计算机科学
COMPUTER SCIENCE

频繁量化模式图挖掘及应用

沙雨济, 王欣, 何艳潇, 钟学燕, 方宇

引用本文

沙雨济, 王欣, 何艳潇, 钟学燕, 方宇. 频繁量化模式图挖掘及应用[J]. 计算机科学, 2023, 50(11A): 230100041-12.

SHA Yuji, WANG Xin, HE Yanxiao, ZHONG Xueyan, FANG Yu. Mining and Application of Frequent Patterns with Counting Quantifiers [J]. Computer Science, 2023, 50(11A): 230100041-12.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

邻域双向聚合与全局感知的TKG链接预测模型

Link Prediction Model on Temporal Knowledge Graph Based on Bidirectionally Aggregating Neighborhoods and Global Aware

计算机科学, 2023, 50(8): 177-183. https://doi.org/10.11896/jsjkx.220900061

基于对比预测的自监督动态图表示学习方法

Self-supervised Dynamic Graph Representation Learning Approach Based on Contrastive Prediction

计算机科学, 2023, 50(7): 207-212. https://doi.org/10.11896/jsjkx.220500093

混合曲率空间用于多关系异构知识图谱链接补全

Mixed-curve for Link Completion of Multi-relational Heterogeneous Knowledge Graphs

计算机科学, 2023, 50(4): 172-180. https://doi.org/10.11896/jsjkx.220500135

基于异构网络表征学习的作者学术行为预测

Author's Academic Behavior Prediction Based on Heterogeneous Network Representation Learning

计算机科学, 2022, 49(9): 76-82. https://doi.org/10.11896/jsjkx.210900078

基于无监督集群级的科技论文异质图节点表示学习方法

Scientific Paper Heterogeneous Graph Node Representation Learning Method Based onUnsupervised Clustering Level

计算机科学, 2022, 49(9): 64-69. https://doi.org/10.11896/jsjkx.220500196

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230100041

频繁量化模式图挖掘及应用

沙雨济　王　欣　何艳潇　钟学燕　方　宇
西南石油大学计算机科学学院　成都 610500
(202025000059@stu.swpu.edu.cn)

摘　要　频繁模式挖掘(FPM)是图数据研究领域的一个经典问题,单一大图上的FPM问题近年来受到了更加广泛的关注。该问题被定义为根据用户给定的频率阈值查找在大图(Graph)中频繁出现的所有模式图(Pattern)。近年来,人们见证了FPM在多个领域的广泛应用,例如社交网络分析、欺诈检测等。然而,面对新兴的应用需求,人们需要更具语义表达力的模式图及其挖掘技术。为此,在传统模式图的基础上,首先提出了量化模式图(Quantified Graph Patterns,QGPs)——一类具有计数量词约束的模式图,实现了模式图语义的扩展;其次设计了一种在分布式场景下挖掘QGPs的算法,提出了量化图模式关联规则(Quantified Graph Pattern Association Rules,QGPARs)及其挖掘技术,用于预测(社交)网络中实体之间的潜在联系,然后利用真实图和合成图数据,通过翔实的实验验证了QGPs挖掘算法的计算效率,通过与经典链接预测方法进行对比,发现QGPARs可以取得更高的链接预测准确性;最后通过与传统图模式关联规则(Graph Pattern Association Rules,GPARs)的链接预测结果进行对比,验证了QGPARs与GPARs之间在链接预测结果方面存在显著差异,也进一步验证了QGPARs在链接预测中的有效性。

关键词:量化模式图;频繁模式挖掘;分布式挖掘;量化图模式关联规则;链接预测
中图法分类号　TP311

Mining and Application of Frequent Patterns with Counting Quantifiers

SHA Yuji,WANG Xin,HE Yanxiao,ZHONG Xueyan and FANG Yu
School of Computer Science,Southwest Petroleum University,Chengdu 610500,China

Abstract　Frequent pattern mining(FPM) is a classical problem in graph theory,more attention has been paid on FPM on single large graphs,which is defined as discovering all the pattern graphs Q with occurrence frequencies above a user defined threshold,in a single large graph G. In recent years,people have witnessed wide applications of FPM,such as social network analysis and fraud detection. However,emerging applications keep calling for more expressive pattern graphs along with their mining techniques to capture more complex structures in a large graph. In light of this,we incorporate counting quantifiers in pattern graphs and introduce quantified pattern graphs(QGPs) which are able to express richer semantics. We then develop a distributive algorithm to mine QGPs in parallel. Furthermore,we introduce quantified graph pattern association rules(QGPARs) for linking prediction on large graphs. We conduct experimental studies to validate the computational efficiency of the QGPs mining algorithm by using real-world and synthetic graph data. By comparing with prior link prediction methods,we find that prediction with QGPARs achieves even higher accuracy. Finally,by comparing with the link prediction results of traditional graph pattern association rules,we verify that there is a significant difference between QGPARs and GPARs in terms of link prediction results,and further verify the effectiveness of QGPARs in link prediction.

Keywords　Quantified pattern graph,Frequent pattern mining,Distribute mining,Quantified graph pattern association rules,Link prediction

1　引言

图(Graph)可以方便地模拟现实世界中各实体间复杂的联系,图中节点表示实体,边表示实体之间的联系。随着应用场景的不断丰富,图数据的规模也随之变得更加庞大、结构变得更加复杂。在图数据研究领域中,频繁模式挖掘(Frequent Pattern Mining,FPM)一直是一个核心任务。受应用场景的驱动,FPM问题的研究演化为两个不同的分支,即基于图数据库的频繁模式挖掘和基于单一大图的频繁模式挖掘。近年来,随着大规模社交网络的普及,基于单一大图的频繁模式挖掘受到了更加广泛的关注,也取得了丰硕的研究成果[1]。

然而,随着应用需求的不断变化,人们需要具有更加丰富语义的模式图,从而捕捉图数据中更为复杂的结构关系。在此背景下,研究人员提出了量化模式图,即带有计数量词(Counting Quantifiers,CQs)的模式图[2]。在一阶逻辑中,计数量词一般被表示为普通的数学符号,它的引入可以扩展语义的表达能力,有助于解决更加复杂的问题。下文通过具体实例来进一步理解引入计数量词所带来的语义变化及其应用。

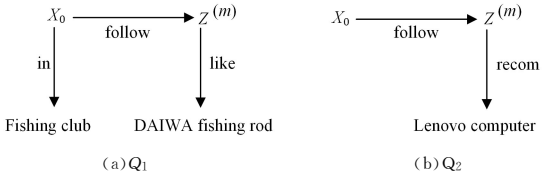
例1　带有计数量词的模式图可以反映社交网络中实体

通信作者:王欣(xinwang@swpu.edu.cn)

之间联系的规律性[2]。图 1 中的两个量化模式图Q_1和Q_2分别表明：

1)如果(i)X_0参加了一个钓鱼俱乐部，并且(ii)在X_0的朋友中至少有 m 个人喜欢并且购买了 DAIWA 品牌的鱼竿，那么X_0也有可能喜欢该品牌的鱼竿。

2)如果X_0的朋友中至少有 m 个人推荐 Lenovo 品牌的电脑，那么X_0有很大的可能去购买该品牌的电脑。



(a)Q_1 (b)Q_2

图 1 量化模式图

Fig. 1 Quantified patterns

不难发现，上述量化模式图通过计数量词 m 实现了量化汇聚，为该模式图在链接预测等问题上带来了独特的优势。以图 1 中的模式图Q_2为例，量词 m 的存在，使得产品推荐的可信度更高，其原因在于，X_0显然更容易接受已经得到 m 个好友推荐的 Lenovo computer。进一步地，信任度的提升将有效地转换为推荐成功率，并取得更好的推荐效果。已有研究表明："90%的顾客信任同伴推荐，而只有 14%的顾客相信传统广告的宣传[3]"，"来自朋友的影响将导致顾客购买产品的概率增加 50%以上[4]"。因此，我们相信，将量化模式图运用到网络营销中，可以取得比传统营销更好的效果。

虽然，量化模式图在社交媒体营销、知识发现等领域有着重要的作用，但截至目前，对于这一类模式图的挖掘、应用等方面的研究尚处于起步阶段。此外，传统的 FPM 挖掘问题没有考虑到计数量词的融入，因此难以通过对已有算法进行简单的改造来挖掘量化模式图。与此同时，随着数据规模的扩大，单一大图往往采用分布式存储的方式；集中式的挖掘计算不仅效率低下，可扩展性差，且往往难以实施。综合以上因素，本文设计了分布式环境下的挖掘算法，实现了量化模式图的高效挖掘；引入了量化图模式关联规则，提出了(社交)网络中实体之间潜在关系预测的新方法。

本文的主要贡献如下：1)对传统模式图进行语义扩展，提出了带有计数量词的量化模式图(Quantified Graph Patterns，QGPs)；2)设计了一种分布式挖掘算法 DisQGPM，高效率地从分布式存储的单一大图中挖掘频繁 QGPs；3)引入了量化图模式关联规则(Quantified Graph Pattern Association Rules，QGPARs)，并利用其在大规模图数据上开展实体关系预测；4)在真实图和合成图数据上验证了算法的性能，并且发现：(1)在大规模图数据上运用 DisQGPM 挖掘频繁 QGPs 是可行的；(2)在关系预测任务上，QGPARs 的平均预测准确率高达 84.3%；(3)QGPARs 的预测结果与传统图模式关联规则(GPARs)的预测结果存在较大差异，相比 GPARs，QGPARs 在链接预测中具有更高的可靠性。

2 相关工作

当前，针对单一大图的频繁模式挖掘问题已经得到较为充分的研究。本文对已有工作进行了分类回顾。

2.1 频繁模式挖掘

频繁模式挖掘是图分析应用中的一个核心问题[5-6]，近年来出现了许多针对大规模数据集的频繁模式挖掘算法。GraMi[7]是一种在单一大图中进行频繁模式挖掘的新框架。GraMi 将子图挖掘问题建模为约束满足问题(Constraint Satisfaction Problem，CSP)，用于评估挖掘所获取的模式图的支持度。在每次迭代期间，它都会求解 CSP，直到找到满足阈值的最小解集。文献[8]提出了一种新的基于 HPC(High Performance Computing)的频繁模式挖掘算法。该算法首先将输入的图根据节点进行分区，然后使用一组操作来最小化节点之间的信息交换。文献[9]提出了 FSSG，其是一种利用图的不变属性和图中存在的对称性来生成候选子图的算法。FSSG 减少了大量候选子图的生成，从而降低了候选生成和频率计算的复杂性。文献[10]介绍了 SOCMI 算法，其核心思想是用 pathgraph 存储模式的外观，这使得 SOCMI 在挖掘时更容易扩展模式和计算频率。文献[11]提出了一种新的 FPM 框架 A-RAFF，通过引入排序度量 FSP-Rank，有效地减少了重复频繁模式的产生。此外，一些基于动态图、加权图等其他类型图的挖掘方法也被广泛研究。文献[12]和文献[13]提出了在加权的单一大图上挖掘加权频繁模式的算法，StreamFSM[14]和 IncGM+[15]是基于动态图的挖掘算法。

