软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(3):663−688 [doi: 10.13328/j.cnki.jos.005444]

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

动态图模式匹配技术综述*

许 嘉[1,2,3], 张千桢[1], 赵 翔[4], 吕 品[1,2,3], 李陶深[1,2]

[1](广西大学 计算机与电子信息学院,广西 南宁 530004)

[2](广西高校并行分布式计算技术重点实验室(广西大学),广西 南宁 530004)

[3](广西高校多媒体通信与信息处理重点实验室(广西大学),广西 南宁 530004)

[4](国防科技大学 系统工程学院,湖南 长沙 410073)

通讯作者: 赵翔, E-mail: xiangzhao@nudt.edu.cn

摘 要: 随着大数据时代的到来,多源异构数据的快速增长已经成为开放性问题,数据之间的内在关联通常可以用图数据的形式来表现.然而在实际应用中,例如网络安全分析和社交网络舆情分析,描述实体对象之间关系的图数据的结构和内容往往不是固定不变的,图数据的结构以及节点和边的属性会随着时间的推移发生更新变化.因此,如何在动态更新的图数据中进行高效的查询、匹配,是目前研究的热点问题.从关键技术、代表性算法和性能评价方面概述动态图模式匹配技术的研究进展.最后,对动态图模式匹配技术的典型应用、面临的挑战问题和未来发展趋势进行了总结和展望.

关键词: 动态图;图模式匹配;子图同构;匹配算法;图搜索

中图法分类号: TP311

中文引用格式: 许嘉,张千桢,赵翔,吕品,李陶深.动态图模式匹配技术综述.软件学报,2018,29(3):663−688. http://www.jos.org.cn/1000-9825/5444.htm

英文引用格式: Xu J, Zhang QZ, Zhao X, Lü P, Li TS. Survey on dynamic graph pattern matching technologies. Ruan Jian Xue Bao/Journal of Software, 2018,29(3):663−688 (in Chinese). http://www.jos.org.cn/1000-9825/5444.htm

Survey on Dynamic Graph Pattern Matching Technologies

XU Jia[1,2,3], ZHANG Qian-Zhen[1], ZHAO Xiang[4], LÜ Pin[1,2,3], LI Tao-Shen[1,2]

[1](School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

[2](Guangxi Colleges and University Key Laboratory of Parallel and Distributed Computing Technology (Guangxi University), Nanning 530004, China)

[3](Guangxi Colleges and University Key Laboratory of Multimedia Communications and Information Processing (Guangxi University), Nanning 530004, China)

[4](College of System and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: With the advent of big data era, the rapid growth of multi-source heterogeneous data has become an open problem. The inherent relationships between these data are usually modeled by the graph model. However, in practical applications, such as network security analysis and public opinion analysis over social networks, the structure and content of the graph data describing relationships

* 基金项目: 国家自然科学基金(61402494, 61402498, 61402513); 广西自然科学基金青年基金(2015GXNSFBA139243, 2016GXNSFBA380182); 广西大学科研基金(XGZ141182, XGZ150322); 广西高等教育本科教学改革工程重点项目(2017JGZ103)

Foundation item: National Natural Science Foundation of China (61402494, 61402498, 61402513); Guangxi Natural Science Foundation (2015GXNSFBA139243, 2016GXNSFBA380182); Scientific Research Foundation of Guangxi University (XGZ141182, XGZ150322); Key Projects of Higher Education Undergraduate Teaching Reform Project in Guangxi (2017JGZ103)

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: 2017-07-31; 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 15:37:11, http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1536.018.html

between entity objects are usually not fixed. To be specific, the structure of the graph data, and the attributes of the nodes and edges in it will vary over time. Therefore, efficient query and match over dynamically updated graph data currently draws extensive research, where many outstanding research works are proposed. In this paper, the research progress of dynamic graph data matching technologies is reviewed from the aspects of key technologies, representative algorithms and performance evaluation. The state-of-the-art applications, the challenging problems and the research trend of dynamic graph matching technologies are summarized.

Key words: dynamic graph; graph pattern matching; subgraph isomorphism; matching algorithm; graph search

在大数据时代,随着信息科技与互联网的快速发展,数据规模不断增长,数据类型不断增多,不同领域关注的实体对象之间的关系变得更加复杂.如何分析和挖掘大数据中蕴含的复杂关系,成为当前的研究热点.图(graph)作为一种广泛使用的数据结构,非常适合刻画这种存在内在关联性的数据,图中的每个节点代表现实世界中的实体对象,节点之间的边表示实体之间的关系.例如:在社交网络(新浪微博、微信等)中,图可以用来刻画用户之间的交互关系;在网络安全领域中,图可以刻画主机间进行通信、用户的登录和 IP 地址的切换等交互关系;在计算生物学领域中,图则可以刻画蛋白质和酶之间发生的复杂交互、调控与代谢关系等.

然而在现实世界中,描述实体对象图数据的结构和内容往往会随着时间的推移而发生变化.以社交网络为例,根据 Facebook 网站 2010 年的年鉴报告,仅 2010 一年期间,用户从 3.37 亿人增加到 5.85 亿人,平均每分钟都会有 47 553 对好友之间建立或者解除关系.2011 年,Google+在上线后的两周时间增长了 1 000 万的用户[1].这些数据表明,现实世界中的实体对象和它们之间的关系无时无刻都可能经历着变化.因此,如何应对大规模图数据中的动态变化,吸引了研究学者的关注.

动态图模式匹配技术(dynamic graph pattern matching techniques)是分析动态图数据上高效查询的重要手段,广泛应用于众多重要领域.动态图模式匹配技术是指在一个实时更新的图中找到与给定模式图相匹配的子图,这里的匹配是指结构相同以及满足特定的语义关系等.例如:在社交网络中,实体对象以及它们之间的关系可以转化为图的形式,公司的 HR 可以通过动态图模式匹配技术在实时更新的社会群体中找到目标客户群体[2];在网络安全领域中,网络管理员可以通过动态图模式匹配技术对一个实时更新的动态网络进行监测,看是否存在异常的网络攻击行为[3];在计算生物学领域中,蛋白质与酶之间的交互作用会使蛋白质的结构不断地发生变化,通过动态图模式匹配技术对已知性质的蛋白质结构进行匹配,来分析变化后的蛋白质的性质[4].可见,动态图匹配技术与人们的生活息息相关,发挥着巨大作用.

目前,研究学者已经对静态图匹配技术进行了大量的研究,例如:Lee 等人[5]采用相同的编程环境以及通用框架,对 5 种经典的静态图匹配算法(VF2,QuickSI,GraphQL, GADDI,SPath)进行了综合对比,并分析解释了在特定图数据上算法性能提升的原因.于静等人[4]从应用出发对图匹配进行分类,包括结构匹配和语义匹配、精确匹配和近似匹配、静态图匹配和动态图匹配以及最优算法和近似算法.文章对动态图模式匹配的描述仅限于简单的介绍,详细描述了静态图的精确图匹配技术,并对匹配算法的性能进行评价.但是传统的静态图匹配模型无法描述图数据随时间发生变化的情况,因而并不适用于解决动态图数据上的模式匹配问题.如何对动态图数据进行有效且高效的匹配分析面临诸多挑战.

1) 数据更新频繁.例如网络社交网络、数据中心网络和金融网络等网络系统对应的图数据每时每刻都经历着更新,传统的静态图匹配技术需要在每一次更新时都对更新后的图数据建立索引并进行匹配,耗时耗力,不适用于解决动态图匹配问题;
2) 数据规模大.由于不断被更新,动态图的规模相对于静态图更加庞大,增加了图匹配问题的难度;
3) 实时分析要求高.在许多实时分析应用中,每当动态图被更新时,需要及时给出更新后的匹配结果,否则会使匹配结果的应用价值降低或丧失.例如,网络安全领域需要对可疑的数据传输模式进行实时匹配监控,若匹配分析结果滞后,可能会导致网络瘫痪.

针对上述问题,现有动态图数据匹配相关研究工作主要从以下几个方面展开.

1) 针对动态图数据频繁更新的特性,研究增量处理技术,仅对动态图数据更新的部分进行分析和匹配;
2) 针对动态图数据规模大的特性,研究如何利用分布式并行图处理框架,例如谷歌的 Pregel[6]、微软的

Trinity[7]、Apache 的 Giraph[8]、Yan 等人的 Blogel[9]以及 Fan 等人的最新研究 GRAPE[10]等,来加速动态图的匹配计算;

3) 针对动态图分析实时性要求高的特性,除了综合运用增量处理技术和分布式并行图处理框架降低匹配处理时延之外,还有学者研究图匹配的近似计算技术,在求解时延和匹配结果假阳性错误率之间做权衡.已有一些关于图模式匹配的综述论文[4,5],但是文献[4]虽然提到动态图匹配,但仅限于对动态图模式匹配的简单介绍,并没有针对动态图模式匹配技术展开详细讨论,也没有对相关研究进展进行介绍;文献[5]则完全是针对静态图匹配算法进行讨论.本文从关键技术、代表性算法和性能评价方面对动态图模式匹配技术的最新研究进展进行综述和讨论.目前,动态图模式匹配还处于起步阶段,但是已经引起了广泛的关注.

在实际应用中,不同的应用场景会产生不同的动态图匹配问题,本文第 1 节详细给出动态图匹配的相关定义,并对动态图匹配问题进行了分类介绍.第 2 节和第 3 节按照动态图的两种不同更新方式,分别对不同类型的动态图匹配问题进行问题描述和研究现状分析.第 4 节对不同类型匹配算法的性能进行比较分析,并给出结论.第 5 节介绍动态图匹配技术的应用现状.第 6 节对动态图匹配技术的未来发展趋势进行展望.

1 动态图匹配问题的定义和分类

1.1 基本动态图匹配问题

通常用三元组(V,E,L)对图数据进行形式化描述,其中,V 表示图中的节点集;E 表示图中边的集合,图中的边和节点可以带有属性信息;L 表示属性映射函数,将节点或边映射到一个或者一组属性上.图中的任意一条边 $e\in E$ 可由节点对(v_i,v_j)表示,其中,$v_i,v_j\in V$.本文用符号 G,P,u,v 分别表示数据图、模式图(或查询图)、单个数据图节点和单个模式图节点.目前,业界所使用的数据图 G 的类型可以分为两种:第 1 种数据图是超大图,例如社交网络就是一张超大图;另一种数据图是由大量的小图组成,例如 AIDS Antiviral 数据集[11]由表示化学物质原子结构的大量小图构成.据调研,目前的动态图匹配问题主要针对数据图是超大图的情况,因而本文集中讨论数据图为超大图情况下的动态图匹配问题.

动态图又称图流(graph streams),是指会随时间发生变化的图.动态图的更新形式可分为以下两类:(1) 图结构更新,随着时间推移,图数据中的节点和边会被插入和删除,从而导致图数据的结构发生变化;(2) 图内容更新,随着时间推移,图数据中的节点和边所关联的数据对象的内容或属性会发生改变,从而导致图数据的内容发生变化.下面给出这两种更新条件下动态图的匹配定义,由于目前的研究工作主要集中于数据图的结构更新,因此仅在图结构更新中给出动态图的定义.

(1) 图结构更新

定义 1(动态图). 已知初始数据图 G,引发数据图更新的操作可以用三元组$\langle op,u_i,u_j\rangle$表示.其中, $op=\{I,D\}$表示操作类型,$op=I$ 时表示增加边操作,$op=D$ 时表示删除边操作;u_i 和 u_j 表示数据图中与操作 op 相关的两个图节点.向数据图中增加一个节点,可以用与该节点相关的一系列增加边的操作来表示.与此类似,向数据图中删除一个节点,可以用跟该节点相关的一系列删除边的操作来表示.若用更新操作集合 $GC_t=\{\langle op_1,u_1,u_2\rangle,\ldots,\langle op_k,u_k,u_{k+1}\rangle\}(k\geqslant 1)$表示数据图 G 在 t 时刻的所有更新操作,用 $GC:G\to G'$表示数据图 G 基于更新操作集合 GC 更新后得到新数据图 G'的过程,则时间域$[0,T]$上的动态图 $G_D^{[0,T]}$ 是一个数据图序列,定义为

$$G_D^{[0,T]}=\{G_0,G_1,G_2,\ldots,G_T\},$$
$$G_0=G,\ GC_0=\varnothing,\ GC_t:G_{t-1}\to G_t(1\leqslant t\leqslant T).$$

定义 2(动态图同构匹配). 已知一个模式图 $P(V_p,E_p,L_p)$和一个动态图 $G_D^{[0,T]}=\{G_0,G_1,G_2,\ldots,G_T\}$,动态图同构匹配问题是指模式图 P 和动态图 $G_D^{[0,T]}$ 在 t 时刻的数据图 G_t 的数据子图 $G_{sub\text{-}t}=(V_{sub\text{-}t},E_{sub\text{-}t},L_{sub\text{-}t})(G_{sub\text{-}t}\subseteq G_t)$之间存在一个双射函数 f,且 f 满足:

- $\forall v\in V_p,L_p(v)=L_{sub\text{-}t}(f(v))$;

- $\forall(u_i,u_j)\in E_p,(f(v_i),f(v_j))\in E_{sub\text{-}t}$,且 $L_p(u_i,u_j)=L_{sub\text{-}t}(f(v_i),f(v_j))$,

其中,$f(v)$表示数据图中与模式图节点 v 满足双射关系的节点,即,数据图中与节点 v 相匹配的节点.

图 1 中的虚线框展示了初始数据图 G 在进行 $GC_1=\{\langle I,u_5,u_8\rangle\}$更新操作变成 G_1 之后,产生的与模式图 P 相匹配的子图 $M=[(v_1,u_3),(v_2,u_6),(v_3,u_5),(v_4,u_8)]$.

