一种基于自适应结构概要的有向标签图子图匹配查询算法

张海威[1),2)] 解晓芳[1)] 段媛媛[1)] 温延龙[1),2)] 张 莹[1),2)] 袁晓洁[1),2)]

[1)](南开大学计算机与控制工程学院 天津 300071)

[2)](南开大学软件学院 天津 300071)

摘 要 有向标签图作为重要的数据表示模型,广泛应用于社交网络、语义网分析等信息技术相关的研究领域,子图匹配查询是图数据管理的重要研究问题,引起了研究者的广泛关注.有向标签图的子图同构和子图模拟匹配查询由于代价极高,不适用于大规模图数据的查询处理.本文针对有向标签图,研究基于自适应结构概要的子图匹配查询算法.首先基于图压缩的思想,提出一种满足顶点"局部双拟"关系且具有自适应更新特性的有向标签图结构概要模型,在缩小数据图规模的基础上,适应查询图的结构;然后采用图模拟方式,提出基于自适应结构概要模型的子图匹配查询算法,根据查询图顶点的标签,对与其匹配的结构概要顶点按照其中包含数据图顶点的数量由小到大排序,根据查询图顶点之间的 *rank* 差值在结构概要模型中实现顶点匹配;最后在真实数据集和模拟数据集上进行实验,结果表明:(1)自适应结构概要模型可根据查询图结构,实现对数据图的最大压缩;(2)可在 $O(|E|\log|V|)$ 的总体时间复杂度内实现结构概要的自适应更新以及基于图模拟方式的子图匹配查询.

关键词 有向标签图;局部双拟;结构概要;自适应更新;子图匹配查询

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2017.00052

An Algorithm for Subgraph Matching Based on Adaptive Structural Summary of Labeled Directed Graph Data

ZHANG Hai-Wei[1),2)] XIE Xiao-Fang[1)] DUAN Yuan-Yuan[1)] WEN Yan-Long[1),2)]
ZHANG Ying[1),2)] YUAN Xiao-Jie[1),2)]

[1)](*College of Computer and Control Engineering, Nankai University, Tianjin* 300071)

[2)](*College of Software, Nankai University, Tianjin* 300071)

Abstract Labeled directed graph, as one of the essential data models, has been widely used in many research areas, such as social network, semantic web analysis and so on. Subgraph matching is an important problem of graph data management and has drawn more and more attention of researchers. Traditional method of subgraph matching such as isomorphism matching and simulation matching cannot handle the large graph for their complexities of time. In this paper, we propose an efficient algorithm for subgraph matching based on adaptive structural summary of graph data. Firstly, we propose adaptive structural summary of graph data based on graph compression. The compressed graph can be adaptively updated according to the structure of query graphs by local bisimulation. The model can reduce the size of graph data and conform to the structure of query

收稿日期:2015-05-23;在线出版日期:2015-11-17.本课题得到国家"八六三"高技术研究发展计划项目基金(2013AA013204,2015AA015401)、国家自然科学基金(61170184,61402243)、天津市自然科学基金(13ZCZDGX02200,13ZCZDGX01098,13JCQNJC00100,16JCTPJC53700)资助.**张海威**,男,1980 年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为半结构化数据管理、图数据库与数据挖掘. E-mail:zhhaiwei@nankai.edu.cn. **解晓芳**,女,1991 年生,硕士研究生,主要研究方向为图数据库. **段媛媛**,女,1989 年生,硕士,主要研究方向为图数据管理. **温延龙**,男,1979 年生,博士,高级工程师,中国计算机学会(CCF)高级会员,主要研究方向为信息检索. **张 莹**(通信作者),女,1986 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为数据挖掘、信息检索. E-mail:yingzhang@nankai.edu.cn. **袁晓洁**,女,1963 年生,博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为数据库技术、数据挖掘、信息检索.

graphs. Secondly, we propose an algorithm for subgraph matching based on adaptive structural summary using graph simulation. According to the nodes in query graphs, each node in the structural summary is ranked in an ascent order. Nodes in structural summary can be matched to those in the query graphs by the distances between ranks of related nodes. Finally, we perform exhaustive experiments on real and synthetic datasets. The results demonstrate that: (1) our adaptive structural summary covers current queries with the lowest space cost; (2) the complexities of time for adaptively updating structural summary and subgraph matching are $O(|E|\log|V|)$.

Keywords　labeled directed graph; local bisimulation; structural summary; adaptive update; subgraph matching

1 引　言

图是一类重要的数据结构，广泛应用于描述社交网络、语义网等数据模型. 目前，随着互联网技术的发展和普及，上述领域的数据规模呈爆炸式增长的趋势，如何高效管理大规模图数据，成为了数据管理领域的热点研究问题，其中，子图匹配查询问题，由于其应用的广泛性而备受关注[1].

本文研究的有向标签图 G，可以定义为三元组 $G=(V,E,L)$，其中：(1) V 是顶点集合；(2) $E\subseteq V\times V$ 为边集合，$(v,u)\in E$ 表示从顶点 v 到顶点 u 的一条有向边；(3) L 为顶点标签函数，$L(v)$ 则表示顶点 v 的标签. 本文提出的子图匹配查询的相关方法，亦可通过简单修改用于边带有标签的有向图. 有向标签图是描述社交网络、互联网页面链接结构、语义网等复杂数据最基本的表示模型并得到了广泛的应用. 有向标签图 G(后文亦称为数据图)的子图匹配查询问题，通常是指给定查询图 G_q，在 G 中查找与 G_q 结构相同或相近的子图. 子图匹配查询通常应用于社交网络、生物信息等领域，如社团查询、蛋白质结构查询分析等问题，通常可以使用子图匹配的相关方法予以解决[2].

目前，子图匹配查询方法主要分为 3 类：同构匹配(Isomorphism Matching)[3-4]、近似匹配(Approximation Matching)[5-13] 和模拟匹配(Simulation Matching)[14-20]. 同构匹配是指在数据图中，找到与查询图结构完全一致的子图集合，该过程已被证明是 NP 完全问题[21]；近似匹配是相对于基于同构的精确匹配而提出的降低匹配条件的查询，通常将图模型转换为查询代价较低的其它模型，并利用专用的查询算法实现查询，而后再将查询结果进一步构建为与查询图匹配的子图，查询过程一般可在较低的时间复杂度内完成；模拟匹配是将子图匹配查询问题模拟为顶点之间匹配关系的查询问题，该过程的时间复杂度为多项式级，但由于模拟匹配的结果集一般无需还原为匹配子图，匹配结果通常与查询图在结构方面差异较大. 上述 3 类子图匹配查询方法，在处理大规模数据图的子图匹配查询时，都会遇到效率瓶颈. 为了提高大规模数据图的子图匹配查询效率，研究者主要采用 3 类方法：

(1) 引入索引，提高数据访问效率. 为数据图构建索引是加速查询的重要手段之一. 文献[22]对图索引进行了分类：基于邻接顶点的索引，边索引，路径索引以及频繁子结构索引. 然而，构建以上几类索引的时间复杂度极高，至多可达到 $O(|V|^4)$ 级别，空间复杂度通常也会超出内存的限制. 文献[23]构建了图数据索引框架 iGraph，在图集数据库中实现了常见的图索引.

(2) 压缩数据图，减少查询对象的规模. 图压缩技术一般可用于图模型计算、Web 和社交网络信息检索与挖掘、图数据查询等问题[24]. 文献[25]提出了一种面向子图匹配查询的压缩图模式，能够覆盖全部可能的子图匹配查询，但由于不具有查询自适应性导致压缩图的规模较之数据图没有显著的降低；基于图压缩技术，文献[8]提出数据图的本体索引，引入了本体库导致数据预处理的过程更加复杂.

(3) 将图数据映射为可在线性时间复杂度内实现数据查询的模型，如关系模型[6]、XML 模型[11]、邻接点模型[7,9]，而后再将查询结果还原为子图集合. 基于模式映射的方法，在数据处理时，需要将图模型转换为其它数据模型，数据预处理比较困难，而且查询时涉及到大量的数据连接操作.

综上所述，目前关于子图匹配查询的方法，主要

存在如下问题:(1)同构匹配查询方式由于其固有的复杂度因而不适用于对大规模图数据进行子图匹配查询;(2)近似匹配查询可以通过放宽匹配条件,用较低的时间代价实现匹配查询,但是通常需要将图数据转换为其它模型,数据预处理以及结果集连接的代价很高;(3)模拟匹配查询方式将子图匹配问题转换为求取查询图顶点集与数据图顶点集之间的二元关系问题.图模拟[26]忽略了子图在边匹配方面的问题,通过该方法匹配的子图在结构方面,通常与查询图差异较大;边界模拟[19]和强模拟[18]分别通过增加边界约束条件以及限制查询范围,提高模拟匹配在结构方面的准确度,但是子图匹配查询的整体代价达到了 $O(|V|^3)$,同样不适用于大规模图数据查询.

本文将针对有向标签图的子图匹配查询问题展开研究,提出一种基于自适应结构概要的子图匹配查询算法,无需高代价的数据预处理过程和查询结果连接过程,兼顾查询匹配的效率与查询子图结构的相似性,论文的主要贡献包括:

(1)基于图压缩技术,提出有向标签图的结构概要模型,在保持图结构的基础上,实现对图数据的最大程度压缩;

(2)提出基于局部双拟关系的结构概要自适应更新算法,使结构概要适应当前的查询需求;

(3)提出基于自适应结构概要的有向标签图子图匹配查询算法,实现对查询图的子图匹配查询.

2 相关工作

目前,子图查询方法根据匹配方式分为同构查询、模拟查询和近似查询,同构查询被认为是 NP 完全问题,模拟查询可在多项式级时间复杂度内实现[25],而近似查询亦被认为是类同构查询,因此也属于 NP 完全问题,但是可以通过放宽相似条件降低时间复杂度.基于图模拟的子图匹配查询策略的主要问题在于匹配子图与查询图之间的结构差异较大,而且此类方法全部用于有向图的查询处理.英国爱丁堡大学 Fan 教授的研究团队长期以来一直致力于子图模拟匹配方法的研究,取得了一系列丰富的成果.文献[19]在图模拟的原始定义[26]基础上,进行了扩展,提出了边界模拟的概念,并重新定义了带有边权重的查询图,支持满足一定语义相似性和路径长度约束的子图匹配.由于匹配条件较之图模拟更加严格,匹配代价也高于图模拟.Ma 等人[18]对边界模拟进行了改进,提出了"强模拟"的概念,通过设置子图匹配半径,对候选集进行剪枝,提高了边界模拟匹配的效率且提高了匹配子图与查询图在结构方面的相似性.Fan 等人[15]在文献[25]图压缩的基础上提出了基于图片段的强模拟匹配方法.其基本思想是将大规模图数据转换为与查询有关的图片段(fraction),将图的片段作为对象(bounded resource)进行子图匹配查询.

近几年,该研究团队将 top-k 策略融入到子图模拟匹配算法的研究中.Fan 等人[16]将子图模拟匹配问题进行了扩展定义,引入查询图的特征顶点 u_0,将子图匹配问题转换为特征顶点匹配,缩减了候选集的规模.在匹配过程中,结合顶点之间的相似度和相异度,计算顶点匹配值.另外,Fan 等人[13]对边界模拟匹配方法进行了改进,提出了基于视图(View)的子图匹配查询方法.视图是根据不同查询目的构建的图数据的子结构模式,当查询图与某个视图等价时,可以提高模拟匹配查询的效率.

除了子图模拟匹配之外,研究者在子图同构匹配和近似匹配查询问题的研究中同样取得了丰富的成果,尤其是近 3 年的研究主要集中在 top-k 子图匹配查询.与图模拟方式不同之处在于,近似匹配查询保证了匹配子图与查询图在结构上的相似性(因此也被认为是类同构查询).另外,为了提高子图匹配效率,近似匹配放宽了对于子图匹配条件的限制,多数情况下通过模型映射的方式,将图模型转换为可用较低时间代价实现查询的模型,而后利用专用的算法实现查询.