2.2 模式图语义扩展

在实践中，传统的模式图不足以对一些复杂的问题进行建模，模式图语义扩展的必要性逐渐显露出来。SPARQLog 使用一阶逻辑(FO)规则扩展了 SPARQL 语言，研究了向 SPARQL 添加规则和量词交替，使其可以识别图中节点的全部量化和存在量化[16]。面向社交网络，SocialScope 和 SNQL 都是对带有数值聚合的节点和边集的图所设计的查询语言[17-18]。此外，研究人员也开展了对社会推荐规则的研究，这些研究引入了数量范围作为约束[19]。

之前的研究表明，现有的扩展仍然具有局限性，在新兴的应用中，需要具有复杂特征的模式，特别是计数量词(CQs)、谓词和否定词，在这些特征中 CQs 得到了更多的关注[20-23]。文献[20]提出了条件图模式(CGP)，它利用计数量词扩展了传统的模式图。QGraph[21]允许用[min，max]形式的 CQs 来注释图形模式的边，这些 CQs 可以表达不同的语义，用户可以以此制定由节点和边组成的查询条件。文献[22]首先提出了量化模式图，通过支持在边上标记简单的计数量词，来实现聚合、存在、普遍以及否定语义，其中计数量词可以表示为"$=p(\%)$"、"$\geqslant p(\%)$"和"$=0$"的形式。文献[22]还引入了一种量化匹配模型，可用于识别社交媒体中的潜在客户。文献[23]提出了一种基于模拟的模式匹配方法，该方法支持图模式边上带有计数量词的情况，使用"[min，max]""≥min"和"≤max"形式的计数量词来注释边，扩展传统模式图。

2.3 并行挖掘计算

随着图数据规模的不断扩大，针对单一大图的分布式 FPM 技术也得到了深入的研究，其中具有代表性的工作如下。SSiGraM[24]是一种面向单一大图的、基于 Spark 的并行频繁模式挖掘算法，能够部署在分布式集群工作站点上并行地进行模式扩展和支持度的评估。文献[25]提出了 DFSME，其是一种使用分布式框架 SPARK 从动态图中发现频繁模式的新方法，它在频繁模式与非频繁模式之间维护了一个模式集，用于减小搜索空间。文献[26]提出了一种在分布式场景下从单个大图中挖掘 top-k 模式的方法，该方法引入

了一种具有提前终止特性的算法，避免了昂贵的时间开销。文献[27]提出了一种基于仿真匹配概念的分布式频繁模式挖掘算法，保证了模式图和数据图之间的点对点匹配，有效避免了冗余的挖掘结果。文献[28]提出了一种使用负载矩阵的新型分布式频繁模式挖掘算法，该算法将数据集垂直分割成多个负载，可用内核进行并行挖掘。PrefixFPM[29]是一个通用的FPM框架，它遵循前缀投影的思想，通过分而治之的方式将FPM的工作负载划分为独立的任务。

与前述工作相比，本文的差异体现在以下方面：本文所提出的量化模式图具有更加丰富的语义，可以有效捕捉传统模式图难以捕捉的匹配，具有良好的应用前景，而相关挖掘技术仍未提出，因此开展这方面的研究必要且迫切。结合分布式技术，设计高效且扩展性强的并行算法是本文的又一特色。

3 基本概念

本章首先回顾传统频繁模式图挖掘问题的基本概念，包括图和子图、分布式图、模式图、图结构匹配、支持度等；随后介绍了带有计数量词的频繁模式挖掘的相关概念。

3.1 频繁模式挖掘

定义 1（图和子图） 图可以表示为 $G=(V,E,L)$，其中 V 表示节点集合，$E\subseteq V\times V$ 是边的集合，L 是节点的标签函数，图中的节点是形如 $(A_1=a_1,\cdots,A_n=a_n)$ 的元组，其中 $A_i=a_i(i\in[1,n])$ 表明节点 v 的 A_i 属性的取值是 a_i，也用 $v.A_i=a_i$ 表示。

给定图 $G=(V,E,L)$ 和图 $G'=(V',E',L')$，如果 $V'\subseteq V$，$E'\subseteq E$，并且对于任意顶点 $v\in V'$，都有 $L'(v)=L(v)$，那么就称 G' 是 G 的一个子图。

定义 2（分布式图） 在实践中，一个大图 G 经常被分割成一系列子图，并存储在不同的站点[30]。给定一个图 $G=(V,E,L)$，它的分割子图集记作 $F=\{F_1,\cdots,F_n\}$，每个分割子图 $F_i=(V_i\cup F_i.O,E_i,L_i)$，其中 1）$(V_1,\cdots,V_n)$ 是 V 的一个划分；2）$F_i.O$ 是节点 v' 的集合，使得 E 中存在一条边 $e=(v',v)$ 或 $e=(v,v')$，$v\subseteq V_i$，v' 属于另一个分割子图；v' 这样的节点被称为虚拟节点（virtual node），e 称为交叉边（crossing edge），cE_i 称为交叉边集；3）$(V_i\cup F_i.O,E_i,L_i)$ 是由节点 $V_i\cup F_i.O$ 组成的 G 的一个诱导子图。

F 中所有的虚拟节点被记作 $V_f=\bigcup_{i\in[1,n]}F_i.O$，所有交叉边记作 E_f，F 中分割图集的数量记作 $|F|$。

定义 3（模式图） 与图的定义相似，模式图被表示为 $Q=(V_p,E_p,L_v)$，其中 V_p 和 E_p 分别是点集和边集；对于任意节点 $u\subseteq V_p$，其标签 $L_v(u)$ 是一组原子公式的连结，其中每个原子公式被定义为“$A=a$”，A 表示节点 u 的一个属性，a 是属性 A 对应的值。通常情况，$L_v(u)$ 表明了节点 u 所表达的查询条件。对于图 2(b)中的模式图 Q_1，其中的节点标签明确了查询条件，如“角色='DBA'”。

例 2 图 2(a)给出了一个社交网络的片段，其中每个节点表示一个用户，并用标签集合（id，姓名，角色）标记。例如，id 为 v_1 的用户姓名为 Denny，角色为 PRG。注意，图中用户的角色共有以下 5 类，分别是 PM（Project Manager），DBA（Database Administrator），PRG（Programmer），BA（Business Analyst）以及 ST（Software Tester）。每条边表示协作关系，例如（Eric，Jason）表示 Eric 和 Jason 之间有协作关系。在图 2(a)中，图 G 被存储在站点 W_1，W_2 和 W_3 中。以 W_1 为例，它不仅保留了本地节点和边，而且还包含了 3 个虚拟节点 v_6，v_7 和 v_8，以及连接这 3 个虚拟节点的交叉边 (v_3,v_6)，(v_3,v_7)，(v_4,v_6)，(v_4,v_7)，(v_4,v_8) 和 (v_5,v_7)。

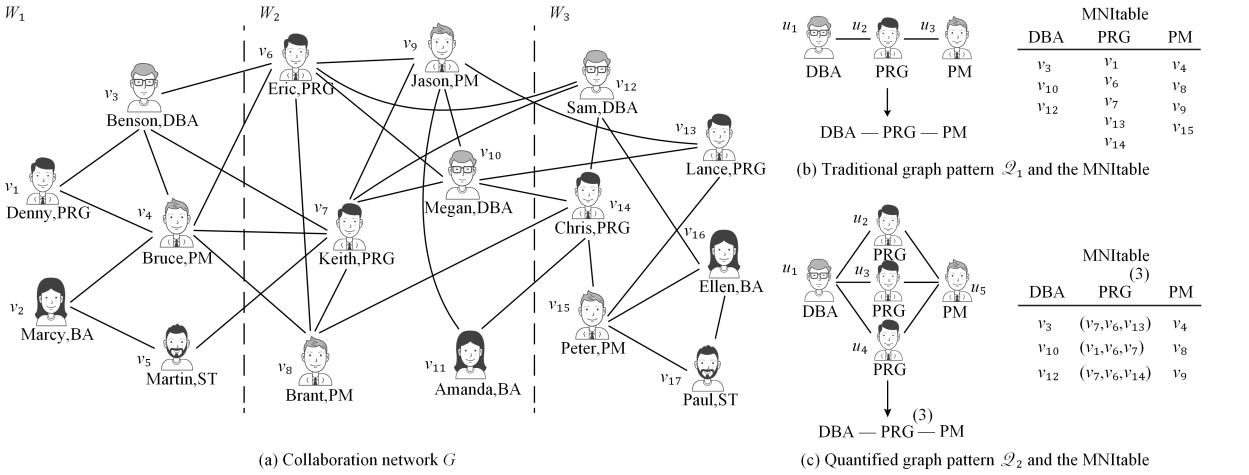


图 2 协作网络图示例

Fig. 2 Example of a collaboration network

定义 4（图结构匹配） 对于图 $G=(V,E,L)$ 和模式图 $Q=(V_p,E_p,L_v)$，如果 G 中节点 v 满足 Q 中节点 u 的查询条件，即：对每一个 $L_v(u)$ 中的原子公式“$A=a$”，在 $L(v)$ 中都有对应的属性 A，使得 $v.A=a$，则称 v 是 u 的匹配，并用“$v\sim u$”表示两者间的匹配关系。

进一步地，图 G 中模式 Q 的“匹配”是一个从 Q 到 G 的同构映射 f，使得：1）对于每个节点 $u\in V_p$，$L_v(u)\sim L(f(u))$；2）对于每个模式边 $(u,u')\in E_p$，当且仅当 $(f(u),f(u'))\in E$。这样当模式 Q 与 G 的子图 $G_s=(V_s,E_s,L_s)$ 存在同构映射关系 f 时，G_s 为 Q 在 G 中的一个匹配。

本文用 $M(Q,G)$ 表示模式图 Q 在图 G 中的所有匹配。

定义 5（MNI 支持度） 支持度表示一个模式图 Q 在图 G 中对应匹配出现的频率，将其记作 $Sup(Q,G)$。假设 ρ 为模式图 $Q=(V_p,E_p,L_v)$ 在 G 中的所有同构映射集合，ρ 集合长度为 m，同时有 $P(u)=\{\rho_1(u),\rho_2(u),\cdots,\rho_m(u)\}$ 为集合 ρ 上每一个 u 去重之后的映射函数，其中 $u\in V_p$，则其 MNI[31] 支持度的计算方式如下：

$$Sup(Q,G)=\min\{|P(u)|,u\in V_p\} \tag{1}$$

不难看出，MNI 支持度将模式图 Q 的支持度定义为 $u\in V_p$ 在图 G 中有效匹配节点数量的最小值。

例 3 如图 2 所示，模式图 Q_1 在图 G 中的同构映射集合通过去重之后得到的结果为：$P(u_1)=\{v_3,v_{10},v_{12}\}$，$P(u_2)=\{v_1,v_6,v_7,v_{13},v_{14}\}$，$P(u_3)=\{v_4,v_8,v_9,v_{15}\}$（如图 2(b)右侧所示），由此可得 Q_1 的支持度为 3。