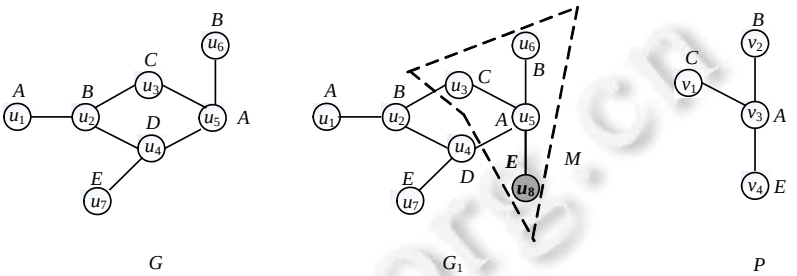


Fig.1 Matching the pattern graph with the updated data graph

图 1 将模式图与更新后的数据图进行匹配

子图同构是 NP 完全问题,基于子图同构的概念完成模式图与数据图之间的匹配,对于社交网络等对匹配精确度要求不严格的应用而言过于苛刻,制约了匹配效率.因而,不少研究工作定义了近似图模式匹配方案,并按照实际应用对匹配近似程度的不同要求由弱到强分为受限模拟(bounded simulation)匹配、图模拟(graph simulation)匹配、双向模拟(dual simulation)匹配、强模拟(strong simulation)匹配和严格模拟(strict simulation)匹配,以上统称为模拟匹配.受限模拟匹配将数据图和模式图边与边之间的严格匹配放松至数据图的一条路径和模式图的一条边进行匹配,数据图节点与模式图节点的匹配条件只要满足节点属性值相同且数据图节点的一个子孙节点与模式图节点的后继节点的属性值相同即可;图模拟匹配与受限模拟相比,要求数据图中的匹配节点保持与模式图中对应节点的后继关系,即,数据图节点存在后继节点与模式图节点的后继节点的属性值相同;双向模拟匹配在图模拟匹配的基础上,还要求数据图中的匹配节点保持与模式图中对应节点的前驱关系;强模拟匹配在双向模拟匹配的基础上,进一步要求匹配节点所在子图(子图中可能包含非匹配点)的半径不大于模式图的直径;严格模拟匹配比强模拟匹配要求更严格,要求完全由匹配节点构成的子图的半径不大于模式图的直径.模拟匹配类是一种可以容忍结果中存在一定噪声和错误的匹配算法,在社交网络分析和 Web 网络分析等应用中发挥着重要作用.模拟匹配的匹配结果与模式图之间存在一定的差别,但通常可以满足实际应用的需求,因此也通常被看做是正确的匹配结果.

下面以图模拟匹配为例,对动态图的图模拟匹配进行定义.

定义 3(动态图的图模拟匹配). 已知一个模式图 $P(V_p,E_p,L_p)$和一个动态图 $G_D^{[0,T]}=\{G_0,G_1,G_2,...,G_T\}$,动态图的图模拟匹配是指模式图 P 和 t 时刻数据图 $G_t(V_t,E_t,L_t)$之间存在二元关系 $S\subseteq V_P\times V_t$,其中,V_P 和 V_t 分别表示模式图节点和数据图节点的集合,且 S 满足:

- 节点约束:if $M=[(v_2,u_2),(v_3,u_3),(v_4,u_4)]$ then $L_p(v)=L_t(u)$;
- 边约束:$\forall(v,v')\in E_p,\exists(u,u')\in E_t:(u',v')\in S$;
- 图约束:$\forall v\in V_p,\exists u\in V_t:(u,v)\in S$.

如图 2 所示:初始数据图 G 中,节点 u_2 的属性与模式图 P 中的节点 v_2 的属性相同;同时,u_2 的后继节点 u_4 的属性与 v_2 的后继节点 v_4 属性相同,因此,u_2 与 v_2 匹配.同理,由于模式图 P 中的节点 v_3 的后继节点为空,则 v_3 与 u_3 相匹配.而 u_4 的后继节点中不存在属性与 v_4 的后继节点 v_3 相同的节点,因此 u_4 与 v_4 不匹配.在 t=1 时刻,初始数据图 G 在进行 $GC_2=\{\langle I,u_4,u_6\rangle\}$更新操作之后,$u_4$ 与 v_4 满足了匹配条件,按照图模拟匹配的定义产生的与模式图匹配的子图 $M_1=[(v_2,u_2),(v_3,u_3),(v_4,u_4)]$和 $M_2=[(v_2,u_2),(v_3,u_6),(v_4,u_4)]$.可以发现,其中 M_1 与模式图 P 的结构不同,但仍然认为是与模式图 P 相匹配的结果.

(2) 图内容更新

图内容被更新的情况指节点或边的属性值或者对图中某一特定对象的评价方式会随着时间发生变化.如图 3 所示:在 t=1 时刻,初始数据图 G 中,节点 u_3 和 u_6 的属性发生变化,此时,虚线框展示了数据图更新后得到的与模式图相匹配的子图 M=[(v_1,u_3),(v_2,u_6),(v_3,u_5),(v_4,u_8)].

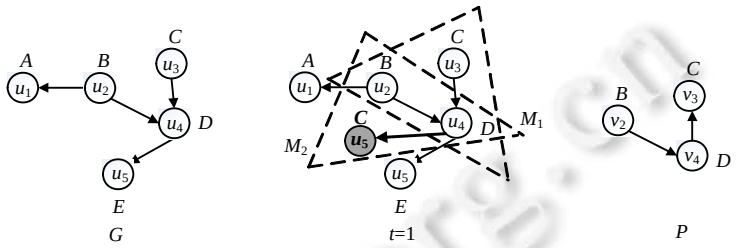


Fig.2 Simulation-Based matching result of the pattern graph P in the data graph G

图 2 基于图模拟匹配技术得到的模式图 P 在数据图 G 中的匹配结果

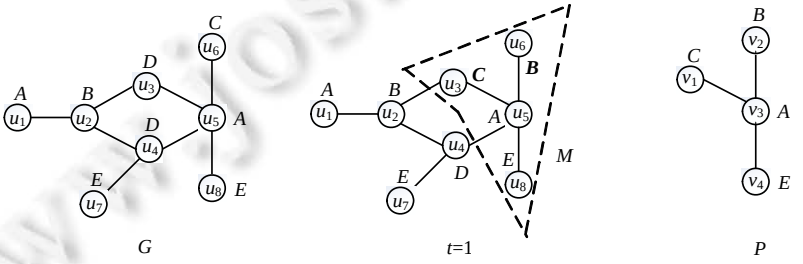


Fig.3 Matching result of pattern graph P over content updated data graph G

图 3 模式图 P 在内容更新后的数据图 G 中的匹配结果

1.2 动态图匹配问题分类

本节从应用出发,根据图数据的不同特点对动态图匹配问题进行分类,并分别介绍每类问题的代表性算法.这里所采用的分类方法并不相互独立,因此,在每类中介绍的匹配算法通常也运用了其他类型的匹配.

(1) 面向结构变化的匹配和面向内容变化的匹配

动态图匹配问题按图中的拓扑结构关系是否发生变化分为面向结构变化的匹配和面向内容变化的匹配.目前,动态图匹配的很多研究工作都围绕着数据图中的节点或边随着时间推移发生动态增删,导致图的拓扑结构发生变化这种情况展开讨论,称为面向结构变化的动态图匹配问题,主要应用于网络异常攻击检测和社交网络关系检测等场景.面向结构变化的动态图匹配问题最早在 2009 年由 Wang 和 Chen 提出,他们构建邻节点树(NNT)[12]并依此对匹配候选集进行过滤,从而有效减少假阳性匹配结果的产生.这之后的代表性算法包括 IncIsoMatch[13]、SJ-Tree[14]、graph simulation(DDST[15]、IncBMatch[13])等,对子图匹配的执行效率进一步提升.随着数据中心的出现,许多研究工作将数据中心的网络拓扑结构抽象为数据图进行相关分析,图中的每个节点代表一台服务器,节点之间的边代表服务器间的链路.在这种情况下,图数据的拓扑结构不会频繁发生改变,而图数据中节点的属性值(例如服务器的空闲内存量)和边的属性值(例如链路的有效带宽)会随时间频繁发生变化,称为面向内容变化的动态图匹配问题.为了有效应对数据图中节点属性值或边属性值的频繁变化, Zhong 等人提出了 Gradin 算法[16],将数据图的频繁子图中的每个节点的属性值作为一个数据维度,从而使一个包含 n 个节点的频繁子图可以表征成 n 维网格索引中的一个 n 维坐标向量,用这种方式索引数据图的所有频繁子图.网格索引适合索引动态变化的数据,因此当频繁子图中的节点属性值频繁发生改变时,也能保证索引的维护性能.基于 n 维网格索引,可以快速对节点标签值不满足匹配要求的中间结果进行过滤,有效提高了数据图和模式图的匹配效率,Gradin 算法也可以直接运用于数据图中边的属性值频繁更新的情况.

(2) 同构匹配和模拟匹配

动态图匹配问题是按照基于双射函数判定还是基于二元关系判定,可以相应地被称为同构匹配(isomorphism matching)和模拟匹配(simulation matching).如果模式图和数据子图之间存在一个双射函数,则称为同构匹配,这种匹配需要保证节点周围有相同的连通结构,要求与模式图完全一致,主要用于蛋白质分子的相互作用[4]、网络异常行为监测[3]等对结构要求比较严格的图数据分析应用.如果模式图和数据图之间存在二元关系,则称为模拟匹配.模拟匹配属于近似匹配,通常先根据数据图节点的标签为模式图中的每一个节点产生匹配候选集,再根据模式图中对节点的前驱和后继的不同近似匹配程度要求过滤掉不匹配的节点.代表性的算法包括 DDST[15]和 IncBMatch[13]等.其中,IncBMatch 算法得到的结果满足即使模式图的结构与数据子图不一样,但是仍然符合匹配条件,效率更高,匹配过程能在多项式时间内完成.除了有效提升匹配效率之外,与同构匹配相比,模拟匹配的灵活性更高,能够识别更多有用的匹配结果.主要用于路网中引发交通事故监测[15]、人群和各类团体 (例如毒品交易关系网络)之间的相互关系监测[13]等更加侧重挖掘节点之间的相互关系的图数据分析应用.

(3) 精确算法和近似算法

按照是否能够获得准确结果,动态图匹配算法分为精确算法和近似算法.精确算法能够保证匹配的结果完全精确,主要应用在网络异常检测和生物数据分析等这类对匹配的结果准确率有严格要求的领域.例如,同构匹配问题的精确算法可以保证计算结果与模式图完全同构,模拟匹配问题的精确算法可以获得与模式图满足二元关系的一系列结果.使用精确算法对数据图和模式图进行匹配的代表性算法包括 IncIsoMatch 算法、SJ-Tree 算法和 Gradin 算法等.另一方面,许多实时性要求高的应用中要求快速返回所有匹配结果,不适合使用计算复杂度高的精确匹配算法,同时,考虑到该类应用可容忍匹配的结果中除了正确的结果之外还可包含一部分错误的结果(即假阳性结果),近似图匹配算法应运而生.近似算法不同于模拟匹配,近似算法通常基于概率统计等数学模型,计算结果中除了正确的结果之外,还包含一部分错误的结果,可以通过参数调整将错误比率控制在一定范围内.以 NNT[12]、Replication mechanism[17]、SSD[18]为代表的研究工作提出了近似图匹配算法,以牺牲匹配精度换取匹配效率的方式,将图匹配问题的转换为较易计算的问题,例如将图匹配问题转化为向量空间的关系检测问题,可将图匹配的时间复杂度降低至多项式级别.

精确算法和近似算法是决定匹配结果准确率的两种算法,而同构匹配和模拟匹配则是不同类型的图匹配模型,概念上有明显区别.

(4) 集中式匹配和分布式匹配

按照是否部署在分布式的平台上,动态图匹配问题分为集中式匹配和分布式匹配.集中式匹配在单台计算机上运行,主要处理规模相对较小的图数据.目前,动态图集中式匹配主要采用基于连接(join)的匹配方法或者基于探索(exploration)[19]的匹配方法实现.基于连接的匹配方法采用将查询图进行分解的策略,对每一个查询分片进行匹配后将得到的结果进行连接,一般需要构建索引.而基于探索的方法是从数据图的一个节点出发,根据模式图的结构关系对数据图进行探索,一般不需要构建索引.分布式动态图匹配技术主要依托于分布式并行图处理框架实现,用来处理规模庞大且计算复杂度高的动态图数据.目前,主流的分布式并行图计算系统的类型包括 3 种,分别是中心节点模型(包括 Pregel[6]、Trinity[7]、Giraph[8]、GraphLab[20]等)、中心块模型(Blogel[9,21])以及自动并行化模型(GRAPE[10]).其中,中心节点模型和中心块模型需要根据模型的特点来改造算法,而 Fan 等人研究的自动并行化模型则不需要考虑模型的特点,用户只需要提供局部匹配、增量计算、结果拼接这 3 种连续的算法即可,不用对算法的逻辑进行修改.这 3 种分布式并行图计算系统都遵循 BSP[22]模型.表 1 展示了不同类型分布式并行图计算系统在最短路径查询方面的性能,采用美国路网作为数据集,可以发现,GRAPE 系统的性能明显优于其他类型系统.由于要实现对匹配结果进行增量更新的需求,因此分布式匹配一般采用基于探索的方法.分布式并行图计算框架包含多个超级步(super-step),每一步只发送当前状态的消息而忽略前次的状态,因此会产生不一致的结果,文献[23]解释了这种不一致结果产生的原因,并提出 Stp(Q)算法来解决因不一致性而产生的结果准确率问题.

综上,算法所针对的应用背景不同,所解决的图匹配问题的类型也就不同.表 2 总结了迄今为止具有代表性的动态图匹配算法以及所解决的图匹配问题的类型.以下将按照面向结构变化和面向内容变化这种分类方式对动态图匹配技术展开细致介绍,将每一类图匹配技术的主要问题、代表性算法和研究现状进行分析对比.由于目前的大部分研究工作都围绕着面向结构变化的动态图匹配问题展开,本文将重点介绍面向结构变化的图匹配技术的最新研究进展.