Zhao 等人[10]将子图匹配转换为顶点匹配,根据 SPath 索引,将查询图与数据图中顶点的 k-hop 邻接点模型进行匹配,由于邻接点模型基于最短路径构建,实际上是对相关顶点的最短路径进行匹配.Khan 等人[7,9]提出了基于信息传递的邻接点模型,以邻接点的特征构建相似度匹配模型,计算顶点的相似度,以顶点相似描述子图相似.Wu 等人[8]通过对有向图数据进行压缩,构建可支持全部可能子图匹配查询的本体索引,在本体索引的基础上提出了 top-k 子图匹配算法 kMatch.kMatch 算法在计算匹配子图的过程中,利用堆结构实现匹配子图的相似性的排序,进而得到 top-k 匹配子图集合.Gupta 等人[3]研究边带有权重的图数据的子图匹配问题,分别利用图拓扑索引、最大元路径权重索引和有序边

列表，按照路径扩展的方式进行顶点标签、边的匹配，实现子图查询，同时，估算匹配度上界，在 top-k 堆结构中进行剪枝，得到 top-k 匹配子图. Yang 等人[13]对子图匹配查询问题进行了扩展，提出了无模式和无结构(schemaless and strcutureless)的图查询解决方案 SLQ，将图数据查询问题表示为概率模型，估计顶点和边的匹配度代价，并通过消息传递机制和迭代策略，实现 top-k 子图匹配. 该方法将机器学习思想融入到图查询中，亦可支持实体匹配查询和语义近似查询.

Cheng 等人[6]将数据图转换为基础表 $R_{(A,D)}$，将查询图转换为基于 t-query 树模型的多维查询空间，将子图查询问题转换为基于 $R_{(A,D)}$ 表的结构连接查询，然后利用小枝模式匹配的思想处理树模型的查询，匹配过程中将 top-k 函数融入基础表的连接查询过程. Zou 等人[12]提出了一种基于距离连接的子图匹配查询方法，将子图匹配查询转换为“边”的查询，利用顶点之间的距离与边进行匹配，同时，利用连接代价估计，对查询过程进行相关优化，得到与查询图匹配的全部子图集合.

文献[24]将目前的图压缩技术归纳为 4 类：一是根据数据图的邻接矩阵或邻接表等存储结构的特征进行压缩的技术[27]，主要应用于查询邻接点等简单的图计算；二是针对表示互联网结构的图数据，基于网页特征，采用合并相似项的方式进行压缩的技术[28]，主要应用于 Web 搜索与挖掘；三是针对社交网络图数据，基于节点特征选取、计算、排序的策略进行压缩的技术[29]，主要应用于社交网络信息搜索；四是针对特定的查询需求，如内外邻查询[30]、可达查询[25]、子图匹配查询[25]、特定结构查询等的压缩技术[31]. Fan 等人[25]针对图的可达查询和模拟查询提出了保持查询结构的图压缩方法，其基本思想是将数据图的全部顶点，按照双拟关系合并为压缩图的一个顶点，从而在保持图结构的前提下，减少数据图的规模，同时保证可达查询和图模拟查询的正确性.

关于图数据的自适应结构概要，目前的研究成果主要应用于 XML 数据查询. Chung 等人[32]为半结构化数据 XML 设计能够随着查询负载进行自适应维护的路径索引模型 APEX，首次提出了自适应索引(Adaptive Index)的概念. Chen 等人[33]引入 APEX 索引的思想，提出了自适应结构概要 $D(k)$-Index，为每个 XML 节点建立路径长度不大于 k 的路径索引，能够根据查询负载的变化进行相应的更新. 自适应结构概要只能够对带有根节点的数据进行路径查询，无法直接应用于一般图数据查询. 但是，自适应结构概要的提出，在压缩查询对象规模并实现高效查询处理方面，具有重要的理论价值.

本文基于图压缩技术，提出有向标签图的结构概要模型以及针对不同查询图的自适应更新算法、子图匹配查询算法. 与已有工作的区别在于实现了对数据图的最大程度压缩，同时，结构概要具有查询自适应性，能够以较低的代价动态更新以满足当前查询需要，并能够高效地实现子图匹配查询.

3　有向标签图的结构概要模型

结构概要是重要的数据压缩模型，最早是针对 XML 数据提出的[34]，旨在约简描述 XML 文档的树型结构. XML 的结构概要也是树型结构，其中不存在路径相同且标签相同的顶点. 图结构较之 XML 文档树型结构更为复杂，而且没有与之匹配的数据模式，因此构建图的结构概要非常困难. 论文将基于图压缩技术，研究有向标签图的结构概要模型的构建以及自适应更新问题，进而研究基于自适应结构概要的子图匹配查询算法. 本节将利用顶点等价类的概念进行图顶点的划分，而后提出有向标签图的结构概要模型构建算法并进行相应的分析.

3.1　图顶点等价类的粗糙划分

构建结构概要模型的基本思想是根据等价关系对数据图顶点进行划分，判断数据图顶点是否等价的依据是顶点的标签与结构特征. 显然，描述顶点的结构特征并由此判定顶点等价较之标签更加困难. 与 XML 树形结构相比，图结构更为复杂，因此无法借助顶点编码等自顶向下的策略描述图顶点的结构特征，针对本文研究的有向标签图，可采用自底向上的策略，从数据图的叶节点，逐层向上计算顶点的结构特征. 采用这种策略，需要保证数据图是具有叶节点的无环图，但是，描述社交网络、网页链接等真实数据的有向图，通常是循环图，因此，当处理对象为有向循环图时，需要利用强连通分量将其转换为无环图.

将有向循环图转换为无环图的方法为：

给定一个有向循环标签图 $G=(V,E,L)$ 及 G 的强连通分量集合 $C=\{c_1,c_2,\cdots,c_m\}$，用函数 $c(v)=c_i$ 描述顶点 v 属于某个强连通分量 c_i，可将 G 转换为有向无环标签图 $G^{scc}=(V^{scc},E^{scc},L^{scc})$，其中：

(1) V^{scc} 是顶点的集合，满足：

$$V^{scc}=\{v|\{v\in V\bigcap v\notin c_i\}\bigcup c_i\}.$$

即 V^{scc} 中的顶点，或者来自于 V 中不属于任何强连通分量的顶点，或者来自于强连通分量中的顶点合并而成的新顶点.

(2) E^{scc} 是边的集合，满足：

$$E^{scc}=\{(v_1,v_2)|\{(v_1,v_2)\in E\}-\{(v_1,v_2)\in E\bigcap v_1,v_2\in c_i\}\}.$$

即 E^{scc} 中的边，其顶点 v_1,v_2 满足如下条件之一：

① v_1,v_2 都不是强连通分量中的顶点；

② v_1,v_2 有一个是强连通分量中的顶点，另一个不是；

③ v_1,v_2 属于不同的强连通分量.

(3) L^{scc} 是顶点标签的集合，对于 G^{scc} 中由 G 的强连通分量构成的顶点，其标签值为特殊的标记“Ψ”.

由上述分析可知，一个有向无环标签图，其相应的 G^{scc} 即为该图本身. 本文采用 Tarjan 算法[35]，经一次遍历即可求得数据图的 G^{scc}.

G^{scc} 即为适用于本文所述相关方法的有向无环图. 而后，根据 G^{scc} 顶点的可达子图，计算顶点的结构特征 *rank*，作为划分顶点等价类的重要依据.

给定一个有向标签图 $G=(V,E,L)$，对于某个顶点 $v\in V$，其可达子图 $g_v=(N(v),E,L(v))$ 表示包括 v 在内，从 v 出发所有的可达顶点构成的 G 的子图. 若该子图无环，则顶点 v 属于良构顶点集合 $WF(G)$，即 $WF(G)=\{v|v\in V,g_v$ 无环$\}$.

无环图 G^{scc} 保证了其中每一个非叶节点顶点可以用其与叶节点之间的距离值 *rank* 来描述其结构特征，由于非叶节点到达叶节点的路径不唯一，因此，计算 *rank* 值的过程较复杂，可借鉴文献[16]中的方法进行 *rank* 值的计算：如果 v 是 G 的叶节点则 $rank(v)=0$；如果 v 不是 G 的叶节点但 $c(v)$ 是 G^{scc} 的叶节点则：$rank(v)=-\infty$，否则：

$$rank(v)=\max\{(1+rank(v'))|(c(v),c(v'))\in E^{scc}\}.$$

通过顶点的标签以及 *rank* 值，可以定义数据图顶点的等价类.

定义 1. 顶点等价类.

顶点等价类是指数据图中，具有相同标签和 *rank* 值的顶点集合.

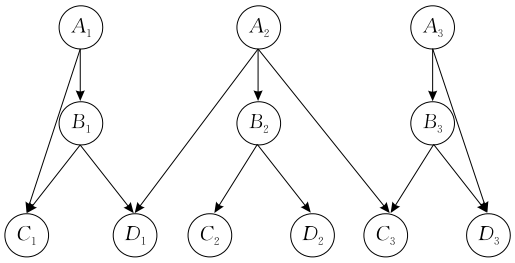
由上述定义，可将有向标签图 G 中的顶点集合 V 表示为若干等价类的集合：

$$Par=\{P_1,P_2,\cdots,P_r\},$$

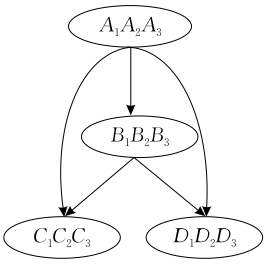
其中，属于同一个等价类 P_i 的任意两个顶点 v 和 u 之间满足：(1) $L(v)=L(u)$；(2) $rank(v)=rank(u)$.

根据该条件，可对数据图进行粗糙的等价类划分. 上述顶点等价类的定义以及划分方法，是最基本的等价类划分方法，也是表示顶点的标签和结构特征的基本方法，旨在实现对数据图的最大程度压缩. 不同的等价类定义方法，会使划分类别发生变化，必要的情况下，可以根据不同的查询目标制定顶点划分的条件，实现不同策略的划分.

图 1 给出了图顶点等价类的划分示例：根据 *rank* 值的计算方法，可得到：$rank(C_1)=rank(C_2)=rank(C_3)=0$，因此顶点 C_1,C_2,C_3 被划分为一个等价类；同理，$rank(A_1)=rank(A_2)=rank(A_3)=2$，顶点 A_1,A_2,A_3 也被划分为同一等价类. 算法 1 展示了划分顶点等价类的具体过程：首先利用 Tarjan 算法(步 1)进行一次深度优先遍历(DFS)，计算出所有强连通分量；接着再进行一次 DFS，根据 G^{scc} 计算出各顶点的 *rank* 值(步 2～步 5)；最后根据顶点标签和 *rank* 值，将原始图数据划分为不同的等价类(步 6～步 9). 显然，算法 1 需要对图数据进行两次深度优先遍历，总体时间复杂度为两次 DFS 的时间复杂度，即为 $O(|V|+|E|)$.



(a) 数据图



(b) 顶点等价类划分

图 1 数据图及其顶点等价类

算法 1. 顶点等价类的粗糙划分.

输入：有向标签图 $G=(V,E,L)$

输出：顶点等价类集合 *Par*

1. 使用 Tarjan 算法计算 G 的强连通分量
2. BEGIN DFS procedure
3. Computing $rank(v)$
4. $r=\max\{rank(v),v\in V\}$
5. END DFS procedure

6. $Par=\{P_i \mid i=-\infty, 0,1,\cdots,r\}$
7. FOR ALL $P_i \in Par$
8. 根据标签进一步划分 P_i
9. END FOR
10. RETURN Par

对图顶点进行等价类划分，是为数据图 G 构建结构概要的基础. 等价类中的图顶点，具有相同的标签和结构特征(由 *rank* 值表示)，但是，从顶点 *rank* 值的定义容易推断出，顶点 *rank* 值描述的是图中顶点与叶节点(出度为 0 的顶点)之间的距离最大值，而无法描述顶点连接关系(即前驱后继顶点)方面的特征. 因此，顶点的 *rank* 值并不能全面反映顶点的结构特征，即顶点的值粗略刻画顶点的结构特征. 基于上述方法对数据图顶点进行划分，属于粗糙划分，旨在最大程度地压缩数据图. 而基于顶点粗糙划分构建的结构概要模型，执行子图匹配查询时需要根据查询图进行进一步的细化，使其能够更精确地描述图顶点的等价关系.

3.2　结构概要模型构建

基于顶点等价类，可定义有向标签图 G 的结构概要模型.

定义 2.　结构概要模型.