3.2 量化模式图挖掘

定义 6（量化模式图，QGP） 一个 QGP 被定义为 $Q^m_{u_0}=(V_p,E_p,L_v,f_v)$，其中 V_p 和 E_p 分别是模式节点集和边集，L_v 是节点的标签函数，而函数 f_v 则为指定的节点 u_0 分配了计数量词 m，即 $f_v(u_0)=m$（m 是自然数）。本文把 u_0 称为模式中的量词约束节点，其余节点称为无量词节点。

直观的说，量化模式图通过纳入计数量词来扩展传统的模式图，它支持单一节点上的简单计数，即对于具有相同标签的两个或两个以上的不同节点，当它们都连接了其他标签的相同节点时，便将这些节点聚合为一个带有计数量词的节点，称之为量词约束节点。

例 4 如图 2(c)所示，Q_2 为一个量化模式图，它的节点 u_2,u_3,u_4 具有相同的职位标签 PRG，并且同时连接了 $u_1(DBA)$ 和 $u_5(PM)$ 两个节点，那么可以将 u_2,u_3,u_4 聚合起来，将这种量化模式记作 $DBA\text{-}PRG^{(3)}\text{-}PM$，其中量词约束节点为 PRG，量词数 $m=3$。

对于一个 QGP，当其量词约束被设置为 1 时，对应的模式图被称为无量词模式图，并记作 Q^w。例如，图 2(b)中的 Q_1 就是图 2(c)中 Q_2 的无量词模式图。

定义 7（QGP 的支持度） 受 MNI 支持度的启发，我们对 QGP 的支持度做出如下定义：

$$Sup(Q,G)=\min\{t\mid t=|MNIcol(u)|,u\in V_p\} \tag{2}$$

其中，$MNIcol(u)$ 指节点 u 在图 G 中的有效匹配集，并且所有 $MNIcol(u),u\in V_p$ 构成一个 MNItable。需要说明的是，当 u 为量词约束节点时，$MNIcol(u)$ 中每个有效匹配是包含多个元素的集合，否则只包含单个元素。

例 5 参考图 2 中的模式图 Q_1 和 Q_2，它们在图 G 中的匹配集记为 $M(Q_1,G)$ 和 $M(Q_2,G)$，其中 $M(Q_1,G)$ 包含 25 个匹配，$M(Q_2,G)$ 包含 4 个匹配，分别在表 1 和表 2 中列出（由于 Q_1 的匹配个数较多，这里仅列举了部分匹配）。对于 Q_2，我们可以得到 $MNIcol(DBA)=\{v_3,v_{10},v_{12}\}$，$MNIcol(PRG^{(3)})=\{(v_6,v_1,v_7),(v_6,v_{13},v_7),(v_6,v_{14},v_7)\}$，$MNIcol(PM)=\{v_4,v_8,v_9\}$，则 $Sup(Q_2,G)=3$，这里包含在 $MNIcol(u_i)$ 中的节点即是模式图节点 u_i 在 G 中的有效匹配；同理我们可以得到 $Sup(Q_1,G)=3$。上述结果所对应的 MNItable 已经在图 2(b)和图 2(c)中展示。

表 1 Q_1 在 G 中的部分匹配结果

Table 1 Part of matching results of Q_1 in G

匹配	DBA	PRG	PM
1	v_3	v_1	v_4
2	v_3	v_6	v_4
3	v_3	v_6	v_8
4	v_3	v_6	v_9
5	v_3	v_7	v_4
6	v_3	v_7	v_8
7	v_3	v_7	v_9
⋮	⋮	⋮	⋮

表 2 Q_2 在 G 中的匹配结果

Table 2 Matching results of Q_2 in G

匹配	DBA	$PRG^{(3)}$	PM
1	v_3	v_6,v_1,v_7	v_4
2	v_{10}	v_6,v_{13},v_7	v_9
3	v_{10}	v_6,v_{14},v_7	v_8
4	v_{12}	v_6,v_{14},v_7	v_8

定理 1 给定 $Q^m_{u_0}$ 和 $Q^{m'}_{u_0}$，它们具有相同的无量词模式图 Q^w，且 $m>m'$，则称 $Q^m_{u_0}$ 蕴含 $Q^{m'}_{u_0}$，记作 $Q^m_{u_0}\Rightarrow Q^{m'}_{u_0}$。并且，$Q^m_{u_0}$ 的每个匹配蕴含了 $Q^{m'}_{u_0}$ 的多个匹配。

证明：对于 $Q^m_{u_0}$ 和 $Q^{m'}_{u_0}$，两者包含了同一类型的量词约束节点 u_0，且具有相同的 Q^w，量词数 $m>m'$，则不难得出 $Q^m_{u_0}$ 可被拆分成若干 $Q^{m'}_{u_0}$，并且有 $MNIcol(u_0^{m'})>MNIcol(u_0^m)$。在上述条件下，若 $Q^m_{u_0}$ 频繁，则 $Q^{m'}_{u_0}$ 一定频繁，即 $Q^m_{u_0}$ 蕴含 $Q^{m'}_{u_0}$。

值得注意的是，对于具有蕴含关系的 QGP，本文着重于识别其中计数量词最大的 QGP。即对于模式 Q^2_B：$A-B^{(2)}-C$ 和 Q^3_B：$A-B^{(3)}-C$，本文只会关注后者，因为：1）$Q^3_B\Rightarrow Q^2_B$；2）根据“反单调”特性，Q^3_B 频繁，则 Q^2_B 也一定频繁。

例 6 参考图 2(a)中结构为 $DBA-PRG^{(m)}-PM$ 的量化模式图，在满足支持度阈值 $\tau=2$ 的情况下，只考虑 $DBA-PRG^{(3)}-PM$，而对于 $DBA-PRG^{(2)}-PM$ 则不再过多讨论。

定义 8（一项集，1-QGP） 对于一个 QGP，当它只含一条边 (u_1,u_2^m)，且 $m>1$ 时，称之为 1-QGP。

定义 9（星型量化模式图，Star-QGP） 星型量化模式图具有星型拓扑结构的特点，该模式结构以一个量词约束节点为中心，其他无量词节点直接与该中心节点相连。

Star-QGP 的生成从一个 1-QGP 开始，然后通过添加相邻 1-QGP 逐步扩展，具体扩展方式为：待扩模式的量词约束节点标签与频繁 1-QGP 的量词约束节点标签相同的情况下，将两者进行合并产生新的 Star-QGP。例如，待扩模式为 $A-B^{(m)}$，另有频繁 1-QGP 为 $C-B^{(n)}$ 且 $m>n$，则合并产生的新模式为 $A-B^{(n)}-C$，以此类推。表 3 列出了本文中用到的符号及其含义。为简化描述，下文中，当没有歧义时，我们用 $\mathcal{Q}$ 来表示量化模式图，而当存在不明指代时则用 $Q^m_{u_0}$ 来表示。

表 3 本文使用的符号定义

Table 3 Definition of symbols used in the paper

符号	含义
$Q^m_{u_0}$	一个量词约束节点为 u_0，量词数为 m 的量化模式图
Q^w	一个量化模式图对应的无量词模式图
$M(Q,G)$	模式 Q 在图 G 中的所有匹配
$Sup(Q,G)$	模式 Q 在图 G 中的支持度
W_c,W_i	协调器和工作站点
$\widetilde{G}[\mathcal{Q}_c]$	模式 $\mathcal{Q}_c$ 的一个虚拟匹配

4 分布式 QGPs 挖掘

前文主要描述了相关研究背景及量化模式图挖掘相关的概念，本章将详细介绍 QGPs 的分布式挖掘算法 DisQGPM。

4.1 问题描述

集中式的解决方案在大图上得不到很好的扩展，因此本文研究了分布式环境下的频繁 QGP 挖掘（DQPM）问题，该问题的形式化描述如下。

输入：分布式图 G（其分割图集为 $F=\{F_1,\cdots,F_n\}$），最小

支持度阈值 τ。

输出:图 G 中频繁 QGP 组成的集合 $\mathcal{B}$,对于 $\mathcal{B}$ 中的任何模式 $\mathcal{Q}$ 都有 $Sup(\mathcal{Q},G)\geqslant\tau$。

注意:根据定理 1,这里的集合 $\mathcal{B}$ 包括了图 G 中所有的频繁 QGP,且不存在任何模式 $\mathcal{Q}'$ 是由 $\mathcal{B}$ 中的另一个模式 $\mathcal{Q}$ 所蕴含的。

4.2 频繁 QGPs 挖掘算法

本文针对 DQPM 问题,提出了一种分布式环境下的并行算法 DisQGPM,如算法 1 所示。该算法首先将“并行挖掘”与“局部评估”相结合,以简化挖掘计算;然后渐进式地识别频繁 QGP。DisQGPM 通过采用一个处理器作为协调器(Coordinator,W_c)和一组处理器作为工作站点(Worker,W_i)来实现并行计算。下文将描述算法的具体细节。

DisQGPM 将一个分布式图 G 的分割图集 $F=\{F_1,\cdots,F_n\}$ 和最小支持度阈值 τ 作为输入,并输出一颗树 T,T 的节点集由图中所有频繁 QGP 组成。在挖掘过程中,W_c 与 W_i 之间存在信息的传播,并应用到了一种策略进行模式的扩展,现在介绍信息传播和模式扩展策略。在分布式环境下,信息在协调器与各工作站点之间传播。具体来说,协调器 W_c 需要向各个工作站点 W_i 广播候选模式,对于每个候选模式 $\mathcal{Q}_c$,各 W_i 需要向 W_c 发送 $\mathcal{Q}_c$ 的无量词节点的局部频率(见式(3))、量词约束节点的匹配集 I_{cq} 和虚拟匹配。下面列出这 3 类信息。

1) $\mathcal{Q}_c$ 在 W_i 的局部频率 $fre(F_i,\mathcal{Q}_c)$,其表达式如下:

$$fre(\mathcal{Q}_c,F_i)=\{(u,|MNIcol(u)|)\mid u\in V_p,u\neq u_0\} \quad (3)$$

2) $\mathcal{Q}_c$ 中量词约束节点在 W_i 的匹配集 I_{cq}。

3) $\mathcal{Q}_c$ 在 W_i 的虚拟匹配 $\widetilde{G}[\mathcal{Q}_c]$,虚拟匹配是一组具有特殊元素 (u,x) 的节点对。这里 (u,x) 表示在分区 F_i 处不存在模式节点 u 的本地匹配,但在其他分区可能会存在一个未知的匹配 x,有关虚拟匹配的细节将在后文说明。

