arXiv:2301.06264v2 [cs.DB] 1 Jul 2023

亮点

一种发现图实体依赖关系（GED）的高效方法

Dehua Liu、Selasi Kwashie、Yidi Zhang、Guangtong Zhou、
Michael Bewong、吴晓莹、Xi Guo、Keqing He、Zaiwen Feng

- 关于图实体依赖关系（GED）发现问题的研究。
- 一种在属性图中发现图编辑依赖（GED）的新型高效方法。
- 一种受最小描述长度启发的图编辑依赖有趣度定义，用于对发现的规则进行排名。
- 对所提出的技术进行了全面的实证评估，并展示了挖掘出的有用规则示例，这些规则与数据质量和管理应用相关。

一种发现图实体依赖关系（GED）的高效方法

Dehua Liu[a]、Selasi Kwashie[b]、Yidi Zhang[a]、Guangtong Zhou[a]、Michael Bewong[c]、Xiaoying Wu[d]、Xi Guo[a]、Keqing He[d]、ZaiwenFeng[a,*]

[a]华中农业大学信息学院，中国湖北武汉[b]人工智能与网络未来研究所，查尔斯特大学，澳大利亚新南威尔士州巴瑟斯特[c]查尔斯特大学计算与数学学院，澳大利亚新南威尔士州沃加沃加[d]武汉大学计算机学院，中国湖北武汉

摘要

图实体依赖关系（GED）是面向属性图的一类新型图约束，可统一键和函数依赖。研究发现，它们在许多真实世界的数据质量和数据管理任务中都很有用，包括社交媒体网络中的事实核查和实体解析。本文研究图依赖的发现问题，即在给定图数据中找到有效图依赖的最小覆盖。我们对该问题进行了形式化，并提出一种有效且高效的方法，以克服图依赖发现中的主要瓶颈。具体而言，我们利用现有的图划分算法，实现快速的 GED 范围发现，并针对规模大到难以处理的候选依赖空间采用有效的剪枝策略。此外，我们基于最小描述长度原理定义了图依赖的趣味性度量，用于对挖掘出的图依赖覆盖集进行评分和排序。最后，我们通过在真实世界基准图数据集上开展大量实验，展示了图依赖发现方法的可扩展性和有效性

*通讯作者网址：Zaiwen.Feng@mail.hzau.edu.cn（Zaiwen Feng）

提交至《知识型系统》的预印本 *2023*年*7*月*4*日

；并展示所发现规则在不同下游数据质量管理应用中的实用价值。

关键词： 图实体依赖、图实体依赖发现、高效算法、数据依赖

---

1. 引言

近年来，研究者提出了适用于属性图的完整性约束（例如键 [1]和函数依赖 （FD） [2]），以便描述各种数据语义，并解决图数据质量与管理问题。图实体依赖（GED） [4, 5]是一种新的基础约束，它将键和函数依赖统一起来，适用于属性图。

图 G 上的GED φ 是一个二元组 $\varphi = （Q[\bar{u}], \ X \to Y）$，用于描述图模式 $Q[\bar{u}]$ 在G中的同态匹配上所满足的依赖关系 $X \to Y$。直观而言，由于图没有固定模式，图模式 $Q[\bar{u}]$ 用于标识 G 中应满足依赖关系 $X \to Y$的实体集合。一般来说，GED能够表达和编码关系型函数依赖与条件函数依赖（CFD）的语义，并且涵盖其他图完整性约束（例如图键 [1] 和GFD [2]）。

图编辑依赖在现实世界的数据质量和数据管理中有许多应用（更多详情请参见 [4, 5] ）。例如，它们已被扩展并应用于：社交媒体网络中的事实核查 [20], 图和关系中的实体解析 [3, 6, 22], 一致性检查 [21], 知识图谱生成 [51],等。

在本研究中，受到 GEDs 在

许多下游应用中的表达能力和实用性的启发，我们研究了如何从给定的属性图中自动学习 GEDs。 据我们所知，目前文献中尚不存在用于发现 GEDs 的算法，尽管已有研究分别在 [12] 和 [16] 中提出了发现图键和 GFDs 的方法。然而， [16] 中的 GFD 挖掘方法针对的是一类特殊的 GEDs（不含任何标识符字面量，并采用同构匹配语义）。此外， [16, 12] 研究了 RDFs 中键约束的发现。因此，仍需要有效且高效的技术来挖掘能够涵盖 GED 完整语义的规则。

发现数据依赖是数据库和数据挖掘领域一项活跃且由来已久的研究课题。事实上，关系数据环境下的这一问题已得到充分研究，关于函数依赖（FDs）已有大量研究成果 [8, 19, 44, 18]，其众多扩展形式（例如条件函数依赖 [7, 26, 24], 、基于距离的函数依赖 [10, 9, 14,15, 13], 等）也不例外。

数据依赖挖掘的总体目标，是找出一组不可再约简的有效依赖关系（又称最小覆盖集），这些依赖关系在给定输入数据上成立，且不包含任何蕴含或冗余。对于大多数依赖关系而言，这是一个具有挑战性且往往难以求解的问题，GEDs 也不例外。事实上，与其他图约束（例如 GFDs 和 GKeys）以及传统关系依赖相比，GEDs 的问题本质上更具挑战性，原因有三：a）存在作为拓扑约束的图模式（关系依赖中没有这种情况）；b）GED 依赖关系的属性集包含三种可能的文字（例如，GFDs 有 2 种，GKeys 有 1 种，CFDs 有 2 种）；c）蕴含与

GEDs 的验证分析——发现过程中的关键任务——分别被证明是 NP完全和 co-NP 完全（详情请见 [5] ）。

本文提出了一种高效且有效的图依赖挖掘方案。我们发现，图依赖挖掘的两大效率瓶颈在于：在大型图中发现模式，以及遍历规模大到难以承受的候选依赖搜索空间。因此，我们利用现有的图划分算法克服第一个挑战，并采用若干有效的剪枝规则和策略来解决第二个挑战。

我们将主要贡献总结如下。1）我们形式化了图编辑距离发现问题（第 3 节）。我们扩展了图依赖中平凡、约简和极小 依赖的概念，并提出了图依赖的形式化覆盖集查找问题。2）我们开发了一种新的两阶段方法，以便在大型图中高效地发现图依赖（第 4 节）。该方法利用现有的图划分算法快速挖掘图模式并找到其匹配。此外，我们设计了一套全面的属性（及实体）依赖发现架构，包含有效的候选生成、搜索空间剪枝和验证策略。此外，我们还为图依赖引入了一种新的有趣度评分，用于对发现的规则进行排序。3）我们在真实世界图上进行了大量实验，证明图编辑距离发现能够应用于大型图，并展示了所提方法的有效性、可扩展性和效率（第 5 节）。

2. 预备知识

本节介绍初步概念和定义。我们用 $\mathbf{A}$，$\mathbf{L}$，$\mathbf{C}$分别表示 属性、标签和常量的全集。

图、图模式和匹配的定义沿用 $[4, 5]$ 中的定义。

图。我们考虑一个有向属性图 $G = (V, E, L, F_A)$，其中：(1) V 是一个有限的节点集合；(2) E 是一个有限的边集合，由 $E \subseteq V \times \mathbf{L} \times V$ 给出，其中 (v, l, v') 是一条从节点 v 指向节点 v'、标签为 $l \in \mathbf{L}$ 的边；(3) 每个节点 $v \in V$ 都有一个特殊属性 id 表示其标识，以及一个取自 L 的标签 $L(v)$；(4) 每个节点 v 都关联一个属性值对列表 $F_A(v) = [(A_1, c_1) ..., (A_n, c_n)]$，其中 $c_i \in \mathbf{C}$ 是一个常量，$A_i \in \mathbf{A}$ 是 v 的一个属性，记作 $v.A_i = c_i$，且 $A_i \neq A_j$ if $i \neq j$。

图模式。图模式记作 $Q[\bar{u}]$，是一个有向图 $Q[\bar{u}] = (V_Q, E_Q, L_Q)$，其中：(a) V_Q 和 E_Q 分别表示模式节点集合和模式边集合；(b) L_Q 是一个标签函数，为每个节点 $v \in V_Q$ 和每条边 $e \in E_Q$ 分配一个标签；(c) $\bar{u}$ 是 V_Q 中的所有节点，称为（模式）变量。所有标签均取自 **L**，包括作为特殊标签的通配符“*”。如果两个标签 l，$l' \in \mathbf{L}$ 满足以下条件，则称它们匹配，记作 $l \asymp l'$：(a) $l = l'$；或 (b) l 或 l' 为“*”。

图模式 $Q[\bar{u}]$ 在图 G 中的匹配是一种从 Q 到 G 的同态 h，满足：(a) 对于每个节点 $v \in V_Q$，$L_Q(v) \asymp L(h(v))$；(b) 对于每条边 $e = (v, l, v') \in E_Q$，在 G 中存在一条边 $e' = (h(v), l', h(v'))$，使得 $l \asymp l'$。我们用 $H(\bar{u})$ 表示 $Q[\bar{u}]$ 在 G 中所有匹配的列表。下面的示例 1 展示了一个图、图模式及其匹配。

示例 1（图、图模式与匹配）。图 *1(a)* 展示了一个简单的图；*(b)–(e)* 是四种图模式的示例。$Q_1[x, y]$ 描述了两个通用（标记为“*”）节点 x 之间的“`is a`”关系

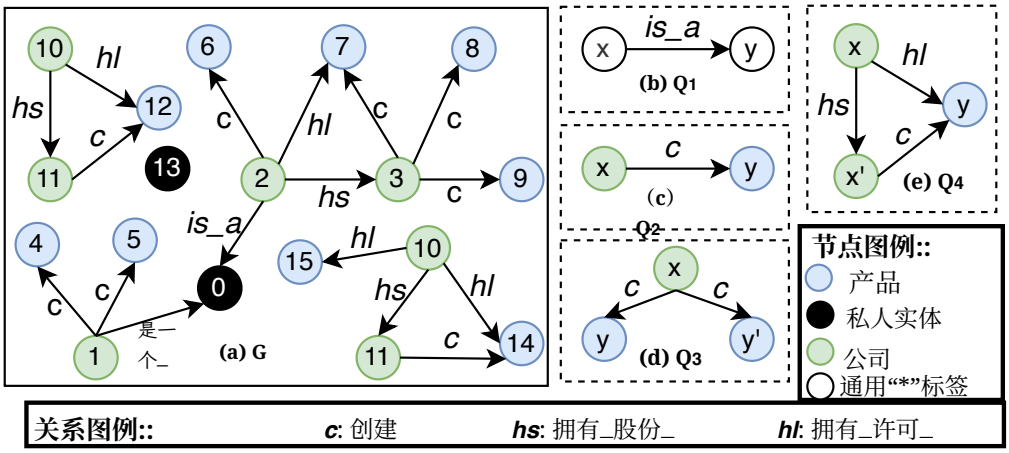


图1：示例：(a) 图；(b)–(d) 图模式

和 y。此模式在示例图中的匹配列表为 $H_1(x, y) = [\{1, 0\}, \{2, 0\}]$。$Q_2[x, y]$ 描述了一个带有与 product 节点 y 之间的 create 关系的 company 节点 x，其在 G 中的匹配为：$H_2(x, y) = [\{1, 4\}\{1, 5\}\{2, 6\}, \{3, 7\}, \{3, 8\}, \{3, 9\}, \{11, 12\}, \{11, 14\}]$。$Q_3[x, y, y']$ 指定了一个与两个 product 节点 y, y' 之间存在 create 关系的 company 节点 x。因此，在 G 中的匹配为 $H_3(x, y, y') = [\{1, 4, 5\}, \{3, 7, 8\}, \{3, 7, 9\}, \{3, 8, 9\}]$。$Q_4[x, x', y]$ 描述了一个与另一个 company x' 之间存在 has shares in 关系、并与由 x' 创建的 product y 之间存在一条 has license to 关系的 company x。$H_4(x, x', y) = [\{2, 4, 7\}, \{10, 11, 12\}, \{10, 11, 14\}]$。