有向标签图 G 的结构概要模型可以表示为四元组：$G_S=(V_S,E_S,L_S,R_S)$，其中：(1) $V_S=\{v_{s1},v_{s2},\cdots\}$ 是结构概要中顶点的集合，每一个顶点 v_{si} 对应图 G 中的一个顶点等价类，即存在图 G 中顶点 v 到 G_S 的映射函数 S，$\exists v\in V, S(v)=v_{si}$；(2) E_S 是结构概要中边的集合，设顶点 $v_{si},v_{sj}\in V_s$，在图 G 中存在一条边 $(v,u)\in E$ 且 $S(v)=v_{si}$，$S(u)=v_{sj}$，则在 G_S 中存在边 $(v_{si},v_{sj})\in E_s$；(3) L_S 是结构概要中顶点标签的映射函数集合 $L(v_{si})=l_i$；(4) R_S 是结构概要中顶点的 *rank* 值映射函数集合 $R(v_{si})=r_i$.

由上述结构概要的定义可知，为有向标签图构建结构概要模型的核心是划分数据图顶点的等价类，一方面，同一等价类的顶点压缩为结构概要中的一个顶点；另一方面，结构概要中的标签映射函数集和 *rank* 值映射函数集也由顶点等价类的相关属性获取，同时，也需要遍历数据图，为结构概要添加符合相关定义的边. 图 1(b)是根据顶点等价类的粗糙划分构建的结构概要模型. 结构概要的构建过程如算法 2 所述.

算法 2.　结构概要模型的构建.

输入：有向标签图 $G=(V,E,L)$，顶点等价类集合 Par

输出：结构概要 G_s

1. $V_s=\varnothing, E_s=\varnothing, L_s=\varnothing, R_s=\varnothing$
2. FOR ALL $P_i\in Par$
3. $v_{si}=P_i$，$V_s=V_s\cup v_{si}$
4. $L_s=L_s\cup L(v)$，$v\in P_i$
5. $R_s=R_s\cup rank(v)$，$v\in P_i$
6. END FOR
7. FOR ALL$(v,u)\in E$
8. IF $v\in v_{si}\cap u\in v_{sj}$
9. $E_s=E_s\cup(v_{si},v_{sj})$
10. END IF
11. END FOR
12. RETURN $G_s=(V_S,E_S,L_S,R_S)$

由算法 2 可知，结构概要构建分为两个过程：一是根据顶点等价类构建结构概要中的顶点集合、标签映射函数集合以及 *rank* 值映射函数集合(步 2～步 6)；二是通过对图 G 的边集合 E 进行遍历，将符合条件的边添加到结构概要的边集合，并最终完成结构概要的构建(步 7～步 12). 其中，第 1 个过程的时间复杂度为 $O(|Par|)$，即与图 G 顶点等价类的数量线性相关；第 2 个过程在遍历图 G 顶点的过程中，每次访问一条边时，还需要判断起止顶点所属的等价类，因此时间复杂度为 $O(|E|\cdot|V|)$，对于大规模数据而言是无法承受的，因此，该过程需要进一步优化. 可将对于结构概要中边的处理过程纳入到对图 G 的深度优先遍历过程. 由等价类划分的过程，可知图 G 中每个顶点所属的等价类，因此，可在对图 G 的再次遍历过程中，每访问到一条边时，直接获取起止顶点所属的等价类信息，进而完善结构概要的边集合. 这样，该过程的时间复杂度可降低至 $O(|E|+|V|)$，因而构建结构概要的总体时间复杂度为

$$O(|Par|+|E|+|V|)\approx O(|E|+|V|).$$

由结构概要的定义可知，它是保留数据图基本结构的一种数据压缩模式.

性质 1.　结构概要是数据图的最大程度压缩.

证明.　根据顶点等价类的定义可知，结构概要模型中的顶点，由数据图中的属于同一个等价类的顶点构成，如果存在更大程度的压缩，势必会将不同等价类合并为一个顶点，显然与结构概要模型的定义不相符；另外，根据结构概要模型的定义可知，数据图中属于两个等价类顶点的边，在结构概要中合并为一条，显然也是对于数据图边集合的最大程度压缩.　　证毕.

由性质 1 可以分析，结构概要实现了对数据图顶点和边的压缩，一方面，结构概要可将标签和 *rank* 值都相同的图顶点压缩为一个顶点；另一方

面,可将图中起止顶点分别属于相同等价类的边压缩为一条边.因此,结构概要在图数据管理中具有重要的意义,例如,可为图数据构建结构概要索引,提高对图数据的处理效率.下一节,将讨论如何利用结构概要进行子图匹配查询处理.

4 基于自适应结构概要的子图匹配查询算法

结构概要已经被证明是数据图的最大压缩模式,结构概要基于数据图的顶点等价类构建,通过结构概要能够访问到数据图中的全部顶点,因此,结构概要为子图匹配查询提供便利,本节将提出一种基于结构概要模型的子图匹配查询算法,首先,通过讨论结构概要与子图匹配查询之间的关系,提出基于局部"双拟"关系的结构概要自适应更新策略(4.1 节和 4.2 节);然后,提出基于自适应结构概要实现子图匹配查询的算法并进行相关的分析(4.3 节).

4.1 结构概要顶点的局部双拟关系

结构概要模型根据数据图顶点的等价类进行构建,但是,根据前文对顶点等价类划分的过程可知,结构概要模型对数据图实现了最大程度的压缩,是根据顶点标签和 *rank* 值进行的粗糙划分,顶点之间的连接关系(前驱与后继顶点)在结构概要中无法得到正确的体现,仅仅通过结构概要模型无法实现子图匹配查询.因此,根据算法 2 构建的结构概要模型称为粗糙模型(raw model),要实现子图匹配查询,必须保证结构概要模型中与查询图可能匹配的顶点之间具有正确的连接关系,因此,需要对原有的顶点等价类进行细化,也就是结构概要模型的更新过程.该过程依赖于查询图的结构,属于自适应更新.通过自适应更新,针对不同查询图可得到结构概要的细化模型,用于对不同的查询图进行子图匹配查询.

本文将基于双拟关系(bisimulation relation)[36],研究子图匹配问题.双拟关系是描述半结构化数据节点相似性的重要标准,广泛应用于 XML 数据查询与图数据查询.双拟关系是指图数据 $G=(V,E,L)$ 上的一个二元关系 $B=V\times V$,定义如下:(1)对于任意 $(u,v)\in B$,$L(u)=L(v)$;(2)对于任意的 $(u,u')\in E$,存在 $(v,v')\in E$,使得 $(u',v')\in B$;(3)对于任意的 $(v,v')\in E$,存在 $(u,u')\in E$,使得 $(v',u')\in B$.满足双拟关系的顶点 u 和 v 记作:$u\approx v$.

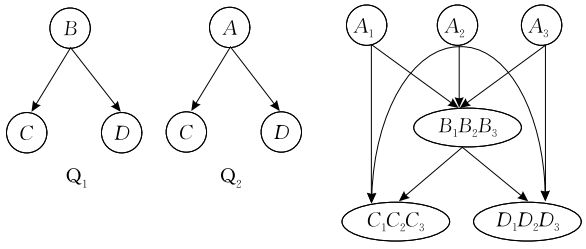
具体来说,$(u,v)\in B$ 成立,当且仅当对于 u 的每一个邻接点 u',存在一个 v 的邻接点 v',并且 $(u',v')\in B$;反之亦然.

根据双拟关系的定义,对于给定的查询图 $G_q=(V_q,E_q,L_q)$,局部双拟关系是指数据图上的一个二元关系 $S\subseteq B$,$\forall(m,n)\in S$,$m\approx n$ 且 $L(m)$,$L(n)\in L_q$.

Fan 等人[25]已证明了双拟关系适用于图模拟匹配查询,局部双拟关系是在结构概要模型的基础上,将与查询图标签相匹配的顶点提取出来并使其满足双拟关系,而无需对数据图全部顶点进行双拟关系的判断以及划分.

根据局部双拟关系,细化数据图的顶点等价类,实现结构概要模型的更新.结构概要的构建是根据顶点标签、层次信息对数据图顶点的粗糙划分,为了使结构概要模型满足子图匹配查询的需求,需在粗糙划分的基础上,对结构概要中与查询图标签匹配的顶点进行细化,细化后的结构概要顶点中所包含的数据图顶点之间全部满足双拟关系,这样,可从细化后的结构概要中找到与查询图顶点匹配的数据图顶点集,进而实现子图查询.

图 1(b)所示的结构概要粗糙模型中,顶点 A 由数据图的 3 个顶点 A_1,A_2 和 A_3 压缩而成,这 3 个顶点的标签都为"A",*rank* 值都为 2,但由于它们的后继关系不同(A_1 的邻接点 B 和 C,A_2 的邻接点为 B,C,D,A_3 的邻接点 B 和 D),因此不满足双拟关系,如果执行如图 2(a)中的查询图 Q_2 时会产生错误的匹配子图,因此应该根据查询图将结构概要的粗糙模型细化为图 2(b)所示的结构以保证子图匹配查询的准确性.另外,当执行图 2(a)中的查询图 Q_1 时,由于查询图可能匹配的顶点 B 包含了数据图顶点 B_1,B_2 和 B_3 满足双拟关系(邻接点都为 C 和 D),因此无需细化结构概要粗糙模型,即可得到正确的匹配子图.因此,在执行查询图 Q_1 时,只需结构概要的局部顶点 B,C 和 D 满足双拟关系即可,而顶点 A 无需满足双拟关系.



(a) 查询图 (b) 更新的结构概要

图 2 根据查询图更新结构概要模型

结构概要顶点细化的过程,是根据查询图的自适应更新过程,将在下一节讨论结构概要自适应更

新的算法.

4.2　结构概要的自适应更新

由结构概要的构建过程可知，直接利用结构概要的粗糙模型进行子图匹配查询可能无法得到精确的匹配子图集合. 因此，进行子图匹配查询时，需要对结构概要模型顶点进行细化，保证与查询图相关的子结构准确地反映到结构概要中. 图 2(b)所示的结构概要即根据查询图 Q_2 对图 1(b)的结构概要进行的细化. 另外，结构概要顶点包含的数据图顶点之间，通常会有边相连，结构概要顶点的细化划分通常会产生新的循环结构，因此，对结构概要顶点的细化时需要保持该顶点的稳定性(stability)[34]. 结构概要的细化过程可以认为是依赖于查询图结构的自适应更新过程.

定义 3.　结构概要模型的自适应更新.

给定数据图 G 的结构概要模型 $G_s=(V_s,E_s,L_s,R_s)$，G_s 可根据查询图 $G_q=(V_q,E_q,L_q)$ 进行自适应更新得到细化后的结构概要模型 $G_{sq}=(V_{sq},E_{sq},L_{sq},R_{sq})$，其中：(1) $V_{sq}=V_{nr}\cup V_r$，$V_{nr}\subseteq V_s$ 表示未细化的等价类构成的结构概要顶点，V_r 表示经过细化后的等价类构成的结构概要顶点；(2) $E_{sq}=E_{nr}\cup E_r$，$E_{nr}\subseteq E_s$ 表示未细化的等价类构成的结构概要顶点之间的边，E_r 表示经过细化后的等价类构成的结构概要顶点之间的边；(3) $L_{sq}=L_s$；(4) $R_{sq}=R_s$.

由上述分析以及定义 6 可知，结构概要模型的自适应更新过程，是数据图顶点等价类的细化过程，包括两个步骤：(1) 根据查询图结构，对结构概要中与查询图顶点标签匹配的顶点根据连接关系进行内部分裂，使该顶点内部达到某个稳定的状态；(2) 由于内部分裂导致结构概要的其它相关顶点进行分裂(外部分裂)，保证分裂后的每个分块包含的数据图顶点具有相同的前驱后继关系(即前驱后继顶点对应的标签相同).

数据图等价类的细化是非常复杂的过程. 图 2(b)所示的结构概要，将标签为 A 的顶点等价类细化为 $\{A_1\}$，$\{A_2\}$ 和 $\{A_3\}$. 但由于数据图及其结构概要的结构通常比较复杂，结构概要的某一个顶点细化后，通常会导致该顶点的前驱或后继顶点不再满足双拟关系，从而需要进行反复的细化，使得结构概要模型的结构不断地发生变化. 为解决该问题，需要确定顶点等价类细化的标准，保证内部分裂达到稳定状态，从而不会影响到其前驱及后继节点的双拟特性.

细化的标准来自于集合划分的稳定性理论，文献[34]对稳定划分进行了定义：对于两个数据图顶点集合 X 和 Y，如果 X 是 Y 的后继顶点集合的子集，或者 X 与 Y 的后继顶点集合不相交，则 X 相对于 Y 是稳定的. Paige 等人在文献[37]中指出，对于给定的基于关系 E 的集合 U 以及 U 上的一个初始划分 P，存在唯一的 P 的细化划分 Q，使得 Q 中所包含的块数最少，并且划分 Q 是稳定的. 根据该结论，结构概要在初始化构建后，$rank$ 值大于负无穷的每个等价类即为集合 U；该等价类中的边关系即为关系 E；在初始化构建时，由外部分裂所引起的划分即为初始划分 P；如果在首次外部分裂中，原等价类没有分裂，则 P 即为该等价类本身. 因此，可以对每个涉及到查询的等价类计算一个细化划分 Q，使得 Q 中所包含的块数最少，并且是稳定的.