定义 10(前向扩展和后向扩展) 在模式图的扩展过程中,如果一条扩展边引入了一个新的节点,则称该扩展边为外边(outer edge),该扩展为前向扩展。若该扩展边未引入新的节点(扩展边的两个节点已存在于该模式图),则称该扩展边为内边(inner edge),该扩展为后向扩展。有关前向扩展和后向扩展的具体介绍,感兴趣的读者可以参考文献[32]。

接下来介绍 DisQGPM 算法,其伪代码如算法 1 所示。

算法 1 DisQGPM /* 在协调器 W_c 处执行 */

输入:图 G 的分割图集 $F=\{F_1,\cdots,F_n\}$,最小支持度阈值 τ

输出:频繁 QGP 的集合

1. initialize $S^{[e]}:=\emptyset$; $S^{[item]}:=\emptyset$; $T_b:=\emptyset$; $L:=\emptyset$; $P:=\emptyset$; $R:=\emptyset$; flag:=false; T as an empty tree;
2. $S^{[e]}:=\bigcup_{i\in[1,n]}S_i^{[e]}$; $S^{[item]}:=\bigcup_{i\in[1,n]}S_i^{[item]}$;
3. remove Q from $S^{[e]}$ if $Sup(Q,G)<\tau$; remove $\mathcal{Q}$ from $S^{[item]}$ if $Sup(\mathcal{Q},G)<\tau$;
4. $T_b:=StarQGPGen(S^{[item]},\tau)$; update T with T_b; /* 产生频繁 Star-QGP,并更新模式树 T */
5. while(flag≠true) do
6. $L:=PatGen(S^{[e]},T)$;
7. $P:=P\cup PatMiner(L,T)$; /* 在每个工作站点 W_i 处执行 */
8. $R:=SuppEva(L,P,T,\tau)$;
9. if $R\neq\emptyset$ then
10. update T with R; update L,P;
11. else
12. flag:=true;
13. return T.

预处理的流程如下。DisQGPM 首先初始化一组参数:集合 $S^{[e]}$ 用于记录频繁单边模式图(其节点不包含计数量词的模式图),集合 $S^{[item]}$ 用于记录频繁 1-QGP;集合 T_b,L,P 和 R 用于记录中间结果;一个布尔变量 $flag$ 和一个空树 T 分别用于控制 while 循环和记录最终结果(第 1 行)。然后 DisQGPM 执行全局单边模式图与全局 1-QGP 的统计(第 2 行),具体来说,W_c 通知各 W_i 执行局部评估,并行识别各 W_i 的单边模式图和 1-QGP 的局部频率。之后,各 W_i 将局部频率的统计结果发送到 W_c,收到这些局部频率之后,W_c 进行全局支持度计算,并分别从 $S^{[e]}$ 和 $S^{[item]}$ 移除支持度低于 τ 的单边模式图与 1-QGP(第 3 行)。接下来,DisQGPM 通过扩展频繁 1-QGP 来生成一组频繁 Star-QGPs(参考定义 9),并使用这一组频繁 Star-QGPs 更新模式树 T,树 T 的每个节点就对应一个具体的 QGP(第 4 行)。

频繁 QGPs 挖掘的流程如下。在这个阶段,DisQGPM 反复迭代产生候选 QGP 并验证它们的支持度。上述迭代过程逐层执行(第 5—12 行)。

在每一轮迭代中,DisQGPM 首先调用程序 PatGen 生成一组候选量化模式图 L,并将 L 广播给所有工作站点 W_i。之后 W_i 将在本地调用程序 PatMiner 进行本地挖掘(第 6、7 行)。在本地挖掘之后,每个 W_i 需要向 W_c 发送前面提到的 3 类响应信息。随后,DisQGPM 调用程序 SuppEva 来评估集合 L 中候选 QGP 的支持度,并将其中频繁的 QGP 记录在集合 R 中(第 8 行)。如果 R 不为空,则更新模式树 T,之后将集合 L 和 P 置为空,以方便下一次计算;否则,将 $flag$ 置为 $true$,表示 while 循环结束,即挖掘停止。

算法 2 PatGen/* 候选模式生成 */

输入:频繁单边模式图集合 $S^{[e]}$,模式树 T

输出:候选模式集 L

1. initialize $L:=\emptyset$; $T_{top}:=\emptyset$; $p_F:=\emptyset$;
2. $T_{top}:=T.getTop()$;
3. for each pattern $\mathcal{Q}$ in T_{top} do
4. if $\mathcal{Q}$ does not contain an inner-edgethen
5. $P_F:=P_F\cup\{\mathcal{Q}\}$;
6. $L:=L\cup FWExt(P_F,S^{[e]})$;
7. $L:=L\cup BWExt(T_{top},S^{[e]})$;
8. return L.

候选模式生成的流程如下。算法 2 给出了候选模式生成的方式。PatGen 通过用 $S^{[e]}$ 中的频繁单边模式图来扩展位于模式树 T 顶层的频繁 QGPs,生成一组候选 QGPs。具体来说,PatGen 首先初始化几个变量来记录 QGPs 产生过程的中间结果,并且获取位于 T 顶层的频繁 QGPs 放置于集合 T_{top} 中(第 1—2 行),T_{top} 中的 QGPs 被用于后续的扩展(第 3—7 行)。首先将 T_{top} 中不包含内边的模式记录下来放在集合 P_F 中,然后调用 FWExt 对 P_F 中的 QGPs 做前向扩展,调用 BWExt 对 T_{top} 中的 QGPs 做后向扩展,最后返回结果集 L。需要说明的是:1)FWExt 和 BWExt 的扩展方式在定义 10 中已说明,因此这里省略了具体细节;2)为了避免产生过多冗余的 QGPs,若一个 QGP 包含内边,它将不再适用于前向扩展。例如,若对图 3 中的 $\mathcal{Q}_{52}$ 进行前向扩展,则会产生与之前已经产生过的 $\mathcal{Q}_{512}$ 相同的 QGP,这显然是冗余的;3)为了避免新

产生的模式在之前已经生成过，本文采用同构检测的方法来处理，如果已经出现过，则将其抛弃。

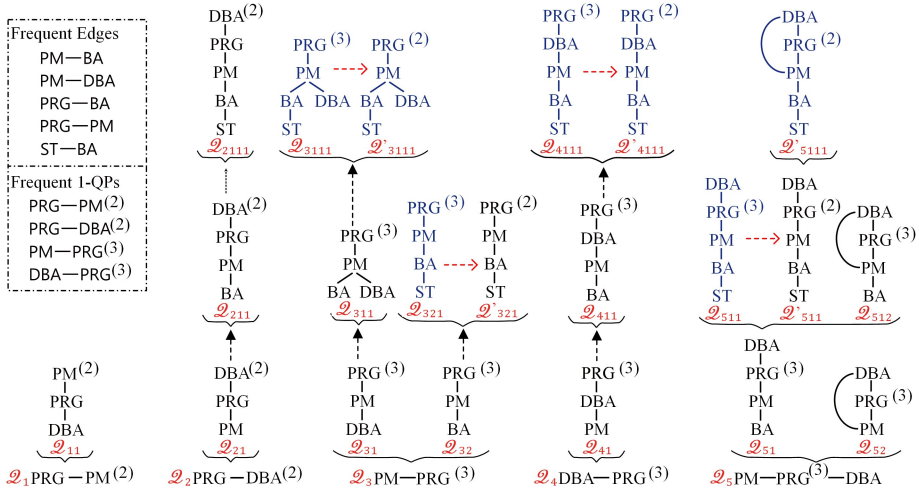


图 3 候选模式生成树 T(电子版为彩图)

Fig. 3 A tree T showing the hierarchical structure of candidate patterns

算法 3 PatMiner /* 在各工作站点W_i并行执行 */

输入：图 G 的一个分割图 F_i，候选模式集 L，模式树 T

输出：集合 H

```
1. initialize an empty map H ;
2. for each candidate pattern 𝒬c in L do
3.    identify a parent 𝒬c' of 𝒬c in T ; Ib:=∅ ;
4.    for each match Gs of 𝒬c' do
5.       if Gs can be locally extended as a match of 𝒬c then
6.          M(𝒬c,Fi):=M(𝒬c,Fi)∪{Gs'} ;
7.       elseif Gs can not be locally extended as a match of 𝒬c then
8.          Ib:=Ib∪{G̃[𝒬c]} ;
9.    send Ib to other workers;
10.   update M(𝒬c,Fi) by including matches received from other workers;
11.   compute fre(𝒬c,Fi) and Icq(𝒬c,Fi) ;
12.   H(Qc):=<Icq(𝒬c,Fi),fre(𝒬c,Fi)> ;
13. return H.
```

本地挖掘的流程如下。如算法 3 所示，当产生一组候选模式 L 时，协调器W_c将 L 广播给所有工作站点W_i，W_i调用程序 PatMiner 进行本地挖掘。PatMiner 以本地分割图F_i、集合 L 和树 T 作为输入，随即展开本地挖掘。PatMiner 首先初始化集合 H 用于维护候选模式中无量词节点的本地频率以及量词约束节点的匹配(第 1 行)，之后对 L 中的每个候选模式$\mathcal{Q}_c$进行计算(第 2－12 行)。具体来说，PatMiner 找出候选模式$\mathcal{Q}_c$在 T 中的父节点模式$\mathcal{Q}_c'$，$\mathcal{Q}_c'$在当前工作站点的匹配将被用于$\mathcal{Q}_c$的支持度评估，之后再初始化一个空集I_b用于记录$\mathcal{Q}_c$的虚拟匹配(第 2－3 行)。对于$\mathcal{Q}_c'$的每个匹配G_s，PatMiner 检查G_s是否可以在当前站点扩展出$\mathcal{Q}_c$的一个匹配G_s'，如果G_s'存在，则将其纳入$\mathcal{Q}_c$的本地匹配集$M(\mathcal{Q}_c,F_i)$中(第 4－6 行)，否则 PatMiner 将生成一个虚拟匹配$\tilde{G}[\mathcal{Q}_c]$，并将其纳入虚拟匹配集I_b中(第 7－8 行)。在$\mathcal{Q}_c'$的所有匹配都被处理之后，PatMiner 将I_b发送给其他工作站点，并且同时接收其他工作站点发送过来的虚拟匹配，然后更新 $M(\mathcal{Q}_c,F_i)$(第 9、10 行)。在此通信结束之后，各工作站点具有候选模式$\mathcal{Q}_c$在本地的一部分匹配，PatMiner 计算出$\mathcal{Q}_c$中无量词节点在本站点的局部频率 $fre(\mathcal{Q}_c,F_i)$，并且计算出量词约束节点的匹配集$I_{cq}(\mathcal{Q}_c,F_i)$，之后用$\langle I_{cq}(\mathcal{Q}_c,F_i),fre(\mathcal{Q}_c,F_i)\rangle$对 $H(\mathcal{Q}_c)$进行初始化(第 11、12 行)。在对所有候选模式进行评估之后，H 作为最终结果被返回(第 13 行)。