Table 1 Graph traversal on parallel systems

表 1 分布式系统中的图遍历

系统	类型	时间(s)	通信开销(MB)
Giraph	中心节点	10 126	1.02×10^5
GraphLab	中心节点	8 586	1.02×10^5
Blogel	中心块	226	2.8×10^3
GRAPE	自动并行化	10.5	0.05

Table 2 Classification of representative dynamic graph pattern matching algorithms

表 2 代表性动态图匹配算法分类

分类 / 算法	结构变化	内容变化	精确算法	近似算法	同构匹配	模拟匹配	集中式匹配	分布式匹配
NNT(2009)	√	×	×	√	√	×	√	×
IncSimMatch(2011)	√	×	√	×	×	√	√	×
IncBMatch(2011)	√	×	√	×	×	√	√	×
BR-Index(2011)	√	×	×	√	√	×	√	×
Vertex-Replication(2012)	√	×	×	√	√	×	×	√
SJ-Tree(2013)	√	×	√	×	√	×	√	×
DeltaGraph(2013)	√	×	×	√	√	×	×	√
DDST(2014)	√	×	√	×	×	√	√	×
Gradin(2014)	×	√	√	×	√	×	√	×
SSD(2014)	√	×	×	√	√	×	×	√
Lary-Search(2014)	√	×	√	×	√	×	√	×
MultiView(2014)	√	×	√	×	√	×	√	×
Distributed-IncSimMatch(2016)	√	×	√	×	×	√	×	√
Graph-View(2016)	×	√	×	√	√	×	√	×
Stp(Q)(2016)	√	×	×	√	√	×	×	√
D-ISI(2016)	√	×	√	×	×	√	×	√
IncISO(2017)	√	×	√	×	√	×	√	×

2 面向结构变化的动态图匹配技术

面向结构变化的动态图模式匹配技术在目前应用最为广泛,其所处理的图数据结构会随着时间发生变化.从算法设计的角度来说,可以将其分为基于快照处理的匹配技术和基于增量处理的匹配技术.基于快照处理的匹配技术就是将每一个时间戳上更新的数据图看成一个静态图来进行匹配处理,通常适用于增加边数目比较多的情形,该情形下可以一次性完成所有增加边操作,并基于更新后的数据图快照进行匹配计算.邻节点树(NNT)算法是最早提出的基于快照处理的动态图匹配算法,之后的代表性算法包括 DeltaGraph[24]和 DDST[15]等.第 2.1 节将重点介绍 NNT 算法,并简要介绍其他相关算法.

基于增量处理的匹配技术仅对数据图中更新的部分进行分析和匹配,避免对整体数据图进行重新匹配所带来的重复计算.Fan 等人[13]最早将增量处理技术应用到动态图匹配计算当中,提出了 IncSimMatch 算法和 IncBMatch 算法,并与各自对应的批处理(快照)算法,即 Match_s 和 Match_{bs},进行对比.实验结果表明:当增加边或者删除操作的数目不超过初始数据图边总数的某一百分比时,基于增量处理的匹配技术的执行效率更高.之后,增量计算的思路广泛应用于动态图模式匹配算法的设计中,其他代表性的基于增量处理的动态图匹配算法包括 SJ-Tree[14]、Lazy-Search[25]和 Distributed-IncSimMatch[26].这些算法是在 Fan 等人提出的增量图匹配技术的基础上加入其他的查询优化策略对匹配过程进行改进,其中以 SJ-Tree 算法最具代表性.动态图增量匹配算法在子图

匹配的过程中可以采用基于连接的匹配技术、基于探索的匹配技术或基于图模拟匹配技术实现,其中,基于连接的匹配技术由于会产生大量的中间结果,因此不适用与分布式并行图处理框架.第 2.2 节将分别介绍这 3 类算法.图 4 总结了面向结构变化的动态图匹配技术的分类与相应代表性算法.

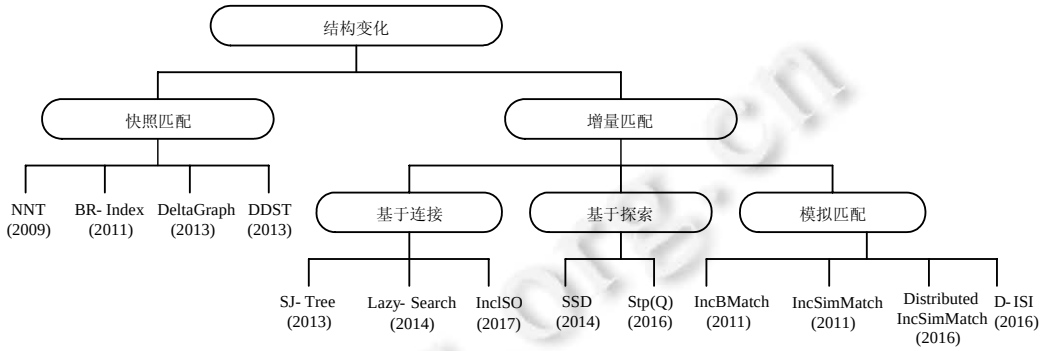


Fig.4 Classification and representative algorithms of structure change-oriented dynamic graph matching techniques

图 4 面向结构变化的动态图匹配技术的分类与对应代表性算法

2.1 基于快照的匹配技术

动态图匹配问题实际上也是大规模的静态图查找问题,可以将每一时刻更新后的数据图看成静态图,然后对这些连续的静态图进行模式匹配.这种基于快照的匹配技术包括两部分:更新操作和匹配操作.下面分别从这两部分进行介绍.

(1) 数据图更新

文献[12]最早提出了关于动态图的模式匹配问题,将不断更新的图数据看成图流的形式,并提出了基于快照的动态图模式匹配算法——NNT 算法,在性能方面取得了较好的结果.NNT 算法采用为数据图 G 和模式图 P 中的每个节点 u 都构造一棵邻节点树,记为 $NNT(u)$.以数据图 G 举例,给定——深度值 L,$NNT(u)(u\in G)$存储了数据图 G 中以 u 为根节点且长度不超过 L 的所有路径.如图 5(a)所示,T_1~T_4 分别为数据图 G 中的节点 1~节点 4 对应的深度值 L=2 的邻节点树,其中,大写字母表示节点的属性.其中,T_1 和 T_2 内容相同.图 5(b)则为这些邻节点树构建倒排索引,用来查找数据图中的节点和边对应的各个邻节点树中的节点和边,其中,*表示倒排索引中的省略部分.

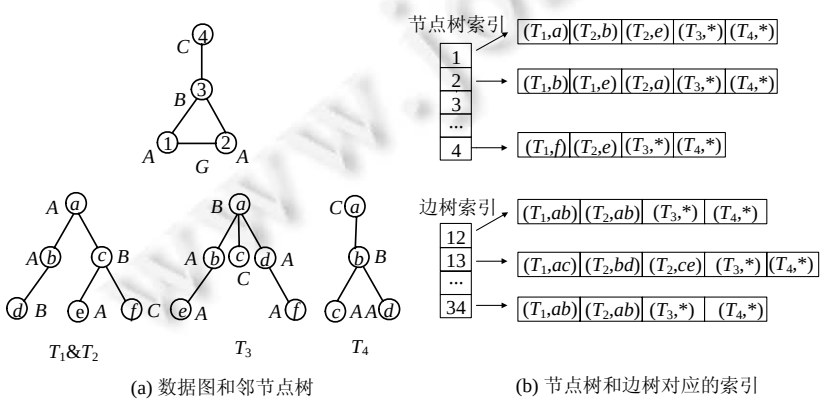


Fig.5 Graph, node-neighbor trees, and inverted index constructed based on node-neighbor trees

图 5 数据图、邻节点树以及基于邻节点树构建的倒排索引

面向结构变化的动态图更新操作包括删除边操作和增加边操作,下面以基于邻节点树构建的倒排索引为例,分别介绍执行这两种操作时如何对索引进行更新.

- 删除边操作:如图 5 所示,假设在 t 时刻删除数据图 G 中的边(1,3),则同时需要从倒排索引中删除该边在所有邻节点树中对应的边,且如果对应的边存在子边,则这些子边也要被删除.此例中,边(1,3)在 T_1&T_2 中对应的边(a,c)以及(a,c)的子边(c,e)和(c,f),以及和这些边相关的节点都要从倒排索引中删除;
- 增加边操作:假设在 t 时刻向图 5 所示的数据图 G 中增加一条边(1,4),首先,根据倒排索引找到节点 1 对应的每棵邻节点树中的节点.然后,从每棵树中与节点 1 相对应的节点开始,将数据图 G 中新产生的 L=2 的路径加入到邻节点树中.如 T_1 中从节点 a 出发增加路径$(a\to g\to h)$,其中,节点 g 和节点 h 的属性分别为 C 和 B.在 T_2 中,从节点 b 增加路径$(b\to g)$,其中,g 的属性为 C.直到所有与节点 1 对应的节点都完成即可.最后,将新加的节点和边添加到倒排索引中.同样的,对节点 4 也需进行相同的操作更新索引结构.

NNT 算法将模式图与数据图每一时刻的快照图进行匹配,并不能体现图随时间的演变过程.文献[27]首次提出一种针对大规模动态图的索引结构 BR-Index,将数据图划分为一系列重叠的索引域,每一个索引域包含若干个相互独立的核心域.提取每个索引域的最大特征,然后构建一个特征点阵用来维护和查找这些特征.BR-Index 还维护了一个基于哈希原理实现的节点查找表,用来存储每一个索引域中的数据图节点以及该节点处于索引域中的哪一个核心域.当数据图更新时,可以基于该查找表快速定位更新部分.文献[15]提出了基于时间窗口的子图匹配算法 DDST,窗口内包含所有满足时间约束的图快照信息.而且数据图中每一条边上都有一个时间常量记录了该边的生成时间.这样就能判断一个数据子图是否在给定的时间窗口内存在(或称“合法”),只有窗口内合法的数据子图满足匹配条件后才会成为匹配的结果.与 NNT 算法相比,使用时间窗口的好处是可以反映出图随时间的一个演变过程,即:只有时间窗口内的图快照满足匹配条件后才是“合法”的结果,更加符合实际的应用.文献[24]提出了一种树形索引结构 DeltaGraph,该索引结构类似二叉树,每个叶子节点都存储按时间顺序排列的一个快照图信息,内部节点存储的信息则是由其孩子节点存储的信息通过取交函数运算得到,其中,根节点表示源节点,叶子节点表示目标节点,每条边都保存有从源节点到目标节点的信息.当数据图执行增加或删除边操作时,这种索引结构只需要将该时刻的数据图快照添加到叶子节点即可.

(2) 子图匹配

在匹配的过程中,如果采用子图同构的方式进行匹配计算,则时间复杂度很高.例如,NNT 算法若采用子图同构的方法进行匹配计算,其时间复杂度高达 $O(n_1\times n_2(|T_1|^{1.5}/\log|T_1|)|T_2|)$[27].其中,$n_1$ 表示模式图节点的数目,n_2 表示每一时刻数据图中的节点数目,$|T_1|$和$|T_2|$分别表示模式图和数据图所对应的邻节点树的最大个数.可见:若采用子图同构的方式进行匹配计算,NNT 算法的执行效率会受到严重制约,不满足动态图匹配的实时性的需求,因此,若应用允许要尽可能避免采用子图同构的方式进行匹配计算.

NNT 算法实际上采用的是近似算法,通过比较两个节点的邻节点树中的路径进行子图匹配计算:如果模式图每一个节点构建的邻节点树中的所有路径在数据子图中与其对应的节点构建的邻节点树中都存在,则模式图与该数据子图相匹配.这种子图匹配计算方法的时间复杂度为 $O(n_1n_2r^l)$,其中,r 表示数据图节点最大的度,l 表示选取的深度值.由于每次匹配都需要对所有可能的节点对进行匹配计算,开销仍然很大.因此,算法还提出了一种编码方式,将邻节点树转化成数字向量,并统计所有邻节点树中不同路径的数目,以达到进一步降低计算开销的目的.如图 5(a)所示,邻节点树 T_1~T_4 共存在 8 种不同的路径,因而每个节点的邻节点树都可以用 8 维向量表示,其中,第 i 维记录了该邻节点树中存在第 i 类路径的数目.这样就可以将模式图和数据子图转化成多维数字向量的形式,如果模式图中每一维度的值都小于等于数据子图中对应维度的值,则满足匹配条件.进行以上优化后的匹配时间复杂度为 $O(\bar{L}n_1n_2)$, 其中, $\bar{L}$ 表示数字向量中非零项的数目.

BR-Index 算法首先提取模式图的最大特征集合,在子图匹配的过程中,根据提取的模式图最大特征集合找到数据图中包含这些特征集合的索引域;然后,在对应的索引域中找到最大特征的候选集;最后将候选集进行组合,即可得到模式图的匹配结果.DDST 算法采用模拟匹配的方式来完成子图匹配计算,因为模拟匹配比同构匹

配更灵活,可以识别更多有用的结果.由于传统的图模拟匹配技术只需要数据图中的匹配节点保持与模式图中对应节点的后继关系,放宽了同构匹配的约束条件,因此很难获取与模式图一致的拓扑结构.DDST 算法在图模拟匹配的基础上加入了两个约束条件 Dual simulation 和 Locality[28],即为模式图构建一个签名,签名包括边和节点的标签以及所有节点的入度和出度信息,若数据图子图的签名与模式图一致,则继续采用二元模拟的匹配技术进行匹配计算.同时提出模式图中的边在时间域上的偏序关系,数据子图中各条边的时间属性只有满足时间域上既定的先后顺序才满足匹配条件.树形索引 DeltaGraph 则采用分布式并行图处理框架完成子图匹配计算,即:将快照信息存储在内存中,给定一个模式图,可以直接从根节点开始根据边上的构建信息找到与该模式图匹配的快照图.

综上,采用子图同构的方式进行子图匹配计算效率低、耗时高,因此可以采用近似匹配算法或者模拟匹配技术来设计基于快照的匹配技术.近似算法用少量假阳性的结果降低匹配时间,模拟匹配则直接采用二元关系匹配来代替子图同构匹配.模拟匹配主要适用于更加侧重于挖掘节点之间关系的应用,用户需针对不同的应用背景来选择合适的子图匹配算法.