进行子图匹配查询时，首先根据查询图的顶点标签寻找结构概要中相关的顶点，然后再根据数据图顶点的连接关系(边)进行分裂，直到达到稳定状态即完成结构概要的一次更新. 对于新的查询，需要重复上述更新过程，使得结构概要能够覆盖新查询图可能的匹配子图. 从结构概要的更新过程可以分析，结构概要的更新是一个使其规模不断增长的过程，以覆盖多样化的子图匹配查询. 当对于数据图的查询，涉及到该图的全部子图时，结构概要的规模达到最大，因此，基于自适应结构概要进行子图匹配查询的意义在于：对于频繁的子图匹配查询，结构概要更新的程度最小甚至无需进行自适应更新，在一定程度上，缩小了查询对象规模，从而提高了查询效率.

算法 3 展示了面向查询图的结构概要模型自适应更新过程. 首先构建查询图的顶点标签集合 $LabelSet$(步 1)并将结构概要中顶点表示为数据图顶点等价类的集合(步 2)，然后从结构概要 G_s 中选择与查询图某个标签匹配的顶点，调用 PT_Split 算法[37]对顶点集合 B_i 进行内部分裂(步 5)，达到某个稳定的状态后，合并相关分块并通过外部分裂得到顶点等价类的细化划分(步 6～步 13)，进而调整结构概要(步 14).

算法 3.　结构概要的自适应更新.

输入：结构概要 $G_s=(V_s,E_s,L_s,R_s)$，查询图 $G_q=(V_q,E_q,L_q)$

输出：更新后的结构概要 G_{sq}

1. $LabelSet=\{L_q(v)\mid v\in V_q\}$
2. $Par=\{B_i\mid B_i\subseteq V_s,\forall v_{si}\in B_i,rank(v_{si})=i,$
$i=0,1,\cdots,\rho\}$

3. FOR ALL $i=0,1,\cdots,\rho$

4. IF $L_s(B_i)\in LabelSet$

5. $PT\_Split(B_i)$

6. Collapse the blocks Split from B_i

7. FOR ALL $n\in B_i$

8. FOR ALL $C\in Par\cap\{\bigcup_{j=i+1}^{\rho}B_j\}$

9. $Par=(Par\backslash C)\cup\left\{\begin{matrix}\{m\in C|(m,n)\in E\},\\ \{m\in C|(m,n)\notin E\}\end{matrix}\right\}$

10. END FOR

11. END FOR

12. END IF

13. END FOR

14. Update V_s, E_s by Par and further construct G_{sq}

15. RETURN G_{sq}

文献[37]已经证明了算法 PT_Split 的时间复杂度为 $O(|E\upharpoonright B_i|\log|B_i|)$，其中 $|E\upharpoonright B_i|$ 表示数据图中与顶点分块 B_i 相关的边的数量. 算法 3 包含了若干次内部分裂和外部分裂的过程，其关键步骤在于待处理的顶点集合 B_i 的数量. 假设结构概要模型 G_s 中，有 n 个顶点与查询图包含的顶点匹配，则算法 3 的总体时间复杂度约为

$$n\cdot O(|E\upharpoonright B_i|\log|B_i|).$$

当 G_s 中每个顶点都需要进行处理时，达到最差时间复杂度，约为 $O(|E|\log|V|)$.

在根据查询图对结构概要进行自适应更新时，每次匹配一个新的查询图，都会根据当前查询图的顶点标签，选择结构概要中相关的顶点进行细化，并在结构概要中保留这种细化，因此，无需恢复结构概要的原始状态. 如果新的查询图中，包含已匹配过的查询图涉及到的顶点标签，且已经完成了这些顶点的细化，则无需重新细化，从而减少新查询图对结构概要的更新代价；如果新的查询图中，不包含任何已细化过的顶点标签，则需要在结构概要中细化相关的顶点. 因此，论文提出的策略，对于匹配频繁查询的子图更具有意义.

性质 2. 基于图压缩的结构概要模型，如果其顶点包含的数据图顶点之间满足双拟关系，则能够保证以图模拟方式进行正确的子图匹配查询.

证明. 根据定义可知，基于结构概要模型，按照图模拟方式的进行子图匹配查询，是求取一个二元关系 $B\subseteq V_q\times V_S$，$\exists(u,v)\in B$，$u\in V_q$，$v\in V_S$，满足以下两个条件即可认为是正确的查询：

(1) 顶点的 u，v 具有双拟关系，即 V_q 与 V_S 具有相同的标签和结构特征. 由于构建结构概要和自适应更新结构概要的过程中，都保证结构概要顶点所包含的数据图顶点具有双拟关系，显然查询图的顶点 u 能够与数据图中的顶点进行正确的匹配；

(2) 顶点之间的连接关系匹配. $\exists(u,u')\in E_q$，如果存在 $u',v'\in V_S$，满足 $u\approx v$，$u'\approx v'$，且 $(u',v')\in E_S$. 根据结构概要模型的定义能够证明，数据图中至少存在 $u_G\in u'$ 和 $v_G\in v'$，且 $(u_G,v_G)\in E$，因此，顶点之间的连接关系通过边匹配方式得到证明. 证毕.

由性质 2 可知，只要保证结构概要中与查询图相关的顶点所包含的数据图顶点全部满足双拟关系，即可以保证基于图模拟方式进行子图匹配查询的正确性. 结构概要模型基于图压缩技术构建，为了保证结构概要模型具有较低的压缩率，必然造成图结构信息的损失. 但是，结构概要的自适应更新过程，能够保证与查询图相关的顶点满足双拟关系，由上述分析可知，这种策略能够保证子图匹配查询的正确性.

4.3 子图匹配查询算法

根据顶点的双拟关系，可按照图模拟的方式定义基于结构概要的子图匹配查询.

定义 4. 基于结构概要的子图匹配.

给定数据图 $G=(V,E,L)$ 及其结构概要 $G_{sq}=(V_{sq},E_{sq},L_{sq},R_{sq})$、查询图 $G_q=(V_q,E_q,L_q)$，基于结构概要的子图匹配是指一个二元关系 $S=V_q\times V_{sq}$，$\forall(u,v)\in S$：

(1) $u\in V_q$，$v\in V_{sq}$，且 $L(u)=L(v)$；

(2) $\forall(u,u')\in E_q$，存在 $(v,v')\in E_{sq}$，且 $(u',v')\in S$.

针对当前查询图进行自适应更新后的结构概要，可用于在数据图中对该查询图进行子图匹配查询. 基于自适应结构概要的子图匹配查询算法，首先对查询图进行处理，对于每一个顶点，在结构概要中找到标签一致的匹配顶点，构建候选匹配顶点集合；然后，根据顶点的 $rank$ 值，对候选集进行修剪，具体过程为：(1) 计算查询图的顶点 $rank$ 值；(2) 对包含于候选集顶点的数据图顶点进行遍历，记录该顶点与其邻居顶点的 $rank$ 差值，若该差值等于查询图中相匹配的顶点与其邻居的 $rank$ 差值，则匹配成功；否则将该候选顶点移出候选集. 直至所有查询顶点匹配完成，算法结束.

算法 4. 子图匹配查询算法.

输入：查询图 $G_q=(V_q,E_q,L_q)$ 及其相应的结构概要 $G_{sq}=(V_{sq},E_{sq},L_{sq},R_{sq})$

输出：子图匹配查询结果集 M_q

1. 对查询图进行 DFS 并计算 $rank(v_q)$

2. 根据标签计算 G_{sq} 中与 $v_q \in V_q$ 匹配的顶点数量并对 $v_q \in V_q$ 按匹配数据图顶点的数量进行升序排序
3. $M_q = \varnothing$
4. FOR ALL $u \in V_q$
5. FOR ALL $v \in V_q$, $v \in u.neighbor$, $u.neighbor = \{u, u' \in V_q | (u, u') \in E\}$
6. $v.MS = \varnothing$ //$v.MS$ 是 V_{sq} 中与 v 匹配的顶点集合
7. IF v 完成匹配，$v.MS = \{m \in V, (m, v) \in S\}$
8. FOR ALL $m \in v.MS$
9. FOR ALL $n \in V$, $n \in m.neighbor$
10. IF ($|R(v) - R(u)| == |R(m) - R(n)|$ && $L(n) == L(u)$)
11. $u.MS = u.MS \cup n$
12. END IF
13. END FOR
14. END FOR
15. END IF
16. IF $u.neighbor.MS = \varnothing$
17. FOR ALL $v \in V$
18. IF $L(v) == L(u)$
19. $u.MS = u.MS \cup v$
20. END IF
21. END FOR
22. END IF
23. END FOR
24. $M_q = M_q \cup u.MS$
25. END FOR
26. RETURN M_q

算法 4 展示了匹配算法的具体过程. 算法步 1 计算查询顶点的 *rank* 值，首先调用 Tarjan 算法求出查询图 G_q 的全部强连通分量，之后根据 *rank* 值的定义求出 G_q 中顶点的 *rank* 值. 从步 4 开始，是匹配算法的执行过程，该过程会对 G_q 中的所有顶点进行扫描，来计算每个顶点的匹配顶点集合. 在计算匹配顶点集合时，对于每一个 $u \in V_q$ 来说，又分为以下两种情况：

(1) 顶点 u 的任意邻接点已经完成匹配：

步 7～步 15 显示了这一过程. 如果当前顶点 u 的任意邻居顶点 v 已经完成匹配顶点集合 $v.MS$ 的计算，则查询顶点 u 的候选集会根据 $v.MS$ 进行剪枝. 其过程为：遍历 $v.MS$，对其中的任意顶点 n，计算其与邻居顶点 m 的 *rank* 值的差值，如果 $|rank(m) - ramk(n)| = |rank(v) - rank(u)|$，并且顶点 n 与顶点 u 的标签一致，则顶点 n 与顶点 u 匹配成功.

(2) 顶点 u 的所有邻接点都未完成匹配：

步 16～步 22 显示了这一过程. 如果当前顶点 u 的所有邻居顶点都未完成匹配顶点集合的计算，则在 G_{sq} 中，所有与 u 标签一致的顶点都被加入到顶点 u 的匹配候选集合中.

根据以上匹配过程可以看出，若当前查询顶点 u 的邻居顶点已经匹配完成，则其邻居顶点的匹配集能够帮助顶点 u 快速定位其匹配候选集，从而显著提高匹配效率. 因此，第 1 个查询顶点候选集的大小影响到整个算法的时间效率. 对此，本文已对搜索匹配算法做出了优化. 前文已经提到，所有顶点在初次匹配时，其匹配集合即为结构概要中，与 u 标签一致的顶点. 因此，在步 2，对构成结构概要顶点的所有等价类，按照其内部顶点数目由小到大进行排序. 在查询时优先选取顶点规模最小的等价类进行匹配.

如算法 4 所示，计算 G_q 顶点 *rank* 值的过程可在 $O(|V_q| + |E_q|)$ 时间复杂度内完成，步 5～步 16 为匹配算法的核心部分. 其中步 5～步 24 根据查询顶点的邻居顶点进行匹配. 对于一个查询顶点 V_q，其邻居顶点数量约为 $|E_q|/|V_q|$；同理对于图数据中的一个顶点，其邻居顶点的个数约为 $|E|/|V|$，因此，该过程的时间复杂度约为

$$O(|V_q| \cdot (|E_q|/|V_q|) \cdot |V| \cdot (|E|/|V|)) = O(|E_q| \cdot |E|).$$

若查询顶点的邻居顶点尚未匹配，则执行步 16～步 19，该过程的时间复杂度为

$$O((|E_q|/|V_q| + |V|/|L|) \cdot |V_q|) = O(|E_q| + |V| \cdot |V_q|/|L|).$$

算法 4 实现子图匹配查询的整体时间复杂度为

$$O(|V_q| + |E_q|) + O(|E_q| \cdot |E|) + O(|E_q| + |V| \cdot |V_q|/|L|) \approx O(|V_q| + |E_q| \cdot |E| + |E_q| + |V|).$$

当查询图包含的顶点和边的数量接近于常数时，该算法的时间复杂度约为 $O(|E| + |V|)$.

