软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(3):689−702 [doi: 10.13328/j.cnki.jos.005449]

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

基于 SQL 的图相似性查询方法*

赵展浩[1,2], 黄斐然[1,2], 王晓黎[3], 卢 卫[1,2], 杜小勇[1,2]

[1](中国人民大学 信息学院,北京 100872)

[2](数据工程与知识工程教育部重点实验室(中国人民大学),北京 100872)

[3](厦门大学 软件学院,福建 厦门 361000)

通讯作者: 王晓黎, E-mail: xlwang@xmu.edu.cn

摘 要: 图作为一种表示复杂信息的数据结构,被广泛应用于社交网络、知识图谱、语义网、生物信息学和化学信息学等领域.随着各领域应用的普及和深入开展,如何管理这些复杂图数据,是目前图数据库技术面临的巨大挑战.图的相似性查询是图数据管理中的热点问题之一,对图查询问题的研究主要包括图的相似性查询等.重点研究基于编辑距离(graph edit distance)的图相似性查询处理问题.首先,通过对目前代表性的问题求解算法分析发现,目前已提出的过滤规则都具有自己的优缺点和适用性.其次,针对已有方法在过滤阶段自身存在的优缺点和适用性的问题,提出一种面向关系型数据库的过滤框架,新的过滤框架可以支持所有已有的过滤规则,从而通过结合不同的过滤规则来优化图相似查询算法以提高查询效率.该方法可以最大程度地保留不同过滤规则的优点并克服其缺点,从而对不同查询具有普遍适用性.最后,基于 PubChem 数据集,通过比较算法在求解查询结果的时间消耗,验证所提出算法的高效性及可扩展性.实验结果表明,所提出的方法优于现有算法.

关键词: 图编辑距离;图相似查询; PostgreSQL;过滤和验证

中图法分类号: TP311

中文引用格式: 赵展浩,黄斐然,王晓黎,卢卫,杜小勇.基于 SQL 的图相似性查询方法.软件学报,2018,29(3):689−702. http://www.jos.org.cn/1000-9825/5449.htm

英文引用格式: Zhao ZH, Huang FR, Wang XL, Lu W, Du XY. SQL-Based solution for fast graph similarity search. Ruan Jian Xue Bao/Journal of Software, 2018,29(3):689−702 (in Chinese). http://www.jos.org.cn/1000-9825/5449.htm

SQL-Based Solution for Fast Graph Similarity Search

ZHAO Zhan-Hao[1,2], HUANG Fei-Ran[1,2], WANG Xiao-Li[3], LU Wei[1,2], DU Xiao-Yong[1,2]

[1](School of Information, Renmin University of China, Beijing 100872, China)

[2](Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), Beijing 100872, China)

[3](School of Software, Xiamen University, Xiamen 361000, China)

Abstract: Graphs are widely used to model complicated data in many areas such as social networking, knowledge base, semantic web, bioinformatics and cheminformatics. More and more graph data are collected such that it has become a rather challenging problem to manage such complex data. The database community has had a long-standing interest in querying graph databases, and graph similarity

* 基金项目: 国家自然科学基金(61502504, 61702432); 中国人民大学科学研究基金(中央高校基本科研业务费专项资金)(15XNLF09); 福建省中青年教师教育科研项目(JAT160003)

Foundation item: National Natural Science Foundation of China (61502504, 61702432); the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China (15XNLF09); the Research Funds of Fujian Province for Young Teachers (JAT160003)

本文由基于图结构的大数据分析与管理技术专刊特约编辑林学民教授、杜小勇教授、李翠平教授推荐.

收稿时间: 2017-08-01; 修改时间: 2017-09-05; 采用时间: 2017-11-07; jos 在线出版时间: 2017-12-05

CNKI 网络优先出版: 2017-12-06 15:23:33, http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1523.013.html

search is one of most popular topics. This paper focuses on the graph similarity search problem with edit distance constraints. Firstly, several state-of-the-art methods are investigated to reveal that all the proposed pruning rules have limitations and none of them can outperform others on various queries. To address this problem, then a novel approach is proposed to support the graph similarity search in the framework of query evaluation using the relational model. The proposed approach develops a novel unified filtering framework by combing all the existing pruning rules. It can avoid limitations on existing pruning rules, and have more widely applications. A series of experiments are also conducted to evaluate the proposed approach. The results show that the new approach can outperform all existing state-of-the-art methods.

Key words: graph edit distance; graph similarity search; PostgreSQL; filter-and-verification

随着互联网、物联网和移动互联等信息技术的发展,图结构数据在当前社会越来越普及,在社交网络、知识图谱、语义网、生物信息学和化学信息学等领域都有广泛的应用[1−3].随着图结构在复杂数据建模方面的广泛应用,图数据管理技术持续得到了广泛关注,而其中,如何从图数据集合中快速检索数据,已经成为一个研究热点[4].在图查询中,相似性查询是一种非常重要的查询方式.相似性查询是指给定一个查询图和阈值 T,返回图数据集合中与查询距离不大于 T 的所有图数据.处理相似性查询时,需要进行图相似度计算.研究学者已经提出了很多种图相似性度量方法,其中最普遍适用的就是图编辑距离[5,6].但是图编辑距离的计算已经被证明是 NP 难题[7].为了快速有效地返回图查询的结果,目前主要采用“过滤+验证”的两阶段处理机制.在这种机制中:

- 首先预定义过滤规则,即,用复杂度较低的方法快速计算出图数据集合中每个图与查询图的编辑距离的范围(用上界和下界表示),其中,编辑距离的上界(下界)描述了两个图距离可能的最大值(最小值);
- 其次,利用下界和上界将集合中不可能出现在结果集(其与查询图的编辑距离的下界>T)或者肯定出现在结果集的图(其与查询图的编辑距离的上界<T)先进行过滤,从而生成规模较小的候选集;
- 最后对候选集进行图编辑距离计算,得到最终的结果集.

现有相似性查询的图过滤方法中具有代表性的有 4 种:基于 label 的图相似查询方法[8]、基于 path 的图相似查询方法[9]、基于 star 的图相似查询方法[10]和基于 tree 的图相似查询方法[11].这些方法采用图中的子结构相似性衡量图编辑距离,通过计算时间复杂度较低的近似距离提高过滤阶段的效率.然而本文通过实验证明:这些近似计算方法往往具有自身的优缺点和适用性,没有一种算法可以在所有应用场景下产生的图数据集合中都具有优越性.为了解决已有方法普遍存在的问题,本文提出一种全新的过滤策略,使得各种编辑距离的近似计算方法可以相互间取长补短,形成更为精确、健壮和实用的图编辑距离的近似度计算方法.