图实体依赖。图实体依赖（GED）[4]φ 是一个二元组（$Q[\bar{u}]$, $X \to Y$），其中，$Q[\bar{u}]$ 是一个图模式，而 X，Y 是 $\bar{u}$ 中两个（可能为空的）文字集合。$\bar{u}$ 的文字 w 是以下三种约束之一：（a）$x.A = c$，（b）$x.A = y.A'$，（c）$x.id = y.id$，其中 x，$y \in \bar{u}$ 是模式变量，A，$A' \in \mathbf{A}$ 是非-id 属性，而 $c \in \mathbf{C}$ 是一个常量。$Q[\bar{u}]$ 和 $X \to Y$ 分别称为模式/范围

以及 φ 的依赖关系。

给定一个图依赖 $\varphi = (Q[\bar{u}], X \rightarrow Y)$，若匹配 $h(\bar{u})$ 满足 $Q[\bar{u}]$ in G *satisfies* 的文字 w of $\bar{u}$，记作 $h(\bar{u}) \models w$，当且仅当：(a) 当 w 为 $x.A = c$ 时，属性 $h(x).A$ 存在，且 $h(x).A = c$；(b) 当 w 为 $x.A = y.A'$ 时，属性 $h(x).A$ 和 $h(y).A'$ 均存在且值相同；(c) 当 w 为 $x.id = y.id$ 时，$h(x).id = h(y).id$。

若匹配 $h(\bar{u})$ 满足文字集合 X of，则 $h(\bar{u})$ 满足其中每个文字 $w \in X$（$h(\bar{u}) \models Y$ 的定义类似）。若$h(\bar{x}) \models X$ 蕴含 $h(\bar{x}) \models Y$，则记作 $h(\bar{u}) \models X \rightarrow Y$。

图 G 满足图依赖 φ，记作 $G \models \varphi$，如果对于 $Q[\bar{u}]$ 在 G 中的所有匹配 $h(\bar{u})$，$h(\bar{x}) \models X \rightarrow Y$。图 G满足图依赖集合 Σ，记作 $G \models \Sigma$，如果对于所有 $\varphi \in \Sigma$，$G \models \varphi$。

我们通过下方图1中的示例图和图模式来说明图编辑依赖的语义。

示例*2*（图编辑依赖）。我们使用图 *1(b)–(e)*中的图模式，定义示例图图 *1(a)*上的示例图编辑依赖。*1)* φ_1：*(*$Q_1[x, y]$*,* $y.A =$*y.A* $\rightarrow$ $x.A$ $=$ $y.A$*)*——此图编辑依赖表示，对于图模式 $Q_1[x, y]$ 在 G 中的任意匹配 *h(x, y)*（即 x `is_a` y），如果节点 *h(y)*具有属性 A，那么 *h(*x*)*在 A上的值必须与 *h(y)* 相同。*2)* φ_2：*(*$Q_2[x, y]$*,* $\emptyset$ $\rightarrow$*y.creator* $=$ $x.name$*)*——对于 Q_2 在 G 中的每个匹配 *h(x, y)*（即 `company`*h(*x*)* `create product` *h(*y*)*），*h(*$y.creator$*)*和 *h (*$x.name$*)*必须具有相同的值。*3)* φ_3：*(*$Q_3[x, y, y']$*,* $\emptyset \rightarrow y.creator = y'.creator$*)*——这表示，对于 G 中的任意匹配 *h(x, y, y′)*（即 `company` *h(*x*)* `create` 两个产品 *h(y, y′)*），y, y' 在其属性 $creator$上的值必须相同。*4)* φ_4：*(*$Q_4[x, x', y]$*,* $x'.country =$ `USA`$\rightarrow$ $x.country =$ `USA`*)*——对于每个匹配

$h(x, x', y)$ 在 G中的 Q_4，如果 $h(x'.country)$ 是 USA，那么 $h(x.country)$ 也必须是 USA。请注意，φ_4 在 G上施加的限制是：美国任何一家公司的股权/许可所有权仅限于美国公司。

3. 问题定义

给定一个属性图 G，一般的图编辑依赖发现问题是找出一组 Σ图编辑依赖，使得 $G \models \Sigma$。然而，与其他依赖关系发现问题类似，这样的一组依赖可能规模庞大且充斥冗余（参见 [16, 22, 44]）。因此，与 [16], 一致，我们希望在持久图模式上找到一组覆盖性的、非平凡且非冗余的依赖关系。下文将正式介绍我们关注的图编辑依赖，并给出问题定义。

持久图模式。图模式可以视为图编辑依赖中的"宽松模式" [5]。因此，为了发现图编辑依赖，在输入图数据中找到持久图模式至关重要。然而，统计子图的频率并非易事，且根据所采用的计数指标（例如有害重叠 [30], 和最大独立集 [29]），这一任务可能难以处理。

由于基于最小映像的支持度具有反单调性和可处理性，我们采用基于最小映像的支持度（MNI） [45]指标来统计给定图中图模式的出现次数。

设 $I = \{i_1, \cdots, i_m\}$ 为模式 $Q[\bar{u}]$ 到图 G 的同构集合。设 $M(v) = \{i_1(v)$ $\cdots$， $i_m(v)\}$ 为集合，该集合包含 G 中函数 i_1，$\cdots$，i_m 映射节点 $v \in V$ 时涉及的（互不相同的）节点。Q 的基于最小映像的支持度（MNI）在

G 中，记为 $mni(Q,\ G)$，定义如下：

$$mni(Q,G) = min\{x \mid x = |M(v)|,\ \forall\ v \in V\}. \tag{3.1}$$

如果图模式 $Q[\bar{u}]$在图 G中满足$mni(Q,\ G) \geq \tau$，我们就称其具有持续性（即频繁），其中 τ是用户指定的最小 MNI 阈值。

代表性图模式。 对于许多真实世界图而言，受其规模影响（参见 [31, 32]），使用通用计算资源挖掘频繁图模式的效率低得难以接受。为缓解这一性能瓶颈，在我们的流水线中，我们考虑输入图中稠密连接社区内频繁出现的图模式。我们将此类图模式称为代表性图模式，并采用常数 Potts 模型（CPM） [35]作为图中社区的质量函数。

形式化地，社区 c 的 CPM 定义为

$$\mathcal{H} = \sum_{c} \left[e_c - \gamma \binom{n_c}{2} \right] \tag{3.2}$$

其中， e_c 和 n_c 分别是 c 中的边数和节点数；$\gamma > 0$ 是（用户指定的）分辨率参数—— γ 越高，社区数量越多，反之亦然。直观而言，分辨率参数 γ限制社区内部密度不低于 γ，并限制社区间密度不高于 γ。

因此，如果图模式 $Q[\bar{u}]$ 在至少一个 γ 稠密社区 $c \in G$ 中是 τ-频繁的，我们就称其具有代表性，或称其为（γ，τ）-频繁的。

平凡 GED。 我们称 GED $\varphi : (Q[\bar{u}],\ X \rightarrow Y)$ 为平凡的，当且仅当：(a) X 中的文字集合无法满足（即 X 的计算结果为 `false`）；或 (b) Y 可由 X 推导得出（即 $\forall\ w \in Y$、 w 可通过等号运算符的传递性从 X 推导得出）。我们考虑非平凡 GED。

¯**约简 GED**。给定两个模式 $Q[\bar{u}] = (V_Q, E_Q, L_Q)$ $Q'[u'] = (V'_{Q'}, E'_{Q'}, L'_Q)$，如果满足以下条件之一，则称 $Q[\bar{u}]$ 将 $Q'[\bar{u'}]$约简，记为 $Q \ll Q'$：(a) $V_Q \subseteq V'_{Q'}$、$E_Q \subseteq E'_{Q'}$；或 (b) L_Q 将 $L'_{Q'}$ 中的某些标签升级为通配符。也就是说，Q 的拓扑约束比 Q'更宽松。

给定两个图编辑依赖，$\varphi = (Q[\bar{u}], X \to w)$ 和 $\varphi' = (Q'[u'], X' \to w')$。如果 φ 能够约简 φ'，记作 $\varphi \ll \varphi'$，则满足：(a) $Q \ll Q'$；以及 (b) $X \subseteq X'$ 和$w = w'$。

因此，若图 G 中的图依赖 φ 满足：(a) $G \models \varphi$；并且对于任意满足 $\varphi' \ll \varphi$ 的 φ'，都有 $G \not\models \varphi'$，则称其为**约简的**。如果图依赖 φ 既是非平凡的又是约简的，则称其为**极小的**。

图编辑依赖覆盖。给定一个图编辑依赖集合 Σ，满足 $G \models \Sigma$。当且仅当对于所有 $\varphi \in \Sigma$，都有 $\Sigma \not\equiv \Sigma \setminus \varphi$ 时，Σ 是极小的。也就是说，Σ 不包含冗余图编辑依赖。如果满足：(a) $\Sigma_c \subseteq \Sigma$；(b) $\Sigma_c \equiv \Sigma$；(c) 所有$\varphi \in \Sigma_c$ 均为极小的；并且 (d) Σ_c 为极小的，则称 Σ_c 是 G 上 Σ 的覆盖。

我们研究以下图编辑依赖发现问题。

定义 1（图编辑依赖的发现）。给定一个属性图 G，以及用户指定的分辨率参数和 *MNI* 阈值 γ、τ：找出覆盖集 Σ_c，其中包含所有在 G 中的(γ、τ)-频繁图模式上成立的有效最小图编辑依赖。

4. 提出的发现方法

本节介绍一种新颖且高效的 GED 发现方法。图 2 展示了我们解决方案的整体流水线，包含两个主要组成部分：(a) 查找图中具有代表性且经过精简的图模式（即“作用域”）及其（同态）匹配；以及 (b) 查找

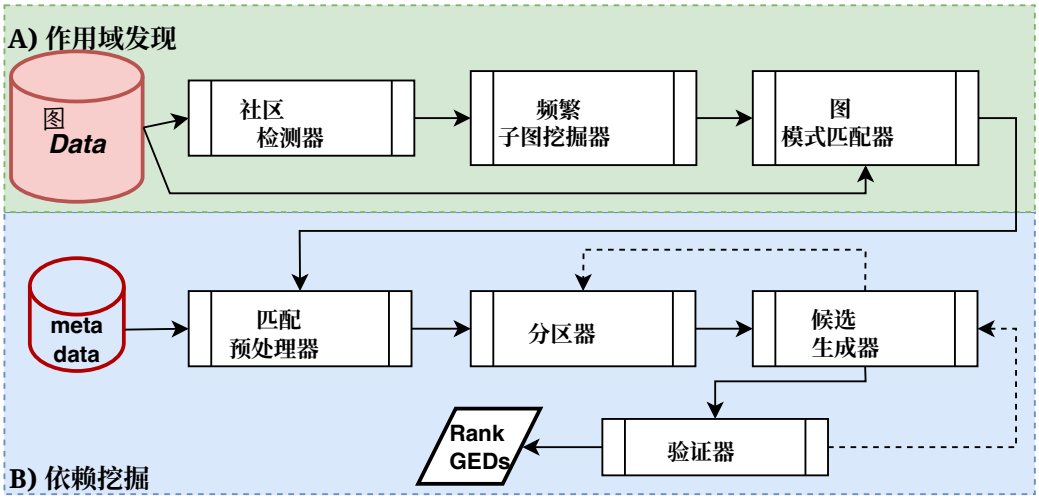


图 2：图编辑距离发现流程总览。

在图模式上成立的最小属性（实体）依赖关系。算法 1 展示了所提出的 GED 发现过程的部分伪代码，其中包含两个主要过程，分别用于执行上述任务（参见算法 1 的第 2 行和第 3–5 行）。该算法所需的输入包括：属性图 G；用户指定的分辨率参数 γ 和最小 MNI 阈值 τ。