5　实验结果与分析

本节设计相关实验，验证论文提出的子图匹配查询相关算法的有效性. 实验分别使用真实数据集和模拟数据集对自适应结构概要和子图匹配查询算法的性能进行评价. 为了便于与已有工作进行比较，论文采用的真实数据集包括 California[25]、Internet[25] 和 Citation[38]，另外，使用图生成工具，构建了模拟数据集 Synthetic. 论文的所有算法用 VC++.Net 2013 编写，在配置为 Intel i5，3.1GHz 处理器，4GB 内存的 PC 机上运行.

5.1 结构概要模型的压缩率

论文提出的自适应结构概要模型 ASSG(Adaptive Structural Summary of Graph data,ASSG),是基于图压缩技术构建的,与文献[25]提出的图压缩模式 Gr 不同之处在于 Gr 面向全部可能的子图匹配查询构建,而 ASSG 只针对当前的查询图构建,其初始状态为最大图压缩模式.本部分实验将压缩率作为评价指标,比较 ASSG 与 Gr 对于数据图的压缩率.测试结构概要模型对数据图的压缩率的意义在于:一方面,验证结构概要模型较之已有方法 Gr 具有更低的压缩率;另一方面,分析影响结构概要模型压缩率的主要因素以及结构概要模型与 Gr 在压缩率方面的关系.

由于数据图通常被表示为顶点和边的集合,因此,需要从顶点和边两个方面的压缩率评价图压缩的效果.将结构概要的顶点和边压缩率定义为 $C_{Vr}=|V_s|/|V|$ 和 $C_{Er}=|E_s|/|E|$,即 C_{Vr} 和 C_{Er} 分别为结构概要的顶点和边数量与数据图顶点和边的数量之间的比值,且压缩率越小,压缩效果越好.

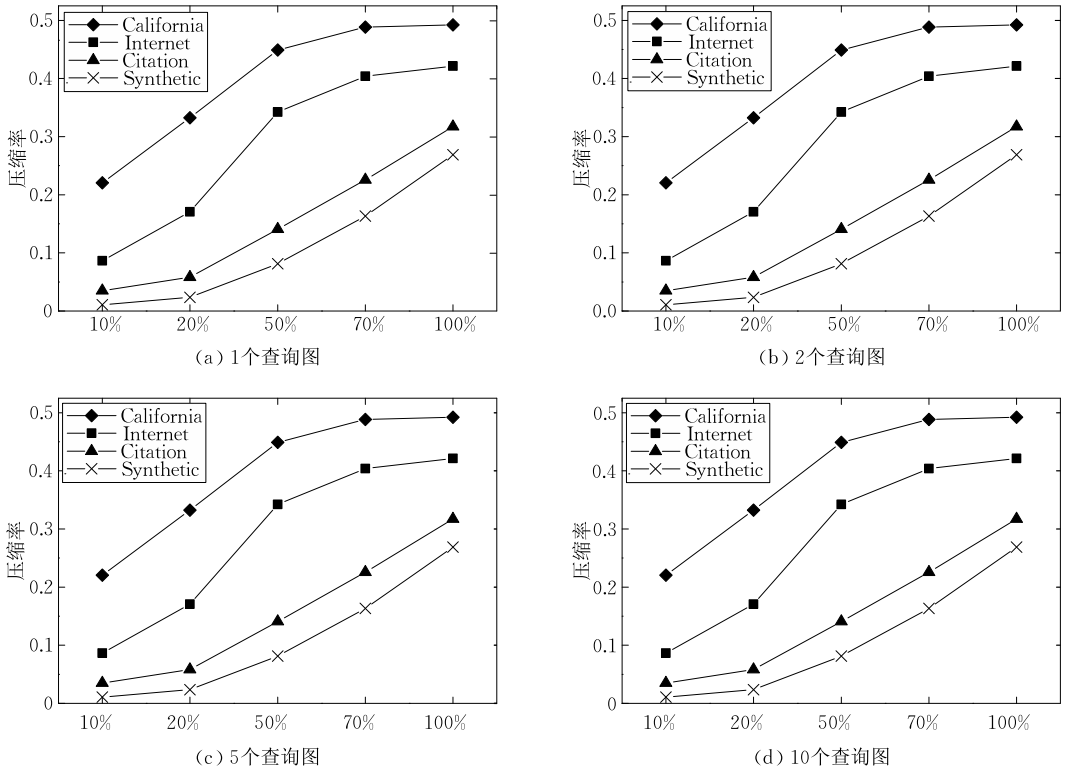
由前文的分析可知,无论是构建压缩图 Gr 还是结构概要 ASSG,都与数据图中顶点标签的数量有关,因此,在实验中,构造一个包含 20%数据图标签的查询图,用来更新初始的结构概要 G_s,即 $|L_q|=20\%\cdot|L|$.自适应结构概要 ASSG 与压缩图 Gr 对于数据图的压缩率如表 1 所示.表 1 中的压缩率是特定实验条件下,得到的实验结果,即针对表中的每一个数据集,在相同的实验环境中分别运行压缩程序,得到相应的顶点和边的压缩率.

表 1 自适应结构概要与图压缩模式的压缩率对比

数据集	$\|V\|$	$\|E\|$	$\|L\|$	ASSG/%		Gr 压缩率/%	
				C_{Vr}	C_{Er}	C_{Vr}	C_{Er}
California	1546	3296	94	33.25	44.40	49.22	66.32
Internet	10878	39994	50	17.80	58.00	42.41	78.37
Citation	21524	48529	67	5.83	15.20	31.71	51.65
Synthetic	100000	547666	60	2.36	9.89	26.89	53.44

从表 1 可以看出,ASSG 平均可将数据图的顶点规模压缩至原规模的 15%,将边压缩至原规模的 32%.在对于数据图的压缩率方面,由于 ASSG 只覆盖当前查询图,其顶点和边的压缩率优于 Gr.

如果增加查询图涉及到的标签数量,由结构概要的定义以及自适应更新算法的过程可以分析出,更新后的结构概要对数据图的压缩率会随着查询图标签数量的增加而增长,直到达到 Gr 的压缩率.图 3 分别展示了查询图数量分别为 1、2、5、10 的情况下,当查询图一共涉及到数据图标签的比例由 10%增加到 100%时,ASSG 压缩率的变化情况.图中横坐标表示所有查询图中包含的标签数量占数据图标签总数的百分比,即 $|L_q|/|L|$.为了使得实验数据具有可比性,假设 $|L_q|/|L|$ 一定时,不同数量的查



(a) 1个查询图 (b) 2个查询图

(c) 5个查询图 (d) 10个查询图

图 3 查询图包含不同比例标签时 ASSG 的压缩率

询图所包含的标签一致，即查询图数量分别为 1、2、5、10 时，这几组查询图具有相同的标签集. 通过图 3 可以得出以下结论：(1) 当$|L_q|/|L|$一定时，在查询图的数量不同但标签集一致的情况下，ASSG 的压缩率相同，因此 ASSG 的压缩率主要取决于$|L_q|$；(2)$|L_q|/|L|$越高，压缩率越大，即 ASSG 的压缩率与$|L_q|/|L|$呈线性正相关，当$|L_q|/|L|=1$时，压缩率最大，且与 Gr 的压缩率相同，这是由于此时的 ASSG 与 Gr 一样覆盖了所有可能的子图匹配查询.

5.2　结构概要模型的构建效率

本部分实验，通过图数据工具生成一组模拟数据集，验证不同规模数据图的自适应结构概要模型构建的时间效率. 数据集如表 2 所示，实验使用的数据图在顶点规模与边规模上进行递增，而标签数量保持不变.

图 4 展示了结构概要模型在该组 Synthetic 数据集上的构建时间，呈线性时间复杂度. 通过 3.2 节对算法 2 的分析可知，构建结构概要模型的时间代价与数据图的顶点数和边数线性相关，图 4 证明了该结论的正确性.

表 2　不同规模的模拟数据集

数据集	顶点数$\|V\|$	边数$\|E\|$	标签数$\|L\|$
S_1	1000	4739	60
S_2	5000	22545	60
S_3	10000	44742	60
S_4	20000	109819	60
S_5	50000	275647	60
S_6	100000	547666	60
S_7	100000	2108253	60
S_8	100000	5342747	60

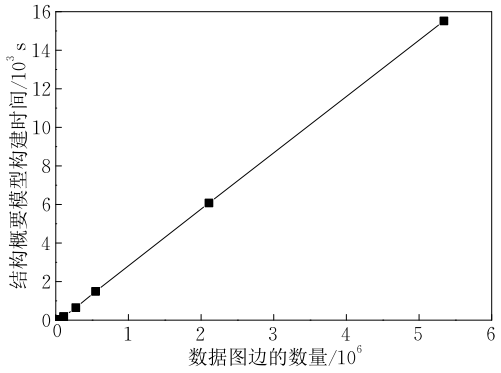


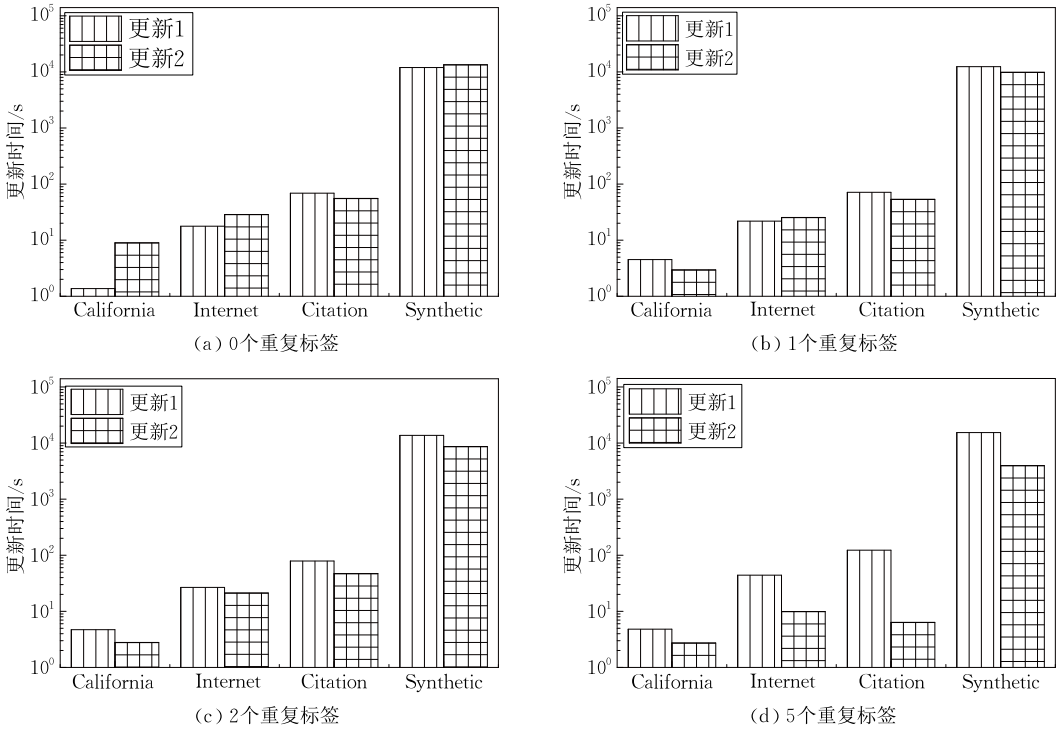
图 4　不同规模的模拟数据集构建结构概要的时间

5.3　结构概要模型的自适应更新

既然 ASSG 只对当前查询图有效，为了进行更多的子图匹配查询，ASSG 需要具备针对不同查询图的自适应更新功能. 为验证算法 3 的有效性，本部分的实验将验证结构概要模型的自适应更新性能.

5.3.1　自适应更新性能的影响因素分析

实验将设计 4 组、每组两个包含不同标签的查询图，每个查询图均包含数据图中 20%的标签. 4 组查询图中，每组的两个查询图分别包含 0 个、1 个、2 个 和 5 个重复的标签，根据这两个查询图在结构概要粗糙模型上进行两次更新. 图 5 展示了结构概



(a) 0个重复标签　(b) 1个重复标签

(c) 2个重复标签　(d) 5个重复标签

图 5　包含不同数量重复标签的查询图更新 ASSG 的时间

要针对上述4组查询图的自适应更新性能，当不同的查询图包含越多重复标签的时候，结构概要自适应更新的效率越高. 图5(b)的两个查询图只包含1个重复标签，结构概要自适应更新以满足第2个查询图的时间代价接近于根据第1个查询图对结构概要的自适应更新；图5(d)所示的两个查询图包含了5个重复标签，根据第2个查询图更新结构概要的时间代价明显低于第1次更新. 由此可以得到的结论是，当子图查询趋于稳定时，即结构概要模型已经能够应对绝大部分子图匹配查询要求时，结构概要进行更新的可能性越小.