考虑到已有的经典过滤方法,如基于图的不同子结构信息 label,path 和 tree 等,很难直接设计一种统一的索引结构来支持已有方法.为了充分结合已有过滤规则来提高过滤效率,本文提出了面向关系型数据库的统一过滤框架.该框架既可以支持已有过滤规则的松耦合结合,又可以利用关系型数据库本身的健壮性和实用性[12]来提高算法的适用性.其主要思想是:

- 首先,基于 PostgreSQL 实现将图数据库转化为关系型数据库进行存储;
- 然后,在 PostgreSQL 中使用 PL/SQL 实现基于 label,star,path 和 tree 这 4 种模型的编辑距离近似计算的算法.

为了缩小相似性查询问题中候选图的规模,本文提出了一种松耦合的过滤方法,对已有的 4 种近似距离计算方法进行有效的结合,从而得到更紧的下界和上界,在过滤阶段剔除更多的非结果图,极大地减少图编辑距离的计算,提高图相似查询的效率.首先,将查询图的规模信息和节点标签信息与图数据库中的图进行比较,利用基于 label 的图过滤规则先将高偏差的图剔除.其次,计算复杂性相对较高的基于 path,tree 或者 star 的近似距离,选取其中最大的下界值和最小的上界值来获得更紧更精确的近似距离,从而得到一个较小的候选集.最后,通过计算候选集中的图编辑距离,返回满足查询要求的图集合.

本文主要贡献为:

- 提出了一种全新的面向关系型数据库的图过滤策略,既可以为已有的过滤方法提供统一的过滤框架,又可以充分利用已有过滤规则的松耦合结合方法获得更加高效的上下界,减少候选集中图的数量,从

而减少实际编辑距离的计算,提高相似性查询的效率;

- 面向关系型数据库的过滤框架,完全基于 SQL 的实现方法,无需修改关系数据库的内核.并且,可以充分利用关系型数据库的健壮性和实用性,提高图相似查询算法在不同领域问题的适用性;
- 相关实验结果证明,本文提出的面向关系型数据库的图过滤策略优于已有的图相似查询算法.

本文第 1 节对已有的图相似查询方法相关工作进行分类总结.第 2 节给出主要问题定义.第 3 节针对本文提出的问题,给出基于已有工作的解决方案,总结分析出这些方法的缺陷,并提出一种更加有效的面向关系型数据库的方法.第 4 节通过一系列实验对本文提出的方法与已有工作进行比较,分析本文方法的有效性.最后总结全文,并提出未来值得关注的研究方向.

1 相关工作

1.1 图相似度计算

图相似查询是一种重要的查询方式,应用广泛.相似性查询是指返回数据库中与查询图相似的图集合.处理相似性查询时,首先需要进行图相似度计算.早期的研究学者主要提出了启发式的图相似性度量方法[8,13−15].文献[13]提出了一种基于节点差异的图相似度量标准.文献[15]基于 random walk 定义了图之间的相似性.文献[8,15]则在文献[13]的基础上进一步利用节点的邻接节点信息来衡量图的相似性.后来的研究学者开始提出基于最大公共子图的相似性度量方法[16]和基于编辑距离的图相似性度量方法[5].文献[20]利用类似的思想,使用 SQL 回答度量空间下的相似度查询,但该工作与本文的研究问题不同.在已有的方法中,最普遍适用的就是图编辑距离.因此,本文采用编辑距离来解决图相似性查询处理问题.

文献[5]完整综述了图编辑距离的计算方法[5],该文把已有计算方法分为两大类:精确计算方法和近似计算方法.精确计算最常用的方法是 A*算法[17].然而,由于编辑距离的计算是 NP 难题[7],这类精确的算法往往因复杂度过高而不具有实用性.为了避免这个问题,一些复杂度较低的近似编辑距离的计算方法被提出,用于过滤偏差较大的图[7,9−11,18].

1.2 图相似查询方法

由于图编辑距离的计算过于复杂,因此很难找到一种高效的计算方法.已有方法采用“过滤+验证”的框架机制来加速图相似查询的速度,其主要思路是:通过一些复杂度较低的近似距离先过滤掉错误结果,然后再通过实际编辑距离的计算对剩下的候选数据进行验证.目前,基于编辑距离的图相似查询算法根据其所采用的不同过滤方式,主要分为 3 大类:基于 label 的图相似查询方法、基于 path 的图相似查询方法和基于 tree 的图相似查询方法.

- 基于 label 的图相似查询方法

早期的方法思路比较直观,就是利用图的节点和边的数量差异作为过滤条件[13],即,两个图之间节点的数量差异与边的数量差异之和是两个图之间编辑距离的一个下界.文献[8]在文献[13]的基础上进一步考虑节点和边的 label 差异,即:先分别将节点和边的 label 进行分类,两个图之间每一类 label 的数量差异之和作为编辑距离的一个下界.由于后面提出的下界比前面的相对较紧,因此本文统一将这类方法称为基于 label 的图相似查询方法.然而,这类方法忽略了图的结构信息,下界往往很松,因此过滤效果有限.

- 基于 path 的图相似查询方法

这类方法的基本思想来源于字符串的 q-gram 数量过滤规则,是一种基于图结构的过滤方法.这类方法中最具代表性的是文献[9].这类方法把图中的子结构 path 作为 q-gram,然后利用字符串 q-gram 数量计算编辑距离的下界.这里的 path 通常定义为图中的一个节点和边序列,其中,任意两个相邻节点之间均有一条边,而边的总数即对应 q-gram 的 q 值.根据这个定义,给定两个图,这类方法首先需要提取两个图的所有 path,构成两个 path 集合;然后,通过计算集合的交集得到两个图的共同 path 数量,只有这个数量查过一定的阈值,两个图的距离才会小于某个阈值.文献[18]指出:由于一个度数较大的节点编辑操作将引起大量的 path 产生差异,因此,基于 q-gram

共同数量的下界可能会很松,从而导致过滤效果有限.

- 基于 tree 的图相似查询方法

基于 tree 的这类方法也是一种基于图结构的过滤方法,然而由于过滤规则计算方法的不同,可以进一步分为两种方法,即,基于 q-gram 数量的过滤方法[11]和基于二分匹配[19]的过滤方法[7,10,18].其中,k-at 算法提出将图中的子结构 k 邻接树作为 q-gram,然后利用两个图之间共同的 q-gram 数量计算编辑距离的下界[11].文献[7,10]提出了基于图中的子结构 star 的过滤方法,这里的 star 可以看做 k=1 的 k 邻接树.然而,这些方法并没有采用很松的 q-gram 数量作为过滤条件,而是提出一种基于二分匹配的下界计算方法.文献[18]也是基于二分匹配计算编辑距离的下界,不同的是,其采用的过滤子结构是 branch,即,去掉所有子节点只包含根节点和邻接边的 star.这类基于 tree 结构的方法所提出的编辑距离的下界的过滤效果也直接受到图的节点度数影响,因此只适用于平均节点度数较低的图数据库.