例 7 如图 2(a)所示，令 $\tau=2$，DisQGPM 采用了一个协调器W_c和一组工作站点W_1，W_2，W_3。W_c首先收集到来自每个W_i的单边模式图和 1-QGP 的局部频率，然后统计出频繁单边模式和频繁 1-QGP，如图 3 所示，虚线框内的内容表示图 G 的频繁单边模式图和频繁 1-QGP(支持度都不低于 2)，用这两者分别初始化集合$S^{[e]}$和$S^{[item]}$。随后，DisQGPM 调用进程 StarQGPGen，利用频繁 1-QGP 生成频繁 Star-QGPs，如$\mathcal{Q}_1-\mathcal{Q}_5$。接下来，DisQGPM 调用进程 PatGen，按照前向扩展和后向扩展的策略生成一组候选模式，例如基于 Star-QGP $\mathcal{Q}_3$，W_c通过频繁单边模式图对$\mathcal{Q}_3$进行扩展，产生了 $L=\{\mathcal{Q}_{31},\mathcal{Q}_{32}\}$。图 3 给出了 4 层没有重复节点标签的候选 QGPs，其中标记为蓝色的 QGPs 是不频繁的(支持度小于 2)，红色的虚线箭头表示量词下降的过程(后文将进行详细介绍)。

频繁 QGPs 识别的流程如下。如算法 4 所示，在获得候选模式集 L 和其在每个工作站点的统计信息集 P 之后，协调器W_c调用程序 SuppEva 来识别 L 中的所有频繁 QGPs。SuppEva 将 L，P，T 和 τ 作为输入，工作方式如下。它首先初始化一组变量，即 D，P_d，$result$ 和T_1(第 1 行)，再执行频繁 QGPs 的识别(第 2－10 行)。具体来说，对于 L 中的每个候选模式$\mathcal{Q}_c$，SuppEva 从集合 P 中检索出$\mathcal{Q}_c$在各站点的相关信息$\langle I_{cq},fre\rangle$，并记录在集合$T_1$中(第 3－5 行)，之后 SuppEva 调用程序 GlobalEva 来计算$\mathcal{Q}_c$全局支持度(第 6 行)。当获得$\mathcal{Q}_c$的全局支持度之后，如果 $Sup(\mathcal{Q}_c,G)\geqslant\tau$，则$\mathcal{Q}_c$将被放入结果集 $result$ 中，并将T_1置为空(第 7、8 行)。若 $Sup(\mathcal{Q}_c,G)<\tau$，但$\mathcal{Q}_c$的量词数大于 2，SuppEva 则调用程序 CQdec 来减小$\mathcal{Q}_c$的量词大小，并将量词减小后得到的 QGP 放入集合 D 中(第 9、10 行)。当 L 中所有的 QGPs 都被处理后，SuppEva 检查 D 是否为空，如果 D 不为空，则将其广播给所有的工作站点，并在每个站点处调用程序 PatMiner 进行本地挖掘，之后递归执行 SuppEva 以进一步识别频繁 QGPs，直到所有频繁的 QGPs 被识别出，最终返回结果集 $result$(第 11－14 行)。

算法 4 SuppEva / * 中心站点S_c执行 * /

输入：候选模式集 L，集合 P，树 T，最小支持度阈值 τ

输出：包含一组频繁 QGPs 的集合

1. initialize D:=∅; P_d:=∅; result:=∅; T_1:=∅;
2. for each candidate pattern $\mathcal{Q}_c$ in L do
3. for each map H in P do
4. retrieve ⟨I_{cq},fre⟩ from H($\mathcal{Q}_c$) ;
5. T_1:=T_1∪⟨I_{cq},fre⟩;
6. GlobalEva(T_1,$\mathcal{Q}_c$);/ * 全局支持度评估 * /
7. if Sup($\mathcal{Q}_c$,G)≥τ then
8. result:=result∪{$\mathcal{Q}_c$} ; update T_1;
9. else if Sup($\mathcal{Q}_c$,G)<τ and CQ>2 then
10. D:=CQdec(D,$\mathcal{Q}_c$) ;
11. if D≠∅ then
12. P_d:=P_d∪PatMiner(D,T);
13. result:=result∪SuppEva(D,P_d,T,τ);
14. return result.

局部评估的流程如下。对于一个候选模式$\mathcal{Q}_c$及其子模式$\mathcal{Q}_c'$，当$\mathcal{Q}_c'$的一个匹配G_s不能在某工作站点扩展出$\mathcal{Q}_c$的匹配时，G_s被称为$\mathcal{Q}_c$的部分匹配，对于这种情况，由于在其他站点可能会通过G_s扩展出$\mathcal{Q}_c$的匹配，因此需要从全局角度对该匹配进行验证。为此，本文引入了一种基于局部评估的方法，具体如下。

针对上述问题，工作站点W_i需要构建一个$\mathcal{Q}_c$的虚拟匹配$\widetilde{G}[\mathcal{Q}_c]$，然后将$\widetilde{G}[\mathcal{Q}_c]$与模式$\mathcal{Q}_c$一起发送到其他工作站点，$\widetilde{G}[\mathcal{Q}_c]$被定义为：

$$\widetilde{G}[\mathcal{Q}_c]=\{(u,\vartheta)\mid u\in V_c',\vartheta\subseteq V_s\}\cup\{(\widetilde{u},x)\mid \widetilde{u}\in V_c\backslash V_c'\}$$

其中，V_c，V_c'和V_s分别代表$\mathcal{Q}_c$，$\mathcal{Q}_c'$和G_s的节点集，x是一个变量，表示$\mathcal{Q}_c$中节点$\widetilde{u}$可能的匹配。需要注意的是，ϑ代表V_c的一个子集，若u为无量词节点，则ϑ只包含一个元素，若u为量词约束节点，则ϑ包含多个元素。

在与其他工作站点交换虚拟匹配集之后，如果$\widetilde{G}[\mathcal{Q}_c]$中的变量$x$能被当前工作站点中具体的节点实例化，则每个工作站点W_i将其添加到本地匹配集$M(\mathcal{Q}_c,F_i)$中。然后，各W_i计算候选模式$\mathcal{Q}_c$中无量词节点本地频率$fre(\mathcal{Q}_c,F_i)$和量词约束节点的匹配$I_{cq}(\mathcal{Q}_c,F_i)$，最后将其发送给协调器$W_c$。

例 8 继续上一个例子。在W_2收到候选模式$\mathcal{Q}_{211}$后，调用程序 PatMiner 识别了$\mathcal{Q}_{211}$在模式树 T 上的父模式$\mathcal{Q}_{21}$以及$\mathcal{Q}_{21}$在W_2处的本地匹配G_{w1}：{(v_3，v_{10}，v_{12})，v_6，v_9}，G_{w2}：{(v_3，v_{10}，v_{12})，v_7 v_9}，G_{w3}：{(v_3，v_{10}，v_{12})，v_6，v_4}，G_{w4}：{(v_3，v_{10}，v_{12})，v_7，v_4}，这里需要注意的是，这 4 个匹配各自蕴含了 3 个$\mathcal{Q}_{21}$的匹配，为了减少内存消耗，此处不会拆开存储。然后 PatMiner 利用这些匹配去计算$\mathcal{Q}_{211}$在W_2的匹配，经过循环，由G_{s1}和G_{s2}产生了$\mathcal{Q}_{211}$的匹配：{(v_3，v_{10}，v_{12})，v_6，v_9，v_{11}}和{(v_3，v_{10}，v_{12})，v_7，v_9，v_{11}}。由于G_{w3}和G_{w4}中的节点v_4在W_2处是虚拟节点，因此在W_2不能扩出相应的匹配，但这并不意味着其他站点也是如此。因此，在W_2处会产生两个虚拟匹配{(v_3，v_{10}，v_{12})，v_6，v_4，x}和{(v_3，v_{10}，v_{12})，v_7，v_4，x}，之后 PatMiner 将这两个虚拟匹配发送到v_4所在的工作站点进行扩展，同时W_2也将收到W_3发送的虚拟匹配{(v_{10}，v_{12})，v_{14}，v_8，x}，可以验证这个虚拟匹配在W_2也不能扩展出$\mathcal{Q}_{211}$的匹配。在处理完接收到的虚拟匹配之后，PatMiner 计算出模式中无量词节点的本地频率{(PRG,2)，(PM,1)，(BA,1)}，并将其与量词约束节点的匹配{(v_3，v_{10}，v_{12})}一起发送到协调器W_c。W_c收到各W_i发送的信息后，调用程序 SuppEva 进行统计分析并识别频繁 QGPs。

5 量化图模式关联规则挖掘及应用

作为 QGPs 的一个应用，本节提出了量化图模式关联规则（Quantified Graph Pattern Associ-ation Rules，QGPAR），用于发现图中实体之间的关联关系，特别是社交网络中潜在客户之间的关联性。

定义 11（量化图模式关联规则，QGPAR） 沿着与文献[32]中定义图模式规则相同的思路，本文将 QGPAR R 定义为形如$\mathcal{Q}_l\rightarrow\mathcal{Q}_r$的形式，其中$\mathcal{Q}_l$是一个 QGP，$\mathcal{Q}_r$是普通模式图（不带计数量词约束），且分别称$\mathcal{Q}_l$和$\mathcal{Q}_r$为 R 的先导和后继。规则 R 指出，在图 G 中，如果存在从$\mathcal{Q}_l$到子图G_1的同构映射h_l，则可能存在另一个从$\mathcal{Q}_r$到子图G_2的映射h_r，使得对于每个节点 $u\in V_l\cap V_r$，如果它被h_l映射到G_1中的节点 v，那么它也被h_r映射到G_2中的同一节点 v。对于一个 QGPAR R，本文将其建模为通过$\mathcal{Q}_r$的边集扩展$\mathcal{Q}_l$得到的模式图，并记为$\mathcal{Q}_R$。

实际上，传统的图模式关联规则（GPAR）是 QGPAR 的特殊情况，即计数量词为 1。带有计数量词的扩展有效地丰富了 QGPAR 的语义，从而能够捕获更多更有意义的规则。

