可以看出,文中所采用的算法,其过滤出的候选图始终小于 Pars,哪怕是双层索引结构都能比 Pars 算法减少近 45% 的假阳性图。原因有两个:首先,参数 $k>1$ 的广义 GED 下界比降级($k=1$)的更紧密;其次,选择性分区方法比 Pars 中的随机分区更有效地生成高选择性的索引分区,这有助于更进一步地过滤假阳性图。另外,可以看到当索引层数增多,被过滤掉的假阳性图也显著增加,这表明多层索引方法具有更出色的过滤能力,与 Pars 的单层随机划分的方法相比,可以保证假阳性图数量的减少。

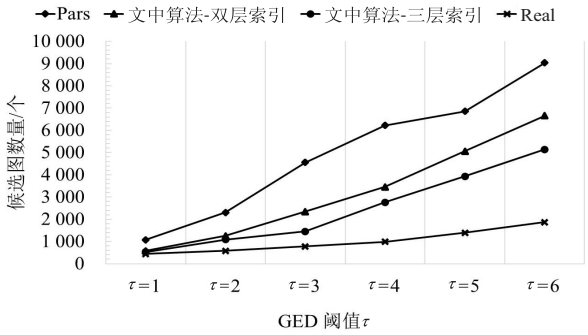

图 6　候选图数量

3.2.3　查询执行时间

最后,评估了 Pars 算法、采用双层索引及采用三层索引的文中算法的总体运行的时间成本。因为评估的是最终的候选图及 GED 验证的时间,故而有无坐标映射对该部分几乎是没有影响的。评估结果如图 7 所示。可以发现,相比于随机划分的单层过滤方法 Pars,文中的多层索引的过滤效果要更好,因为坐标轴是按指数增加的,所以就算看起来没有增加多少,但实际上是差距越来越大的,这主要是因为多层索引所生成的候选集明显更少,这样总体运行时间才会更少,说明选择性图划分方法的多层索引结构明显具有更好的,更精确的过滤效果。

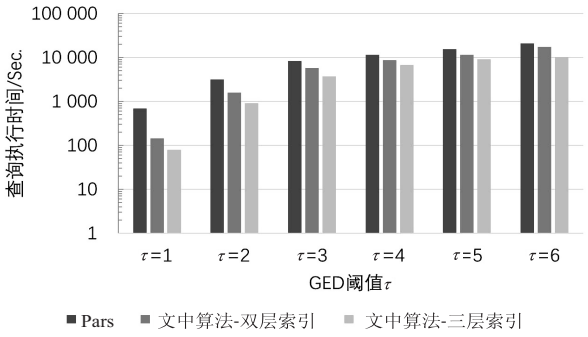

图 7　查询执行时间

综合上述的实验结果可以得出以下结论:首先,基于坐标映射的预过滤方法可以有效减少参与后续细粒度过滤方法的数据图数量,从而明显减少细粒度过滤方法的索引构建时间;其次,虽然在索引构建阶段,文中算法因为需要对分区质量进行调整,所以整体索引构建时间要略长于随机划分的 Pars 算法,但是图 6 和图 7 的实验结果表明,其过滤性能与精度是要明显优于 Pars 算法的,从整体的查询速度来看,文中算法的整体查询速度是要明显优于 Pars 算法的。

4 结束语

相似性搜索问题在管理和查询图结构数据中起着最基本的和最关键的作用,并且在现实世界中大规模的图数据库中有广泛的应用。该文探讨了在图编辑距离的约束下的相似性搜索问题,并在 Pars 算法的基础上进一步引入多层索引结构来解决图相似性搜索这一问题,同时针对多层索引结构的构建时间开销过大的问题,提出了坐标映射的批量图处理方式来有效减少过滤阶段的时间成本。首先采用坐标映射来过滤掉一大批从规模上就超出相似度阈值的图,然后使用参数化的、选择性图划分方式的 GED 下界来过滤假阳性图,以此使得图划分变得更加可控,同时通过倒排索引和标签频率表来减少子图同构的时间开销,最终通过生成多个索引集来对批量过滤后的图再次进行交叉检查,以此提升过滤的精度。这样,在保证多层索引过滤精度的同时,使用坐标映射大大减少了时间成本。模拟实验结果证明,坐标映射提升了算法的时间性能,而多层索引的引入提升了算法过滤的精度。

参考文献:

[1] CAO L. Data science:a comprehensive overview[J]. ACM Computing Surveys,2017,50(3):1-42.