综上所述,现有的方法提出的过滤规则均具有自身的优缺点和适用性,很难有一种算法可以适用于各种图数据库或者优于所有其他算法.

2 预备知识

2.1 问题定义

本文主要研究无向简单图的相似查询问题.无向简单图指无向带标签的无环连通图,可以通过四元组 $g=(V,E,l_v,l_e)$表示,其中,V 代表 g 图中所有节点的集合,E 代表所有边的集合,l_v 代表节点标签的集合,l_e 代表边的标签的对应集合.本文的其他标注见表 1.

Table 1 Paper notations

表 1 文中标注

标注	描述
$\|V(g)\|$	代表图 g 的大小,即为节点总个数
$\|V(g)_l\|$	代表标签为 l 的节点个数
$Deg(v)$	节点 v 的度数
$\delta(g)$	图 g 的最大度
$ged(r,s)$	图 r 与图 s 的编辑距离
D	图的集合
$\|D\|$	图的集合的大小,即为图的总个数

定义 1(图编辑距离). 给定两个图 r 和图 s,两图之间的编辑距离 $ged(r,s)$即为将图 r 转换成图 s 所需的最小编辑操作数目.图的编辑操作主要包括以下 6 种.

- 在图中插入一个孤立的节点;
- 从图中删除一个孤立的节点;
- 在图中修改一个节点的标签;
- 在图中插入一条连接两个节点的边;
- 从图中删除一条边;
- 在图中修改一条边的标签.

定义 2(基于编辑距离的图相似查询问题). 给定一个图的集合 $D=\{g_1,g_2,\ldots,g_{|D|}\}$,待查询的图 q,阈值 T,找到所有图 g_i 与待查询图 q 的编辑距离满足不等式 $ged(g_i,q)\leqslant T$ 的图的集合.

图 1 给出了 3 个化学结构,依次为甲烷(g_1)、乙烷(g_2)、乙烯(g_3),显然均是无向简单图.

给定图的集合包含图 g_1,g_2 和 g_3,我们指定待查询图为 $q=g_1$,其中,可以计算得到 $ged(g_1,q)=0$,$ged(g_2,q)=7$,$ged(g_3,q)=6$.如果编辑距离的阈值是 $T=6$,那么由于 $ged(g_1,q)\leqslant T$ 和 $ged(g_3,q)\leqslant T$,g_1 和 g_3 将出现在结果集里,而 g_2 将不在结果集里.

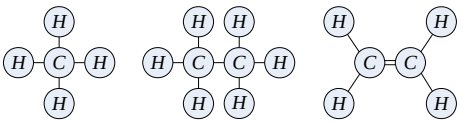


Fig.1 Example of three data graphs g_1, g_2 and g_3

图 1 图集合示例

2.2 现有代表性算法

如定义 2 所示:要解决该问题,一般需要遍历数据集 D 中所有的图,计算每一个图与待查询图的编辑距离,然后再根据给定的阈值筛选出相似图.然而,图编辑距离的计算已被证明是 NP 难题[7],计算的复杂度非常大,目前尚未有任何方法能快速计算出图的编辑距离.因此,对数据集内全部的图进行遍历计算显然不切实际.为了提高图相似查询的速度,现有的方法往往采用“过滤+验证”的两阶段处理机制.现有的图过滤方法中,最具代表性的有 4 种:基于 label 的过滤方法、基于 star 的过滤方法、基于 path 的过滤方法和基于 tree 的过滤方法.以下具体介绍这 4 种经典的图相似查询算法.

2.2.1 基于 label 的过滤方法

通过两个图的标签来度量相似性是最为基础的一种过滤方法[8].根据图编辑操作的定义,我们可以认为:如果两个图具有不同个数的边和节点,则显然需要通过一定的编辑操作才能使得图同构.因此,可以得到公式(1):

$$ged(r,s)\geqslant abs(V(r)|-|V(s)|)+abs(E(r)|-|E(s)|) \tag{1}$$

图 r 与图 s 的编辑距离一定小于或等于两个图的节点个数之差加上边条数之差.因此,简单的基于 label 的过滤规则定义为:找到集合 D 中的所有满足不等式 $abs(V(r)|-|V(s)|)+abs(E(r)|-|E(s)|)\leqslant T$ 的图.

再进一步考虑,由于节点和边的标签不同,可能也会产生相应的编辑操作.因此,我们对公式(1)可以进行相应的优化,即:先通过标签进行分类,然后再对每一类进行计算.见公式(2):

$$ged(r,s)\geqslant \sum abs(|V(r)_l|-|V(s)_l|)+abs(|E(r)_l|-|E(s)_l|),l\in l_v,l_e \tag{2}$$

2.2.2 基于 star 的过滤方法

根据图的结构进行子图分解也是一种近似计算图编辑距离的思路.首先,我们将图拆解为多个星形结构[7].星型结构即为由一个中心节点以及与它有边相连的节点所构成的深度为 1 的树结构.星型结构可以用三元组 $s=(r,L,l)$表示,其中,r 表示根节点,即中心点,L 代表所有的叶子节点集合,l 为叶子节点对应的标签集合.所以,对于有 n 个节点的图 g,就会得到 n 个对应的星型结构,星型结构的集合用 $S(g)$表示.图的编辑距离计算也就转化为对星型结构进行编辑距离计算.

我们给出星型结构的编辑距离(star editdistance,简称 SED)计算公式,见公式(3):

$$sed(s_1,s_2)=T(r_1,r_2)+d(L_1,L_2)=T(r_1,r_2)+abs(|L_1|+|L_2|)+\max(|L_1|,|L_2|)-||L_1|\cap|L_2|| \tag{3}$$

如果 s_1 与 s_2 的根节点具有相同的标签,则 $T(r_1,r_2)$的值为 0;否则为 1.将其与图编辑距离的转换见公式(4)、公式(5):

$$md(r,s)=\min_P \sum_{s_i\in S(r)} sed(s_i,P(s_i)) \tag{4}$$

$$\frac{md(q,g)}{(\max\{4,[\max\{\delta(q),\delta(g)\}+1]\})}\leqslant T \tag{5}$$

图编辑距离的计算其实转化为了对图 r 和图 s 的星型结构集合找到最优匹配,要找到编辑距离的下界,使得最后匹配得到的 SED 之和最小,我们用 mapping distance(MD)来代表这个最优解.因此,最优匹配的求解思路为:首先,将图 r 和图 s 的星型结构两两计算 SED,从而构成一个权重矩阵,如果两个图所具有的星型结构个数不同,则通过插入空节点的方法构成矩阵[10].之后,将所得到的矩阵作为输入参数,采用匈牙利算法[19]找到最优匹配.