支持度：通过将 QGPAR R 视为模式图$\mathcal{Q}_R$，将 QGPAR R 的支持度定义为：

$$Sup(R,G)=Sup(\mathcal{Q}_R,G)$$

置信度：遵循传统关联规则的置信度度量，将 QGPAR R 在图中的置信度定义为：

$$Conf(R,G)=\frac{Sup(\mathcal{Q}_R,G)}{Sup(\mathcal{Q}_l,G)}$$

本文研究的 QGPAR 需满足：1）$\mathcal{Q}_R$，$\mathcal{Q}_l$和$\mathcal{Q}_r$是连通的；2）$\mathcal{Q}_l$和$\mathcal{Q}_r$是非空的；3）$\mathcal{Q}_l$和$\mathcal{Q}_r$没有公共边。例如，图 4(a)是在图 2(a)中发现的一个 QGPAR，表示为 R：$\mathcal{Q}_l\rightarrow\mathcal{Q}_r$。通过用$\mathcal{Q}_r$的边集扩展$\mathcal{Q}_l$，可以将 R 表示为图 4(a)右边的$\mathcal{Q}_R$。由例 5 知$Sup(\mathcal{Q}_l,G)=3$，同样，$\mathcal{Q}_R$在 G 中的匹配为{v_3，(v_6，v_1，v_7)，v_4}和{v_{10}，(v_6，v_{13}，v_7)，v_9}，则 $Sup(\mathcal{Q}_R,G)=2$。因此，R 的置信度$Conf(R,G)=\frac{2}{3}$。

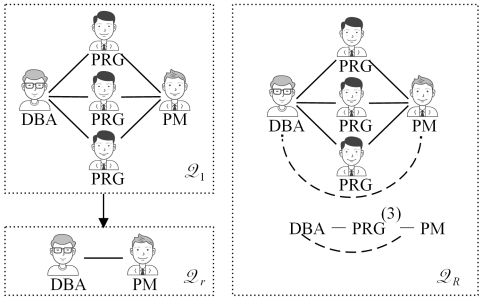
图 4(b)列举了两种 QGPARs 的应用场景。其中规则的先导表示为模式图中具有实线的部分，后继用虚线表示。从这些规则中可以看出 QGPARs 的作用。

例 9 1）可以用于发现社交网络中的潜在客户。R_1表明，如果X_0参加了一个音乐俱乐部，并且在他的朋友中有 m 个人都喜欢并且买了一张专辑，那么X_0也有很大的可能买下这张专辑。

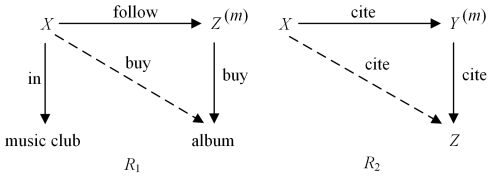
2）可以发现某些协作网络中不同实体之间的联系，R_2描述了 X 引用了 m 篇 Y 方向的论文，这 m 篇论文引用了一篇 Z 方向的论文，可以推测 X 与论文 Z 之间大概率也存在引用关系。

QGPAR 生成过程如下：给定一个频繁 QGP $Q_{u_0}^m$，枚举其非空子图$\mathcal{Q}_l$（包含量词约束节点u_0），以及$\mathcal{Q}_r=Q_{u_0}^m\backslash\mathcal{Q}_l$（不含$u_0$），然后可以产生规则$R$：$\mathcal{Q}_l\rightarrow\mathcal{Q}_r$。本文将所有满足支持度和

信任度阈值的规则称为有效规则。



(a)QGPAR R: $Q_l -> Q_r$



(b)QGPAR

图 4 量化图模式关联规则及示例

Fig. 4 QGPARs and two samples

6 实验结果与分析

使用真实图数据以及合成图数据，本文进行了 3 组实验以评估：1)算法 DisQGPM 的效率；2)QGPAR 在大型真实图以及合成图中链接预测的准确性；3)QGPAR 与传统图模式关联规则(GPAR)的链接预测结果的差异性。

本文采用分图算法[33]将图数据划分为 n 个部分，并将它们分布到 n 个工作站点。每个站点配置相同：操作系统为 Centos7.6，CPU 为 x86 32 核 2.0GHz，128GB 内存。

6.1 实验数据

整个实验一共使用了表 4 中的 6 个数据集，数据集描述和相关处理如下。

1)Lasftm_asia：该数据集是 LastFM 用户组成的社交网络，于 2020 年 3 月从公共 API 中收集。节点是来自亚洲国家的 LastFM 用户，边缘是它们之间的相互追随者关系。根据用户喜欢的艺术家提取顶点特征。此目标特征源自每个用户的国家/地区字段。

2)Facebook：该数据集是 Facebook 页面的数据，于 2017 年 11 月从公共 API 中收集。节点表示页面，边缘是页面之间的链接关系。由于原始数据集并不含有节点和边的标签，因此随机地对节点和边进行标签的添加，标签分布服从高斯分布。

3)Mico：该数据集对 Microsoft 合作信息进行建模，并由具有 10 万个节点和 100 万条边的无向图组成。节点代表作者，并标有作者感兴趣的领域。边代表两位作者之间的合作，并标有合作论文的数量。Mico 数据集是 Elseidy[7]从 Academic.research.microsoft.com 爬取的计算机科学协作图。

4)DBLP：该数据集是 DBLP 数据库中论文作者合作的信息，其顶点表示论文的作者，顶点的标签为该作者所在的学术领域，边表示两个作者之间存在合作关系，边的标签为两个作者之间合作的次数。由于原始数据集并不含有节点和边的标签，因此随机地对节点和边进行标签的添加，标签分布服从高斯分布。

5)Amazon：该数据集是亚马逊商城的商品信息，节点表示商品，边 i-j 表示购买商品 i 的同时购买商品 j，节点标签表示商品 ID，由于原始数据集上不包含节点标签，因此随机地对节点进行标签的添加，标签分布服从高斯分布。

6)AstroPh：涵盖提交至天体物理学类别的作者论文之间的科学协作。如果论文 i 与论文 j 之间存在引用关系，则该图包含从 i 到 j 的边。

表 4 实验数据集的描述

Table 4 Description of datasets

数据集	点数量	点标签数量	边数量	边标签数量
Lasftm_asia	7624	49	27806	1
Facebook	50515	33	819306	1
Mico	100000	29	1080298	1
DBLP	425957	65	1049866	1
Amazon	334863	30	925872	1
AstroPh	18772	50	396160	1

6.2 算法 DisQGPM 的性能评估

本文评估了算法 DisQGPM 的性能，分别实现了在集中式环境下与分布式环境下的挖掘过程，并且在 Lasftm_asia，Facebook，Mico，DBLP，AstroPh 和 Amazon 数据集上做了对比实验。

图 5(a)－图 5(f)显示了在不同数据集、不同实验设置下的挖掘时间性能，横轴为最小支持度 τ 的取值，纵轴为算法的运行时间(RunningTime，RT)，其中 DisQGPM(n)表示 DisQGPM 被部署在 n 个工作站点，DisQGPM(1)表示 DisQGPM 的集中式版本。

1)变化阈值 τ。本文在不同的数据集上对最小支持度 τ 设置了不同范围，例如在 Amazon 数据集上 τ 以 1×10^2 的增量从 3×10^3 增长到 3.4×10^3。

在工作节点数 n 保持 4 不变的情况下，比较不同最小支持度下挖掘的运行时间 RT，结果如图 5(a)－图 5(f)所示。图 5(a)－图 5(f)给出了在 6 个不同数据集下运行时间和最小支持度的关系。可以发现 DisQGPM 在 τ 较小的情况下，需要更长的运行时间，这是因为运行过程中需要验证更多的候选模式及其匹配。随着最小支持度 τ 的不断增加(或减少)，候选模式数量急剧减少(或增加)，因此耗时的评估成本保持下降(或上升)，运行时间也一直保持下降(或上升)。上述趋势在其他数据集上也是如此。

2)变换工作站点数 n。图 5(a)－图 5(f)还给出了另一个关键因素的影响，即并行度参数 n。图 5(a)－图 5(f)给出了同一最小支持度 τ 下并行度参数 n 对性能的影响。(1)当数据规模较小时，集中式环境下的挖掘性能可能比在分布式环境中的性能更好，如图 5(a)所示，在最小支持度 τ 相同时，相比 DisQGPM(4)和 DisQGPM(2)，集中式版本 DisQGPM(1)表现出了更好的性能。(2)在大规模数据集中，挖掘计算涉及的工作站点(处理器)越多，DisQGPM 所需的运行时间更短，这是因为 DisQGPM 通过并行计算节省了时间开销。例如，本文设置了 Mico 的最小支持度为 2.35×10^3，DBLP 的最小支持度为 3×10^2，Amazon 的最小支持度为 3.2×10^3，对于这 3 个数据集，DisQGPM(4)的运行速度分别是 DisQGPM(1)的 3.23，3.48 和 1.95 倍，DisQGPM(2)的运行速度分别是 DisQGPM(1)的 1.35，2.14 和 1.24 倍。

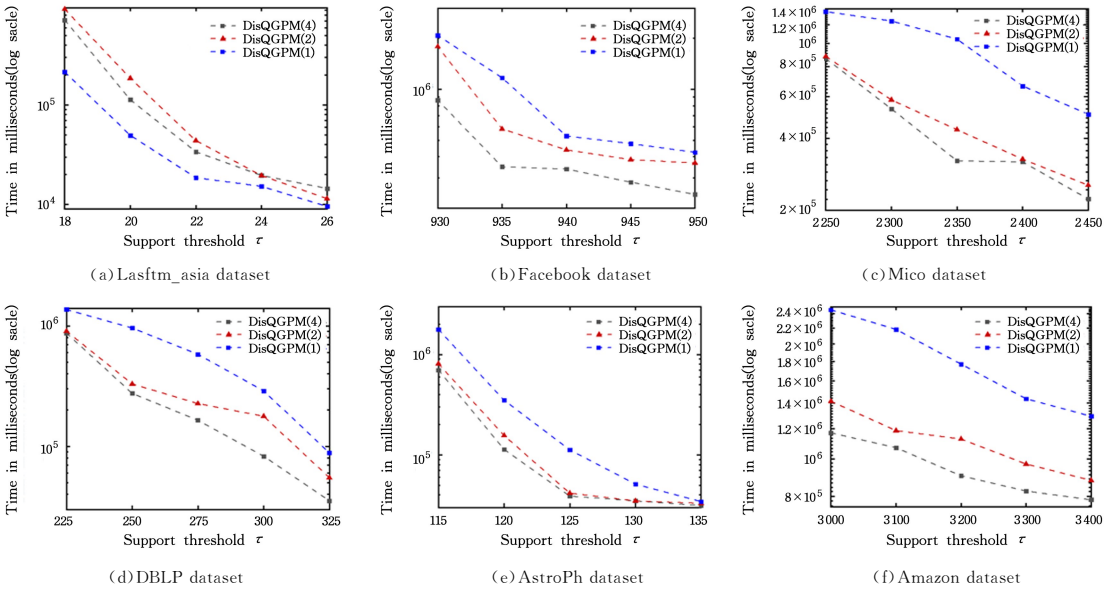


图 5　DisQGPM 在真实图上的性能对比

Fig. 5　Performance comparison of DisQGPM on real-life graph

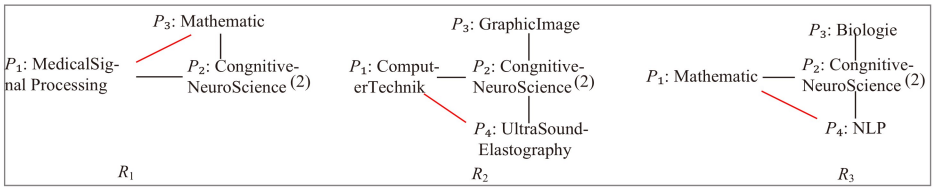
6.3　预测精度

在这组测试中，本文评估了 QGPAR 与 Jaccard，SimRank[34]，Adamic-Adar(AA)，Resource Allocation(RA)，Hub Depressed Index（HDI）和 Node Clustering Coefficient (CCLP)[34]的预测准确性，实验设置如下。