4.1. 图模式发现与匹配

发现过程的首要任务是找出代表性图模式及其在给定图中的匹配。我们提出了一个包含三个阶段的任务流程，即：(i) 检测输入图中的稠密社区；(ii) 在社区中挖掘频繁图模式；(iii) 在输入图中查找已发现（并经过缩减）的图模式的匹配。

过程 1，mineScopes(G，γ，τ)，给出了这项任务的伪代码（工作流程如图 2 A 所示）。

算法 1 DisGEDs

要求： 一个图，G；CPM 分辨率参数，γ；MNI 阈值，τ。

确保： GED 的一个覆盖集 Σ_c（包含 τ 频繁图模式），位于 G

1: $\Sigma_c := \emptyset$; $\Sigma := \emptyset$; $\mathcal{S} := \emptyset$

2: $\mathcal{S} \leftarrow mineScopes(G, \gamma, \tau)$ ▷ 查找图模式及其匹配项

3: **对于每个** $(Q_i, H(Q_i, G)) \in \mathcal{S}$ **执行** ▷ 挖掘最小图编辑依赖

4: $\Sigma_i \leftarrow mineDependencies(Q_i, H(Q_i, G))$

5: $\Sigma \leftarrow \Sigma \cup \Sigma_i$

6: $\Sigma_c \leftarrow findCover(\Sigma)$ ▷ 移除所有传递蕴含的图编辑依赖

7: **返回** Σ_c

4.1.1. 稠密社区检测

我们方法的第一步很简单：对输入进行划分将图 G 划分为多个稠密社区，$\mathcal{C} = \{c_1, \cdots, c_k\}$。具体而言，我们采用方程 3.2 中的质量函数，以确保每个社区 $c_i \in \mathcal{C}$，$(i \in [1, k])$ 的密度不低于 γ。通常，任何高效的图聚类算法都可以用于此任务；我们采用 `Leiden` 图聚类算法 [37]

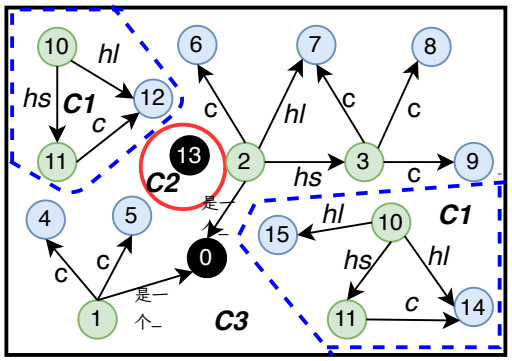


图 3：示例图 G 中的 0.01 稠密社区：$C1$，$C2$，$C3$ 将输入图划分为 γ -稠密的

使用 CPM 质量函数进行划分社区（参见过程 1 第 2 行）。

例如，来看图1 (a) 中的示例图。当分辨率值为 $\gamma = 0.01$时，如图 3 所示，会生成三个稠密社区。

这包括： c_1，由蓝色虚线圈出的部分图； c_2，单个节点 13；以及 c_3，其余部分图。

过程 1 mineScopes(G, γ, τ)

要求： 图， G；CPM 分辨率参数， γ；MNI 阈值， τ

确保： 约简模式 Q_i 及其同态匹配 (,) 在 中的集合 $\mathcal{S} = \{(Q_1, H(Q_1, G)) \cdots\}$

同态匹配 $H(Q_i, G)$ 在 G 中

1: $\mathcal{C} := \emptyset; \mathcal{Q} := \emptyset; \mathcal{S} := \emptyset; G' = \emptyset$

2: $\mathcal{C} \leftarrow$ `Leiden` (G, γ) ▷ 挖掘稠密社区

3: </style>对 $c_i \in \mathcal{C}$ 执行 ▷ 挖掘频繁超图模式

4: $\mathcal{Q}_i \leftarrow$ `GraMi`$(c_i, \ \tau)$

5: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{Q}_i$

6: $\mathcal{Q}' \leftarrow reduceGraphPatterns(\mathcal{Q})$

7: </style>对 $Q \in \mathcal{Q}'$ 执行 ▷ 查找同态匹配

8: $G' \leftarrow$ `filter`$(G, \ Q)$ ▷ 按边/节点筛选 G w.r.t. Q

9: $H(Q, G) \leftarrow$ `match`(Q, G')

10: $\mathcal{S} \leftarrow \mathcal{S} \cup (Q, \ H(Q, \ G))$

11: **返回** $\mathcal{S}$

4.1.2. 代表性图模式挖掘

接下来，我们将发现的 γ-稠密社区 $\mathcal{C} = \{c_1, \cdots, c_k\}$ 作为输入，并返回一组频繁 且经过精简的图模式，使用同构匹配和 MNI度量（见公式 3.1）对社区中的模式出现次数进行计数（参见过程 1 的第 3–6 行）。

我们对 `GraMi` 图模式挖掘算法 [28] 进行了调整，以找出所有在 $c_1, \cdots,$ c_k中满足 τ频繁条件的模式所构成的集合$\mathcal{Q}$。正如预期的那样， $\mathcal{Q}$ 可能

包含冗余模式。因此，我们设计了一种简单而有效的算法——函数 1，用于剔除 $\mathcal{Q}$中的所有冗余模式，并返回由 G 中的代表性约简模式构成的集合 $\mathcal{Q}'$（即不存在任何一对模式 $Q, Q' \in \mathcal{Q}'$ 满足 $Q \ll Q'$ 或 $Q' \ll Q$）。

函数 1 $reduceGraphPatterns(\mathcal{Q})$

1: $n \leftarrow |\mathcal{Q}|$; $\mathcal{Q}' := \emptyset$; $\vec{T} \leftarrow [false, false, ..., false]$
2: $Sort(\mathcal{Q})$ ▷ 按模式大小降序排列
3: **对于** $i \leftarrow 0$ 到 $n-1$，**执行**
4: **if** $\vec{T}[i] == false$ **然后**
5: **对** $j \leftarrow i+1$ 执行 $n-1$ **执行**
6: **if** $\vec{T}[j] == false$ **则**
7: **如果** count(VF2U($\mathcal{Q}(i)$ $\mathcal{Q}(j)$)) > 0 **则**
8: $\vec{T}[j] \leftarrow true$
9: $\mathcal{Q}' \leftarrow \mathcal{Q}' \cup \mathcal{Q}(i)$
10: **返回** $\mathcal{Q}'$

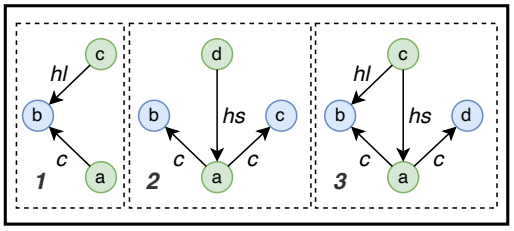
函数 1 的输入是一组 $\mathcal{Q}$ 个 τ-频繁图模式，这些模式来自 γ-稠密社区（即 (γ, τ)-频繁图模式或代表性图模式）；输出则是一组 $\mathcal{Q}'$ 个没有冗余的图模式。

令 n 为 $\mathcal{Q}$的大小 $|\mathcal{Q}|$; $\vec{T}$ 为长度为 n的向量，其中索引 $i \in [0, n-1]$ 处的每个位置 $\vec{T}[i]$ 都存储一个布尔值。每个条目 $\vec{T}[i]$都与一个图模式 $Q_i \in \mathcal{Q}$相关联，并表示相应的图模式是否冗余。

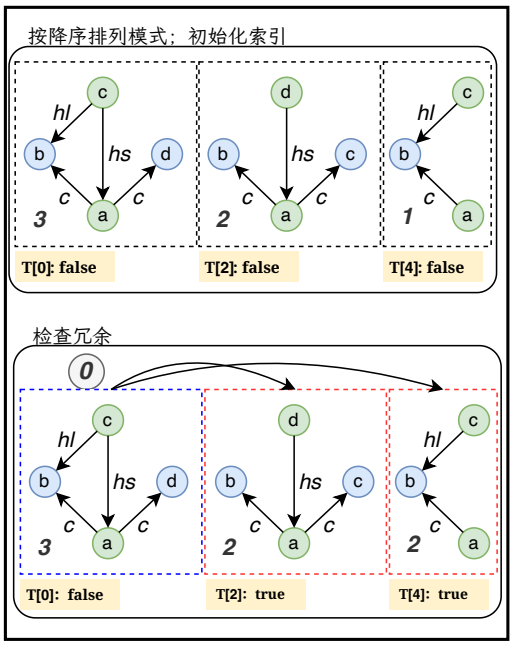
$\vec{}$变量 n, $\mathcal{Q}'$ 和 T 在函数 1 第 1 行中初始化。随后， $\mathcal{Q}$ 中的图模式按照每个图模式中边的数量从多到少排序（第 2 行）。

量。之后，我们评估

用于冗余判断的图模式对（第 3–9 行）。具体而言，函数 VF2U和 count 用于检查索引为 i、j 的 $\mathcal{Q}$中的模式之间的同构匹配，并分别计算返回的同构匹配数量。我们通过放宽一些条件，对 VF2 算法[43]的高效实现[1] 进行了扩展，如 VF2U所示。主要区别在于，在 VF2U中，查询图模式查询的不是诱导子图，而是目标图模式中与其同构的子图。



(a) 示例图 G（图 1(a)）中的 2-频繁模式示例



(b) 频繁图模式冗余剪枝示意图
频繁图模式

图 4：频繁子图剪枝

对于图模式的两两比较，将索引为 j 的模式视为查询模式，而将索引为 i 的模式视为

目标模式。算法从 $\mathcal{Q}$ 中最大的模式开始迭代（第 3 行）。在第 i 轮迭代中，索引为 i 的最大模式被用作 $\mathcal{Q}(i)$，如果

[1]https://github.com/MorvanHua/VF2

它带有 $false$ 标记。由于 $\mathcal{Q}$（j），会检查索引从 $i+1$ 到 $n-1$ 的每个模式，以判断其是否与 $\mathcal{Q}$（i）同构。 VF2U 返回从 $\mathcal{Q}$（j）到 $\mathcal{Q}$（i）的同构， count 返回所得集合的大小。

在每次迭代中，如果索引为 $\mathcal{Q}(j)$ 的 $\mathcal{Q}(i)$ 与索引为 $\mathcal{Q}(j)$ 的 $i+1$ ($n-1$) 同构，则将索引为 j ($j > i$) 的模式标记为 $true$（即第 8 行），并且在后续迭代中，索引为 $\mathcal{Q}(j)$ 的 $i+1$ ($n-1$) 不会成为目标图。检查完索引从 $\mathcal{Q}$ 到 i 的所有模式后，将 $\mathcal{Q}'$ (i) 添加到 $\mathcal{Q}'$ ，本轮迭代结束。最后，处理完所有图模式后，返回集合 作为输出，该集合存储了所有无冗余的图模式。