因此,基于 star 的过滤规则定义为:给定阈值 T,过滤得到集合 D 中的图 g 与 q 满足不等式(5)的图集合.

2.2.3 基于 tree 的过滤方法

基于 tree 的过滤规则[11]主要通过图中的子树进行构建.通过深度优先搜索,指定一个开始节点,可以将与其相邻的节点作为叶子节点,通过递归的方式构建无限深度的树,我们把这个树称为邻接树(adjacent tree,简称 AT).因为邻接树结构的复杂性,在计算中会带来不必要的效率问题,所以我们限定了邻接树的深度 k,将指定深度的邻接树(k-AT)作为基于 tree 的过滤规则核心.

因此,过滤规则的不等式见公式(6):

$$|k\text{–}ATs(r)\cap k\text{–}ATs(s)|\geqslant|V(r)|-T\cdot2(\delta(r)-1)^{k-1} \tag{6}$$

将图 r 和图 s 转化为从图中每个节点出发的 k-AT 的集合.不等式的左半边即为两个集合所具有相同结构 k-AT 的个数.

2.2.4 基于 path 的过滤方法

基于 path 的过滤规则[9]也是一种基于图结构的过滤方法.首先,需要找到图中指定长度 q 的所有路径.本文中采用深度优先搜索算法进行 path 提取.因为路径有开始节点和结束节点,分别从开始节点和结束节点出发,将会得到两条序列,对这两条序列我们保留字典序较小的一条.将图 g 中每个节点出发的路径组成的集合用 $P(g)$ 表示,则通过过滤的图需要满足公式(7),不等式左半边为图 r 和 s 的路径集合,右半边为取到两个图中度数减去阈值乘以最大度的较大值.

$$|P(r)\cap P(s)|\geqslant\max(|P(r)|-T\cdot\delta(r),|P(s)|-T\cdot\delta(s)) \tag{7}$$

2.3 图相似查询框架

如图 2 所示,图的相似查询框架主要分为特征提取、过滤和验证这 3 个部分.首先,根据对应的过滤算法的所需参数,对待查询的数据集 D 进行特征抽取,如对图进行遍历得到路径,统计图中顶点和边的个数等;然后就进入过滤部分的操作,即,调用对应的过滤算法产生通过过滤的候选数据集,从而将完全不相似的图丢弃;最后,将候选集中的图进行精确的图编辑距离计算,从而得到最终的相似查询结果集合.

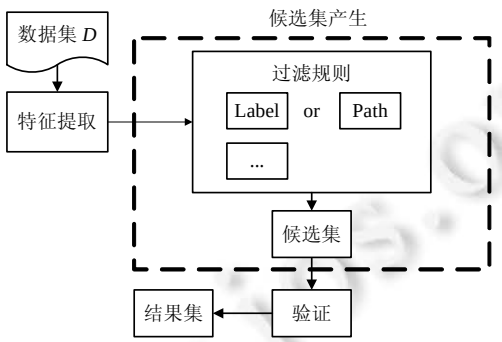


Fig.2 Framework of graph similarity search based on edit distance

图 2 基于编辑距离的图相似查询框架

这种较为经典的过滤框架是一种解决图相似查询的比较好的方法,可以较大地提升图相似查询的效率.但是作为一种实验框架,对于实际的图结构应用环境不能做到很好的支持.如在社交网络的应用中,根据社交关系所构建的图结构会不停地更新,那么每次更新都需要对图的特征进行重新提取,进而通过“过滤+验证”来进行图相似查询,这种串行的处理模式会使得实际应用场景下,该框架的效能较低.同时,针对当前大数据环境下的图结构所具有的数据量巨大的特点,框架没有给出一种很好的存储解决方案来保证对图数据的有效管理.因此,对现有的图相似框架进行优化是非常有必要的.

通过分析图中基于编辑距离的图相似查询过程,我们识别出影响搜索性能最重要的影响因素(如图 2 的过滤规则部分所示),即,图过滤规则的设计(见第 3 节).本文在现有的图过滤方法中选取最经典的 4 种(见第 2.2 节),

并一一实现其图相似查询算法.我们从 PubChem 数据集中抽样得到实验数据集,将实现的 4 种算法在抽样的图集上进行实验(实验结果见第 4.1 节),总结分析实验结果发现:已有的规则本身具有优缺点和适用性,这 4 种方法没有一种能够绝对地优于其他的方法.为了最大程度地保留已有方法的优点,避免缺点,本文提出了一种全新的面向关系型数据库的过滤框架,新的过滤框架可以将这 4 种图的编辑距离的近似计算方法进行有效结合,求得更紧的上下界,形成了能够进行非常高效地进行剪枝的方法.

3 面向关系型数据库的图相似查询算法

本节重点阐述面向关系型数据库的图相似查询算法,首先,简要介绍所提出的全新过滤框架以及新的过滤规则.然后,将其在 PostgreSQL 这一开源的关系型数据库中的实现思路进行描述.最后,综合介绍面向关系型数据库的图相似查询框架,并且提出了基于图相似框架的图数据管理方法.

3.1 图结构在关系型数据库中的关系模式

作为一种最为普遍的结构化数据存储工具,关系型数据库与结构化查询语言(SQL)的结合一直在数据操作方面表现良好.但是当前,随着图结构的不断发展,在关系型数据库中对图结构的操作仍然支持不够.因此,本文实现一种基于 SQL 的图相似性过滤方法,充分利用关系型数据库的特性,增强了关系型数据库对图相似性搜索的支持.

我们将所有的图存储在关系数据库管理系统 PostgreSQL 中,存储图的 Graph 表具有以下 3 个基本字段:编号(rid:integer)、节点(vertex:integer[])、边(edge:integer[]),每一个图在关系数据库中存储为一条记录,假设某条记录存储了图 g_p,则该记录可以表示为 $(p,\{V_1,l_1,V_2,l_2,...,V_n,l_n\},\{E_1,E_1',E_2,E_2',...,E_n,E_n'\})$.通过编号唯一地标志一个图,并将顶点和边采用数组的形式存储,(v_k,l_k)代表一个顶点,包括了顶点编号和顶点标签两个属性,(E_k,E_k') 代表一条边,用边的两个端点表示.例如,对于图 1 中的甲烷结构,我们将其存储为表 2 中的数据格式.