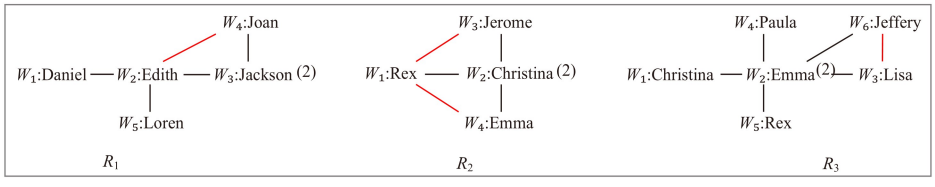
QGPAR 的预测准确性通过交叉验证进行测试，也就是说，给定一个图 G，本文将其划分成两个片段 F_1 和 F_2，从 F_1 中挖掘 QGPAR 并根据度量 Gain[35]（θ 设置为 0.4）和 Interest[36]进行排序，选择前 10，20，30，40，50 个 QGPAR，在 F_2 中去验证这些规则的精度。对于 F_2 中的每一个 QGPAR R 来说，其预测精度定义为 $Acc(R)=\frac{sup(\mathcal{Q}_R,F_2)}{sup(\mathcal{Q}_l,F_2)}$。

本文选取了从 DBLP 和 AstroPh 两个数据集中挖掘出的具有代表性的 QGPARs。图 6(a)给出了从 DBLP 发现的 3 个典型的 QGPARs。具体来说，R_1 表明，一篇医学信号处理方向的论文 P_1 引用了 2 篇认知神经科学方向的论文 P_2，并且这两篇认知神经科学方向的论文引用了一篇数学方面的论文 P_3，那么这篇医学信号处理方向的论文 P_1 很有可能引用了这篇数学方向的论文 P_3。R_2 和 R_3 所表达的意思与 R_1 相近。

图 6(b)给出了 AstroPh 数据中 3 个典型的 QGPARs。1）R_1 表明如果论文作者 W_2 与作者 W_3 进行了多次合作，并且 W_3 与 W_4 也进行了多次合作，那么 W_2 和 W_4 之间也可能进行了论文之间的合作。2）R_2 表明作者 W_1，W_3 以及 W_4 都与作者 W_2 进行了多次论文合作，则 W_1 和 W_3，W_1 和 W_4 之间也可能存在合作关系。R_3 与前两种规则类似。



(a) Real-life QGPARs: DBLP



(b) Real-life QGPARs: AstroPh

图 6　真实图上的量化模式关联规则

Fig. 6　QGPARs on real-life graph

Jaccard，Simrank，AA，RA，HDI 和 CCLP 的测试是通过 4 折交叉验证进行的。具体如下：

1）由于 Simrank 不适合在整个大图上运行，本文分别从 Amazon，AstroPh，DBLP 提取了大小为（10000，30729），(5000，88822)，(10000，5678)的子图 G_F。

2）对于每一个 G_F，将其边集随机分为 4 部分；每次选择一部分作为测试集，将其余 3 部分作为训练集。交叉验证的过程重复 4 次，其中 4 个子集中每一个都会用来做一次测试集，剩余的 3 个则作为训练集。

3）每次在训练集上运行 Jaccard，Simrank，AA，RA，HDI 和 CCLP，获得一个节点对列表，按照计算出来的相似度进行排序，选择 Top-L（L 的范围是 50～250，增量为 50）个节点对

作为预测边的节点对，验证精度定义为 $Acc=\frac{L_r}{L}$，其中 L_r 为预测正确的边的数量。

4）对于 Simrank，参数 C 设置为 0.8，迭代次数为 5。

表 5 列出了 QGPAR，Jaccard，Simrank，AA，RA，HDI 和 CCLP 的预测精度，对比发现：1）挑选前 50 个 QGPARs，可以在 3 个数据集上以高达 84.3%的平均准确率预测缺失关系，并且 k 值越高，$Acc(R)$ 越低，因为度量“Interest”可能会导致较低的预测精度；2）本文方法在预测精度方面优于 Jaccard，Simrank，AA，RA，HDI 和 CCLP。以数据集 Amazon 为例，当使用度量“Interest”，k 设置为 50 时，QGPAR 的预测精度（即 71.7%）分别是 Jaccard，Simrank，AA，RA，HDI 和 CCLP（$L=250$）的 170.71%，320.0%，201.40%，194.84%，229.81%和 186.23%。这里需要注意的是：1）QGPAR 用于预测模式，这可能会涉及到多条边的计算，而其他方法都只能预测单条边，虽然比较条件不一致，但是仍然可以看到 QGPARs 实现了更高的整体预测精度；2）通过深入分析 Jaccard，Simrank，AA，RA，HDI 和 CCLP 的预测过程，发现这些方法所取得的较低的预测精度部分是由于所分析图的稀疏特征引起的。

表 5 预测准确度对比

Table 5 Comparison of prediction accuracy

（单位：%）

		Amazon					AstroPh					DBLP				
		Top-k($Acc(R)$)					Top-k($Acc(R)$)					Top-k($Acc(R)$)				
		10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
QGPAR	Gain	83.4	81.1	78.6	75.9	73.7	92.9	93.3	93.4	92.2	83.0	90.9	90.1	88.9	86.9	83.8
	Interest	79.1	76.9	76.8	73.8	71.7	92.7	91.9	90.8	90.0	89.7	91.5	86.1	86.1	85.1	82.7
		Top-L(Acc)					Top-L(Acc)					Top-L(Acc)				
		50	100	150	200	250	50	100	150	200	250	50	100	150	200	250
Jaccard		43.5	42.5	43.6	43.4	42.0	56.5	43.0	43.0	33.5	27.0	47.0	49.7	48.7	48.4	48.1
SimRank		18.0	20.2	22.1	25.2	22.4	13.0	14.0	13.8	17.7	21.1	73.5	44.2	31.0	23.6	19.6
AA		46.0	45.0	40.7	30.0	35.6	54.0	64.0	70.7	71.0	68.8	56.0	51.0	50.7	50.5	46.8
RA		44.0	38.0	36.7	39.5	36.8	56.0	69.0	74.7	75.0	74.0	56.1	52.0	53.3	49.0	50.4
HDI		50.0	35.5	31.3	29.5	31.2	32.0	25.0	30.0	22.5	18.0	48.0	53.0	49.3	50.0	50.0
CCLP		47.9	41.5	40.3	43.0	38.5	58.7	71.2	76.0	77.3	75.8	61.0	55.6	55.1	54.7	51.5

本节通过对比实验，证明了本文方法在准确性方面的优势。

6.4 QGPAR 的有效性

本节对不同数据集中部分量化模式图和它对应的无量词模式图的匹配集进行了比较，并分析对比了两类不同的模式图所产生的图模式关联规则在链接预测性能方面的差异。

图 7 给出了从数据集 Amazon，AstroPh 和 DBLP 中选出的 3 组代表性频繁量化模式图 $\mathcal{Q}_1$，$\mathcal{Q}_2$，$\mathcal{Q}_3$；图 8 则给出了这 3 组量化模式图（QGP）及其所对应的无量词模式图（GP）在对应图数据中匹配结果集的大小，其中横坐标对应模式图的序号，纵坐标代表匹配集大小。不难看出，在 Amazon，AstroPh 和 DBLP 图上，$|M(Q_i,G)|/|M(Q_i^w,G)|(i\in[1,3])$ 的均值分别为 0.156，0.413，0.403。由此发现，量化模式图的匹配结果集相对无量词模式图要小很多，这对于用户的理解及使用是十分有益的。

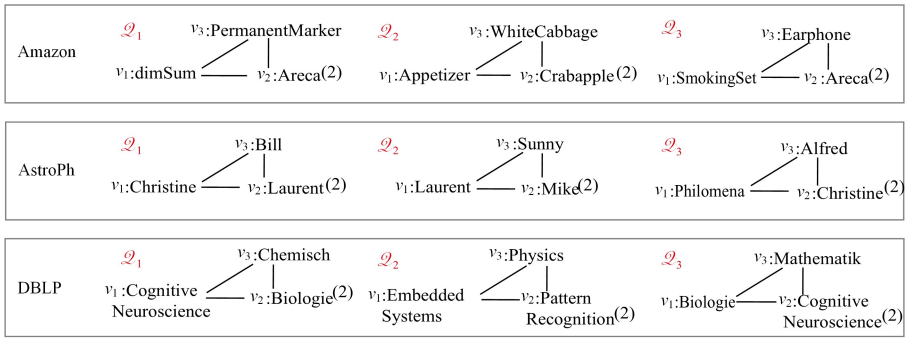
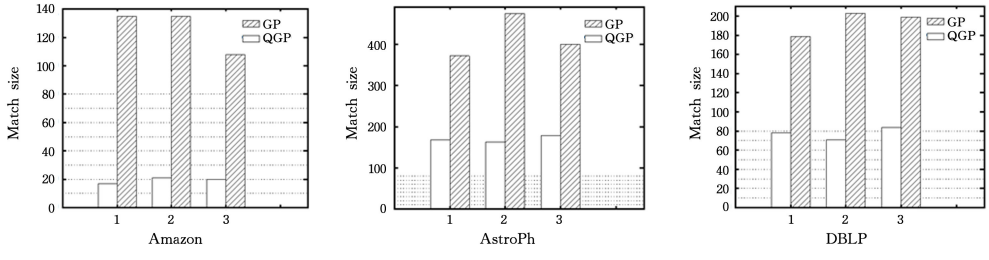