2.2 基于增量处理的匹配技术

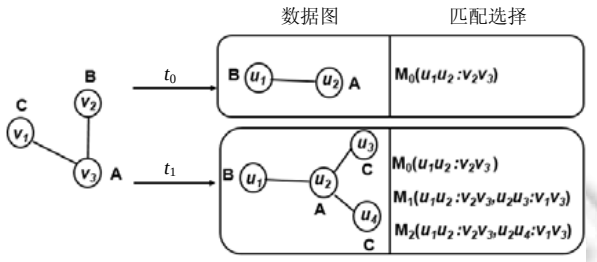
动态图模式匹配属于 NP 完全问题.在匹配的过程中,如果按照基于快照的重复查找策略,需要在每一次数据图更新后都对完整的数据图进行匹配,当更新操作数量较少时,造成大量冗余计算,无法满足动态图高效分析处理的潜在需求.因此在设计算法的时候可以采用增量处理技术,通过数据图局部匹配技术或者利用之前匹配的结果来减少冗余的匹配计算.

文献[13,29]介绍了增量图计算的思想,增量图计算是指给定一个数据图 G、模式图 P、初始的模式图匹配结果 $P(G)$、数据图的更新 ΔG,计算数据图更新后新增加的匹配结果集 ΔO,即 $P(G\oplus\Delta G)=P(G)\oplus\Delta O$.其中,$\oplus$表示操作符,用来将变化的内容加入到原始数据当中.文献[13]提出增量图模拟匹配(IncSimMatch)算法,在局部匹配的基础上,充分利用之前的匹配结果,并提出辅助的数据结构 $match(v)$和 $candt(v)$加速匹配计算.其中:v 表示模式图节点,$match(v)$表示数据图中与节点 v 匹配的节点;$candt(v)$表示数据图中与节点 v 属性相同但是不满足其他匹配条件的节点.与基于快照处理的匹配技术相同,数据图更新时对匹配结果的影响可基于增加边和删除边这两种情况分别展开讨论.

- 删除边操作:IncSimMatch 算法中,当从数据图中删除边时不会导致匹配结果增加.只有当删除的边,例如删除(u_i,u_j),满足 $u_i\in match(\cdot),u_j\in match(\cdot)$才会导致匹配结果的减少,即,将之前的匹配结果集中所有包含该边的匹配结果移除即可;
- 增加边操作:IncSimMatch 算法中,当从数据图中增加边时不会导致匹配结果的减少.只有当增加的边,例如增加(u_i,u_j),满足 $u_i\in candt(\cdot),u_j\in match(\cdot)$或者 $u_i\in candt(\cdot),u_j\in candt(\cdot)$时才会产生新的匹配结果.此时,可以通过局部匹配的方法来找出所有匹配结果.

文献[29]介绍了 4 种图查询问题,分别是 RPQ(regular path queries)、SCC(strongly connected components)、KWS(keyword search)和 ISO(subgraph isomorphism),并提出了针对这 4 种图查询问题的增量计算方法.其中,前 3 类图查询问题与本文的主旨子图模式匹配不符,因此不做详细讨论.与本文相关的是第 4 类 ISO 子图同构查询.文献[29]针对 ISO 查询问题提出了增量计算方法 IncISO,当数据图更新时,只需对更新边周围模式图直径范围内的数据图节点进行重新匹配即可,避免对整个数据图的重复计算.

朴素增量匹配是一种基于增量图计算的思想,通过局部匹配的方式,检查数据图中每一条新增加的边是否与模式图中的边相匹配,即,检查模式图中是否有与数据图新增加边的属性相同的边,而且数据图中新增加的边的端点与模式图中相对应边的端点的属性也是相同的.满足上述条件后,检查新得到的局部匹配结果图与已经存在的局部匹配结果图之间能否组合成一个更大的匹配结果图.如图 6 所示,在 t_0 时刻,找到所有模式图中与数据图中边(u_1,u_2)相匹配的边.同理,在 t_1 时刻找到所有模式图中与数据图中边(u_2,u_3)和(u_2,u_4)相匹配的边.t_1 时刻的匹配结果能够与 t_0 时刻的匹配结果结合产生更大的匹配结果 M_1 和 M_2,且 M_1 和 M_2 均是与模式图 P 相匹配的结果.



(a) 模式图 (b) 查询的过程以及匹配结果

Fig.6 Naïve incremental ieda for dynamic graph pattern matching

图 6 动态图模式匹配的朴素增量思想

在子图匹配过程中,动态图模式匹配与静态图模式匹配相似,因此可根据静态图模式匹配的分类方法,按是否需要将模式图分解,可分为基于连接的匹配技术和基于探索的匹配技术.除此之外,在动态图子图匹配中,运用基于图模拟的匹配技术也得到了较好的实验结果.下面分别从这 3 个方面对动态图模式匹配技术进行深入介绍.

2.2.1 基于连接的动态图数据匹配技术

研究发现:当模式图规模较大时,如果每次数据图更新时都对整个模式图进行查找会浪费大量时间,因此可以将模式图进行分解.如图 7 所示:将模式图 P 分解成一系列更小的子模式图,表示为 $p_1,\ldots,p_k$.接着,分别对这些子模式图进行匹配,将匹配的结果进行连接整合即可得到整个模式图匹配的结果.

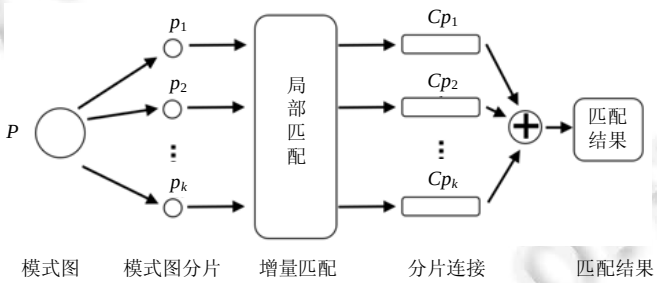


Fig.7 Technology framework for join-based dynamic graph pattern matching

图 7 基于连接的动态图数据匹配技术框架

在静态图模式匹配中,基于连接的方法应用比较广泛,代表性算法包括 gIndex 算法[30]和 GraphGrep 算法[31]等.基于连接的方法通过挖掘数据图中具有识别能力的特征,例如路径[31,32]、树[33,34]和子图[30,35,36]等,一方面根据这些特征在数据图上构建索引;另一方面,利用这些特征分解模式图.但是,在将该方法直接应用到动态图匹配过程中,则会产生一系列问题.例如,gIndex 算法[30]需要在每一时刻挖掘数据图子图的特征,这显然不适用于实时分析要求高的动态图匹配处理.GraphGrep 算法[31]可以满足实时性需求,但是其仅仅使用路径这种识别度不高的特征进行核实过滤,可能会产生大量错误的匹配结果.可见:将基于连接的方法运用于动态图匹配分析时,如何实现高效数据图特征提取,是核心挑战问题.

在社交网络应用中构建的数据图往往是多关系图,即:实体之间边的属性不仅表示连通性,还表示实体之间的相互关系.因此在这类应用中,可以用数据图实体之间的相互关系(即边的类型)作为数据图特征对模式图进行分解.文献[14]提出了一种树形索引结构 SJ-Tree.SJ-Tree 树是一棵二叉树,它的根节点表示模式图,内部节点表示模式图的子图,内部节点的孩子节点由该节点中的子图进一步分解得到,叶子节点则表示将模式图最终分解后得到的分解结果.如图 8 所示,SJ-Tree 树中的每一个节点都存储了它的兄弟节点和父节点的信息并维护了一个哈希表存放该节点的匹配结果.当数据图被更新时,在 SJ-Tree 树上迭代搜索所有叶子节点以获取包含新增

加边的匹配结果,并将匹配结果存放在 SJ-Tree 树相应节点的哈希表中.同时,检查该叶子节点的兄弟节点所维护的哈希表中存储的兄弟节点的匹配结果能否与其连接整合成更大的匹配结果.如果存在更大的匹配结果,则将该结果存放至父节点的哈希表中,直到产生与整个模式图相匹配的结果.例如图 8 中,SJ-Tree 叶子节点的候选集{(“George”,“friend”,“Join”)}和{(“Join”,“like”,“Santana”)}相连接整合,可以得到更大的匹配结果.

{(“George”,“friend”,“Join”),(“Join”,“like”,“Santana”)}.

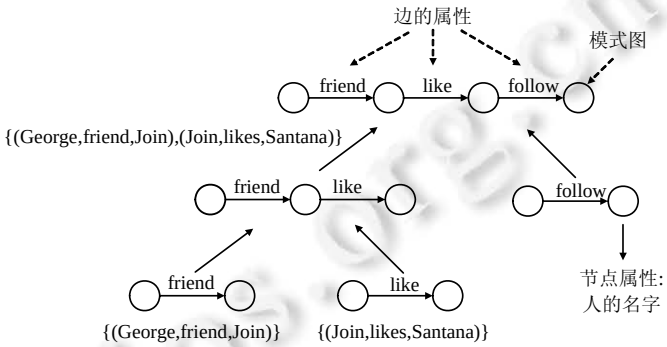
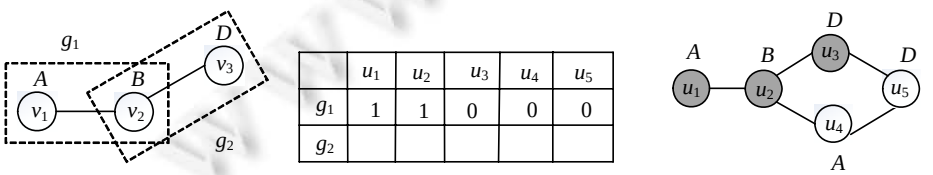


Fig.8 Decomposition of social query in SJ-Tree

图 8 将社会关系按照 SJ-Tree 进行分解

如图 8 所示:由于模式图包含 friend 关系,在子图匹配的过程中,如果数据图中 friend 关系频繁出现,则需要追踪所有的与 friend 相匹配的边,这无疑将花费大量时间.因此,可以选择将模式图的分解结果 friend 边推迟匹配,优先匹配在数据图中出现相对不频繁的模式图的分解结果.基于此思想,文献[37]提出一种基于选择度的 Lazy-search 算法.给定初始的数据图 G,则图 G 中的 k 边子图 g 的选择度为 g 在数据图 G 中出现的次数与 G 中所有 k 边子图数的比值.其中,g 被称为选择度基元.为了限制子图同构的计算代价,同时也为了使选择度基元在数据图变化的过程中仍然有效,一般选择单边或者双边子图作为选择度基元.Lazy-search 算法用离线方式计算出数据图包含的所有类型的单边或双边子图的选择度,并将这些选择度基元按其选择度值递增排序,排在前面的选择度基元的选择度值越低,识别能力则越高.后续将查询图按照这些选择度基元的先后顺序进行分解.在匹配的过程中,为了保证 SJ-Tree 树中相邻叶子节点的匹配结果在数据图上仍然相邻,构建了一个位图索引结构,如图 9(b)所示.图 9(b)中,行表示数据图中所有节点,列表示满足邻接关系的模式图分解结果.图 9(a)展示了模式图 P 的两个相邻分解结果 g_1 和 g_2.如果数据图节点在查询分片 g_1 的匹配结果中,则相应比特位置 1,其他比特位置 0.例如,g_1 与数据图中的边(u_1,u_2)相匹配,则位图中 g_1 对应的 u_1 和 u_2 的比特位分别置 1,其比特位置 0.对 g_2 进行匹配时,只需要在数据图中 g_1 所在位图向量对应值为 1 的节点(即 u_1 和 u_2)周围查找与 g_2 满足匹配的结果即可.这样匹配的好处是:利用模式图分解结果相邻的特性,保证匹配结果在数据图上仍然相邻,能够避免产生不满足邻接关系匹配结果,减小查找空间,加快匹配速度.



(a) P (b) 基于位图的索引 (c) G

Fig.9 Bitmap-Based index structure

图 9 基于位图的索引结构

基于连接的匹配技术采用子图同构的方式进行子图匹配计算,可以得到精确的匹配结果,但是代价高昂.虽

然利用查询分解的方法可以控制待匹配子图的规模从而限制子图同构的代价,但是仍会产生很多无效的中间结果.如果再对数据图构建索引以降低无效中间结果的数目,也需要考虑索引的构建和维护代价.同时,如何在频繁更新的动态图数据中挖掘特征结构进行查询分解,仍是当前需要重点解决的问题.

2.2.2 基于探索的匹配技术

文献[19]提出了一种基于探索的匹配技术,该方法得到的结果是近似结果而不是精确结果.探索的过程如图 10 所示:给定模式图 P 和数据图 G,首先从模式图中的节点 a 出发,找到数据图 G 中与其相匹配的节点 a_1;然后从节点 a_1 开始探索数据图 G,找到节点 b_1,发现数据图中的边(a_1,b_1)与模式图中的边(a,b)相匹配;接着从节点 b_1 出发探索数据图 G,找到节点 c_1 和 c_2,发现数据图中的边(b_1,c_1),(b_1,c_2)与模式图中的边(b,c)相匹配.这样就可以获得数据图中和模式图相匹配的子图,而不用像基于连接的匹配技术那样需要对大量的中间匹配结果进行连接计算,也不需要在数据图上构建和维护索引.

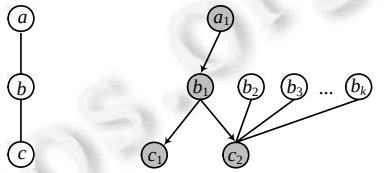


Fig.10 Example of graph pattern matching using exploration-based method

图 10 基于探索的方法示例

相比基于连接的动态图匹配技术,基于探索的动态图匹配技术更适合于在分布式并行图处理框架中实现.主流的分布式并行图处理框架,以谷歌的 Pregel[6]系统为例,均采用以节点为中心的计算模型框架.该模型框架中:每个节点都维护一个信息输入队列和一个信息输出队列;每个计算任务由一系列超级步(super-step)组成,且每个超级步中节点从信息输入队列接收输入信息后,会按用户自定义的脚本程序对信息进行处理,最后将处理结果输出至信息输出队列.由于这种以节点为中心的计算模型强调存储和处理单元均是节点,考虑到基于连接的动态图匹配技术会产生大量中间结果,这无疑加重了模型框架中节点的处理、存储和数据传输代价,因而基于连接的动态图匹配技术不太适用于在分布式并行图处理框架中实现.相反,基于探索的方法因为不需要构建索引且不会产生和处理大量的中间结果,则更加适用于在分布式并行图处理框架中实现,并且天然满足增量计算的潜在需求.PathMatch 算法是在分布式并行图处理框架中采用的基于探索的匹配技术,该算法首先找到模式图中的一条哈密顿路径,哈密顿路径指能够访问到图中每一个节点的路径,然后基于该路径进行信息传递来完成探索匹配.然而,将基于探索的动态图匹配技术在分布式并行图处理框架中实现仍然面临以下问题:

- 基于探索的方法得到的结果是近似匹配结果而不是精确匹配结果;
- 在解决一些动态图匹配问题时,使用基于探索的动态图匹配技术会比使用基于连接的动态图匹配技术的计算代价昂贵得多;
- 并不是所有的图匹配步骤仅仅依靠探索的思路就可以完成,例如检查需要探索的下一个节点是否是最开始的匹配节点.