Table 2 DataSchema

表 2 图存储模式

Rid	Vertex	Edge
1	{1,6,2,1,3,1,4,1,5,1}	{1,2,1,3,1,4,1,5}

对于过滤方法中需要用到的图特征结构,本文通过如下的关系模式将提取得到的特征结构进行存储,存储图的 Graph 表具有以下 4 个基本字段:编号(rid:integer)、路径(path:integer[])、树结构(tree:integer[])、星型结构(star:integer[]).分别对 Path,Tree,Star 运用一个属性存储下来,并用 rid 标志这些特征所属的图.对于图 1 中的甲烷结构,对其进行特征抽取后,指定 path 的长度为 2,tree 的深度为 1,得到的数据格式见表 3.

Table 3 Graph structure schema

表 3 图特征存储模式

Rid	Path	Tree	Star
1	{1,2,1,3,1,4,1,5}	{1,2,1,3,1,4,1,5,2,1,3,1,4,1,5,1}	{1,2,1,3,1,4,1,5}

3.2 面向关系型数据库的图相似过滤方法

因为在图相似查询过程中,过滤部分是非常重要的环节,所以本文对原有的过滤流程进行了重新设计,提出了一种分层过滤框架,如图 3 所示.框架的整体思想是,希望通过多层过滤较好地规避时间成本和空间成本.因为原有的过滤方法都不是完全最优的,因此,我们考虑将他们结合起来进行使用,即:最先用过滤粒度最粗,但时间复杂度最低的方法进行过滤,得到相应的候选集后,由下一层的过滤方法在所得候选集中进行过滤,以此类推,通过这种层层过滤的方式,使得每个方法都能发挥最好的过滤效果.因此,下文中统一使用 combined-filter 表示本文所使用的过滤方法.

本文通过大量的实验得出了一种较为高效的分层过滤顺序,即:先对数据集运用基于 label 的过滤,再对得

到的候选集采用基于 path 的过滤,然后对得到的候选集采用基于 tree 的过滤,基于 star 结构的过滤放到最后进行.这一过滤顺序具有非常好的过滤能力,并且时间复杂度较低,流程图如图 3 所示.

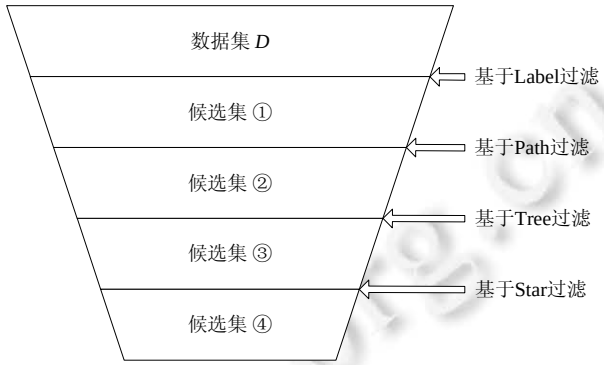


Fig.3 Novel framework of graph similarity search based on RDBMS

图 3 面向关系型数据库的全新图相似过滤框架

下面介绍 combined-filter 在关系型数据库中的实现思路.我们在 PostgreSQL 中使用 PL/SQL 实现了基于 label,star,tree,path 这 4 种图编辑距离过滤算法,并通过函数调用.因为 4 种方法之间是相互独立的,所以这种松耦合的方式使得方法的普适性和稳定性都较强,可以方便地嵌入到各类关系型数据库中,而不需要修改数据库的内核.对于这 4 种方法,我们采用统一的过滤算法框架进行实现,以此保证算法的一致性,保证分层过滤的效率.由此得到的过滤算法框架见算法 1.

算法 1. 过滤算法框架.

```
1   function FILTER(q,D,T,Cand)
2       Input:q(待搜索图),D(数据库中的图结构数据集),T(编辑距离阈值);
3       Onput:Cand(候选集合).
4       Cand←∅
5       for g∈D do
6           if filter-startegy() then
7               Add(g,Cand)
8           end if
9       end for
10      return Cand
11  endfunction
```

3.3 基于新型过滤框架的图相似查询算法

本文提出的新型图相似查询算法流程见算法 2 所示.给出一个待搜索的图 q,并给出图数据集 D,编辑距离阈值 T,在过滤阶段中,通过 label,path,tree,star 的顺序进行层次过滤,得到候选的图集合 *Cand*.然后,通过调用精确的图编辑距离计算方法,将 *Cand* 中的图一一验证得到最终的结果集合 R.因为几个过滤规则间是解耦的,所以混合的过滤规则可以有多种组合,因此提升了图相似查询的灵活性.

算法 2. 基于新型过滤框架的图相似度搜索算法.

```
1   function COMBINED-GRAPH-SEARCH(q,D,T)
2       R←∅
3       CREATE VIEW Candidate AS
4       SELECT*FROM StarFilter(q,TreeFilter(q,PathFilter(q,LabelFilter(q,D))));
```

```
5        for g in SELECT*FROM Candidatedo
6            if ExactGED(g)≤T then    //计算精确编辑距离
7                Add(g,R)
8            end if
9        end for
10   end function
```

因此,在调用时图相似查询方法时,可以通过简单的 SQL 查询语句进行调用.通过对 SQL 函数传入 3 个参数:待查询图的顶点数组(vertex-array)、待查询图的边数组(edge-array)、查询阈值(*T*),即可返回查询得到的图结果集.示例查询语句如下所示:

SELECT *Combined-Graph-Search*(*Vertex-Array*,*Edge-Array*,*T*).

3.4 SQL-Based图特征提取

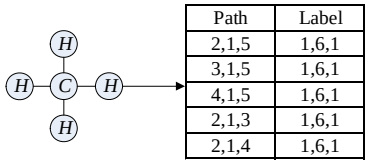
对于特征抽取操作,我们结合 SQL 语言的特性和广度优先搜索的思想,设计了一种效率较高的图遍历算法,见算法 3.

算法 3. 图的深度优先搜索算法.

```
1    function BREADTH-FIRST-SEARCH(edge,node)
2        //edge 边集合,node 顶点集合
3        WITH RECURSIVE transitive-closure(a,b,distance,paths,labels) AS
3        (
4        SELECT fromnode, tonode, 1 AS distance,
5        array[fromnode] $||$ array[tonode] AS paths,
6        array[n1.nname] $||$ array[n2.nname] AS labels
7        FROM edge left join node n1 on fromnode=n1.nid left join node n2 on tonode=n2.nid
8        UNION ALL
9        SELECT tc.a, e.tonode, tc.distance +1,
10       tc.paths $||$ array[e.tonode] AS paths,
11       tc.labels $||$ array[n3.nname] AS labels
12       FROM edge AS e left join node n3 on n3.nid=e.tonode
13       JOIN transitive-closure AS tc ON e.fromnode=tc.b
14       WHERE tc.paths::text NOT LIKE '%'||e.tonode||'%'
15       )
16       SELECT tc1.paths,tc1.labels FROM transitive-closure as tc1 where tc1.distance=q
17       and tc1.a$<$tc1.b ORDER BY labels;
18   end function
```

运用 WITH,UNION 关键字在 SQL 语言中实现了高效的递归查询,从而查询得到指定长度的图中所有路径集合,distance 字段代表路径长度,paths 代表顶点编号组合的路径,labels 代表顶点标签组合的路径,如图 4 所示,即为对甲烷结构进行长度为 3 的路径提取示例.根据此图搜索算法,也可以方便地处理得到 tree,star 等结构.