我们通过示例 3 简要说明这一简化过程。

示例 3（代表性模式）。给定图 *3* 中发现的 0.01 密集三社区，设最小支持度为 $\tau = 2$，用于在各社区中挖掘频繁子图。图 *4*（*a*）展示了三个频繁但冗余的子图。下面结合图 *4*（*b*）中的示意图说明剪枝策略。

首先，我们按边数递增的顺序对子图排序，并将向量 $\vec{T}[i]$ 中对应的索引 i 初始化为 false。接下来，我们遍历该向量，从位于 i 的最大子图和位于 $j = i+1$ 的下一个子图开始，检查 j 处的模式是否包含于 i 处的模式中。如果 i 处的模式包含该模式，则将 $\vec{T}[j]$ 处的索引设为 true 。

最后，我们返回索引为 k 的所有模式（如果 $\vec{T}[k]$ 为 false ，在此情况下，索引为 0的模式 3）。 □

4.1.3. 同态模式匹配

给定上一步得到的约简代表图模式集合 $\mathcal{Q}'$，我们在输入图中查找它们的同态匹配（参见过程 1 的第 7–10 行）。为确保对任意模式$Q \in \mathcal{Q}'$进行高效匹配，我们根据 Q中的节点和边类型对输入图进行剪枝，以生成输入图的简化/过滤版本；然后使用 Q在简化图上执行同态子图匹配。

基于给定图模式 Q 的简化图 G'，是指移除了其中未出现在 Q 中的节点类型和边类型（标签）的图

这些类型和标签已从原始图 G 中移除。相对于图模式而言，简化输入图可显著缩小候选匹配的搜索空间，并提高匹配过程的效率。我们采用 [38]中介绍的高效最坏情况最优连接（WCOJ）算法，在 G' 中查找 Q 的所有同态匹配。

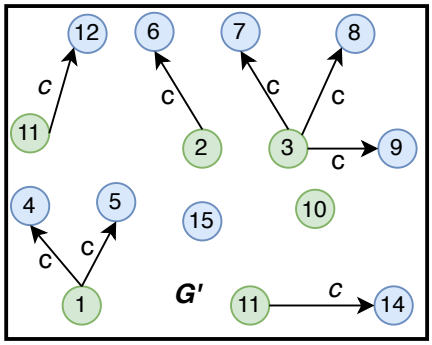


图5：相对于 Q_3 的简化 G

以我们的示例为例，为简洁起见，假设我们分别要将图1 (a) 和 (d) 中的示例图 G 相对于图模式而言， $Q_3[x, y, y']$ 进行简化。得到的简化图 G' 如图5所示。

对于基于某个图模式的每组子图匹配记录，通过组合图模式中实体的所有属性，以及每个匹配子图中实体的所有属性值，可以生成一个大型属性关系数据表。该伪关系表存储图模式中实体的所有属性值。第一行表示图模式中包含的实体的所有属性，其余每一行均为一个匹配子图的信息，

表示匹配子图中实体的所有属性值。图模式 Q_3 上的伪关系示例如图6所示（来自图1中的运行示例）。

4.2. 属性（实体）依赖发现

在此，我们针对上一步发现的图模式所产生的匹配对应的伪关系表挖掘依赖关系。如图2(B)所示，依赖发现架构由四个关键模块组成，分别是：(a) 匹配预处理器，(b) 分区器，(c) 候选生成器，以及 (d) 验证器。

下面我们将依次讨论各个模块的功能，并说明如何针对图中模式 Q 的匹配 $H(Q,\ G)$ 执行整体依赖挖掘算法（如过程 2 所示）。

4.2.1. 模式匹配预处理

这是依赖挖掘过程中的一个重要（但可选的）第一步。如果存在关于输入图的任何领域知识（例如以元数据的形式），便可利用这些知识来增强发现过程。例如，给定任意图模式及其匹配列表，我们可以根据可用的元数据确定：(a) 每个模式变量需要考虑哪些属性集；(b) 生成变量和 id 文字时需要考虑哪些模式变量对。

也就是说，此阶段允许对图模式匹配结果的伪关系表进行任何相关的预处理，以便高效地开展后续

过程 2 挖掘依赖关系($Q, H(Q, G)$)

要求： 图模式 Q及其在 G中的同态匹配 $H(Q, G)$

确保： 在 G中， Q 上的最小图编辑依赖集合 Σ

1: $\Sigma := \emptyset; \mathbb{L}[0] := \emptyset$; Q中唯一属性的数量为 $N :=$

2: $\mathcal{T} \leftarrow$ 在 Q上构成 $H(Q, G)$的伪关系

3: $\mathbb{L}[1] \leftarrow generateLiteralSetPartitions(Q[\bar{u}], \mathcal{T})$

4: 左侧 $\leftarrow \mathbb{L}[0]$; 右侧 $\leftarrow \mathbb{L}[1]$

5: $\Sigma \leftarrow validateDependencies$(左侧,右侧)

6: **对于** $i = 2$ 到 $i = N - 1$ **执行**

7: **如果** $|\mathbb{L}[i-1]| > 1$ **则**

8: **对于每个**对 $(n_1, n_2) \in \mathbb{L}[i-1] \times \mathbb{L}[i-1]$ **执行**

9: $n = n_1 \cup n_2$

10: **如果** n 是可接受的**则** ▷ 引理 2

11: $\mathbb{L}[i] := \mathbb{L}[i] \cup n$

12: 左侧 $\leftarrow \mathbb{L}[i-1]$; 右侧 $\leftarrow \mathbb{L}[i]$

13: $\Sigma := \Sigma \cup validateDependencies$(左侧，右侧)

14: **返回** Σ

挖掘任务。

4.2.2. 分区

接下来，我们将输入表转换为适合高效挖掘规则的数据结构。在下文中，我们将等价类和划分的概念扩展到模式匹配上的属性值对。这使我们能够调整并使用相关的数据结构，并高效地验证候选依赖。

Table 1: Pseudo-relation over pattern (Q3) variable attributes

Pattern variables	x		y					y'			
Attributes / Matches	id	Name (A1)	id	Name (A2)	Genre (A3)	Year (A4)	Price (A5)	id	Name (A2)	Genre (A3)	Year (A4)
h1	1	GL	4	AF9	Racing	2018	£50	5	AF11	Racing	2018
h2	3	EA	7	F20	Soccer	2019	£55	8	F20	Soccer	2019
h3	3	EA	7	F20	Soccer	2019	£55	9	F21	Soccer	2020
h4	3	EA	8	F20	Soccer	2019	£60	9	F21	Soccer	2020

图6：图1(d)中模式 Q_3 上伪关系的示例

定义2（等价类、划分）。令 H 为图模式 $Q[\bar{u}]$ 在 G 中所有匹配的集合，X 为 $\bar{u}$ 中的文字集合。用 $\pi(X, H)$ 表示由文字集合 X 对 H 进行的划分，即由非空且互不相交的子集构成的集合，其中每个子集包含所有满足 X 且取值相同的匹配。为简洁起见，在上下文清楚时，我们将 $\pi(X, H)$ 记作 $\pi(X)$。

给定 X, H: $\pi(X, H) = \bigcap_{w \in X} \pi(w, H)$，其中 w [2] 是 $Q[\bar{u}]$ 中模式变量上的文字。两组文字 X, X' 关于 H等价，当且仅当：$\pi(X, H) = \pi(X', H)$。 □

根据图依赖满足关系的语义和定义2，以下引理成立。

引理 1。依赖关系 $X \to Y$ 在模式 $Q[\bar{u}]$ 上于 G 中成立，当且仅当：$\pi(X, H) = \pi(X \cup Y, H)$。

在函数2中，我们展示了如何对给定伪关系表 $\mathcal{T}$ 中的 $Q[\bar{u}]$ 模式变量进行划分的过程。第 2–5 行生成常量

[2]for $x, y \in \bar{u}$, w can be: $(x.A = c)$, or $(x.A = y.A')$, or $(x.id = y.id)$

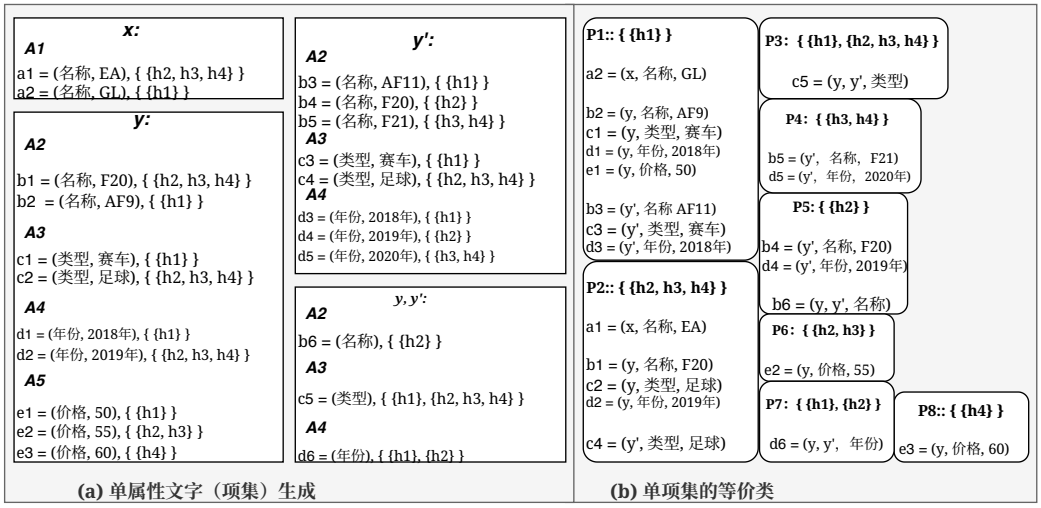


图 7：模式变量划分与等价类示例。

函数 2 $generateLiteralSetPartitions(Q[\bar{u}], \mathcal{T})$

1: $\mathbb{L}[1] = \emptyset$

2: **对于每个**模式变量 $x \in Q[\bar{u}]$ **执行** ▷ 常量字面量

3: **对每个** 属性 $x.A \in \mathcal{T}$**执行**

4: $L := \{X = (c,\ x.A) \mid \forall\ c \in adom(x.A) \wedge \pi(X,\ H(Q,\ G)) \neq \emptyset\}$

5: $\mathbb{L}[1]$.add(n): $n = (X, \pi(X, H))$

6: **对每个** 配对 $x, y \in Q[\bar{u}]$ **执行** ▷变量字面量

7: $L := \{X = (x.A = y.A') \mid A, A' \in \mathcal{T} \wedge \pi(X, H(Q, G)) \neq \emptyset\}$

8: $\mathbb{L}[1]$.add(n): $n = (X, \pi(X, H))$

9: **返回** $\mathbb{L}[1]$

文字，而函数的第 6–8 行生成变量和 id 文字。我们指出，前一步进行的适当预处理（基于领域知识）可确保仅考虑语义上有意义的模式变量对来生成变量文字。

在下面的示例中，我们展示一个简单示例中的划分和等价类。

示例 4. 假设图 *1* 中模式 Q_3 的同态匹配的属性由图 *6* 中的伪关系表示。图 *7* 展示了由此得到的划分和等价类。值得注意的是，在此示例中，由于图 *6* 中的节点标识符唯一，因此不会生成 id文字。

4.2.3. 候选依赖生成

在此，我们将讨论生成候选依赖所用的数据结构和策略。我们通过属性格 [39] 扩展并建模可能依赖关系的搜索空间。我们采用逐层自顶向下搜索方法来生成候选规则、测试/验证这些规则，并利用有效规则剪枝搜索空间。