针对以上问题,可以采用基于连接和基于探索相结合的方式解决.即:宏观上使用基于连接的动态图匹配技术的框架,而在子图匹配的过程中,使用基于探索的方法找到匹配结果.换言之,在子图匹配的过程中仍采用模式图分解的方法,基于模式图中信息的传递而不是基于数据图中提取的特征子图对模式图进行分解.接着,将每个模式图的分解结果存储到分布式并行图处理框架的一个节点上,并编写节点的处理脚本,让节点基于探索的匹配方法实现图匹配.例如,STwigMatch 算法[19]即为基于连接与基于探索相结合的方式,将模式图分解成一系列小的 twigs,然后对这些小的 twigs 按照基于探索的匹配技术进行匹配.这种方法的好处是可以权衡匹配计算代价和匹配结果准确率这两个重要因素.

文献[18]采用了分布式框架 Giraph,并提出了基于探索的 SSD 算法.下面以图 11 为例说明 SSD 算法的计算过程:首先,将模式图中度最大的节点定为汇点;然后从汇点出发,利用宽度优选搜索策略决定边的方向.这样可将模式图转化为单汇点有向无环图(DAG).图 11(b)给出了图 11(a)所示的模式图进行转化后得到的 DAG 图,其中,节点 3 为汇点,可以接收到所有其他边传递来的信息.基于汇点可以将图 11(b)所示的 DAG 图分解成 3 个子 DAG 图,且每一个子 DAG 中会存在一个入度为 0 的节点,被称为源点,例如,0 号子 DAG 图中的节点 0 即是该子 DAG 图的源点.图 11(c)展示了子图匹配的过程,在 0 号子 DAG 图中,信息从节点 0 出发,传递到其所有邻节点,即节点 1 和节点 2.信息在这两个节点中经过转换后,继续向下游的邻节点传递并最终到达节点 3,从而得到从源点开始的信息传递规则.然后,将这种从源点开始的信息传递规则映射到数据图中,并基于该规则从与源点属性相同的节点中开始检查后续节点是否匹配.例如图 11(c)中,源点 0 与数据图节点 *a* 相匹配,接下来由一系列超级步来完成匹配任务:在第 1 超级步中,节点 *a* 将信息传递到其相邻的节点;在第 2 超级步中,节点 *b*, *c* 收到信息后并将信息传递给节点 *d*.节点 *d* 发现收到的来自相同节点的两条信息,与 0 号子 DAG 图的信息传递规则相同,则可证明匹配子图的存在.

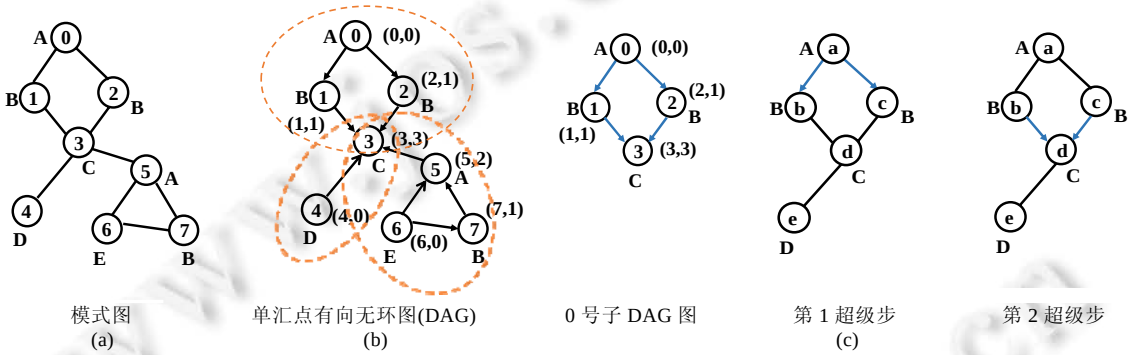


Fig.11 Illustration of the SSD algorithm

图 11 SSD 算法过程演示

SSD 算法存在一个问题,即在 Super-step 2 中,如果在节点 *b* 和节点 *c* 将信息传递给节点 *d* 的同时,边(*a*,*b*)被删除,则节点 *d* 收到信息后会返回过期的不准确匹配的结果.为了解决这个问题,文献[23]提出了 Stp(Q)算法.

基于探索的匹配技术更加适用于处理规模庞大的图,但是由于匹配的不一致性会产生不精确的匹配结果,因而该类方法适用于那些对结果准确率要求不严格的应用,例如社交网络分析等.如何将基于探索与基于连接这两种方法有效地结合,用以权衡子图匹配效率与子图匹配准确率这两个重要因素,是目前研究的热点问题之一.

2.2.3 基于模拟的匹配技术

基于连接的匹配技术可以产生精确的匹配结果但是连接操作代价较为昂贵,基于探索的匹配技术则会产生不精确的匹配结果,且这两种方法都采用 NP 难的子图同构图匹配计算方式,限制了这两类技术在执行效率上的提升.因此,基于模拟的匹配技术成为当前动态图匹配技术研究领域的研究热点之一.基于模拟的匹配技术通常先根据节点的标签为模式图中的每个节点产生一个匹配候选集,然后根据模式图中节点前驱和后继的不同近似程度要求过滤掉不匹配的节点(详见本文第 1.1 节模拟匹配定义).

文献[13]在原始图模拟匹配的定义上进行了扩展,提出了受限模拟(bounded simulation)的概念,重新定义了带有边权重的模式图.模式图中每一条边都维护了常量 *k*,数据图中的一个节点只要在 *k* 跳范围内能与模式图中对应节点的后继节点匹配,就认为该节点与对应节点匹配.因此,即使匹配得到的数据子图的结构和模式图的结构不一样,仍然符合匹配的条件.受限模拟匹配的好处是可以将匹配过程在多项式的时间内完成,有效提高了子图匹配的效率.该作者同时将受限模拟匹配技术应用到动态图模式匹配中,提出了增量 IncBMatch 算法.增量 IncBMatch 算法与本文第 2.2 节提到的 IncSimMatch 算法的增量计算思路相同,解决了同构匹配复杂度高的问

题,取得了较好的效果.文献[26]首次将 IncSimMatch 算法应用到以节点为中心的分布式并行图处理框架中,实现了动态图模拟匹配的分布式并行计算.在匹配过程中,用框架中的一个节点存放更新的图数据,并编写执行脚本让该节点过滤掉那些肯定不会产生新匹配结果的边.接着,再用框架中的一个节点检查这些边是否满足图模拟匹配中对边的约束关系,当主进程收到来自所有节点的处理信息之后,检查这些子图是否满足图模拟匹配中边的约束关系,即可得到最终匹配结果.

基于模拟的匹配方式由于采用二元关系进行匹配,会产生与模式图结构不一致的结果,因此主要适用于处理类似社交网络分析这类对图结构匹配要求不是很严格的应用.模拟匹配也可以用来进行候选集的剪枝,文献[39]提出了一种分布式的动态图剪枝算法 D-IDS,采用构建在图模拟匹配为基础的双向模拟匹配方式对数据图进行剪枝,双向模拟要求当前节点的所有孩子节点符合二元关系的同时,其父节点也要符合二元关系.当数据图更新时,通过二元模拟的思想对数据图进行剪枝,可以将一个大的数据图剪枝成相对较小的数据图,并持续地维护该数据图,在匹配的过程中只需对小图进行增量匹配即可.

模拟匹配的匹配效率高,不论是分布式环境还是集中式环境下都适用,具有其独特的优势,这种优势在处理动态图匹配中更加明显.其中,构建在以图模拟匹配基础的双向模拟匹配更能很好地权衡模拟匹配结果的有效性和及时性,同时获得与模式图结构一致性相对更高的匹配结果,从而有效弥补了基于连接的匹配技术和基于探索的匹配技术的缺陷.

3 面向内容变化的动态图匹配技术

除了面向拓扑结构变化的动态图模式匹配技术之外,面向内容变化的动态图模式匹配技术也得到了广泛应用.例如:基于该技术,可以在频繁更新的数据中心网络[40]中找到满足用户需求的服务器部署方案;可以在频繁变化的社交网络中帮助广告商找到互相联系紧密的人员团体以便于投放广告等.这些应用中,节点或者边的属性会随时间发生频繁变化.目前,对面向内容变化的动态图模式匹配技术的研究工作相对较少,下面对几个典型的研究工作进行介绍.

以数据中心网络中的资源分配为应用背景,文献[16]提出了 Gradin 算法解决了用户按需定制的资源模式图和拓扑结构相对稳定、节点或边的属性频繁变化的数据中心网络图之间的动态匹配问题.在数据中心网络图中,节点和边的属性分别表示随时间频繁变化的服务器的剩余内存值和网络的有效带宽值.匹配过程包含以下两个阶段.

- 离线索引构建

在离线索引构建的过程中,考虑到图的拓扑结构固定不变,可以采用基于连接的匹配技术在数据图中挖掘具有辨识力的频繁子图作为索引元素[30],然后基于这些频繁子图在数据图上构建倒排索引.同时,将模式图基于这些频繁子图进行分解,得到若干模式图分片,便于后续基于倒排索引进行检索和匹配.

- 在线查询

由于数据中心网络图的拓扑结构相对稳定,而节点或边的属性则会发生频繁变化,因此可以采用网格索引(grid index)结构对数据图中的频繁子图构建索引.具体而言,Gradin 算法将带有节点/边的属性值的频繁子图转化为多维向量,然后将多维向量映射至多维网格结构当中.如图 12 所示:s_1 表示一个从数据图中挖掘得到的频繁子图,将数据图中所有与 s_1 结构相同的子图基于由它们两个节点的属性值构成的二维向量可映射至二维网格索引中.如图 12 中所示,网格索引中的每个圆点均表示数据图中与 s_1 结构相同的子图.同时,也按此原理将模式图中与 s_1 结构相同的子模式图映射至该网格索引中,如图 12 中的方点所示.由于模式图和数据图的子图进行匹配时要求它们之间相应节点的属性值满足偏序关系,因此,根据方点位置可以将该二维网格划分为 3 个区域:落在阴影区域 R_1 的圆点所对应的数据图子图肯定满足子模式图的匹配要求;落在阴影区域 R_3 的圆点所对应的数据图子图肯定不满足子模式图的匹配要求,因此只需对落在阴影区域 R_1 和阴影区域 R_3 之间的区域(即 R_2 区域)内的圆点所对应的数据图子图进行匹配计算,即可得到匹配结果.Gradin 算法平均需要进行 $\frac{dn_S}{\lambda^d}\left(\frac{\lambda+1}{2}\right)^{d-1}$

次匹配,其中,λ为网格索引的密度,d 为网格索引的维度,n_s 表示离线索引构建过程中模式图分片的数目.

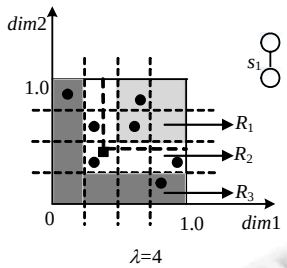


Fig.12 Mapping a frequent subgraph s_1 derived from the data graph to the grid index

图 12 将数据图中的频繁子图 s_1 映射至网格索引中

当数据图的节点或边的属性值遭遇频繁更新时,网格索引比其他索引结构的维护代价更低,平均每一次图更新只需对网格索引中的数据图分片进行 $2(1-1/\lambda^d)$次操作.Gradin 算法也可以直接运用于数据图中边的属性值频繁更新的情况.除了采用数据图中的频繁子图对模式图进行分解,也可以采用目前学术界通用的高等级子图结构对模式图进行分解.文献[41]设计了 CASQD 系统,该系统整合了基于探索的图匹配技术和基于连接的图匹配技术,并选择团、星、双边团等这些具有高识别能力的结构对模式图进行分解.这样设计的好处是不需要对数据图进行频繁子图挖掘,且高等级结构可以并入到任何支持子图模式匹配的查询语言当中.

4 动态图模式匹配算法的性能比较

4.1 面向拓扑结构变化的动态图匹配算法性能比较

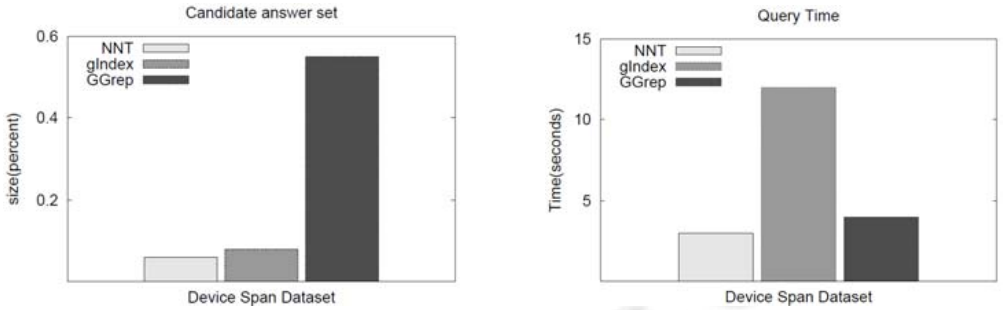
近年来,动态图匹配技术的应用得到了学术界和工业界的广泛关注.越来越多的动态图匹配算法应运而生,不同类型的算法有不同的评价标准.按照第 2 节的分类方式,将基于拓扑结构变化的动态图匹配技术分为基于快照技术的匹配和基于增量技术的匹配两类,本节将对这两类算法的性能进行介绍.