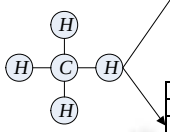
对于递归查询所需要的 Edge 和 Node 表,我们通过临时表的形式进行构建,Node 表具有两个字段:顶点编号(nid:integer)、顶点标签(nname:integer);Edge 表具有 3 个字段:开始顶点(fromnode:integer),结束顶点(tonode:integer),边标签(ename:integer).两张表结合起来,可以完全表示出一张图具有的结构和特征.在数据处理时,先从 graph 表中查询某个具体图,然后将其用 Node 表和 Edge 表组织起来,因为关系型数据库的支撑,在其基础上进行了查询等操作就能够具有较高的效率.例如,甲烷结构可以用图 5 的形式在数据库中表示出来.



Path	Label
2,1,5	1,6,1
3,1,5	1,6,1
4,1,5	1,6,1
2,1,3	1,6,1
2,1,4	1,6,1

Fig.4 Path extract example

图 4 路径提取示例



Nid	Name
1	6
2	1
3	1
4	1
5	1

Fromnode	Tonode	Ename
1	2	1
1	3	1
1	4	1
1	5	1

Fig.5 Node and Edge relation schema

图 5 Node 和 Edge 关系模式

4 实验与结果分析

本节对图相似查询算法进行了实验验证,并对实验结果和分析进行了展示.我们所有的实验都在 Intel(R) Xeon(R) CPU E5-4607@2.20GHz,8GB 内存的服务器上完成,操作系统为 CentOS Linux 7.0.所有的算法都通过 PL/SQL 语言进行实现,并在 PostgreSQL 9.4 版本上执行通过.

我们选用了公开的 PubChem 数据集作为实验数据.本数据集中包含 100 万个真实的化学结构,平均的顶点个数为 23.98,平均边数为 25.76,属于较为稀疏的简单图结构,在图中不存在环等复杂结构,符合本文实验的开展条件.在每次实验中,本文将数据集中编号最小的图作为待查询图,将数据集中剩余的图结构作为图集合,通过调用不同的过滤方法查询图集合中与待查询图相似的图结构,即可得到结果集以及过滤所需要的时间.通过结果集大小以及过滤时间,可以很好地衡量不同算法的过滤效果.下面给出了其对应的定义.

- 结果集(candidate size),指所有图经过过滤规则作用,过滤掉无用结果之后得到的候选集.结果集大小即为结果集所包含图结构的个数;
- 过滤时间(response time),指过滤算法执行的总时间,这里,我们忽略候选集进行具体图编辑距离计算的验证时间,因为验证时间直接与候选集大小成正比,从候选集大小就可以衡量出验证时间.

本文首先对 4 种已有过滤方法的过滤效果进行了对比实验分析.通过在 PubChem 数据集中随机抽取 0.1k 个、1k 个、10k 个有机物结构,组成了 4 组实验数据集,通过实验对比了 4 种方法的过滤效果.然后,通过随机生成多组 100k 的数据集,对本文提出的分层过滤模型进行了实验验证,并与 4 种已有过滤方法的过滤效果进行对比,从而分析本文方法的过滤效率.

4.1 4种过滤方法效能分析

基于 path 的过滤算法需要指定 path 长度 q,因此,我们在 0.1k,1k,10k 这 3 个数据集对取不同阈值、不同 path 长度环境下的过滤效果进行了测试.结果如图 6 所示.图 6(a)~图 6(c)分别代表 0.1k,1k,10k 数据集,指定不同阈值所得到的过滤结果集大小(candidate size);图 6(d)指出了在 10k 数据集,path 提取的时间(response time)随 path 的增长有增长趋势,所以 q 值应该在保证过滤能力的情况下尽可能小.可以看到:当 path 长度取 3 时,在不同大小的数据集,不同阈值下的过滤表现都较为良好,并且因为有机物结构大小不统一,并且当 path 长度超过 3 时,会出现结构丢失的情况.综上,在后续实验中,我们指定 path 的长度统一取 3.