通常，格中的一个节点 n 是一对 $n = (X, \pi(X))$。不过，为了非冗余地生成和剪枝候选集合 X, Y，我们会按照各自的模式变量，将每个节点的文字集合分别组织起来。例如，图 8(a) 展示了用于生成候选依赖的格的快照（对应图 6 和图 7 中的示例）。此外，图 8 的部分 (b)、(c) 和 (d) 分别展示了第 1 层的所有节点，以及第 2 层和第 3 层中的部分节点。

为了生成格中的第一级节点 $\mathbb{L}[1]$，我们使用上一步中得到的单属性变量划分（即函数2的输出）。此外，为了推导格中更高层级的节点，我们遵循引理 2 中的以下原则，以确保生成的文字集合无冗余且有效。

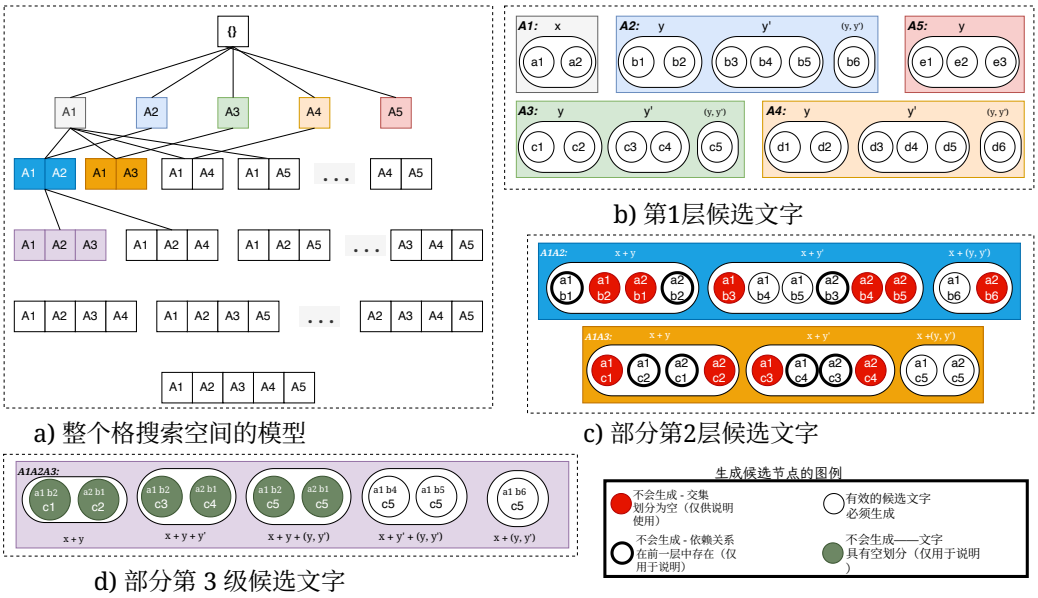


图 8：图依赖格：生成、搜索与遍历（彩色查看效果最佳）

引理 *2*（可接受节点）。设 $\mathbb{L}[i]$ 为格的当前层级 i 上的节点列表，其中至少包含两个节点 $\mathbb{L}[i] = [n_1 = (X_1, \pi(X_1))\ n_2 = (X_2, \pi(X_2))]$。我们使用 n_1、n_2 生成新节点 $n = (X, \pi(X)) \in \mathbb{L}[i+1]$，该节点具有以下性质：***(a)*** $X = X_1 \cup X_2$；且 ***(b)*** $\pi(X) = \pi(X_1) \cap \pi(X_2)$。当且仅当满足以下条件时，我们称 n 为可接受节点：***(i)*** $|X| > i$；***(ii)*** $\nexists\, w, w' \in X$，使得 $x.A \in w, w'$；以及 ***(iii)*** $\pi(X) \neq \emptyset$。 □

对于每一对节点 n_1，$n_2 \in \mathbb{L}[i]$，若由此产生一个可接受节点 $n \in \mathbb{L}[i+1]$，我们便建立边 $n_1 \to n$ 和 $n_2 \to n$ 来表示候选依赖关系。下一步将测试所有候选依赖关系的有效性。下面通过贯穿全文的示例说明候选依赖生成过程。

函数 3 $validateDependencies$(左侧,右侧)

1: $\Sigma := \emptyset$

2: **对于每个** 节点 $n_i = (X_i, \pi(X_i)) \in$ 左侧 **执行**

3: **对于每个** 节点 $n_j = (Y_j, \pi(Y_j)) \in$ 右侧 **执行**

4: **如果** $X_i \subset Y_j$ 且 $\pi(X_i) = \pi(Y_j)$ **则**

5: $\sigma := (Q[\bar{u}], \quad X_i \to (Y_j \setminus X_i))$

6: $\Sigma := \Sigma \cup \sigma$

7: 右侧 := 右侧 $\setminus n_j$

8: **返回** Σ

例 5（生成和搜索格空间）。图 *8* 展示了通用属性格（左）以及与部分属性集相关联的扩展文字集合节点（右）。为简洁明了起见，我们省略节点的划分集，并用颜色标示不允许节点。

$\mathbb{L}[0], \mathbb{L}[1]$ 中的候选规则形式为： $\emptyset \to Y$，其中 Y是 $\mathbb{L}[1]$ 中的单属性文字（即每个第*1*层节点）。 $\mathbb{L}[1], \mathbb{L}[2]$ 中边 $A1 \to A1A2$ 上的候选规则集合形式为： $X \to Y$，其中$X \in \{\{a1\}, \{a2\}\}$ 和 $Y \in \{\{a1, b1\}, \{a1, b2\}, \cdots, \{a1, b6\}, \{a2, b6\}\}$。 □

4.2.4. 依赖验证

我们采用对搜索空间进行逐层遍历的方式，验证格中相邻两层之间生成的候选规则。也就是说，对于任意两层 $\mathbb{L}[i]$ 和 $\mathbb{L}[i+1]$，它们分别对应左侧和右侧候选项的集合。对于左侧、右侧中的任意边 $X \to Y$，我们测试依赖关系$X \to Y \setminus X$。函数 3 执行此任务，其过程不言自明。在第 4 行，$X_i \subset Y_j$ 对所有候选依赖关系均为真（即该边

$X_i \to Y_j$ 存在）。如果依赖关系成立（即根据引理 1，$\pi(X_i) = \pi(Y_j)$），我们就将图依赖添加到集合中，并相应地剪枝节点搜索空间（第 5–7 行）。

在例 6 中，我们展示了运行示例中一些候选 GED 的验证。

例 6（候选 GED 的验证）。考虑例 *5* 中格的第*1*层里的候选 *GED* $\mathbb{L}[0], \mathbb{L}[1]$ 。很容易验证这些候选都无效，因为唯一的文字集合 $X \in \mathbb{L}[0]$ 具有性质$X = (\emptyset, \pi(\emptyset, H))$。注意，$(\pi(\emptyset, H) = \{h1, h2, h3, h4\}$。更具体地说，$\pi(X) \neq \pi(X \cup Y)$ $\forall\, Y \in \mathbb{L}[1]$。

同样，我们可以验证 *GED* $X \to Y$ 在 $\mathbb{L}[1], \mathbb{L}[2]$ 中成立，其中 $X = (\{b1\}, \pi(\{b1\}))$ $Y = (\{a1, b1\}, \pi(\{a1, b1\}))$，因为 $\pi(\{b1\}) = \pi(\{a1, b1\})$。因此，我们可以构造 *GED* $(Q_3[x, y, y'], y.name = \texttt{F20} \to x.name = \texttt{EA})$，并剪枝节点 $n = (\{a1, b1\}, \pi(\{a1, b1\})) \in \mathbb{L}[2]$。 □

4.3. 图编辑依赖的最小覆盖集发现与排序

在图编辑依赖挖掘过程的这一阶段，我们会进一步分析上一步在 Σ 中挖掘出的所有规则，以确定并消除所有冗余规则或蕴含规则。此外，我们引入一种有趣度度量，用于对图编辑依赖的最终覆盖集 Σ_c 进行评分和排序。

4.3.1. 最小覆盖集

根据上述内容，每个 GED $\varphi \in \Sigma$ 都是最小的（参见第 3 节）。然而，由于 GED 的传递性，Σ 可能并非最小，因为可能存在冗余图编辑依赖。具体而言，考虑 Σ 中的以下依赖关系：$= \{\varphi_1, \varphi_2, \varphi_3\}$；其中 $\varphi_1 = (Q[\bar{u}], X \to Y)$ $\varphi_2 =$

函数 4 $findCover(\Sigma)$

1: $\Sigma_c := \emptyset, \Gamma := \emptyset$

2: **对于每个** $\varphi = (Q[\bar{u}], X \to Y) \in \Sigma$ **执行**

3: 在 Γ 中创建边 e: $X \to Y$

4: **如果** e 完成一个三角形 $t \in \Gamma$ **则**

5: 分析传递性并剪枝 t

6: **对于每条**剩余边 $e \in \Gamma$ **执行**

7: 添加从 e 到 Σ_c对应的图依赖

8: **返回** Σ_c

$(Q[\bar{u}], Y \to Z)$ 和 $\varphi_3 = (Q[\bar{u}], X \to Z)$。我们称 φ_3 在 Σ 中可由传递性推出，记作 $\varphi_3 \vDash \Sigma$ 为 $\Sigma \setminus \varphi_3 \equiv \Sigma$。也就是说，集合 $\Sigma' := \{\varphi_1, \varphi_2\} \equiv \Sigma$，且 Σ' 是最小的。

因此，只需从 Σ 中消除所有由传递性蕴含的图依赖，即可得到 Σ的最小覆盖 Σ_c。我们在函数 4 中采用一种简单但有效的方法，消除 Σ中所有由传递性蕴含的图依赖，并生成 Σ_c。我们创建一个图 Γ，其中包含 Σ 中的所有图依赖，使每个节点对应于依赖关系中唯一的文字集合（即第 2、3 行）。需要指出的是，如果 Σ中存在传递性，Γ 中就会形成三角形（即 3- 团）（参见第 4–5 行），并返回一个最小集合 Σ_c（第 6–8 行）。

4.3.2. GED 排序

我们提出了一种衡量图编辑依赖有趣度的方法，以及一种对给定最小覆盖中的规则进行排序的方法 Σ_c。我们采用最小描述长度（MDL）原理 [46] 来编码图编辑依赖的相关性。

模型 m 的最小描述长度为:

$$mdl(m, \text{data}) = cost(\text{data}|m) + \alpha \times cost(m),$$

其中， $cost$(数据$|m$)、 $cost$(m)分别表示模型对编码错误进行编码的成本，以及对模型本身进行编码的成本； α 是用于权衡这两项成本的超参数。

我们在下方的公式 4.1 中提出一种度量方法，用于评估图 G 中图依赖 $\varphi = (Q[\bar{u}], X \to Y)$ 的趣味性，记作 $rank(\varphi, G)$:

$$rank(\varphi, G) = \alpha \times (1 - \varsigma(\varphi)) + (1 - \alpha) \times \frac{k}{N}, \tag{4.1}$$

其中：1) $\varsigma(\varphi)$ 是比率 $\frac{|\pi_{(X \cup Y} + \in [0,1]}{|H|}$；2) k 是 $X \cup Y \in [0, N]$ 中唯一属性的数量；3) N 是 $Q[\bar{u}]$的所有匹配 H 中属性的总数；4) $\alpha \in [0, 1]$ 是用户指定参数，用于调整这两项的权衡。