因此，自适应结构概要针对以下两类子图匹配查询的处理效率较高：一是子图匹配查询集中于数据图的少部分顶点；二是频繁查询的子图.

即使查询图新引入的标签种类在数量上一致，而新引入的标签种类不一致时，ASSG更新的时间以及压缩率仍然可能不同. 表3给出了在California数据集上，一个查询图新引入的标签种类逐渐递增时，ASSG更新的时间与压缩率.

表3 ASSG更新的时间与压缩率

新引入标签种类	ASSG更新时间/s	ASSG更新后的压缩率/%
5	1.34	20.89
10	1.66	22.06
15	4.31	33.25
30	13.05	40.04
50	14.79	44.89

对比表3第1行与第2行，不难得出更新时引入了5种新的标签，ASSG细化这5个标签相关等价类所需时间为0.32 s(1.66 s−1.34 s)，其更新后的压缩率提高了1.17%(22.06%−20.89%)；而对比第2行与第3行，ASSG细化这5个标签所需时间为2.65 s(4.31 s−1.66 s)，其更新后的压缩率提高了11.19%(33.25%−22.06%). 同样在查询时引入5个新的标签，但ASSG的更新效率以及压缩率却有较大差别.

从理论角度分析，ASSG的这一特性不难理解. 因为带有不同标签的顶点在数目上可能有较大差异，因此在相关等价类被细化时，所需要的时间与空间复杂度也会相差较多. 本文同样在其它数据集上进行了类似实验，实验结果与表3给出的结果相一致. 综上所述，ASSG更新的时间性能与更新后的压缩率不仅与新引入标签的数量有关，还与包含新引入标签的顶点数量有关.

5.3.2 自适应更新的性能分析

本实验采用Synthetic数据集，其统计信息已经在表2中给出.

首先，选取4组(S_3～S_6)顶点规模较大、但顶点平均出度较小的数据集作为测试数据，即顶点规模为10 000～100 000，每个顶点的平均出度约为5. 构造6个查询图Q_1～Q_6，如表4所示，每个查询图均包含60个顶点和229条边，标签数量分别为5，10，15，30，45和60. 对于每一个查询，皆在表2所示的Synthetic数据集S_3～S_6上进行测试，用以验证ASSG在不同数据规模下的更新性能.

表4 实验使用的查询图

查询图	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
标签比例/%	10	15	25	50	75	100

图6展示了实验的整体测试结果. 其横坐标已经进行过处理，即为$|E|\log|V|$，纵坐标为引入查询图时ASSG的更新时间. 其中，Q_1～Q_6为表4所示的查询图. 从图6中可以明显看出，以ASSG的更新时间为纵坐标，以数据图的$|E|\log|V|$为横坐标的顶点连线为一条直线，这表明ASSG的更新时间与数据图的$|E|\log|V|$正相关，从而验证了ASSG的更新时间复杂度为$O(|E|\log|V|)$.

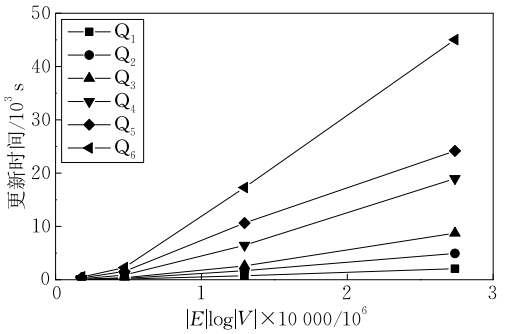


图6 针对低密度图，包含不同标签数的查询图更新ASSG的时间

然后，在高密度图上进行测试，从表2中选取3组(S_6～S_8)数据集，每个数据集顶点规模均为100 000个，顶点的平均出度分别为5，20，50. 构造与上一组测试相同的6个查询图，验证ASSG在不同密度数据图中的更新性能. 测试结果如图7所示，虽然高密度图每个顶点平均出度较大，结构更加复杂，但是结构概要的自适应更新时间仍然与$|E|\log|V|$线性正相关.

5.4 子图匹配查询算法

本文基于双拟关系，按照图模拟的思想对子图匹配查询问题进行了定义. 本部分实验将验证论文提出的基于自适应结构概要的子图匹配查询算法的总体性能. 子图匹配查询由两个过程构成：一是结构概要根据查询图进行自适应更新的过程；二是基于

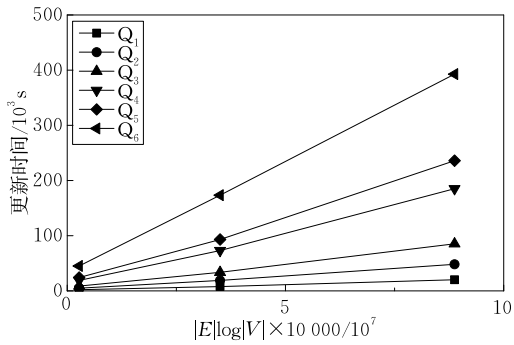


图 7　针对高密度图，包含不同标签数的查询图更新 ASSG 的时间

更新后的结构概要进行子图匹配查询的过程. 因此，将设计 3 个实验：一是在不同规模数据集上，处理少量包含相同比例标签的查询图时，验证结构概要的自适应更新性能以及相应的子图匹配查询性能；二是在不同规模数据集上，处理大量随机生成的查询图时，验证结构概要的自适应更新性能以及相应的子图匹配查询性能；三是将论文提出的算法(ASSG 算法)与已有的关于子图模拟匹配的强模拟算法(Strong Simulation，SS)[18]进行比较.

5.4.1　针对少量查询图的总体性能验证

在表 2 所示的不同规模 Synthetic 数据集 S_1～S_6 上，对包含不同比例标签的查询图 Q_1～Q_6(如表 4 所示)，验证在数据图中进行子图匹配查询时，结构概要模型的更新效率和子图匹配查询的效率.

图 8 展示了在不同规模数据集上进行子图匹配查询时，结构概要自适应更新的运行时间. 由图 8 可知，在不同规模数据集上，对相同的查询图执行查询时，相应结构概要的自适应更新代价与 $|E|\log|V|$ 线性正相关；而在相同规模数据集上，执行包含不同比例标签的查询图时，结构概要的更新时间也有区别，这与查询图包含的标签数量有关(见 5.3.1 节的相关分析). 图 9 展示了在更新后的结构概要上进行子图匹配查询的时间效率. 在不同规模数据集上，对

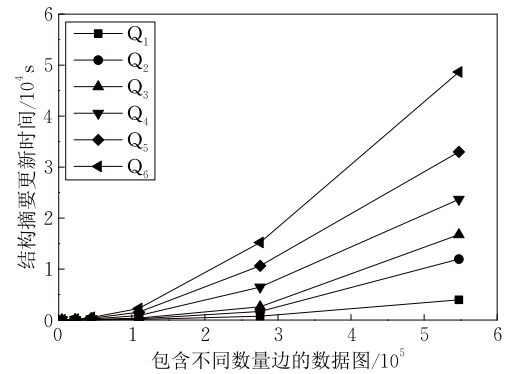


图 8　包含不同数量标签的查询图更新 ASSG 的时间

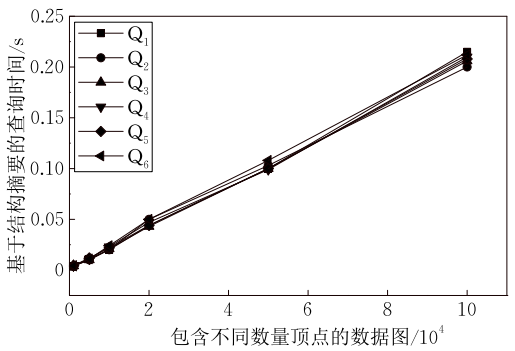


图 9　针对不同规模数据图的查询时间

相同的查询图执行子图匹配查询的代价接近于线性时间复杂度；而在相同规模数据集上，对包含不同比例标签的查询图进行子图匹配查询的时间效率几乎相同，这表明，子图匹配查询算法(算法 4)只依赖于数据图的规模($O(|E|+|V|)$)，而与查询图包含的标签数量无关，从而验证了 4.3 节对子图匹配算法时间复杂度的分析.

5.4.2　针对大量查询图的总体性能验证

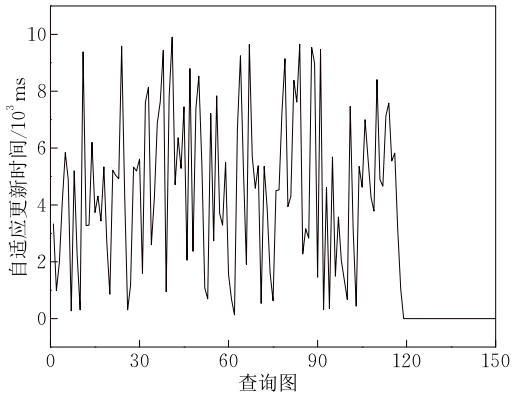
本部分实验，在表 1 所示的 4 组数据集上，验证论文提出的方法在处理大量查询图时，在结构概要自适应更新和子图匹配查询方面的性能.

(1)在不限制查询图顶点标签范围的情况下，根据数据图的顶点标签随机生成 150 个包含 10 个顶点和 30 条边的查询图.

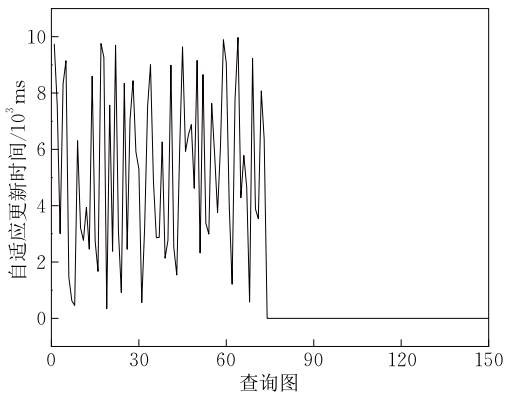
由于顶点标签的分配具有随机性，因此，从图 10 中可以看到，顶点更新的时间代价也具有随机性，但是，当生成一定数量的查询图时，这些查询图已经覆盖了数据图的全部顶点标签，再处理后续的查询图时，无需进行结构概要的更新(更新时间为 0)，而此时的结构概要模型的压缩率达到最大值，接近于 Gr(参见 5.1 节对于图 3 的分析). 在更新后的结构概要中，对这 150 个查询图进行子图匹配查询处理的时间代价(如图 11 所示)具有随机性，但都比较接近，这是因为子图匹配查询算法只与数据图的顶点和边数有关.

(2)选取数据图顶点标签集合中 20%的标签，随机构建 100 个包含 10 个顶点和 30 条边的查询图，以模拟“子图匹配查询集中于 20%的标签”或者“有 20%的标签被频繁地查询”的情况.