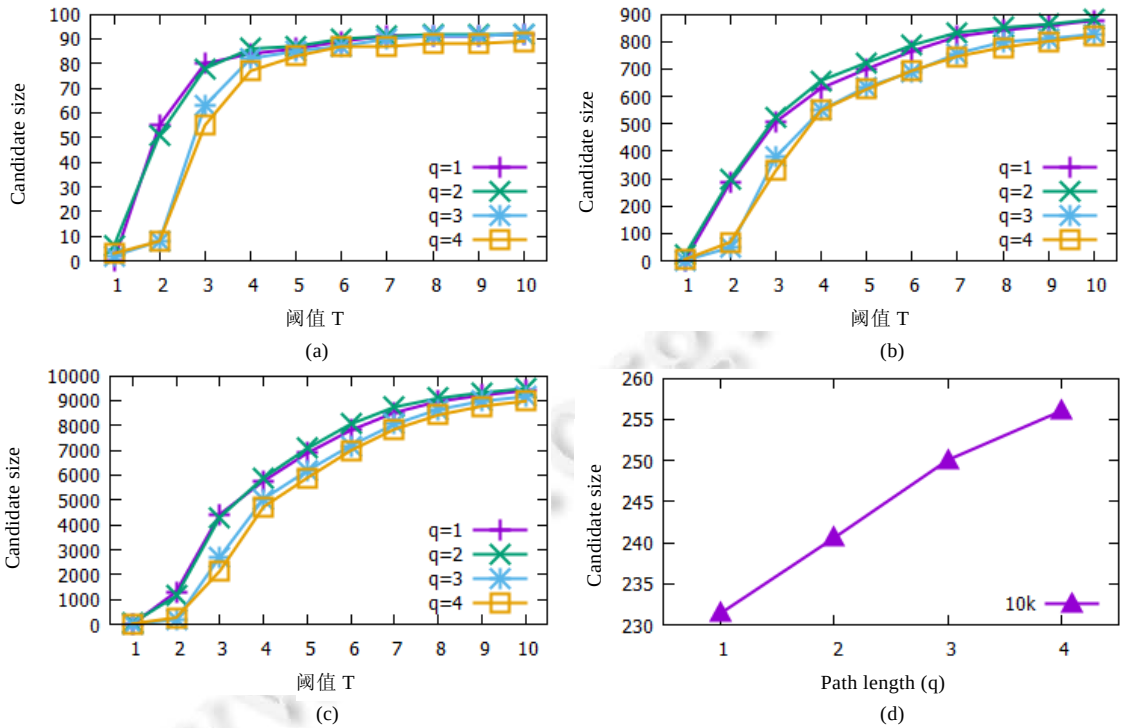


Fig.6 Path-Based filtering method analysis

图 6 基于 path 的过滤方法分析

基于 tree 的过滤方法中,需要指定生成树的深度 *k*,通过实验综合考虑过滤结果集以及响应时间两个影响因素,本文设定 *k*=1,具体过程不再赘述.

我们从 PubChem 数据集中随机抽取了 5 组 10k 的数据,来作为对 4 种过滤方法进行对比分析的实验数据.通过将 5 组实验结果取平均值的方法,得到的结果集大小随阈值变化如图 7(a)所示,响应时间如图 7(b)所示.

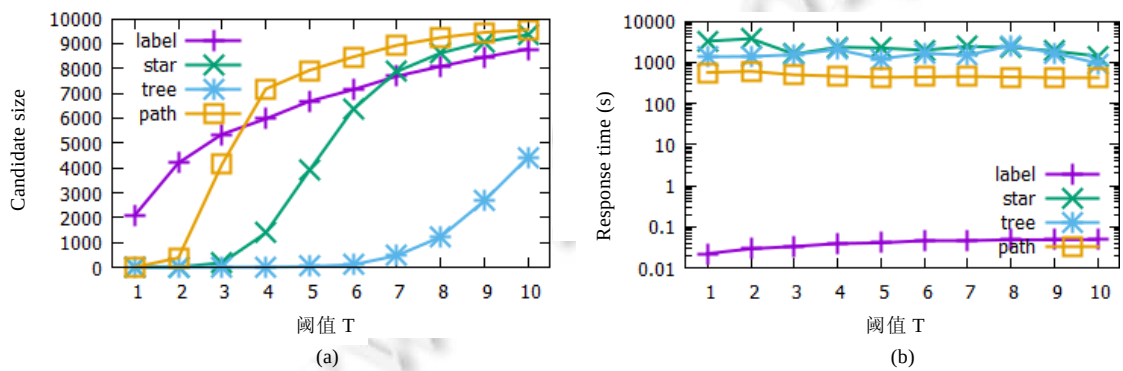


Fig.7 Four filtering methods' performance

图 7 4 种方法的过滤效果对比分析

从图 7(a)可以分析得出:4 种方法过滤得到的候选集大小都会随着阈值的增大而增大,但都会表现出不同的特性,即,随着阈值的变化,结果集最小的方法会发生改变.例如:当阈值小于 7 时,基于 label 的过滤方法较 star 方法较差;而当阈值大于 7 时,则相反.因此,很难给出一种方法在任何情况下能够优于另一种方法的结论.通过对图 7(b)所示相应时间的分析,基于 label 的过滤耗费时间相较于其他 3 种方法较少,所以作为一种初步筛选的方

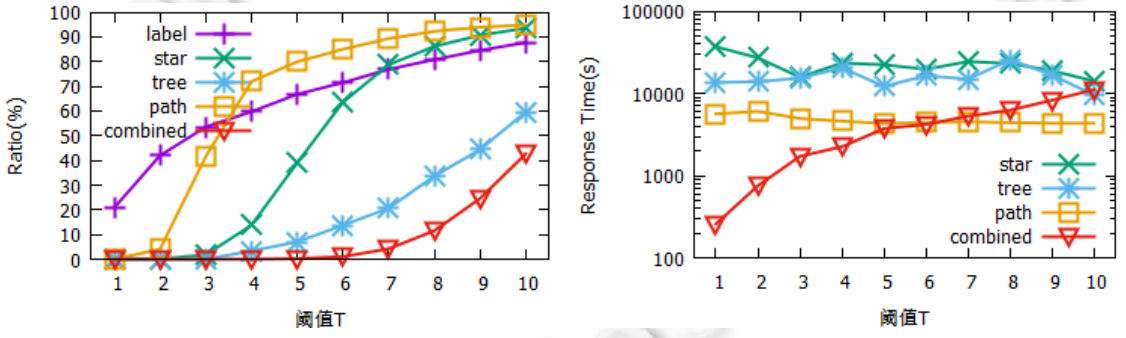
法较为合适.基于 path 的方法过滤时间较基于 star 和 path 的方法较少,且结合过滤效果来看,其在阈值较小的情况下表现较好,阈值较大的情况下表现较差,较于剩余两种方法过滤能力不够稳定,因此作为第 2 步的过滤较为合理.最后,基于 star 和基于 tree 的过滤方法都需要较长的过滤时间,根据实验结果中 star 方法过滤时间较长的结论,因此先进行基于 tree 的过滤后再进行基于 star 的过滤是效果较好的.需要说明的是:本次实验中,因为每次实验的输入数据集大小相同,都为 10k 个图,每个阈值下都会对数据集中的图进行遍历操作,而响应时间主要由数据集的大小决定,因此相应时间随阈值的变化基本维持不变.

4.2 新型图相似查询方法效能分析

在本次实验中,我们通过候选集和与总数据集合大小的比值来反映过滤能力,比值的计算见公式(8)(比值与过滤能力成反比关系,过滤能力越强,比值越小):

$$Ratio=|Cand|/|D| \tag{8}$$

我们基于 100k 数据集进行对比实验,并通过随机提取 3 组 100k 的数据集进行重复实验的方法,使得实验的结果相对准确.实验结果通过取平均值的方式给出,*Ratio* 值随阈值的变化如图 8(a)所示,平均响应时间如图 8(b)所示.在图 8(a)中可以看到:本文提出的过滤方法可以得到较小的结果集,过滤效率基本都超过 50%,分层混合过滤要优于任何一种单一过滤方法.当阈值为 10 时,分层混合过滤比单一的 label 过滤效果好 50%,比单一的基于 tree 的过滤效果好 10%.在过滤时间方面,如图 8(b)所示:由于本文层次过滤的思路,每次过滤得到的结果集都能得到很好的收敛,因此对于响应时间较长的方法,传入的候选图集合大小相对于响应时间短的方法较小,从而极大地减小了响应时间.因为基于 label 的过滤作为一种较为粗略且用时很短的过滤方法,因此在图 8(b)中将其图像省略.综上所述,本文所提出的分层过滤方法大幅地缩小了候选集,从而减小精确计算编辑距离的次数,也较好地缩短了过滤时间,因此具有较好的过滤效果,对提升图查询操作效率有一定的借鉴意义.