与最小描述长度（MDL）一致，图依赖的排名越低，趣味性越高。我们将在下一节对该分数进行实证评估。

4.4. 所提方法的复杂度。

这里，我们概述所提出的图依赖挖掘方法的时间复杂度。该方法的关键任务包括：频繁子图挖掘（FSM），以及在频繁子图的匹配结果上进行属性（实体）依赖挖掘（ADM）。因此，下面分别介绍这两项任务的理论复杂度和实际复杂度。

FSM 复杂度。 </style}令 $|G|$ 和 k 分别表示给定图中的节点数和最大的图模式大小。总体而言，在最坏情况下，

FSM 过程的复杂度为 $\mathcal{O}(2^{|G|^2} \cdot |G|^k)$（详情参见 [28]）。这种关于 $|G|$ 的指数级时间复杂度是主要的性能瓶颈，正如第 1 节所述。因此，我们采用基于稠密社区检测的图划分启发式策略来降低$|G|$ （参见第 4.1 小节）。因此，在实践中，我们的 FSM 时间复杂度为 $\mathcal{O}(C \cdot 2^{|N|^2} \cdot |N|^k) \approx \mathcal{O}(2^{|N|^2} \cdot |N|^k)$，其中 $C << |N| << |G|$是最大的社区大小，C 是社区总数。

此外，从一个大小为 k 的子图到 G 执行同态匹配，在最坏情况下可在 $\mathcal{O}(|G|^2 \cdot k)$ 时间内完成（参见 [38] 和 [50]）。因此，FSM 的总体时间复杂度大致为 $\mathcal{O}(2^{|N|^2} \cdot |N|^k + |G|^2)$。

ADM 复杂度。这里我们考察在图模式上搜索并验证所有依赖关系的复杂度。设 $|\bar{u}|$、m、$|H|$ 分别为给定图模式 $Q[\bar{u}]$ 中的节点数、每个模式变量的最大属性数，以及 G 中 $Q[\bar{u}]$ 的匹配总数。

接着，按 $\bar{u}$ 的文字对 H 进行划分的复杂度为 $\mathcal{O}(|\bar{u}|^2 \cdot m \cdot |H|) \approx \mathcal{O}(m|H|)$；因为在实践中，$|\bar{u}| << m << |H|$（参见函数2）。此外，候选 GED 的搜索空间为 $2^{m \cdot |\bar{u}|^2}$。因此，在 H 中的 $Q[\bar{u}]$ 上生成、搜索和验证候选 GED，并对其进行缩减，在最坏情况下需要 $\mathcal{O}(|\bar{u}|^2 m|H| + 2^{m|\bar{u}|^2}|H| + |\Sigma| + |E(\Gamma)|) \approx \mathcal{O}(mH + 2^m|H| + |\Sigma|)$（参见函数3和4），其中 $|\Sigma|$ 为有效规则的数量，$|E(\Gamma)| \leq |\Sigma|$ 为规则图 Γ 中的边总数。

5. 实证研究

本节将使用真实世界图数据集评估所提出的发现方法。下文将讨论

表 2：数据集汇总

数据集	#节点	#边	#节点类型	#边类型	密度
DBLP	300K	800K	3	3	8.65×10^{-6}
IMDB	300K	1.3M	5	8	1.25×10^{-5}
YAGO4	4.37M	13.3M	7764	83	8.70×10^{-6}

实验设置，考察所提算法的可扩展性，将我们的方法与文献中的相关研究进行比较，并展示挖掘出的一些图编辑依赖在各种下游应用中的实用性。

5.1. 实验设置

本文介绍所使用的数据集以及实验设置。

数据集。 我们使用了三个具有不同特征和规模的真实世界基准数据集，汇总于表 2 中。DBLP [40] 是一个引文网络，包含 0.3M 个节点（3 种类型）和 0.8M 条边（3 种类型）；IMDB [41]是一个知识图谱，包含 0.3 M 个节点（5 种类型）和 1.3M 条边（8 种类型）；YAGO4 [42] 是一个知识图谱，包含 4.37M 个节点（7764 种类型）和 13.3M 条边（83 种类型）。与 [27], 一致，我们使用每个属性在活动域中出现频率最高的前五个属性值来构造常量字面量。

实验。 本文提出的所有算法均使用 Java 实现；实验则在一台配备 128GB 内存、运行 Linux 操作系统的 2.20GHz 英特尔至强处理器计算机上进行。我们通过实验评估所提出的方法，以展示其：(1) 可扩展性；(2) 相较于文献中其他（相关）图约束/依赖关系挖掘方法的性能相对于 其他图

约束/依赖关系；(3) 挖掘出的图编辑依赖的潜在用途。除非另有说明，所有实验均将分辨率参数 γ设为 1.0，最小 MNI 阈值 τ 设为 5000。所有实验均重复进行二十五（25）次，并报告所有结果的平均值。

5.2. 提案方案的可扩展性

根据第4.4节对我们提出的方案进行的复杂度分析，影响时间性能的关键参数包括：输入图的大小 $|G|$、图模式的大小 k，以及每个模式变量的属性数量 m 。因此，在这组实验中，我们针对以下参数变化，研究所提方案在全部三个数据集上的时间性能：a) 输入图大小 $|G|$ （实验1a）；b) 图模式大小 k （实验1b）；以及 c) 节点上的属性数量 m （实验1c）。

实验-1a。 在本实验中，我们将三个数据集中的每个节点的属性数量均固定为七（7）个。我们使用了 DBLP 的完整图，并对 IMDB 和 YAGO4 图进行采样，使其规模具有可比性。图9（a）展示了三个数据集不同规模对应的发现时间（以秒为单位，采用对数刻度）曲线，其相应的标准差（SD）列于表3中。

表3：图9（a）中时间的标准差

	100K	200K	300K
DBLP	2.927	5.266	7.236
YAGO4	0.095	0.413	6.741
IMDB	4.491	0.749	23.066

表4：图9（c）中时间的标准差

	2	3	4	5
DBLP	0.09	2.16	1.90	2.67
YAGO4	5.12	11.94	10.76	19.0
IMDB	7.06	14.30	23.07	20.95

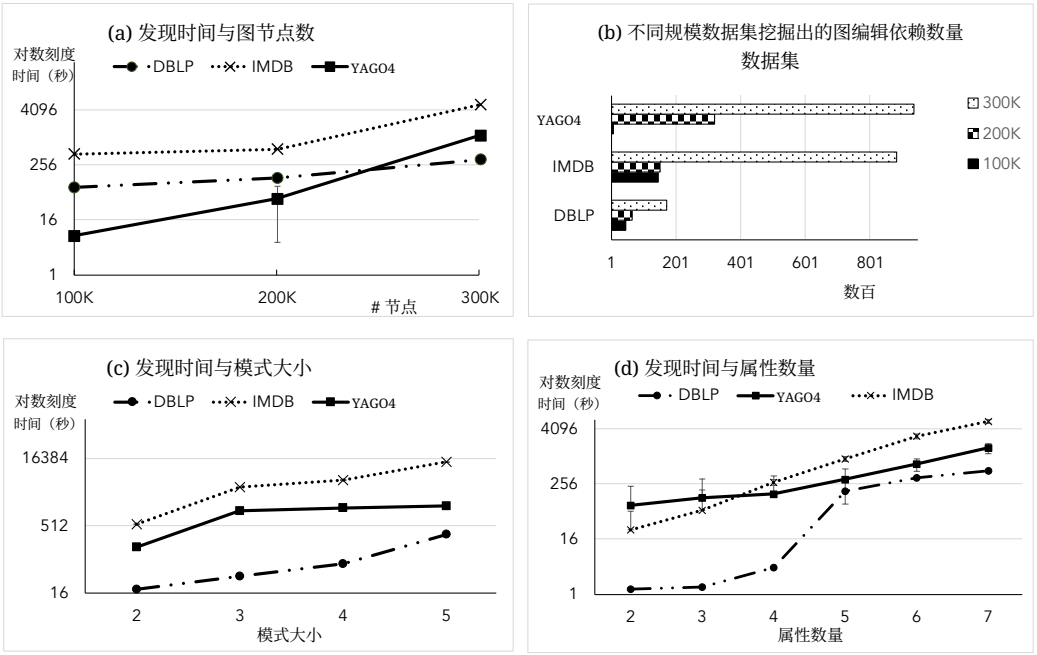


图9：所提方法的可扩展性

总体而言，DBLP 数据中的图依赖发现效率最高，因为与其他数据集相比，其频繁模式产生的匹配数量最少。这一特征可通过图的密度（见表 2）体现出来。换言之，图越稠密、连通性越强，就越有可能为图模式找到更多匹配。因此，图依赖发现所需的时间也越长。

不同大小的三个数据集所发现的图编辑依赖的分布如图9（b）所示——在几乎所有情况下，YAGO4 和 DBLP 数据集产生的图编辑依赖数量分别最多和最少。

实验1b. 这里，我们考察图模式大小对发现时间的影响。我们以图模式中的节点数作为其大小，考察大小为 2 至 5 的模式。对于每个数据集，我们在

大小为 2 至 5 的模式上挖掘图编辑依赖，每个节点最多包含七（7）个属性，并使用完整的图。

不同大小模式的时间性能结果如图9 (c)所示，相应的标准差见表4。在该实验中，IMDB和YAGO4数据集产生了相近的时间性能，但大小为5的模式除外。在这一特殊情况下，我们发现IMDB数据中大小为5的模式匹配数量明显更多，这一点也反映在图中。

实验1c。 这组实验考察了属性数量对所有数据集发现时间的影响，将图中每个节点的属性数量从2调整到7。三个数据集中所有大小的图模式的时间性能取平均值，作为最终结果。

图9 (d)中的图展示了发现平均时间的特征

对于属性数量不断增加的三个数据集（模式大小为2至5），相应的标准差见表5。从中可以看出

表5：图9 (d)中的时间标准差

	2	3	4	5	6	7
DBLP	0.04	0.02	0.03	0.07	0.04	0.27
YAGO4	0.02	0.07	0.06	0.13	0.27	0.60
IMDB	0.05	0.12	0.88	2.76	2.03	2.32

对于这三张折线图，正如对特征的分析所预期的那样（参见第4.4节），属性数量会直接影响时间性能。

5.3. 与挖掘其他图约束的比较

GED 的概念对图函数依赖（GFD）和图键（GKey）进行了推广并涵盖了它们。因此，我们在这组实验中展示，我们的方案通过以下两项简单调整，可用于挖掘 GFD 和 GKey：

首先，根据 GFD 和 GKey 的定义，将图模式匹配语义从同态改为同构。其次，在挖掘图键时仅考虑标识文字右端项，而在挖掘 GFD 时不考虑标识符字面量。

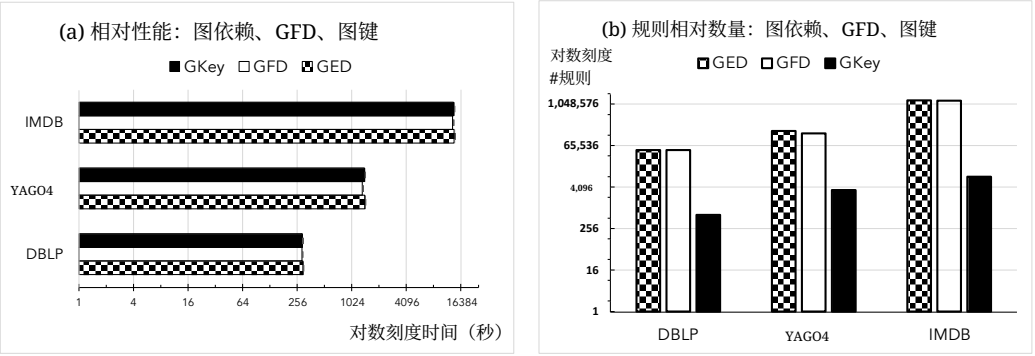


图 10：图依赖、GFD 和图键挖掘的性能比较

在图 10 中，我们总结了图依赖、图函数依赖和图键挖掘的比较分析。图中的部分 (a) 分别展示了在 DBLP、YAGO4 和 IMDB 数据集上挖掘这三种不同图约束的相对时间性能。在每项实验中，我们展示了最多五个图模式的性能（使用每个数据集中频率最高的五个模式）。