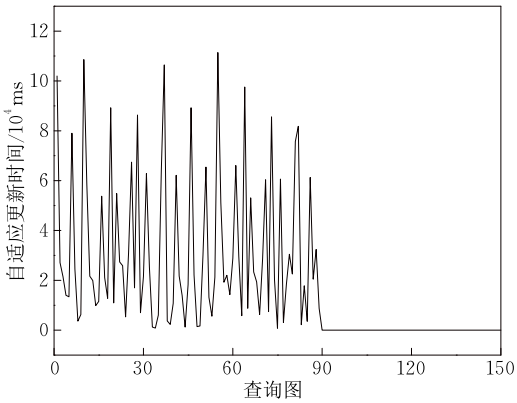
从图 12 中可以看出，当处理到 8 个左右的查询图时，结构概要已经覆盖了指定的 20%数据图顶点标签，再处理后续的查询图时，无需进行结构概要更新，而此时的结构概要模型的压缩率，小于 Gr(参见 5.1 节对于图 3 的分析). 图 13 展示了对于每个查



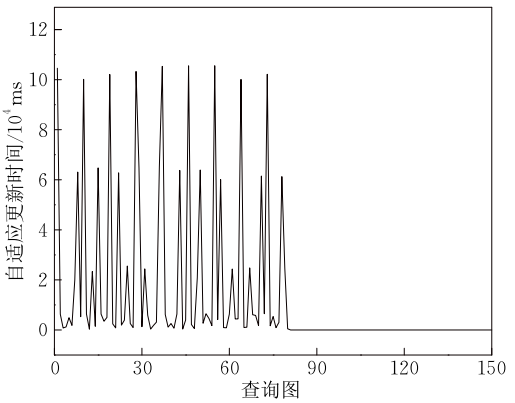
(a) California



(b) Internet

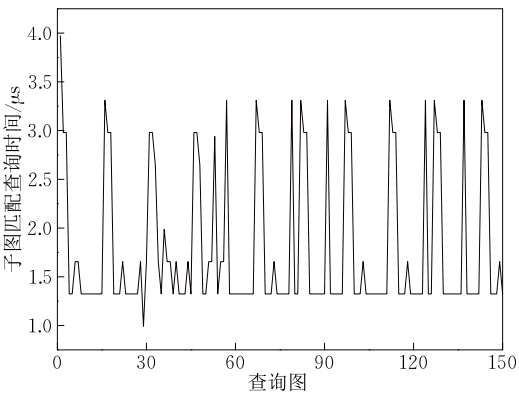


(c) Citation

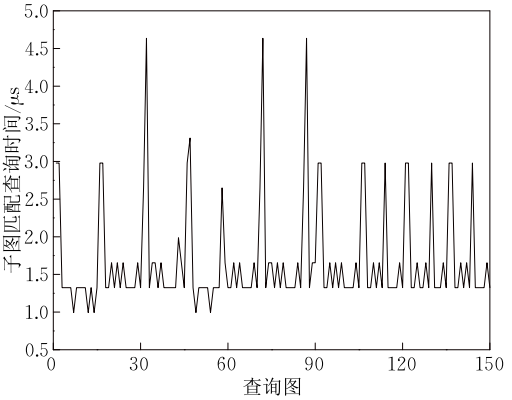


(d) Synthetic

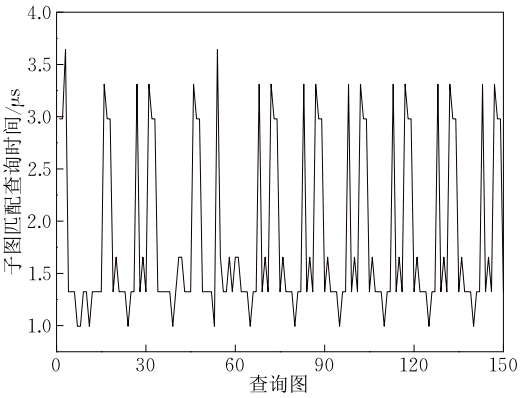
图 10 不限定标签范围,处理 150 个随机查询图时自适应结构概要的自适应更新代价



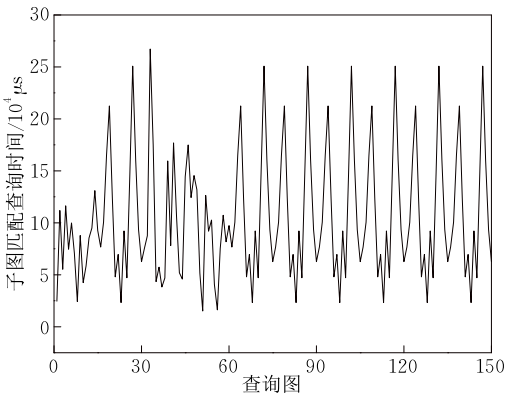
(a) California



(b) Internet



(c) Citation



(d) Synthetic

图 11 不限定标签范围,处理 150 个随机查询图时的子图匹配查询时间代价

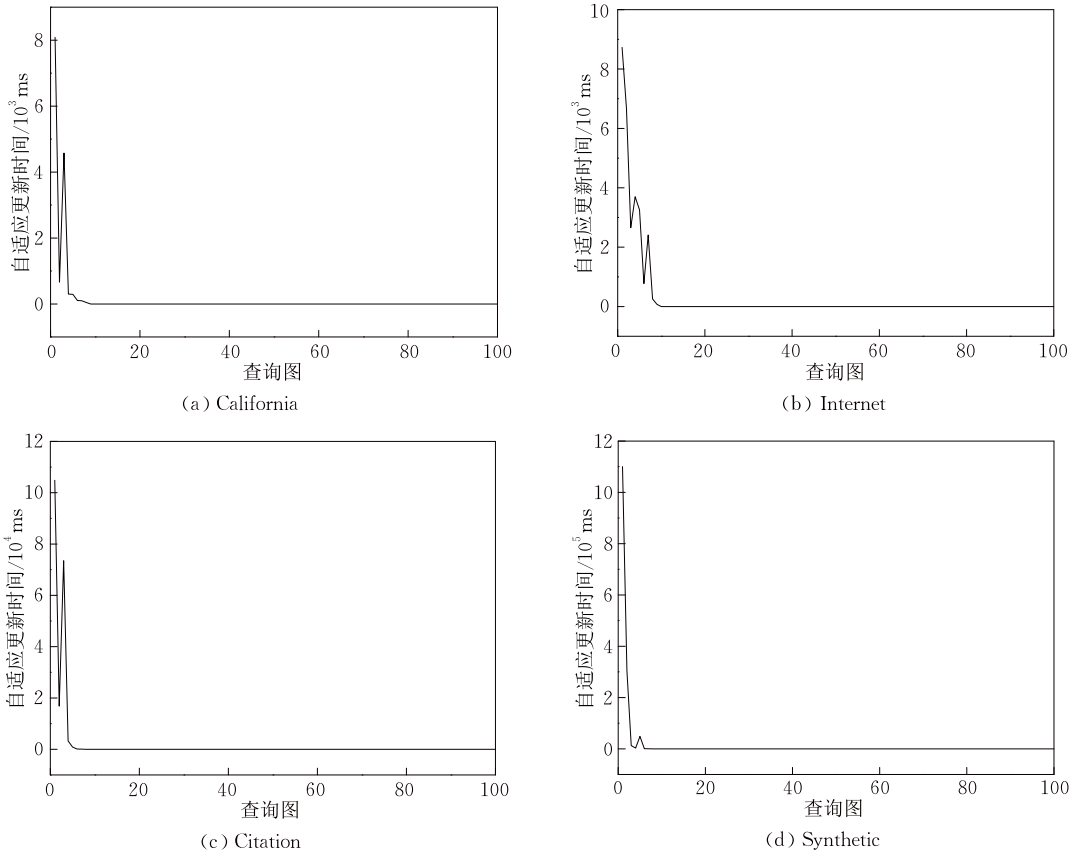


图 12　限定标签范围，处理 100 个随机查询图时自适应结构概要的自适应更新代价

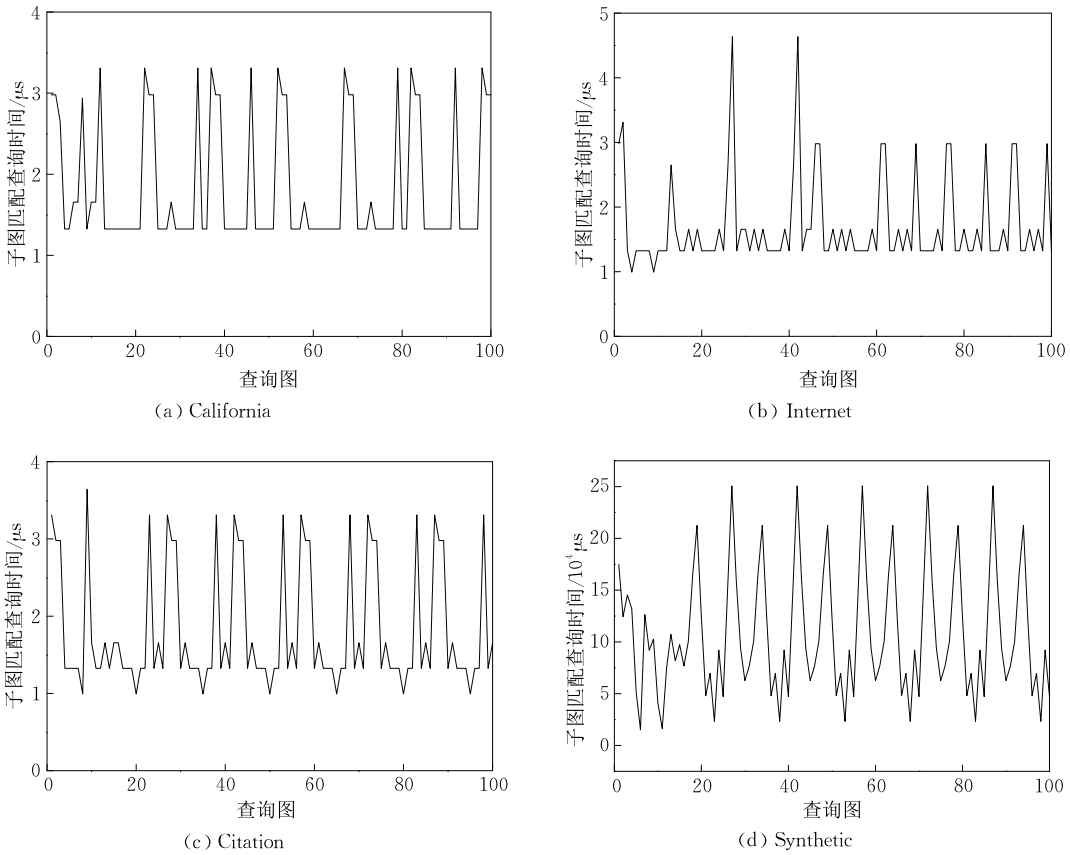


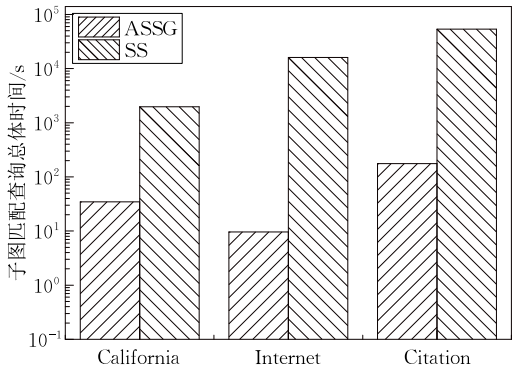
图 13　限定标签范围，处理 100 个随机查询图时的子图匹配查询时间代价

询图进行子图匹配的时间代价，仍然具有随机性，且比较接近.

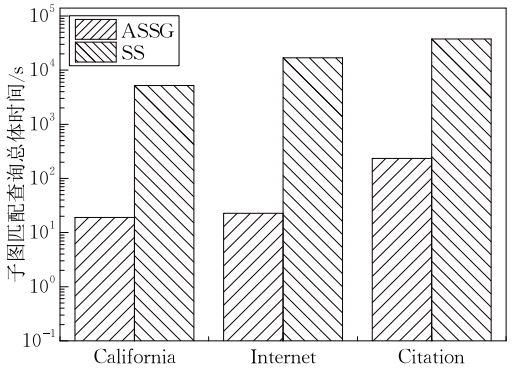
通过实验及其分析可知，论文提出的方法，在处理频繁查询的标签时，具有较低的压缩率和较高的子图匹配查询处理效率.

5.4.3 ASSG 算法与强模拟算法的对比实验

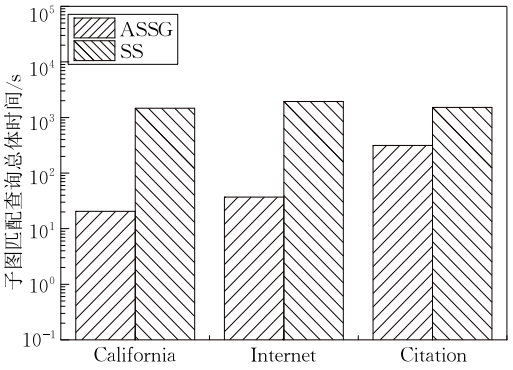
图 14 和图 15 是论文提出的基于自适应结构概要进行子图匹配查询的算法 ASSG 与强模拟算法 SS 的运行效率对比. 由于强模拟匹配算法是一个包含数据预处理的完整过程，因此，实验过程中使用包含结构概要构建、自适应更新以及查询 3 个过程的总体时间进行对比. 仍然使用表 4 所示的查询图进行实验. 图 14 是在真实数据集上，当查询图的标签数量分别为 5、10 和 15 的情况下，ASSG 算法和 SS 算法运行时间的比较. 从图 14 中可知，ASSG 算法的总体时间效率远远高于 SS 算法，这是由于 ASSG 算法运行的基础是结构概要模型，只需根据查询图进行相应顶点等价类的细化更新而后即可进行查询；而 SS 算法每次运行都需要从数据图中直接进行模拟匹配.