(a) (b)

Fig.8 Original filtering methods' performanceanalysis

图 8 新型图过滤算法效果分析

5 结束语

本文重点研究基于编辑距离的图相似查询问题.首先,通过对现有具有代表性的 4 种算法进行分析,发现现有算法存在过滤效果不稳定和适用性有限的问题;其次,针对已有方法在过滤阶段存在的问题,提出面向关系型数据库的全新过滤策略.该策略可以在过滤阶段灵活结合已有过滤规则来获取更加紧的编辑距离的上下界,从而过滤更多的无用结果,减少需要验证的候选集合的大小.本文经过调研,选取具有代表性的关系型数据库系统 PostgreSQL 来实现所提出的新策略,设计并实现了 4 种已有过滤规则,并实现不同过滤规则松耦合结合策略;最后,基于 PubChem 数据集,通过比较算法在求解查询结果的时间消耗,验证本文提出算法的高效性及可扩展性.实验结果表明,本文提出的策略优于现有方法.

References:

[1] Cai D, Shao Z, He X, Yan X, Han J. Community mining from multi-relational networks. In: Proc. of the Knowledge Discovery in Databases (PKDD 2005). Berlin: Springer-Verlag, 2005. 445−452. [doi: 10.1007/11564126_44]

[2] Yang Q, Sze S H. Path matching and graph matching in biological networks. Journal of Computational Biology, 2007,14(1):56−67. [doi: 10.1089/cmb.2006.0076]

[3] Willett P, Barnard JM, Downs GM. Chemical similarity searching. Journal of Chemical Information and Computer Sciences, 1998, 38(6):983−996. [doi: 10.1021/ci9800211]

[4] Shasha D, Wang JTL, Giugno R. Algorithmics and applications of tree and graph searching. In: Proc. of the twenty-first ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. New York: ACM Press, 2002. 39−52. [doi: 10.1145/543613.543620]

[5] Bunke H, Allermann G. Inexact graph matching for structuralpattern recognition. Pattern Recognition Letter, 1983,1(4):245−253. [doi: 10.1016/0167-8655(83)90033-8]

[6] Sanfeliu A, Fu KS. A distance measure between attributedrelational graphs for pattern recognition. IEEE Trans. on Systems, Man, and Cybernetics, 1983,13(3):353−362. [doi: 10.1109/TSMC.1983.6313167]

[7] Zeng Z, Tung AKH, Wang J, Feng J, Zhou L. Comparingstars: On approximating graph edit distance. Proc. of the VLDB Endowment, 2009,2(1):25−36. [doi: 10.14778/1687627.1687631]

[8] Khan A, Li N, Yan X, Guan Z, Chakraborty S, Tao S. Neighborhood based fast graph search in large networks. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2011). New York: ACM Press, 2011. 901−912. [doi: 10.1145/1989323.1989418]

[9] Zhao X, Xiao C, Lin X., Wang W. Efficient graph similarity joins with edit distance constraints. In: Proc. of the 2012 IEEE 28th Int'l Conf. on Data Engineering (ICDE 2012). Washington: IEEE Computer Society, 2012. 834−845. [doi: 10.1109/ICDE.2012.91]

[10] Wang X, Ding X, Tung AKH, Ying S, Jin H. An efficient graph indexing method. In: Proc. of the 2012 IEEE 28th Int'l Conf. on Data Engineering (ICDE 2012). Washington: IEEE Computer Society, 2012. 210−221. [doi: 10.1109/ICDE.2012.28]

[11] Wang G, Wang B, Yang X, Yu G. Efficiently indexing large sparse graphs for similarity search. IEEE Trans. on Knowledge and Data Engineering, 2010,24(3):440−451. [doi: 10.1109/TKDE.2010.28]

[12] Le TH, Elghazel H, Hacid MS. A relational-based approach for aggregated search in graph databases. In: Proc. of the Database Systems for Advanced Applications (DASFAA 2012). Berlin: Springer-Verlag, 2012. 33−47. [doi: 10.1007/978-3-642-29038-1_5]

[13] Tian Y, McEachin RC, Santos C, States DJ, Patel JM. SAGA: A subgraph matching tool for biological graphs. Bioinformatics, 2007,23(2):232−239. [doi: 10.1093/bioinformatics/btl571]

[14] Tong H, Faloutsos C, Gallagher B, Eliassi-Rad T. Fastbest-Effort pattern matching in large attributed graphs. In: Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2007). New York: ACM Press, 2007. 737−746. [doi: 10.1145/1281192.1281271]

[15] Tian Y, Patel JM. Tale: A tool for approximate largegraph matching. In: Proc. of the 2008 IEEE 24th Int'l Conf. on Data Engineering (ICDE 2008). Washington: IEEE Computer Society, 2008. 963−972. [doi: 10.1109/ICDE.2008.4497505]

[16] Yan X, Yu PS, Han J. Graph indexing: A frequent structure-basedapproach. In: Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2004). New York: ACM Press, 2004. 335−346. [doi: 10.1145/1007568.1007607]

[17] Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. on Systems Science & Cybernetics, 2007,4(2):100−107. [doi: 10.1109/TSSC.1968.300136]

[18] Zheng W, Zou L, Lian X, Wang D, Zhao D. Graph similarity search with edit distance constraint in large graph databases. In: Proc. of the 22nd ACM Int'l Conf. on Information & Knowledge Management (CIKM 2013). New York: ACM Press, 2013. 1595−1600. [doi: 10.1145/2505515.2505723]

[19] Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics (NRL), 1955,2(1−2):83−97. [doi: 10.1002/nav.3800020109]

[20] Lu W, Hou J, Yan Y, Zhang M, Du Y, Thomas M. MSQL: Efficient similarity search in metric spaces using SQL. The VLDB Journal, 2017,26(6):829−854. [doi: 10.1007/s00778-017-0481-6]

赵展浩(1995－),男,浙江诸暨人,硕士生,主要研究领域为数据库,大数据管理系统.

黄斐然(1992－),男,硕士生,主要研究领域为数据库.

王晓黎(1985－),女,博士,助理教授,CCF 专业会员,主要研究领域为医疗健康大数据分析,图搜索,文本自动注解.

卢卫(1981－),男,博士,副教授,CCF 专业会员,主要研究领域为云计算与大数据管理,空间与文本数据库管理,索引技术.

杜小勇(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库系统,智能信息检索.