正如预期，由于我们优化了：a）同态与同构的匹配效率；以及 b）文字数量，因此所有情况下的时间性能差异都微乎其微。

表6：图 10 (a) 中时间的标准差

	GED	GFD	GKey
DBLP	0.71	0.35	0.60
YAGO4	1.92	0.86	0.87
IMDB	15.67	14.85	16.20

针对每种依赖/约束类型进行考察。表6展示了所有数据集上时间性能的标准差。

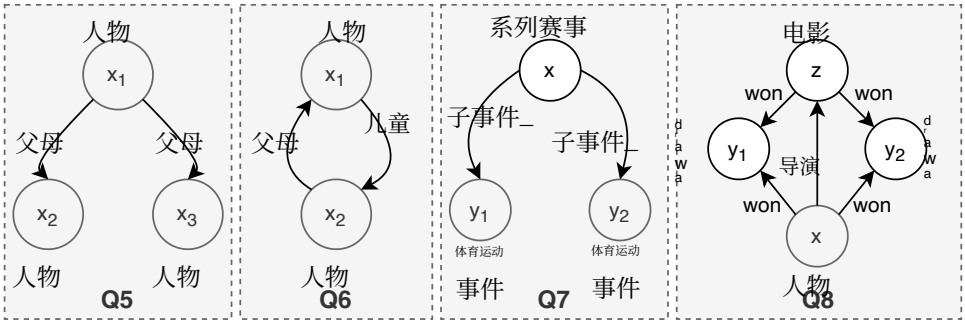


图 11：YAGO4 中若干图依赖的图模式

此外，图 10 (b) 展示了所有数据集中每种约束发现的数量。具体而言，在所有数据集中，图编辑依赖与图函数依赖的数量始终相近；而图键是规模最小的规则，因为其右侧仅限于标识符字面量。

5.4. 挖掘规则的实用性

本节将通过一些示例介绍发现的图编辑依赖，并讨论它们在现实世界的数据质量和数据管理应用中的潜在用途。此外，我们还将讨论挖掘出的图编辑依赖的有趣性排序。

发现的图编辑依赖示例。 我们人工检查了挖掘出的图编辑依赖，并验证了它们的实用性。我们展示了在 YAGO4 数据中发现的 3 个图编辑依赖，其图模式如图 11 所示：

σ_1： $Q_5[x_1, x_2, x_3]$, $\{x_1.\texttt{dob} = \texttt{1777-04-24}, x_2.\texttt{dod} = \texttt{1792-03-01}, x_3.\texttt{name} = \texttt{`Leopold II, Holy Roman Emperor'}\} \rightarrow x_2.\texttt{id} = x_3.\texttt{id}$。

σ_2： $Q_6[x_1, x_2]$, $\emptyset \rightarrow x_1.\texttt{surname} = x_2.\texttt{surname}$。

σ_3： $Q_7[x, y_1, y_2], x.\texttt{name} = \texttt{`2018 Wuhan Open'} \rightarrow x_1.\texttt{prefixname} = x_2.\texttt{prefixname}$.

σ_4: $Q_8[x, y_1, y_2, z]$, x.name = 'Akira Kurosawa' $\wedge$ y_1.name = 'Golden Lion' $\wedge$ y_2.url = 'http:www.labiennale.org' $\wedge$ z.name = 'Rashomon' $\rightarrow$ y_1.id = y_2.id

图依赖 σ_1 指出，对于 YAGO4 图中模式 Q_5 的任何匹配，如果 person x_1 出生于 1777-04-24，并且有一位父母 x_2 的死亡日期为 1792-03-01 ，且另一位父母 x_3名为 'Leopard II, Holy Roman Emperor'，那么 x_2, x_3 指的是同一个人物。该图依赖可验证为真（参见此处的维基百科记录[3]）——因为人物 x_1 是 'Archduchess Maria Clementina of Austria'Leopard II的儿童 Holy Roman Emperor'。在 YAGO4 数据中发现 σ_1 ，表明'Leopard II 和 Holy Roman Emperor'存在重复的 person 实体。因此，该图依赖的一个潜在用途是对图中的上述实体进行消歧或去重（实体链接）。

此外， σ_2 指出，对于亲子模式Q_6的任何匹配，儿童 x_1 和父母 x_2 必须拥有相同的姓氏。图依赖 σ_3声称，对于事件系列 2018 Wuhan Open x 中的任意两个体育赛事 y_1, y_2 y_1, y_2 ，它们必须属于同一事件 prefixName。事实上，这些图依赖简单易懂，可通过验证依赖关系，用于检测图中的违规或不一致情况。例如， σ_2 可用于检查所有姓氏不同的亲子关系，而 σ_3 可用于查找 '2018 Wuhan Open' 中违反前缀命名约束的任何子赛事。

[3]https://en.wikipedia.org/wiki/奥地利女大公_玛丽亚_克莱门蒂娜_的_奥地利

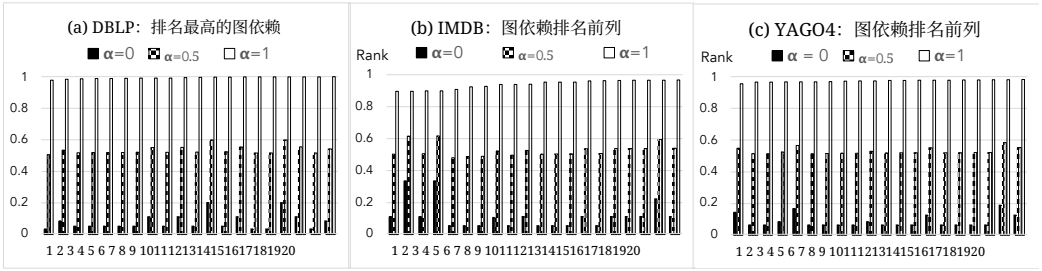


图12：按 α 设置排名最高的图编辑依赖

图依赖 σ_4与 σ_1一样，是图键的一个示例（即，具有右侧标识符字面量）。因此，它可用于实体解析或去重。

图依赖的排名。 接下来，我们展示在三种不同的 α 取值（即 0、0.5 和 1）下，所发现图编辑依赖的有趣度评分图。为简洁起见且受篇幅所限，我们仅在图12中展示排名前 $K = 20$ 的图编辑依赖。有趣度评分越低，图依赖越好，反之亦然。该图显示，有趣度评分会受到 α取值的影响，而对图依赖进行排序可以帮助用户更轻松地找到有意义的图依赖。对于 $\alpha = 0$，排名分数反映了图依赖的复杂度，而 $\alpha = 1$ 反映了规则在数据中的持久性。 $\alpha = 0.5$ 则以相同权重结合了两者。

6. 相关工作

本节回顾文献中关于发现图数据约束的相关研究。近年来，图数据中的约束挖掘日益受到关注。下文将从两个大类介绍一些相关研究。

6.1. 非属性图中的规则发现。

大多数关于非属性图数据分析与挖掘的研究都聚焦于 XML 和 RDF 数据（参见 [47]）。例如， [48, 49, 52] 研究了 RDF 数据和知识库（KB）中的（路径）关联规则挖掘问题，而 [53, 54, 55, 56] 则提出了基于归纳逻辑程序设计的规则挖掘器，用于处理 RDF 数据和知识库。

与上述研究不同，本文研究属性图中函数依赖（包括条件依赖和实体依赖）的挖掘。不过，我们提出的方法可以调整用于挖掘 RDF 数据中的规则，尤其是包含常量文字的图编辑依赖。

6.2. 属性图中的规则发现。

与本研究更相关的是属性图中的规则发现技术。该领域一些值得关注的工作包括：1） [17, 27, 57, 58, 59], 研究了属性图中关联规则的发现；以及 2） [12, 16, 22, 60] 研究了属性图中的键和依赖关系挖掘，与本研究最为接近。具体而言， [12] 提出了一种基于频繁子图扩展的方法，用于挖掘属性 RDF 中的键；而 [16] 则提出了面向大型属性图的高效顺序和并行图函数依赖发现方法。 [16]中的 GFD 发现方法与我们的工作主要区别在于处理效率瓶颈的方式。具体来说， [16] 采用固定参数化来限制待挖掘模式的大小，而我们则对图进行调节，并从图中的稠密社区挖掘图模式。此外，我们设计了一种逐层搜索策略，通过项集格来发现并验证规则

而 [16] 中的顺序技术则通过生成树的垂直和水平扩展来发现 GFD。

此外， [60]研究了时序GFD 的发现——即在一段时间内适用于属性图的 GFD； [22] 研究了属性图上的图差分依赖关系（GDD）的发现——即具有距离语义的图编辑依赖。尽管相关，但这项工作不同于时序 GFD 的挖掘，因为我们只考虑图的一个时间段。此外，我们不考虑 GDD 所刻画的数据差异语义。从本质上说，GED 可以视为 GDD 的一种特殊情形，即所有属性上的距离均为零。

7. 结论

本文提出了一种新的图依赖挖掘方法。所开发的发现流程将图划分、非冗余且频繁的图模式挖掘、同态图模式匹配以及属性/实体依赖挖掘无缝结合起来，以从属性图中发现图依赖。我们开发了有效的剪枝策略和技术，以确保返回的图依赖集合是最小且不存在冗余的。此外，我们提出了一种有效的基于最小描述长度的度量，用于评估图依赖的有趣程度。最后，我们在大型真实世界图上进行了实验，以展示所提出方法的可行性、有效性和可扩展性。实验结果表明，我们的方法有效、可扩展且高效，并能发现语义上有意义的规则。

致谢

本研究部分获得以下项目资助：内蒙古重点科技项目（项目编号：2021ZD0046）；海洋防御技术创新中心创新基金（项目编号：JJ-2021-722-04）；湖北洪山实验室重大项目（项目编号：2022HSZD031）；湖中农业大学作物遗传改良国家重点实验室开放基金；湖中农业大学农业微生物学国家重点实验室开放基金；武汉大学杂交水稻国家重点实验室2021年开放项目；以及中央高校基本科研业务费专项资金（项目编号：2662020XXQD01和2662022JC004）。

参考资料

[1] Fan, W., Fan, Z., Tian, C. and Dong, X.L., 2015年. 图的键. VLDB基金会会刊,8(12),pp.1590-1601.

[2] Fan, W., Wu, Y. and Xu, J., 2016年, 6月. 图的函数依赖. In Proceedings of the 2016年 International Conference on ManagementofData (pp.1843-1857).

[3]Ma, H., Alipourlangouri, M., Wu, Y., Chiang, F. and Pi, J., 2019年. 基于本体的实体匹配 in 属性图. Proceedings of theVLDB基金会会刊, 12(10), pp.1195-1207.