4.1.1 基于快照技术的匹配算法性能比较

算法的性能可以通过有效性和效率两方面进行评价.文献[4]将提出的 NNT 算法与 gIndex[30]以及 GraphGrep[31]算法进行了比较,采用真实数据与合成数据作为数据集,其中,真实数据来自 MIT 媒体实验室的手机通信信息[42],使用 2004 年 1 月~2005 年 5 月间 97 个固定人员的手机通信信息的子集作为实验的数据集,总共产生了由 300 个图快照组成图流,每一个快照代表当前时刻的动态图状态.合成数据用图生成工具[43]产生.实验结果表明:

- 在有效性方面(如图 13(a)所示),GraphGrep 算法会将 50%以上匹配的结果对作为候选集,剪枝能力差,有效性偏低;而 NNT 算法和 gIndex 算法的有效性相差不大,其中,NNT 算法会产生低于 6%的候选集,gIndex 算法的候选集则在 6%~10%之间;
- 在效率方面(如图 13(b)所示),gIndex 算法的查询时间则明显高于另外两种算法.这是因为 gIndex 算法采用的是频繁子图挖掘的方法,虽然使得匹配结果的有效性高,但是在每一个快照都需要挖掘数据图的频繁子图,因此耗费了大量的时间.而 NNT 算法与 GraphGrep 则不需要挖掘子图特征,因此算法效率较高.

与运用在静态图上基于连接的匹配技术(gIndex 算法等)相比,基于快照的动态图匹配可以从两方面进行改进:第一,设计有效的索引结构,可以高效地处理数据图的更新,同时利用索引的剪枝能力来有效约减匹配候选集的规模;第二,采用近似算法,牺牲部分准确率来换取更高的匹配效率.



(a) 算法有效性对比 (b) 算法执行效率对比

Fig.13 Comparison of effectiveness and efficiency

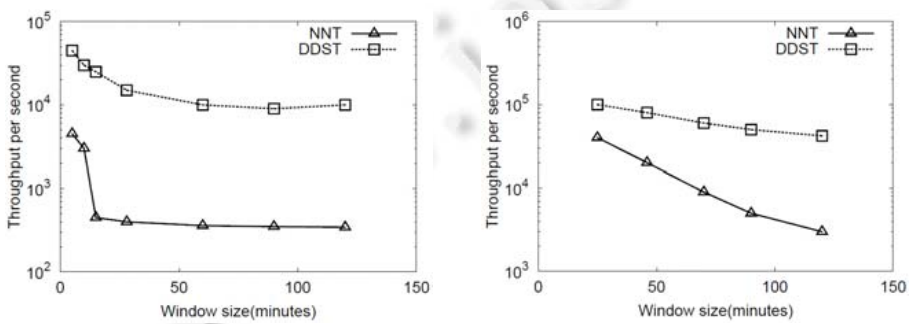
图 13 算法有效性以及效率比较

文献[15]介绍了动态图的真实数据集,包括路网交通数据、电话通信数据、专利引用数据以及推特数据,并对数据集的最大入度/出度、增加边的平均时间间隔、节点以及边的数目进行了描述,表 3 对这些数据进行了详细统计.同时,将提出的 DDST 算法与 NNT 算法进行了比较,两者采用的模型不同,NNT 算法没有考虑窗口下的时间约束关系,采用的是基于同构匹配问题的近似算法,得到的匹配结果中存在错误的结果.而 DDST 算法不仅考虑了数据图更新边上的时间约束关系,且采用的是基于模拟匹配问题的精确算法,结果不存在准确率的问题.由于算法的性能需要在相同的实验环境下进行对比,因此在比较的过程中,需要忽略时间上的约束关系以及结果的准确率信息,仅从吞吐率方面对算法的效率进行对比,吞吐率表示算法每秒能处理的数据图边的数目,可以衡量算法的效率.如图 14 所示,实验结果表明:无论是在 Road 数据集还是 Phone 数据集,DDST 算法的吞吐率都明显高于 NNT 算法;同时,随着窗口范围的增大,NNT 算法吞吐率下降的幅度相对更大.这是因为 NNT 算法在索引的维护上要花费大量的开销,对数据的增加比较敏感,且采用同构匹配的方式,限制性强.而 DDST 则采用的是图模拟的方法,因此,DDST 在吞吐率方面(效率)更好一些.

Table 3 Statistics on the datasets

表 3 动态图数据集统计

	out_deg	in_deg	inter_arrival	# edges	# nodes
Road	3 336	2 224	0.48 sec	686 104	605
Phone	1 725	1 661	3 min	52 050	6 809
Patent	770	779	0.79 min	16 522 438	3 774 768
Twitter	308 636	10 997	6.7 ms	495 544 069	34 664 679



(a) 改变窗口大小 *w*(road 数据集) (b) 改变窗口大小 *w*(phone 数据集)

Fig.14 Comparison of throughput

图 14 算法吞吐率比较

综上,为了满足实时性的需求,基于快照的匹配技术可以从两方面进行设计.

- 若采用同构匹配,需要设计高效的索引结构,在数据图更新的过程中,不会产生过高的索引维护代价;同时,需要设计一个近似算法来提升算法的效率;
- 若采用模拟匹配,可以在图模拟的基础上加上一些限制条件,更加符合实际应用的需求;同时,设计基于该模拟匹配二元关系的精确算法来提升算法的匹配效率.

需要注意的是:同构匹配和模拟匹配的选择,需要根据具体的应用背景来决定.

4.1.2 基于增量技术的匹配算法性能比较

随着增量技术在动态图匹配中出现,越来越多的研究工作开始围绕着基于增量技术的动态图匹配进行,而增量技术也更加适合实时更新的图数据.基于增量技术的匹配算法可以分为基于连接、基于探索以及基于图模拟匹配这 3 类,下面分别对这 3 类动态图匹配算法性能进行分析比较.

(1) 基于连接的匹配技术

文献[14]将 SJ-Tree 与 IncIsoMatch 算法进行了比较,采用真实的 2011 年 8 月~10 月的《纽约时报》数据集,包含 39 523 的节点、68 682 条边.并用递增的方式来处理 1 000 条《纽约时报》更新的数据,将处理的时间作为评价指标.如图 15 所示,实验结果表明:使用 SJ-Tree 进行查询的时间更低,速度更快,而且这种性能差距会随着数据图的增加变得更大.因为增量匹配技术在匹配的过程中需要在每一个新到来的边的周围进行查找.若采用 IncIsoMatch 算法,当一条新的边到来时,要查找该边端点周围所有 k 跳范围内的节点,其中,k 表示模式图的直径.当数据图比较密集时,k 跳范围内的子图将会累积大量的边,从而使查找的代价更加昂贵,查询速率更慢.而 SJ-Tree 树这种基于查询分解的方法将一个大的匹配转化成一系列小的匹配,从而使算法的效率更高.

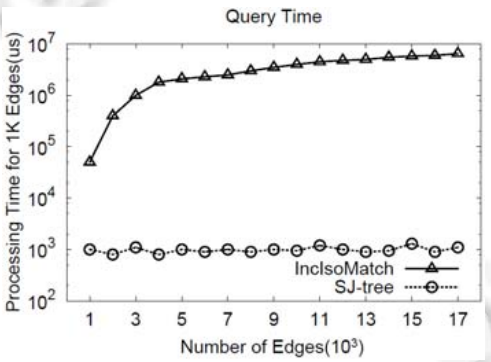


Fig.15 Comparison of IncIsoMatch and SJ-tree

图 15 IncIsoMatch 和 SJ-tree 的性能比较

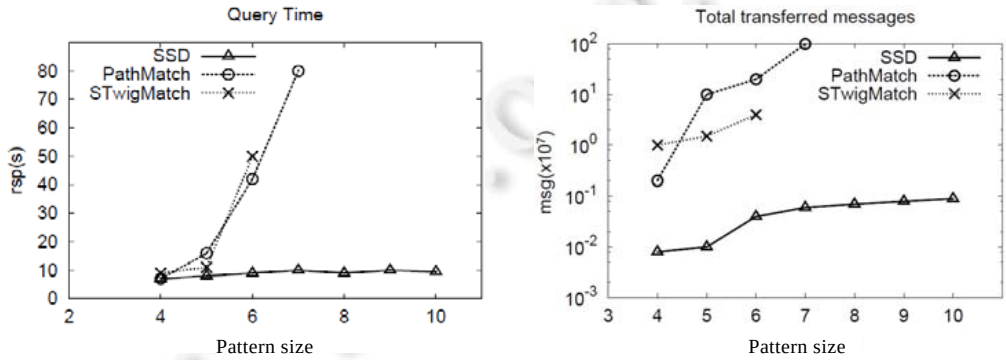
文献[25]在 SJ-Tree 的基础上提出了一种基于选择度的方法 Lazy-Search,首先选取具有识别能力的结构进行过滤,减少查找空间.实验结果表明:在查询的过程中,子图同构操作占据了总查询时间的 95%.因此,减少子图同构的次数,能够有效增加算法的性能,其查找速度是 VF2 算法[44]的 10 倍~100 倍.

从上述的实验结果可以看出:在进行增量匹配时,如果将整个模式图进行匹配效率很低,尤其对密集的数据图.因此,可以采用将模式图分解的方法,但是这种方法会产生很多无效的中间结果.同时,基于连接的匹配技术采用的是子图同构的核实方法,无效的中间结果会导致匹配效率降低.因此,可以选择特征结构进行查询分解,从而过滤无效的中间结果,加快匹配速率.

(2) 基于探索的匹配技术

文献[18]在分布式的平台 Giraph[8]上进行实验,将 SSD 算法与文献[19]的工作以及 PathMatch 方法进行比较,并采用真实数据集 livejournal.livejournal 是综合型 SNS 交友网站,有 4 847 571 个节点、68 993 773 条边.其中,文献[19]采用的是基于 STwig 的分解框架以及基于探索的匹配方法,用 STwigMatch 表示;而 PathMatch 方法采用精确的基于探索的匹配方法.实验以查询响应时间以及总的传递信息作为评价标准.如图 16 所示,实验结

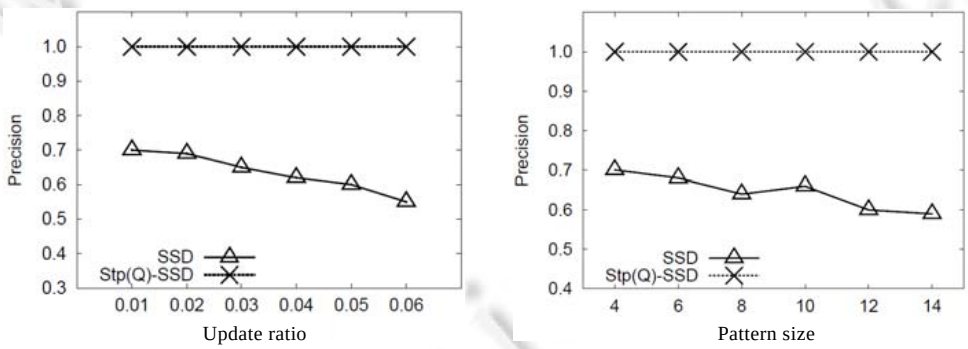
果表明:SSD 算法与 STwigMatch 方法的性能要整体优于 PathMatch 方法,且 SSD 算法需要传递的信息最少.由于 PathMatch 方法与 STwigMatch 方法传递信息巨大,导致耗费了大量的时间;而 SSD 算法采用 DAG 分解的方式,可以有效减少消息的传递,降低查询响应时间.除此之外,无论是 STwigMatch 方法还是 PathMatch 算法,在测试使用的数据集中都无法处理大小超过 7 的模式图,否则会超出内存的限制,导致查询过程很慢.文献[23]在 SSD 的基础上提出了 Stp(Q),解决了 SSD 算法在数据图变化过程中每一个超级步结果的不一致性问题.如图 17 所示,其中,Update ratio 表示数据图的更新比率,例如:0.03 表示数据图中有 3%新加入的边,同时有 3%的边被移除.实验结果表明:SSD 算法的准确率会随着数据图更新程度的增加或者模式图的增大相应降低;而 SSD 算法采用 Stp(Q)之后,准确率可以一直保证达到 100%,有效保证了结果的准确率.



(a) 算法执行效率对比 (b) 算法信息传递量对比

Fig.16 Comparison of SSD and other methods

图 16 SSD 与其他方法的性能比较



(a) 改变数据图更新的比率 (b) 改变模式图的大小

Fig.17 Precision of SSD and Stp(Q)-SSD

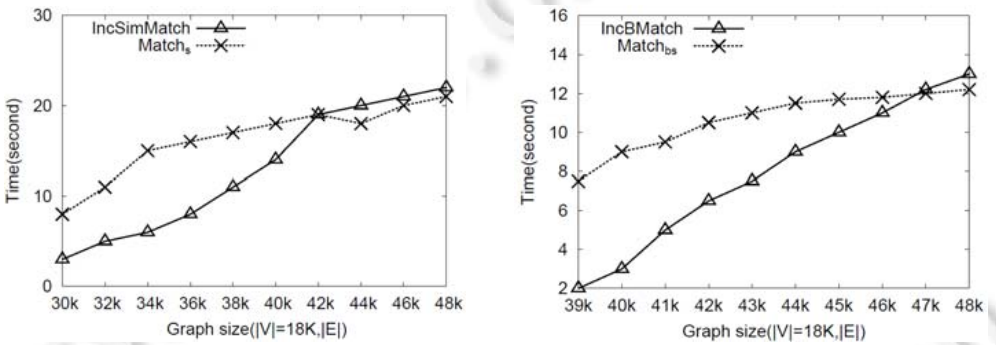
图 17 SSD 和 Stp(Q)-SSD 的正确率对比

从实验结果可以看出,仅仅使用基于探索的匹配方式性能很差.因此,可以将基于连接的方法与基于探索的方法相结合,采用基于连接方法的查询分解框架,在匹配的过程中采用基于探索的方法.如何有效地利用信息的传递方式对模式图分解,是这种方法的关键部分,好的分解方式可以使总的信息传递量减少,从而提高查询效率.同时,也需要设计高效的算法来提高结果的准确率.