(a) 查询图包含5个标签



(b) 查询图包含10个标签



(c) 查询图包含15个标签

图 14 真实数据集上包含不同数量标签的查询图的总体查询时间

图 15 展示了在模拟数据集(表 2 中的模拟数据集 $S_1 \sim S_5$)上运行 ASSG 算法和 SS 算法的时间. 对于一般的子图匹配查询，查询图涉及到的标签数量不多，因此本实验构建的查询图包含的标签数固定为 5. 由图 15 可知，ASSG 算法的总体运行效率远高于 SS 算法. 一方面，ASSG 算法将数据图进行了压缩，构建结构概要模型，使得查询对象的规模减小；另一方面，就子图匹配查询的过程而言，ASSG 算法的总体时间代价约为 $O(|E|\log|V|)$；而 SS 算法的时间代价约为 $O(|V|^2(|V|+|E|))$，为多项式级时间复杂度，当数据集规模达到一定程度时，算法运行

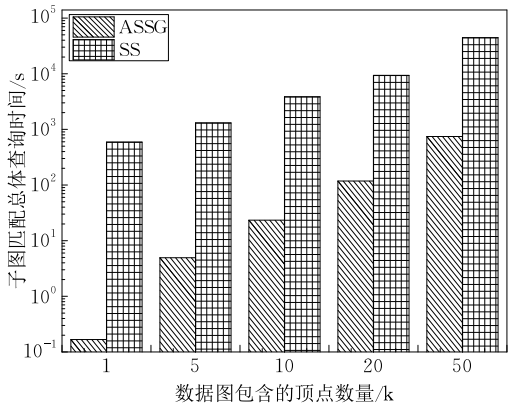


图 15 模拟数据集上对不同规模数据图的总体查询时间

非常困难，论文使用的实验环境下无法得出子图匹配查询结果.

6 结　论

结构概要是一类压缩图结构，在子图匹配查询中具有重要的作用.论文首先基于图压缩技术，提出了以顶点等价类为核心的有向标签图结构概要模型，该模型能够根据当前查询图的结构自适应更新，通过顶点细化分裂的方式，使更新后的结构概要覆盖当前查询图在数据图中的全部匹配子图；然后，提出基于自适应结构概要的子图匹配查询算法，可高效地实现基于图模拟方式的子图匹配查询；最后，通过设计实验，验证论文提出方法的运行效率和有效性，结果表明，自适应结构概要较之经典的图压缩算法得到更低的压缩率，可在 $O(|E|\log|V|)$ 时间复杂度内实现结构概要的自适应更新以及子图匹配查询.

未来仍有很多挑战性的工作：(1) 论文选用的实验数据，无论是图数据自身，还是构建的结构概要模型都能够在内存中存储，但是存在着更大规模的数据，需要进行更为复杂的处理；(2) 论文的工作针对有向标签图，相关思想如何应用于无向图甚至更加复杂的图数据，需要进一步的研究.

参　考　文　献

[1] Yu Ge, Gu Yu, Bao Yu-Bin, Wang Zhi-Gang. Large scale graph data processing on cloud computing environments. Chinese Journal of Computers, 2011, 34(10): 1753-1767(in Chinese)
(于戈，谷峪，鲍玉斌，王志刚，云计算环境下的大规模图数据处理技术. 计算机学报，2011，34(10)：1753-1767)

[2] Yu Jing, Liu Yan-Bing, Zhang Yu, et al. Survey on large-scale graph pattern matching. Journal of Computer Research and Development, 2015, 52(2): 391-409(in Chinese)
(于静，刘燕兵，张宇等. 大规模图数据匹配技术综述. 计算机研究与发展，2015，52(2)：391-409)

[3] Ullmann J R. An algorithm for subgraph isomorphism. Journal of the ACM, 1976, 23(1): 31-42

[4] Cordella L P, Foggia P, Sansone C, et al. A (sub) graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(10): 1367-1372

[5] Gupta M, Gao J, Yan X, et al. Top-k Interesting subgraph discovery in information networks//Proceedings of the 30th IEEE International Conference on Data Engineering. Chicago, USA, 2014: 820-831

[6] Cheng J, Zeng X, Yu J X. Top-k graph pattern matching over large graphs//Proceedings of the 29th IEEE International Conference on Data Engineering. Brisbane, Australia, 2013: 1033-1044

[7] Khan A, Wu Y, Aggarwal C C, Yan X. NeMa: Fast graph search with label similarity. Proceedings of the VLDB Endowment, 2013, 6(3): 181-192

[8] Wu Y, Yang S, Yan X. Ontoloty-based subgraph querying//Proceedings of the 29th IEEE International Conference on Data Engineering. Brisbane, Australia, 2013: 697-708

[9] Khan A, Li N, Yan X, et al. Neighborhood based fast graph search in large networks//Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. Athens, Greece, 2011: 901-912

[10] Zhao P, Han J. On graph query optimization in large networks. Proceedings of the VLDB Endowment, 2010, 3(1-2): 340-351

[11] Zeng X, Cheng J, Yu J X, Feng S. Top-k graph pattern matching: A twig query approach//Proceedings of the 13th International Conference on Web-Age Information Management. Harbin, China, 2012: 284-295

[12] Zou Lei, Chen Lei, Ozsu M T, Zhao Dong-Yan. Answering pattern match queries in large graph databases via graph embedding. Proceedings of the VLDB Endowment, 2012, 21(1): 97-120

[13] Yang S, Wu Y, Sun H, Yan X. Schemaless and structureless graph querying. Proceedings of the VLDB Endowment, 2014, 7(7): 565-576

[14] Fan W, Wang X, Wu Y. Answering graph pattern queries using views//Proceedings of the 30th IEEE International Conference on Data Engineering. Chicago, USA, 2014: 184-195

[15] Fan W, Wang X, Wu Y. Querying big graphs within bounded resources//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. Snowbird, USA, 2014: 301-312

[16] Fan W, Wang X, Wu Y. Diversified Top-k graph pattern matching. Proceedings of the VLDB Endowment, 2013, 6(13): 1510-1521

[17] Fan W, Li J, Ma S, et al. Adding regular expressions to graph reachability and pattern queries//Proceedings of the 27th IEEE International Conference on Data Engineering. Hannover, Germany, 2011: 39-50

[18] Ma Shuai, Cao Yang, Fan Wen-Fei, et al. Capturing topology in graph pattern matching. Proceedings of the VLDB Endowment, 2011, 5(4): 310-321

[19] Fan W, Li J, Ma S, et al. Graph pattern matching: From intractable to polynomial time. Proceedings of the VLDB Endowment, 2010, 3(1-2): 264-275

[20] Fan W, Li J, Ma S, Wu Y. Graph homomorphism revisited for graph matching. Proceedings of the VLDB Endowment, 2010, 3(1-2): 1161-1172

[21] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, USA: Freeman W H & Co., 1990

[22] Sun Z, Wang H, Wang H, et al. Efficient subgraph matching on billion node graphs. Proceedings of the VLDB Endowment, 2012, 5(9): 788-799

[23] Han W, Lee J, Pham M, Yu J X. iGraph: A framework for comparisons of disk based graph indexing techniques. Proceedings of the VLDB Endowment, 2010, 3(1-2): 449-459

[24] Zhang Yu, Liu Yan-Bing, Xiong Gang, et al. Survey on succinctrep resentation of graph data. Journal of Software, 2014, 25(9): 1937-1952(in Chinese)
(张宇, 刘燕兵, 熊刚等. 图数据表示与压缩技术综述. 软件学报, 2014, 25(9): 1937-1952)

[25] Fan W, Li J, Wang X, Wu Y. Query preserving graph compression//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, USA, 2012: 157-168

[26] Henzinger M R, Henzinger T A, Kopke P W. Computing simulations on finite and infinite graphs//Proceedings of the 36th Annual Symposium on Foundations of Computer Science. Milwaukee, USA, 1995: 453-462

[27] Claude F, Navarro G. A fast and compact web graph representations. ACM Transactions on the Web, 2010, 4(4): 1-31

[28] Boldi P, Rosa M, Santini M, Vigna S. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks//Proceedings of the 20th International Conference on World Wide Web. Hyderabad, India, 2011: 587-596

[29] Hernández C, Navarro G. Compression of Web and social graphs supporting neighbor and community queries//Proceedings of the 5th ACM Workshop on Social Network Mining and Analysis (SNAKDD). San Diego, USA, 2011: 1-10

[30] Maserrat H, Pei J. Neighbor query friendly compression of social networks//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, USA, 2010: 533-542

[31] Ahnert S E. Power graph compression reveals dominant relationships in genetic transcription networks. Molecular BioSystems, 2013, 9(11): 2681-2685

[32] Chung C, Min J, Shim K. Apex: An adaptive path index for XML data//Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. Madison, USA, 2002: 121-132

[33] Chen Q, Lim A, Ong K W. $D(k)$-index: An adaptive structural summary for graph-structured data//Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego, USA, 2003: 134-144

[34] Wan Chang-Xuan, Liu Xi-Ping. XML Database Technology. 2nd Edition. Beijing: Tsinghua University Press, 2008(in Chinese)
(万常选, 刘喜平. XML 数据库技术. 第 2 版. 北京: 清华大学出版社, 2008)

[35] Tarjan R E. Depth-first search and linear graph algorithms. SIAM Journal on Computing, 1972, 1(2): 146-160

[36] Dovier A, Piazza C, Policriti A. A fast bisimulation algorithm//Proceedings of the 13th Conference on Computer Aided Verification. Paris, France, 2001: 79-90

[37] Paige R, Tarjan R E. Three partition refinement algorithms. SIAM Journal on Computing, 1987, 16(6): 973-989

[38] Tang J, Zhang J, Yao L, et al. Arnetminer: Extraction and mining of academic social networks//Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Las Vegas, USA, 2008: 990-998

ZHANG Hai-Wei, born in 1980, Ph. D., lecturer. His research interests include semi-structured data management, graph database and data mining, etc.

XIE Xiao-Fang, born in 1991, M. S. candidate. Her research interest is graph database.

DUAN Yuan-Yuan, born in 1989, M. S. Her research interest is graph data management.

Wen Yan-Long, born in 1979, Ph. D., senior engineer. His research interests include information retrieval, etc.

ZHANG Ying, born in 1986, Ph. D., associate professor. Her research interests include data mining, information retrieval, etc.

YUAN Xiao-Jie, born in 1963, Ph. D., professor, Ph. D. supervisor. Her research interests include database, data mining, information retrieval and big data, etc.

Background

With the increasing popularity of large graphs which usually contain millions of nodes and billions of edges, e. g. social network, bioinformatics, and semantic web, subgraph matching has attracted much attentions in various domains and becomes one of the most fundamental operations in many applications.

Existing three solutions, graph isomorphism, graph approximation and graph simulation, are expensive where

graph isomorphism is NP-complete, graph approximation needs complex preprocessing and graph simulation takes quadratic time. All of above solutions will encounter bottlenecks while dealing with the large scale graph data.

Graph isomorphism and approximation matching maintained the similar structures between query graphs and matched subgraphs while graph simulation did not. Recent works such as bounded simulation and strong simulation constrained the structural similarities but increased the cost to cubic time.

Graph simulation transformed the task of finding isomorphism subgraphs into that of finding binary relations of nodes between query graphs and data graphs. And it can decrease the complexity of time from exponential to polynomial. Basing on the idea of simulation, we explore an efficient algorithm for subgraph matching in this research. We propose adaptive structural summary of graph data based on graph compression. And design algorithms of updating structural summary and matching subgraphs with the complexities of $|E|\log|V|$. Our algorithms have a lower time complexity compared with exiting works.

This work is mainly supported by the National High Technology Development 863 Programs of China under Grant Nos. 2013AA013204 and 2015AA015401. The work is also supported by the National Natural Science Foundation of China (Nos. 61170184 and 61402243) and the Tianjin Municipal Science and Technology Commission (Nos. 13ZCZDGX02200, 13ZCZDGX01098, 13JCQNJC00100 and 16JCTPJC53700).

In recent years, the authors have devoted to the researches of graph data mining and management. We have gained achievements of frequent subgraph mining and reachability queries on graphs. In this work, we focus on subgraph matching, and explore the algorithms to solve it in an approximate linear time. The time complexity is an important measure of subgraph matching and RDF data queries for above projects.