[4] Fan, W. 和 Lu, P.， 2017年5月。《图的依赖》。载于第36届 ACM SIGMOD-SIGACT-SIGAI PODS研讨会（第403–416页）。ACM。

[5] Fan, W. 和 Lu, P.，2019年。《图的依赖》。ACM数据库系统汇刊（TODS），44（2），第1–40页。

[6] Fan, W., Geng, L., Jin, R., Lu, P., Tugay, R. 和 Yu, W.，2022年5月。跨关系和图链接实体。载于《2022年 IEEE 第38届国际数据工程大会（ICDE）》（第634-647页）。IEEE。

[7] Liu, J., Li, J., Liu, C. 和 Chen, Y.，2010年。从数据中发现依赖关系——综述。IEEE知识与数据工程汇刊，24(2)，第251-264页。

[8] Abedjan, Ziawasch、Lukasz Golab 和 Felix Naumann。《关系型数据剖析：调查综述》。《The VLDB Journal》24，第4期（2015年）：557-581。

[9] Caruccio, Loredana、Vincenzo Deufemia 和 Giuseppe Polese。《宽松函数依赖——方法综述》。《IEEE知识与数据工程汇刊》28，第1期（2015年）：147-165。

[10] Vezvaei, A.、Golab, L.、Kargar, M.、Srivastava, D.、Szlichta, J. 和 Zi- hayat, M.，2022年。《使用参数微调依赖关系》。载于 EDBT（第2-393 页）。

[11] Zada MS、Yuan B、Anjum A、Azad MA、Khan WA、Reiff-Marganiec S。基于图概率依赖关系（GPD）的 大规模数据集成。载于 2020年 IEEE/ACM 大数据计算、应用与技术国际会议（BDCAT），2020年12月7日（第 27-36 页）。IEEE。

[12] Alipourlangouri, M. 和 Chiang, F.，2018年。Keyminer：发现图的键。发表于 VLDB 研讨会 TD-LSG。

[13] Kwashie, S.、Liu, J.、Li, J. 和 Ye, F.，2015年9月。条件差分依赖（CDDs）。发表于东欧数据库与信息系统进展会议（第 3-17 页）。Springer，Cham。

[14] Kwashie, S., Liu, J., Li, J. 和 Ye, F.，2014年，7月。差分依赖挖掘：一种子空间聚类方法。载于《澳大拉西亚数据库会议》（第50-61页）。Springer，Cham。

[15] Kwashie, S., Liu, J., Li, J. 和 Ye, F.，2015年，6月。通过关联规则挖掘高效发现差分依赖。载于《澳大拉西亚数据库会议》（第3-15页）。Springer，Cham。

[16] 范文飞、胡春明、刘雪莉和卢平。2020年。图函数依赖的发现。《ACM数据库系统汇刊》45卷第3期，文章15（2020年9月），42页。

[17] 范文飞、王鑫、YinghuiWu 和 JingboXu。2015年。带图模式的关联规则。《VLDB基金会会刊》8卷第12期（2015年8月），1502–1513页。

[18] Thorsten Papenbrock 和 Felix Naumann。2016年。一种用于函数依赖发现的混合方法。载于《2016年国际数据管理会议论文集》（SIGMOD ’16）。计算机协会，美国纽约州纽约，821–833。

[19]Papenbrock, Thorsten, Sebastian Kruse, Jorge-Arnulfo Quian´e-Ruiz 和 Felix Naumann。“基于分而治之的包含依赖关系发现。”VLDB基金会会刊 8，第 7 期（2015年）：774-785。

[20] Lin, P., Song, Q. 和 Wu, Y.，2018年。利用本体子图模式对知识图谱进行事实核查。《数据科学与工程》，3(4)，第341-358页。

[21] Fan, W., Lu, P., Tian, C. and Zhou, J.，2019年。《推导图的若干修复方案》。VLDB基金会会刊，12(7)，第752-765页。

[22] Kwashie S、Liu L、Liu J、Stumptner M、Li J、Yang L。《Certus：一种基于图差分依赖（GDDs）的有效实体解析方法》。VLDB基金会会刊。2019年2月1日；12(6)：653-66。

[23] Jixue Liu、Selasi Kwashie、Jiuyong Li、Feiyue Ye 和 Millist Vincent。《近似差分依赖的发现》。计算机科学·数据库。2013年。

[24] Chiang、Fei 和 Ren´ee J. Miller。《发现数据质量规则》。VLDB基金会会刊 1，第1期（2008年）：1166-1177。

[25] Cosmadakis, Stavros S.、Paris C. Kanellakis 和 Nicolas Spyratos。“关系的划分语义。”《计算机与系统科学杂志》33卷，第2期（1986年）：203-233。

[26] Rammelaere, Joeri 和 Floris Geerts。“重新探讨条件函数依赖发现：将‘C’与‘FD’分开。”载于《机器学习与数据库中的知识发现》欧洲联合会议论文集，第552-568页。Springer，Cham，2018年。

[27] Fan, W., Fu, W., Jin, R., Lu, P. 和 Tian, C.，2022年。从大规模图中发现关联规则。《VLDB基金会会刊》，15(7)，第1479-1492页。

[28] Mohammed Elseidy、Ehab Abdelhamid、Spiros Skiadopoulos 和 Panos Kalnis。2014年。GraMi：在单个大型图中进行频繁子图和模式挖掘。《VLDB基金会会刊》7(7)（2014年3月），517–528。

[29] Fiedler, M. 和 Borgelt, C., 2007年10月。单个大型图中的子图支持度。载于《第七届IEEE国际数据挖掘会议研讨会》（ICDMW 2007）（第399-404页）。IEEE。

[30] Kuramochi, M. 和 Karypis, G., 2005年。在大型稀疏图中发现频繁模式。《数据挖掘与知识发现》，11(3)，第243-271页。

[31] Ribeiro, P., Paredes, P., Silva, M.E., Aparicio, D. 和 Silva, F., 2021年。子图计数综述：概念、算法，以及在网络模体和图元中的应用。《ACM计算调查》（CSUR），54(2)，第1-36页。

[32] Jiang, C., Coenen, F., &Zito, M.（2013）。频繁子图挖掘算法综述。《知识工程评论》，28(1)，75-105。doi:10.1017/S0269888912000331

[33] Kasra Jamshidi、Rakesh Mahadasa 和 Keval Vora。2020年。Peregrine：一种模式感知图挖掘系统。载于《第十五届欧洲计算机系统会议论文集》（EuroSys '20）。计算机协会，纽约，纽约州，美国，文章编号 13，1–16。

[34] Vinicius Dias、Carlos H. C. Teixeira、Dorgival Guedes、Wagner Meira 和 Srinivasan Parthasarathy。2019年。Fractal：一个通用图

模式挖掘系统。载于《2019年国际数据管理会议论文集》（SIGMOD '19）。计算机协会，纽约，纽约州，美国，1357–1374。

[35]Traag, V. A., Van Dooren, P. & Nesterov, Y. 无分辨率极限社区检测的适用范围狭窄。《物理评论E》84, 016114, https://doi.org/10.1103/PhysRevE.84.016114（2011年）。

[36] 严熙锋和韩家炜，“gSpan：基于图的子结构模式挖掘”，2002年IEEE国际数据挖掘会议，2002年。会议录，2002年，721-724。

[37]Traag, V.A., Waltman, L. & van Eck, N.J. 从 Louvain 到 Leiden：确保社区具有良好的连通性。《科学报告》9, 5233（2019年）。https://doi.org/10.1038/s41598-019-41695-z

[38]Mhedhbi, A. 和 Salihoglu, S., 2019年。通过结合二元连接和最坏情况最优连接来优化子图查询。VLDB基金会会刊，12(11)，第1692-1704页。

[39] R.Agrawal 和 R. Srikant. 在大型数据库中挖掘关联规则的快速算法。载于第*20*届超大型数据库国际会议论文集，VLDB '94，第 487–499 页，旧金山，加利福尼亚州，美国，1994 年。Morgan Kaufmann出版社。

[40] DBLP数据集。https://dblp.uni-trier.de/xml/

[41] IMDB数据集。http://www.imdb.com/interfaces。

[42] YAGO4数据集。https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[43] Luigi P. Cordella、Pasquale Foggia、Carlo Sansone 和 Mario Vento，《一种用于匹配大型图的（子）图同构算法》，IEEE《模式分析与机器智能汇刊》，第 26 卷，第 10 期，2004 年。

[44] Huhtala、Yka、Juha K¨arkk¨ainen、Pasi Porkka 和 Hannu Toivonen。“TANE：一种用于发现函数依赖和近似依赖关系的高效算法。”《计算机杂志》第 42 卷，第 2 期（1999 年）：100–111。

[45] Bringmann B、Nijssen S. 单个图中什么是频繁的？载于：亚太知识发现与数据挖掘会议，2008 年 5 月 20 日（第 858–863 页）。Spri nger，柏林，海德堡。

[46]Peter D. Grunwald 和 Jorma Rissanen，《最小描述长度原理》，麻省理工学院出版社，2022年。

[47] Abedjan, Z., Golab, L., Naumann, F. 和 Papenbrock, T., 2018年。数据剖析。《数据管理综合讲座》，10(4)，第1-154页。

[48] Barati, M., Bai, Q. 和 Liu, Q., 2017年。从 RDF数据中挖掘语义关联规则。《知识型系统》，133，第183-196页。

[49] Abedjan, Z. 和 Naumann, F., 2013年。通过关联规则挖掘改进RDF数据。《数据库光谱》，13(2)，第111-120页。

[50] Bi、F.、Chang、L.、Lin、X.、Qin、L. 和 Zhang、W.，2016年6月。通过延后笛卡尔积实现高效子图匹配。载于《

2016年国际数据管理会议论文集》（第1199—1214页）。

[51] Feng, Z., Mayer, W., He, K., Kwashie, S., Stumptner, M., Grossmann, G., Peng, R. 和 Huang, W., 2020年。一种由模式驱动的合成知识图生成方法，采用扩展图差分依赖关系（GDDxs）。IEEE Access，9，页码 5609-5639。

[52]Sasaki, Y., 2022年。路径关联规则挖掘。arXiv 预印本 arXiv:2210.13136。

[53] Gal´arraga, L.A., Teflioudi, C., Hose, K. 和 Suchanek, F., 2013年5月。AMIE：在本体知识库中基于不完整证据进行关联规则挖掘。载于第 22 届万维网国际会议论文集（第 413-422 页）。

[54] Chen, Y., Wang, D.Z. 和 Goldberg, S., 2016年。ScaLeKB：针对大型知识库的可扩展学习与推理。VLDB Journal，25(6)，页码 893-918。

[55] Lajus, J., Gal´arraga, L. 和 Suchanek, F.，2020年5月。使用 AMIE 3 进行快速且精确的规则挖掘。载于《欧洲语义网会议》（第 36–52 页）。Springer，Cham。

[56] Ortona, S., Meduri, V.V.和 Papotti, P., 2018年，4月。在知识库中稳健地发现正规则与负规则。载于2018年 IEEE 第34届国际数据工程会议（ICDE）论文集（第1168-1179页）。IEEE。

[57] Namaki, M.H., Wu, Y., Song, Q., Lin, P. 和 Ge, T., 2017年，11月。发现图时序关联规则。载于2017年 ACM 信息与知识管理会议论文集（第1697-1706页）。

[58] Fan, W., Jin, R., Liu, M., Lu, P., Tian, C.and Zhou, J., 2020年. 捕获图中的关联。《VLDB基金会会刊》，13(12)，第1863-1876页。

[59] Huang, C., Zhang, Q., Guo, D., Zhao, X. and Wang, X., 2022年. 在时序网络中发现关联规则与图模式。《清华大学学报（英文版）》，28(2)，第344-359页。

[60] Noronha, L. and Chiang, F., 2021年. 时序图函数依赖关系的发现。载于第30届ACM国际信息与知识管理会议论文集（第3348-3352页）。