(3) 基于模拟的匹配技术

文献[13]将 IncSimMatch 算法、IncBMatch 算法和对应的批处理算法 $Match_s$ 与 $Match_{bs}$ 进行比较,采用真实的 YouTube 和 citation network 数据集,其中:YouTube 有 14 829 个节点、58 901 条边,每一个节点代表一个视频

对象,记录了视频的长度、种类等属性;citation network 有 17 292 个节点、61 351 条边,每一个节点代表一篇论文对象,记录了论文的题目、作者和发表年份这几个属性.如图 18 所示,实验结果表明:对于 IncSimMatch 算法(如图 18(a)所示),当一次边的增加或删除更新所涉及边的数目不超过数据图边总数的 40%时,增量算法要优于批处理算法.对于 IncBMatch 算法(如图(18b)所示),当一次边的增加或删除更新所涉及边的数目不超过数据图边总数的 20%时,增量算法要优于批处理算法.当数据图更新部分的大小相同时,IncBMatch 算法中增加边操作比删除边操作处理的时间要长很多.文献[39]将提出的剪枝算法 D-IDS 与 IncBMatch 算法进行比较,IncBMatch 通过受限模拟的方式对数据图进行剪枝,但产生的结果与模式图结构一致性较低,而 D-IDS 算法通过二元模拟匹配则会产生与模式图更加一致的结果.实验结果表明,D-IDS 算法平均可以将数据图的大小减小 60%.对于直径较小的数据图,性能提升较高.对于直径较大的数据图,效率也会有所提高,但不够明显.文献[38]将分布式的增量算法与批处理算法进行比较,实验结果表明:在第 12 次更新之后,采用增量算法的匹配速率是批处理算法的 3 倍~10 倍,并且可以过滤掉 60%以上的无效更新.



(a) 基于 YouTube 数据集测试 IncSimMatch 算法 (b) 基于 YouTube 数据集测试 IncBMatch 算法

Fig.18 Incremental match vs. batch counterpart

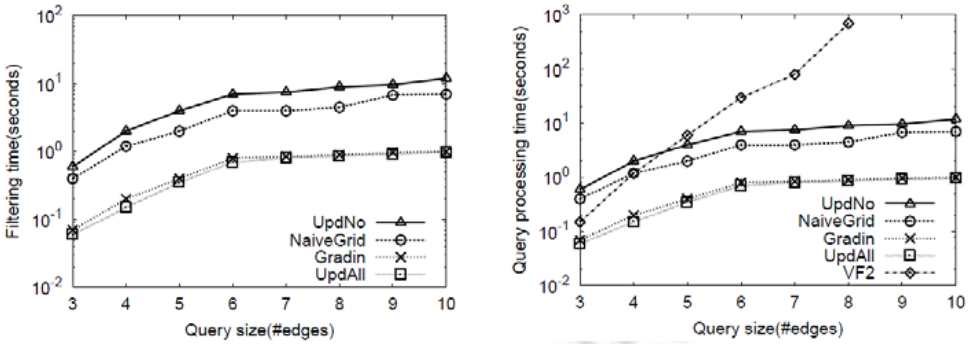
图 18 增量匹配和批处理对比

综上,基于增量技术的动态图模式匹配研究越来越广泛,一般从 3 个方面考虑算法的性能提升:第一是设计增量算法,充分利用之前的匹配结果过滤掉无效的更新;第二是设计高效的增量子图匹配算法,在新增加的边周围找到可能是匹配结果的候选集;第三是设计一种剪枝策略,进一步在匹配的候选集中过滤掉无效的结果.

针对不同的应用背景,也需要采取合适的方法.从准确率方面来说,基于连接的方法可以获得精确的匹配结果,适合处理对图的拓扑结构要求比较严格的应用.从效率方面来说,基于连接的匹配技术会产生大量的中间结果,效率很低.而基于探索的方法会提高匹配效率,但是准确率不高,因此可以采用两者相结合的匹配方式.特别地,针对那些对拓扑结构要求不是严格的应用,可以采用基于图模拟的匹配方式,而一些扩展的图模拟匹配则可以获得与模式图结果一致性更高的匹配结果,从而弥补基于连接与基于探索方法的不足.

4.2 面向内容变化的动态图匹配算法性能比较

基于内容变化的动态图匹配算法一般从查询时间、可扩展性方面对算法的性能进行比较.文献[16]将 Gradin 与 VF2,UpdAll,NaiveGrid 以及 UpdNo 算法进行比较,其中,VF2 是无索引的子图匹配算法,UpdAll 将查询分片部署到多维查找树中,NaiveGrid 是指使用传统核实算法的网格索引,而 UpdNo 则采用倒排索引.实验采用真实的数据集 BCUBE,BCUBE 是数据中心的一个网络体系结构,选择其中的 3 000 个节点作为实验的数据集.如图 19 所示,实验结果表明:Gradin 算法的剪枝速率、查询时间与 UpdAll 算法相似,但是索引构建时间是 UpdAll 算法的 4 倍~10 倍;同时,Gradin 算法的剪枝速率是 UpdNo 算法的 10 倍,是 NaiveGrid 算法的 5 倍之多.Gradin 算法采用基于连接的匹配方式,可以加速图匹配过程,使得其查找速度最快.



(a) 算法过滤时间对比 (b) 算法执行效率对比

Fig.19 Comparison of Gradin and other methods

图 19 Gradin 与其他方法的性能比较

目前面向内容变化的动态图模式匹配研究相对较少,主要用于数据中心方面.一般从两方面进行考虑:第一,对于属性更新不是很频繁的应用可以采用基于连接的匹配技术,类似于静态图匹配;第二,对于属性频繁更新的应用,如果采用基于连接的匹配方式则会不断产生大量的中间结果,因此可以采用基于探索的匹配方式.

5 动态图匹配技术的应用分析

近年来,图数据广泛地应用于刻画现实世界中各类实体间的复杂关系,且这种复杂的关系在现实世界中时刻经历着变化.因此,与传统的静态图模式匹配技术相比,动态图模式匹配技术的应用场景更加广泛.本节对动态图数据匹配技术的应用现状进行了总结.

(1) 犯罪行为分析

目前,毒品交易、恐怖袭击事件等犯罪行为给国家造成了严重的危害.可建立以人物为节点、活动关系为边的行为关系图作为数据图,同时定义犯罪团体行为关系图作为模式图,从而利用模式图和动态数据图之间的模式匹配技术发现和预测潜在的犯罪团伙.

例如,文献[13]采用面向动态图的受限模拟匹配技术分析有潜在毒品贩卖行为团伙.如图 20 所示:P_0 为模式图,图中的节点表示犯罪人员,边表示他们之间的贩卖关系,边(*AM*,*FW*)上的边属性值为 3,意味着从 *AM* 这个人到 *FW* 这个人需要满足在 3 跳之内贩卖约束.基于该模式图,在大规模动态变化的行为关系图中进行匹配查找,即可及时发现潜在的毒品贩卖关系团伙,为犯罪案件的侦破提供有力依据.据悉,美国著名的情报公司 Palantir 还利用该技术完成了特定场景的侦查分析任务.

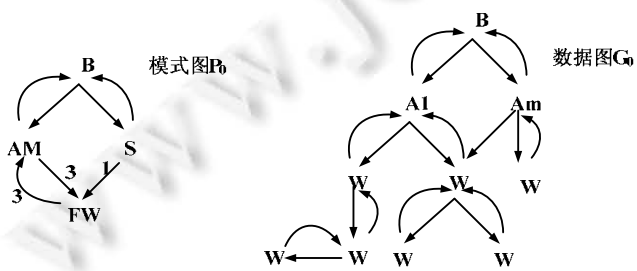


Fig.20 Potential drug trafficking social network analysis based on dynamic graph pattern matching

图 20 基于动态图模式匹配的潜在贩毒品团伙分析

(2) 路网监测

城市的道路交通监测是动态图模式匹配技术的重要应用.城市路网可以表示为图的形式,其中,节点表示路

口,边表示路段.同时,用户可以根据交通事故发生后周围的典型路况定义模式图,将该模式图与动态变化的路网数据图进行匹配,以完成对交通事故的实时监测.以文献[15]中的路网检测为例,如图 21 所示,其中,图 21(a)表示交通事故发生后周围的典型路况,图中边的属性表示该边所对应的道路的拥堵情况(cong.表示拥堵,smooth.表示畅行).在匹配的过程中,需要满足时间上的偏序关系,这种时间上的偏序关系可以定义成如图 21(b)所示的时间关系图,图中的每一个节点对应图 21(a)中的每一条边.模式图和数据图的相应子图之间只有满足图 21(b)定义的时间域上的先后顺序,才能成为最终匹配结果.每一次的道路交通拥塞报告都可以看做是对路网图数据的更新,且更新的部分需要在如图 21(c)所示的时间窗口范围内才认为有效.图 21(d)和图 21(e)分别表示路网数据图在不同时间点上的两个快照,图中括号内的数字表示时间.可以将图 21(a)所示的交通事故发生模式图和不同时刻动态变化的路网数据图(例如图 21(d)和图 21(e))进行模式匹配,完成对交通事故的实时检测.此例中,图 21(e)存在与模式图相匹配的子图.

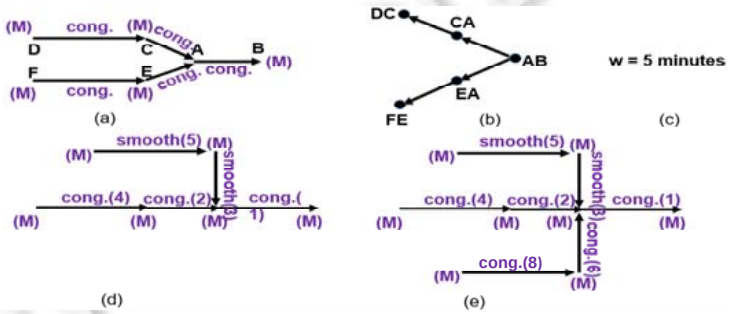


Fig.21 Traffic accident monitoring based on dynamic graph pattern matching

图 21 基于动态图模式匹配的交通事故监控

(3) 网络安全监测

随着互联网行业的快速发展,网络安全问题越发突出.以文献[25]中的网络攻击监测为例,图 22 展示了 3 个用户定义的网络攻击行为模式图,图中的节点表示主机,边表示主机间的通信、用户登录等交互关系.基于动态图模式匹配技术将网络攻击行为模式图和动态变化的网络图进行模式匹配,可及时发现网络中的网络攻击事件或潜在网络攻击事件,达到对网络攻击行为进行检测甚至预测的效果.

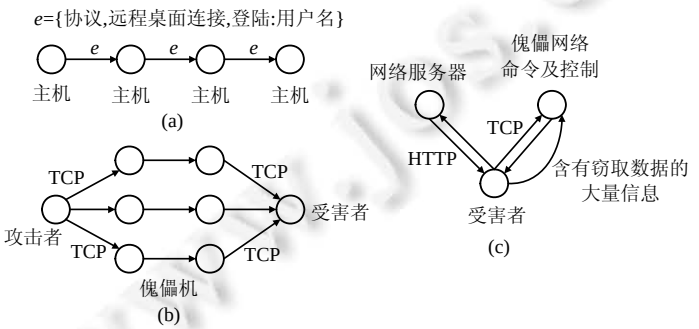


Fig.22 Cyber attack detection based on dynamic graph pattern matching

图 22 基于动态图模式匹配的网络攻击行为检测

(4) 计算生物学数据分析

生物学中研究的分子结构数据可以表示为图的形式.在蛋白质交互作用网络(protein-interaction network)中,蛋白质可能会与某种酶发生反应导致变异.通过将已知性质的蛋白质网络和动态变化的蛋白质交互作用网络进行匹配,可以从蛋白质交互作用网络中快速地找到发生变异的蛋白质结构,为研究生物组织的结构及功能

提供重要依据.例如,文献[45]在对包含 1.8 万个蛋白质(节点)和 4.4 万种蛋白质之间的相互关系(边)的蛋白质交互作用网络进行研究的过程中,通过计算蛋白质交互作用网络的连通性、中心性以及匹配已知性质的结构,发现介数中心性(betweenness)大且连联通性小的蛋白质在人类基因中是冗余的蛋白质,这无疑为分析基因和蛋白质功能提供了重要的依据.

6 结论与展望

动态图作为目前广泛使用的数据模型,对动态图数据匹配方面的研究具有十分重要的理论意义和应用前景,引起了学术和工业界越来越多的关注.随着大数据时代的到来,数据规模的激增,数据更新越发频繁,数据之间的关系越发复杂,这无疑给动态图数据匹配技术的研究和应用提出了新的挑战,也产生了新的机遇,可以从以下几个方面开展研究.