[2] AGGARWAL C C,WANG H. Graph data management and mining:a survey of algorithms and applications[J]. Managing and Mining Graph Data,2010,40(2):13-68.

[3] COOK D J,HOLDER L B. Mining graph data[M]. New Jersey:Wiley,2006:36-42.

[4] 张志浩,孙保华,韩 韬,等. 基于图数据库的配电网供电范围分析应用研究[J]. 机电信息,2023(3):1-5.

[5] 陈根奇,黄振华,王少春,等. 基于图数据库和图算法的转供电方案在配电网智能操作票系统的研究和应用[J]. 电力学报,2023,38(1):73-82.

[6] 张百平. 基于电子病历系统关系数据库构建患者诊疗图谱[J]. 医学信息学杂志,2022,43(12):45-49.

[7] PRUSTI D,DAS D,RATH S K. Credit card fraud detection technique by applying graph database model[J]. Arabian Journal for Science and Engineering,2021,46(9):8849-8868.

[8] LIBKIN L,MARTENS W,VRGOČ D. Querying graphs with data[J]. Journal of the ACM,2016,63(2):1-53.

[9] AHMADI Z,PARAND F A,MATINFAR F. A fuzzy logic-based approach for fuzzy queries over NoSQL graph database[J]. Concurrency and Computation:Practice and Experience,2022,34(1):e6542.

[10] BI F,CHANG L,LIN X,et al. Efficient subgraph matching by postponing cartesian products[C]//Proceedings of the 2016 international conference on management of data. New York:Association for Computing Machinery,2016:1199-1214.

[11] KADIAM C,FREEDMAN C S,SCHALL D G,et al. Techniques for optimizing graph database queries:U. S. 11,093,499[P]. 2021-08-17.

[12] 周显春. 基于模糊图神经网络的最大频繁子图相似匹配系统设计[J]. 现代电子技术,2021,44(5):84-88.

[13] 陈梓扬,王 璿,周军锋,等. 一种高效的图编辑距离计算方法[J]. 智能计算机与应用,2020,10(12):94-98.

[14] WANG Guoren,WANG Bin,YANG Xiaochun,et al. Efficiently indexing large sparse graphs for similarity search[J]. IEEE Transactions on Knowledge and Data Engineering,2012,24(3):440-451.

[15] CHENG J,KE Y,FU A W C,et al. Fast graph query processing with a low-cost index[J]. The VLDB Journal,2011,20(4):521-539.

[16] ZHU Y,QIN L,YU J X,et al. Answering Top-k graph similarity queries in graph databases[J]. IEEE Transactions on Knowledge and Data Engineering,2019,32(8):1459-1474.

[17] WESKAMP N,HULLERMEIER E,KUHN D,et al. Multiple graph alignment for the structural analysis of protein active sites[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2007,4(2):310-320.

[18] SHERVASHIDZE N,BORGWARDT K M. Fast subtree kernels on graphs[C]//Proceedings of the 22nd international conference on neural information processing systems. New York:Curran,2009:1660-1668.

[19] WANG X,SMALTER A,HUAN J,et al. G-hash:towards fast kernel-based similarity search in large graph databases[C]//Proceedings of the 12th international conference on extending database technology. New York:[s. n.],2009:472-480.

[20] GOUDA K,ARAFA M. An improved global lower bound for graph edit similarity search[J]. Pattern Recognition Letters,2015,58(Jun. 1):8-14.

[21] GAO X,XIAO B,TAO D,et al. A survey of graph edit distance[J]. Pattern Analysis and Applications,2010,13:113-129.

[22] JUSTICE D,HERO A. A binary linear programming formulation of the graph edit distance[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2006,28(8):1200-1214.

[23] ZHAO X,XIAO C,LIN X,et al. A partition-based approach to structure similarity search[J]. Proceedings of the VLDB Endowment,2013,7(3):169-180.

[24] ZHAO X,XIAO C,LIN X,et al. Efficient graph similarity joins with edit distance constraints[C]//2012 IEEE 28th international conference on data engineering. Washington:IEEE,2012:834-845.