图 7 量化模式图实例

Fig. 7 Examples of QGP



(a) Amazon dataset (b) Astroph dataset (c) DBLP dataset

图 8 匹配集大小对比

Fig. 8 Comparison of matching set size

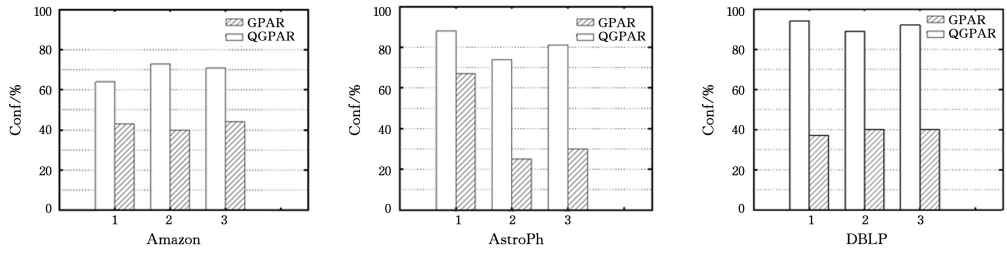
进一步地，基于上述量化模式图及其对应的无量词模式图，我们产生了相应的量化图模式关联规则(QGPAR)和图模式关联规则(GPAR)，并利用这些规则进行了链接预测性能(即置信度)的比较。表 6 列出了生成的 QGPAR 与 GPAR 的链接预测结果集的 Jaccard 值，从结果可以看出，这两类规则所发现的链接预测结果有较大差异。进一步地，图 9 给出了不同数据集上产生的 QGPAR 和 GPAR 的置信度指标。其中，横坐标代表对应的(量化)模式图所生成的 QGPAR 和 GPAR，纵坐标代表规则的置信度。实验结果表明，QGPAR 的置信度普遍比 GPAR 的置信度高，这显示出基于 QGPAR 的推荐更加可靠。从以上两点可以看出，QGPAR 与 GPAR 在链接预测方面存在较大差异，尤其是 QGPAR 的预测结果置信度显著高于 GPAR，这对于推荐场景下的使用具有重要意义。

表 6　QGPAR 与 GPAR 的预测结果的 Jaccard 值

Table 6　Jaccard values for predicted results of QGPAR vs GPAR

(单位：%)

	Amazon			AstroPh			DBLP		
	$(Q_1, Q_1^{\overline{xu}})$	$(Q_2, Q_2^{\overline{xu}})$	$(Q_3, Q_3^{\overline{xu}})$	$(Q_1, Q_1^{\overline{xu}})$	$(Q_2, Q_2^{\overline{xu}})$	$(Q_3, Q_3^{\overline{xu}})$	$(Q_1, Q_1^{\overline{xu}})$	$(Q_2, Q_2^{\overline{xu}})$	$(Q_3, Q_3^{\overline{xu}})$
Jaccard	43.9	45.3	27.8	8.3	13.5	11.3	25.0	33.4	30.9



(a) Amazon dataset　(b) Astroph dataset　(c) DBLP dataset

图 9　规则置信度对比

Fig. 9　Comparison of rule confidence

结束语　本文研究了量化模式图(QGP)挖掘的问题，并提出了一种分布式环境下的挖掘算法——DisQGPM。该算法采取了“并行挖掘”与“局部评估”相结合的策略来识别频繁 QGPs。此外，作为 QGP 的应用，本文还提出了量化图模式关联规则(QGPAR)，它可以对社交实体之间更复杂的关系进行建模。最后，本文通过实验验证了算法的性能、有效性和可扩展性。

目前我们对量化模式图的研究仍处于起步阶段，未来工作是将计数量词扩展到多个节点上，使模式图更加多样化，进一步研究提升算法运行效率的可能性也是未来工作的一部分。

参 考 文 献

[1] FAN W F. Graph pattern matching revised for social network analysis[C]// Proceedings of the 15th International Conference on Database Theory. Berlin: ACM, 2012: 8-21.

[2] FAN W F, WANG X, WU Y H, et al. Association rules with graph patterns[J]. Proceedings of the VLDB Endowment, 2015, 8(12): 1502-1513.

[3] BAPNA R, UMYAROVA. Do your online friends make you pay? A randomized field experiment on peer influence in online social networks[J]. Management Science, 2015, 61(8): 1902-1920.

[4] Nielsen global online consumer survey [OL]. https://www.nielsen.com/wp-content/uploads/sites/2/2019/04/pr_global-study_07709.pdf.

[5] LIAQAT M, KHAN S, YOUNIS M S, et al. Applying uncertain frequent pattern mining to improve ranking of retrieved images [J]. Applied Intelligence, 2019, 49(8): 2982-3001.

[6] SONG Q, WU Y, LIN P, et al. Mining summaries for knowledge graph search[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(10): 1887-1900.

[7] ELSEIDY M, ABDELHAMID E, SKIADOPOULOS S, et al. Grami: Frequent subgraph and pattern mining in a single large graph[J]. Proceedings of the VLDB Endowment, 2014, 7(7): 517-528.

[8] TALUKDER N, ZAKI M J. A distributed approach for graph mining in massive networks[J]. Data Mining and Knowledge Discovery, 2016, 30(5): 1024-1052.

[9] KAVITHA D, HARITHA D, PADMA Y. Optimized Candidate Generation for Frequent Subgraph Mining in a Single Graph [C]// Proceedings of International Conference on Computational Intelligence and Data Engineering. Springer, Singapore, 2021: 259-272.

[10] LI L, DING P, CHEN H, et al. Frequent pattern mining in big social graphs[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2021, 6(3): 638-648.

[11] UR REHMAN S, LIU K, ALI T, et al. A graph mining approach for ranking and discovering the interesting frequent subgraph patterns[J]. International Journal of Computational Intelligence Systems, 2021, 14: 1-17.

[12] ASHRAF N, HAQUE R R, ISLAM M, et al. WeFreS: weighted frequent subgraph mining in a single large graph[C]// 19th Industrial Conference Advances in Data Mining-Applications and Theoretical Aspects(ICDC). 2019: 201-215.

[13] LE N T, VO B, NGUYEN L B Q, et al. Mining weighted subgraphs in a single large graph[J]. Information Sciences, 2020, 514: 149-165.

[14] RAY A, HOLDER L, CHOUDHURY S. Frequent subgraph discovery in large attributed streaming graphs[C]// Proceedings of the 3rd International Workshop on big data, streams and heterogeneous source mining: algorithms, systems, programming

models and applications. New York: JMLR, 2014: 166-181.
[15] ABDELHAMID E, CANIM M, SADOGHI M, et al. Incremental frequent subgraph mining on large evolving graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2710-2723.
[16] BRY F, FURCHE T, MARNETTE B, et al. SPARQLog: SPARQL with rules and quantification[M]. Berlin: Springer, 2009: 341-370.
[17] AMER-YAHIA S, LAKSHMANAN L, YU C. Socialscope: Enabling information discovery on social content sites[C]// Fourth Biennial Conference on Innovative Data Systems Research. Asilomar, CA: CIDR, 2009.
[18] SAN MARTIN M, GUTIERREZ C, WOOD P T. SNQL: A social networks query and transformation language[C]// Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management. Santiago: CEUR, 2011.
[19] LiN W Y, ALVAREZ S A, RUIZ C. Collaborative recommendation via adaptive association rule mining[J]. Data Mining and Knowledge Discovery, 2000, 6(1): 83-105.
[20] MAHFOUD H. Expressive top-k matching for conditional graph patterns[J]. Neural Computing and Applications, 2021: 1-17.
[21] BLAU H, IMMERMAN N, JENSEN D. A visual language for querying and updating graphs[J]. University of Massachusetts Amherst Computer Science Technical Report, 2002, 37: 2002.
[22] FAN W F, WU Y H, XU J B. Adding counting quantifiers to graph patterns[C]// Proceedings of the 2016 International Conference on Management of Data. San Francisco: ACM, 2016: 1215-1230.
[23] FAN W F. Graph pattern matching revised for social network analysis[C]// Proceedings of the 15th International Conference on Database Theory. Cambridge, MA: IEEE, 2012: 8-21.
[24] QIAO F, ZHANG X, LI P, et al. A parallel approach for frequent subgraph mining in a single large graph using spark[J]. Applied Sciences, 2018, 8(2): 230.
[25] SENTHILSELVAN N, SUBRAMANIYASWAMY V, VIJAYAKUMAR V, Et al. Distributed frequent subgraph mining on evolving graph using SPARK[J]. Intelligent Data Analysis, 2020, 24(3): 495-513.
[26] WANG X, XIANG M, ZHAN H, et al. Distributed Top-k Pattern Mining[C]// Asia-Pacific Web(APWeb) and Web-Age Information Management(WAIM) Joint International Conference on Web and Big Data. Cham: Springer, 2021: 203-220.
[27] HUA G, ZHANG J, CUI L, et al. D-colSimulation: A Distributed Approach for Frequent Graph Pattern Mining based on colSimulation in a Single Large Graph[C]// 2020 IEEE International Conference on Services Computing(SCC). IEEE, 2020: 76-83.
[28] SAHOO A, SENAPATI R. A Novel Approach for Distributed Frequent Pattern Mining Algorithm using Load-Matrix[C]// 2021 International Conference on Intelligent Technologies (CONIT). IEEE, 2021: 1-5.
[29] YAN D, QU W, GUO G, et al. PrefixFPM: a parallel framework for general-purpose frequent pattern mining[C]// 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1938-1941.
[30] FARHAN HUSAIN M, DOSHI P, KHAN L, et al. Storage and retrieval of large rdf graph using hadoop and mapreduce[C]// IEEE International Conference on Cloud Computing. Springer, Berlin, Heidelberg, 2009: 680-686.
[31] BRINGMANN B, NIJSSEN S. What is frequent in a single graph? [C]// Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining. Berlin: Springer, 2008: 858-863.
[32] WANG X, XU Y, ZHAN H Y. Extending association rules with graph patterns[J]. Expert Systems with Applications, 2020, 141: 112897.
[33] RAHIMIAN F, PAYBERAH A H, GIRDZIJAUSKAS S, et al. Ja-be-ja: A distributed algorithm for balanced graph partitioning [C]// 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems. Philadelphia: IEEE, 2013: 51-60.
[34] KUMAR A, SINGH S S, SINGH K, et al. Link prediction techniques, applications, and performance: A survey[J]. Physica A: Statistical Mechanics and its Applications, 2020, 553: 124289.
[35] ROBERTO J, BAYARDO JR, AGRAWAL R. Mining the most interesting rules[C]// Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. San Diego: ACM, 1999: 145-154.
[36] VO B, BAC L E. Interestingness measures for association rules: combination between lattice and hash tables[J]. Expert Systems With Applications, 2011, 38(9): 11630-11640.

SHA Yuji, born in 1993, postgraduate. His main research interests include data mining and machine learning.

WANG Xin, born in 1981, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include knowledge discovery in database, artificial intelligence, machine learning and data mining.