- 面对图数据规模激增的现状,可以研究如何基于主流的分布式并行图处理框架系统,例如谷歌的 Pregel 和 Apache 的 Giraph,来实现图匹配的并行分布式处理,提高匹配技术应对大规模图数据的可扩展性.这些主流的分布式并行图处理框架系统通常采用以节点为中心的处理模式.以节点为中心的处理模式的基本计算单元是节点,其将数据图进行划分,由计算框架中的每个节点存储和处理一个分区的图数据.文献[29]还提出了一种动态图处理框架 BLADYG,提出用以块为中心的分布式并行计算模型来处理大规模数据图.中心块计算模型的基本计算单元是块,每一个块存储和处理的是数据图的一个连通子图.这两种分布式并行图处理模型各有优劣,研究人员可以根据应用需求灵活选择;
- 面对图数据更新越发频繁的现状,采用快照的匹配方法很难满足动态图匹配的实时性需求,因此需要研究高效的面向动态图特点的匹配方式和查询优化策略.当前,研究者可以从增量算法设计和候选集剪枝两方面进行研究,从而缩小查询范围,提高匹配效率;
- 基于连接的匹配技术是实现动态图模式匹配的有效手段,从模式图或数据图中提取优质的特征,是该技术面临的核心问题之一.为了避免发生数据图更新导致特征失效的情况,目前的研究工作一般选择单边子图或双边子图这类简单子图作为数据图的特征.然而,这类简单子图的可识别特性不强,导致中间匹配结果激增,加重了子图连接阶段的处理代价.因而,如何提取更加有识别度的特征,同时设计高效的特征维护方案,是当前研究的主要问题.另一方面,基于探索的动态图匹配方式有其特有的优势,但却面临着匹配结果准确率偏低的问题,可以研究基于探索的动态图匹配技术与基于连接的动态图匹配技术相结合的新型处理方案,重点考虑结合过程中如何发挥各自技术类型的优势,从而弥补各自技术类型的缺憾.基于模拟的匹配技术是当前研究的热点,它避免了复杂的子图同构匹配计算,对模拟匹配技术进行扩展研究,将是另一个研究关注点.另外,设计面向应用需求的图匹配相似度度量模型、开发更有效的图匹配近似算法,也是未来的研究方向;
- 目前,大部分研究涉及的动态图数据匹配问题都是针对模式图不变、数据图随时间动态变化的情况,然而在实际生活中,模式图发生变化的情况也非常普遍.例如:网络安全领域,病毒经常发生变种,网络攻击模式也在不断发生演变;在计算生物学领域,蛋白质变性、流感病毒变异也时常发生.文献[38]研究了数据图不变、模式图发生动态变化时的图模式匹配增量算法.研究者们一方面可以开拓研究这方面的增量处理技术;另一方面,也可以将模式图变化与数据图变化结合起来,拓展动态图数据匹配技术的应用领域.

References:

[1] Gong NZ, Xu W, He Y, Huang L, Mittal P, Stefanov E, Sekar V, Song D. Evolution of social-attribute networks: Measurements, modeling, and implications using Google+. In: Proc. of the 2012 ACM Internet Measurement Conf. Libraries: ACM Press, 2012. 131−144. [doi: 10.1145/2398776.2398792]

[2] Fan W, Wang X, Wu Y. Fan W, Wang X, Wu Y. ExpFinder: Finding experts by graph pattern matching. In: Proc. of the 29th Int'l Conf. on Data Engineering. Brisbane: IEEE Computer Society, 2013. 1316−1319. [doi: 10.1109/ICDE.2013.6544933]

[3] Choudhury S, Holder L, Chin G, Ray A, Beus S, Feo J. StreamWorks: A system for dynamic graph search. In: Proc. of the 2013 ACM SIGMOD Int'l Conf. on Management of Data. NewYork: ACM Press, 2013. 1101−1104. [doi: 10.1145/2463676.2463697]

[4] Yu J, Liu YB, Zhang Y, Liu MY, Tan JL, Guo L. Survey on large-scale graph pattern matching. Journal of Computer Research and Development, 2015,52(2):391−409 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2015.20140188]

[5] Lee J, Han WS, Kasperovics R, Lee JH. An in-depth comparison of subgraph isomorphism algorithms in graph databases. Proc. of the VLDB Endowment, 2013,6(2):133−144. [doi: 10.14778/2535568.2448946]

[6] Malewicz G, Austern MH, Bik AJC, Dehnert JC, Horn I, Leiser N. Pregel: A system for large-scale graph processing. In: Proc. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data. Indianapolis: ACM Press, 2010,18(18):135−146. [doi: 10.1145/1807167.1807184]

[7] Shao B, Wang H, Li Y. Trinity: A distributed graph engine on a memory cloud. In: Proc. of the 2013 ACM SIGMOD Int'l Conf. on Management of Data. NewYork: ACM Press, 2013. 505−516. [doi: 10.1145/2463676.2467799]

[8] Apache girph. 2017. http://incubator.apache.org/giraph

[9] Yan D, Cheng J, Lu Y, Ng W. Blogel: A block-centric framework for distributed computation on real-world graphs. Proc. of the VLDB Endowment, 2014,7(14):1981−1992. [doi: 10.14778/2733085.2733103]

[10] Fan WF, Xu JB, Wu YH, Yu WY, Jiang JX, Zheng ZY, Zhang B, Cao Y, Tian C. Parallelizing sequential graph computations. In: Proc. of the 2017 ACM SIGMOD Int'l Conf. on Management of Data. Chicago: ACM Press, 2017. 495−510. [doi: 10.1145/3035918. 3035942]

[11] Postech Database Lab. NASA, Yeast, and Human datasets. 2014. http://dtp.nci.nih.gov/docs/aids/aids_data.html

[12] Wang C, Chen L. Continuous subgraph pattern search over graph streams. In: Proc. of the 25th Int'l Conf. on Data Engineering. Shanghai: IEEE Computer Society, 2009. 393−404. [doi: 10.1109/ICDE.2009.132]

[13] Fan WF, Li JZ, Luo JZ, Tan ZJ, Wang X, Wu YH. Incremental graph pattern matching. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data. Athens: ACM Press, 2011,38(3):925−936. [doi: 10.1145/2489791]

[14] Choudhury S, Holder L, Chin G, Feo J,. In: Proc. of the Workshop on Dynamic Networks Management and Mining. NewYork: ACM Press, 2013. 1−8. [doi: 10.1145/2489247.2489251]

[15] Song C, Ge T, Chen C, Wang J. Event pattern matching over graph streams. Proc. of the VLDB Endowment, 2014,8(4):413−424. [doi: 10.14778/2735496.2735504]

[16] Zong B, Raghavendra R, Srivatsa M, Yan X, Singh AK, Lee KW. Cloud service placement via subgraph matching. In: Proc. of the 30th Int'l Conf. on Data Engineering. Chicago: IEEE Computer Society, 2014. 832−843. [doi: 10.1109/ICDE.2014.6816704]

[17] Mondal J, Deshpande A. Managing large dynamic graphs efficiently. In: Proc. of the 2012 ACM SIGMOD Int'l Conf. on Management of Data. Scottsdale: ACM Press, 2012. 145−156. [doi: 10.1145/2213836.2213854]

[18] Gao J, Zhou C, Zhou J, Yu JX. Continuous pattern detection over billion-edge graph using distributed framework. In: Proc. of the 30th Int'l Conf. on Data Engineering. Chicago: IEEE Computer Society, 2014. 556−567. [doi: 10.1109/ICDE.2014.6816681]

[19] Sun Z, Wang H, Wang H, Shao B, Li J. Efficient subgraph matching on billion node graphs. Proc. of the VLDB Endowment, 2012, 5(9):788−799. [doi: 10.14778/2311906.2311907]

[20] Low Y, Gonzalez JE, Kyrola A, Bickson D, Guestrin C, Hellerstein J. GraphLab: A new framework for parallel machine learning. https://www.mendeley.com/research-papers/graphlab-new-parallel-framework-machine-learning/

[21] Aridhi S, Montresor A, Velegrakis Y. BLADYG: A graph processing framework for large dynamic graphs. Big Data Reaearch. 2017. [doi: 10.1016/j.bdr.2017.05.003]

[22] Valiant L. A bridging model for parallel computation. Communications of the ACM, 1990,33(8):103−111. [doi: 10.1145/79173.79181]

[23] Gao J, Zhou C, Yu JX. Toward continuous pattern detection over evolving large graph with snapshot isolation. The VLDB Journal, 2016,25(2):269−290. [doi: 10.1007/s00778-015-0416-z]

[24] Khurana U, Deshpande A. Efficient snapshot retrieval over historical graph data. In: Proc. of the 29th Int'l Conf. on Data Engineering. Brisbane: IEEE Computer Society, 2013. 997−1008. [doi: 10.1109/ICDE.2013.6544892]

[25] Choudhury S, Holder L, Chin G, Mackey P, Agarwal K, Feo J. Query optimization for dynamic graphs. Corrsionence & Technology Protection, 2014. http://xueshu.baidu.com/s?wd=paperuri%3A%285125477c2fae3f27651d70c5d9193b26%29&filter=

sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fde.arxiv.org%2Fpdf%2F1407.3745&ie=utf-8&sc_us=2466112609285908482

[26] Kao JS, Chou J. Distributed incremental pattern matching on streaming graphs. In: Proc. of the ACM Workshop on High Performance Graph Processing. Kyoto: ACM Press, 2016. 43−50.

[27] Yang J, Jin W. BR-Index: An indexing structure for subgraph matching in very large dynamic graphs. In: Proc. of the Scientific and Statistical Database Management. 2011. 322−331. [doi: 10.1007/978-3-642-22351-8_20]

[28] Ma S, Cao Y, Fan W, Huai J, Wo T. Capturing topology in graph pattern matching. Proc. of the VLDB Endowment, 2011,5(4): 310−321. [doi: 10.14778/2095686.2095690]

[29] Fan WF, Hu C, Tian C. Incremental graph computations: Doable and undoable. In: Proc. of the 2017 ACM SIGMOD Int'l Conf. on Management of Data. Chicago: ACM Press, 2017. 155−169. [doi: 10.1145/3035918.3035944]

[30] Yan X, Yu PS, Han J. Graph indexing: A frequent structure-based approach. In: Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data. Paris: ACM Press, 2004. 335−346. [doi: 10.1145/1007568.1007607]

[31] Giugno R, Shasha D. GraphGrep: A fast and universal method for querying graphs. In: Proc. of the IEEE Int'l Conf. on Pattern Recognition. Quebec: IEEE Computer Society, 2002. 112−115. [doi: 10.1109/ICPR.2002.1048250]

[32] Zhao P, Han J. On graph query optimization in large networks. Proc. of the VLDB Endowment, 2010,3(1-2):340−351. [doi: 10.14778/1920841.1920887]

[33] Shang H, Zhang Y, Lin X, Yu JX. Taming verification hardness: An efficient algorithm for testing subgraph isomorphism. Proc. of the VLDB Endowment, 2008,1(1):364−375. [doi: 10.14778/1453856.1452899]

[34] Yang J, Zhang S, Jin W. DELTA: Indexing and querying multi-labeled graphs. In: Proc. of the 20th ACM Int'l Conf. on Information and Knowledge Management. Glasgow: ACM Press, 2011. 1765−1774. [doi: 10.1145/2063576.2063832]

[35] Jiang H, Wang H, Yu PS, Zhou S. GString: A novel approach for efficient search in graph databases. In: Proc. of the 23rd Int'l Conf. on Data Engineering. Istanbul: IEEE Computer Society, 2007. 566−575. [doi: 10.1109/ICDE.2007.367902]

[36] Cheng J, Ke Y, Ng W, Lu A. FG-Index: Towards verification-free query processing on graph databases. In: Proc. of the 2007 ACM SIGMOD Int'l Conf. on Management of Data. Beijing: ACM Press, 2007. 857−872. [doi: 10.1145/1247480.1247874]

[37] Choudhury S, Holder L, Chin G, Agarwal K, Feo J. A selectivity based approach to continuous pattern detection in streaming graphs. Computer Science, 2015,93(8):939−945. [doi: 10.5441/002/edbt.2015.15]

[38] Zhang LX, Wang WP, Gao JL, Wang JX. Pattern graph change oriented incremental graph pattern matching. Ruan Jian Xue Bao/ Journal of Software, 2015,26(11):2964−2980 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4891.htm [doi: 10.13328/j.cnki.jos.004891]

[39] Wickramaarachchi C, Kannan R, Chelmis C, Prasanna VK. Distributed exact subgraph matching in small diameter dynamic graphs. In: Proc. of the IEEE Int'l Conf. on Big Data. Washington: IEEE Computer Society, 2017. 3360−3369. [doi: 10.1109/BigData.2016.7840996]

[40] Raghavendra R, Lobo J, Lee KW. Dynamic graph query primitives for SDN-based cloudnetwork management. In: Proc. of the 1st Workshop on Hot Topics in Software Defined Networks. Helsinki: ACM Press, 2012. 97−102. [doi: 10.1145/2342441.2342461]

[41] Mondal J, Deshpande A. CASQD: Continuous detection of activity-based subgraph pattern queries on dynamic graphs. In: Proc. of the 10th ACM Int'l Conf. on Distributed and Event-Based Systems. Irvine: ACM Press, 2016. 226−237. [10.1145/2933267.2933316]

[42] Reality mining dataset. 2017. http://reality.media.mit.edu

[43] Kuramochi M, Karypis G. Frequent subgraph discovery. In: Proc. of the IEEE Int'l Conf. on Data Mining. San Jose: IEEE Computer Society, 2001. 313−320. [doi: 10.1109/ICDM.2001.989534]

[44] Cordella LP, Foggia P, Sansone C, Vento M. A (sub)graph isomorphism algorithm for matching large graphs. IEEE Trans. on Pattern Analysis & Machine Intelligence, 2004,26(10):1367−1372. [doi: 10.1109/TPAMI.2004.75]

[45] Bader DA, Madduri K. A graph-theoretic analysis of the human protein-interaction network using multicore parallel algorithms. Parallel Computing, 2008,34(11):627−639. [doi: 10.1016/j.parco.2008.04.002]

附中文参考文献:

[4] 于静,刘燕兵,张宇,刘梦雅,谭建龙,郭莉.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. [doi: 10.7544/issn 1000-1239.2015.20140188]

[38] 张丽霞,王伟平,高建良,王建新.面向模式图变化的增量图模式匹配.软件学报,2015,26(11):2964−2980. http://www.jos.org.cn/1000-9825/4891.htm [doi: 10.13328/j.cnki.jos.004891]

许嘉(1984－),女,山东荣成人,博士,副教授,CCF 专业会员,主要研究领域为数据库理论与技术,图数据分析,数据隐私保护,大数据分布式并行计算.

张千桢(1992－),男,硕士生,CCF 学生会员,主要研究领域为图数据管理.

赵翔(1986－),男,博士,讲师,CCF 专业会员,主要研究领域为图数据管理与挖掘,基于大数据的情报智能.

吕品(1983－),男,博士,副研究员,CCF 专业会员,主要研究领域为无线网络与移动计算,物联网,网络虚拟化,网络数据分析.

李陶深(1957－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为无线 Mesh 网络,云计算与大数据,网络计算与信息安全,分布式工程数据库.