

# 时序图事件预测研究

范文飞<sup>1,2,3</sup>, 金若春<sup>1</sup>, 陆平<sup>3</sup>, 田超<sup>4</sup>, 徐瑞琪<sup>5</sup>

<sup>1</sup>爱丁堡大学<sup>2</sup>深圳计算科学研究所<sup>3</sup>北京航空航天大学

<sup>4</sup>中国科学院<sup>5</sup>新加坡国立大学

{wenfei@inf, ruochun.jin@} ed.ac.uk, luping@buaa.edu.cn, tianchao@iscas.ac.cn, ruiqi.xu@nus.edu.sg

## 摘要

本文提出了一类用于事件预测的时间关联规则, 用TACOs表示。与以前的图形规则相反, TACOs监视图形的更新, 并可用于捕获推荐中的暂时兴趣, 以及捕获响应行为变化的欺诈行为等。TACOs是根据变化模式和(时间)条件在时间图上定义的, 并且可以携带用于时间事件预测的机器学习(ML)谓词。我们解决了TACOs的推理复杂性问题, 包括其可满足性、蕴涵和预测问题。我们开发了一个系统, 称为TASTE。TASTE通过在创建者-评论家框架中基于生成式ML模型迭代训练规则创建者来发现TACOs。此外, 它通过应用发现的TACOs来预测事件。使用真实数据集和合成数据集, 我们实验验证了TASTE在TACO发现方面的平均速度是传统数据挖掘方法的31.4倍, 并将最先进的事件预测模型的准确性提高了23.4%。

## PVLDB参考格式:

面向时序图的事件预测。Pvldb, 15(9): 1861- 1874,2022. doi: 10.14778/3538598.3538608

## 1 介绍

事件预测是预测现实世界中与特定话题相关的、将在特定时间发生的事件[109]。事件的范围从大规模(例如, 疾病爆发), 到中等规模(例如, 系统故障), 再到小规模(例如, 欺诈检测)。事件预测在疾病控制和商业智能等多个领域都很重要。举例来说, 考虑在线推荐, 这是事件预测的一个特殊案例(“销售事件”), 向用户推荐物品。推荐模型主要通过学习用户-项目交互分为协同过滤(CF), 通过评估用户和项目内容特征的相似度分为基于内容(CB), 以及通过将两者整合为混合(hybrid)[107]。然而, 在现实世界中, 电商公司经常监测交易图的变化, 并采用规则来捕捉用户的时间兴趣和检测欺诈行为。

例1:下面是来自某电商平台的两条规则。

(1)如果一部电影获得了电影奖提名, 并且有用户在提名后两周内观看了这部电影, 那么推荐

8/3538598.3538608电影发给用户的朋友之间的提名日期和颁奖典礼的日期。在这里, 提名表明了一个在时间间隔内触发推荐的“变化”。这样的案例超越了传统的CF和CB模型。

(2)如果用户在6月份搜索“烧烤”至少m次, 那么在未来两个月向用户推荐肉类。

时间变化也被用来预测其他事件。

(3)如果一个地区在过去两周内至少报告了一种传染病的病例, 那么为那里的人提供z型疫苗。

(4)如果设备M在一小时内只对k个账户进行了一次访问, 并且这些账户在过去的一个月内都被其他设备定期访问过至少M次, 则设备M很可能进行了账户接管攻击。2

挑战。无论多么重要, 事件预测都具有挑战性。虽然已经对图的规则进行了研究, 如图函数依赖(GFDs)[12]、图关联规则(GARs)[21]和Hom规则[25], 但它们是在静态图上定义的, 不能表达时间变化。这里有几个开放的问题。我们如何在逻辑规则中指定图形和用户时序兴趣的变化模式?我们能否提高具有时间条件的事件预测ML模型的准确性, 并解释它们的预测?用时态条件和嵌入的ML模型来推理规则的成本有多高?我们如何从现实生活的图中高效地发现这样的规则?当我们应用这些规则来预测事件时, 是否有可能对大型图进行缩放?

**贡献&组织。本文解决了这些问题。**

(1)时间规则(第2节)。我们提出了一类时间关联规则, 称为TACOs(时间事件预测规则), 用时间条件丰富事件预测ML模型。TACOs是根据变化模式、时间条件和ML预测模型(作为谓词)在时间图上定义的。与以前的图规则相比, TACOs应用于时间图的更新, 时间图通常比整个图小得多, 以监视变化并预测事件。

(2)复杂性(第3节):我们研究了TACOs的经典问题, 包括(a)可满足性(检查一组TACOs是否没有冲突), (b)决定一组TACOs是否需要另一个TACO的隐含性(决定一组TACOs是否需要另一个TACO)和(c)预测(预测事件是否会发生)。我们证明它们分别是 $\Sigma_2^P$ -完全、 $\Pi_2^P$ -完全和 $NP$ -完全。也就是说, 尽管TACOs增加了表现力, 但与之相比, 它并没有使我们的生活变得更加艰难[21,22]。

(3) TASTE系统(第4节)。尽管很棘手, 我们还是开发了一个称为TASTE(时间系统)的系统。TASTE (a)使用生成式ML模型发现高质量的TACOs, (b)将发现的TACOs并行应用于时间事件预测。

(4)规则发现(第5节)。从概率分布学习的角度出发,我们开发了一个创造者-评论家框架来发现TASTE中的TACOs,这是我们的主要贡献。规则创建者采用GAN和LSTM语言模型生成具有候选模式和时间条件的TACOs,而评论家则在时间图上对这些TACOs规则进行评估,收集高质量的规则,并向创建者提供反馈,以便在下一轮改进其质量。与GAN中传统的判别器不同,“评论家”是一种预定义的评分算法,不需要训练。该方法避免了在指数级大空间中穷尽的逐级搜索[20],能够找到具有大图模式的高质量TACOs。

(5)并行预测(第6节):我们开发了一种基于TACOs的时间事件预测并行算法。与现有的图分割方法(如edge-cut或vertex-cut[3,77])相比,我们建议基于时间局部性对时序图进行分割,从而使事件预测可以实现无通信。我们表明,当给予更多处理器时,并行事件预测算法保证运行得更快,即并行可扩展性。

(6)实验研究(Section 7):使用真实数据集和合成数据集验证了TASTE的准确性、效率和可扩展性。我们发现了以下内容。平均而言,(1)TASTE在时间事件预测和动态推荐方面的准确率分别比之前的方法提高23.8%和23.0%。(2)TASTE的创造者-评论家发现方法比分级算法快31倍以上。它发现TACOs在1639年有20条边的图案,与传统方法相比。(3)使用少量的训练数据和训练轮数,TASTE可以找到84.76%的分层方法挖掘的规则。(4)TASTE中的事件预测算法具有并行可扩展性,当使用32个处理器而不是4个处理器时,其运行时间缩短了3.2倍。

新鲜事物。本文提出了一种通过在逻辑规则中丰富ML模型的时序逻辑条件来进行事件预测的方法,以及一种用大型图模式来圆盘化规则的实用方法。该工作的新颖性包括:(1)TACOs,一类用于动态图上事件预测的规则;(2)TACOs的可满足性、蕴涵性和预测性问题的复杂度界;(3)创造者-评论家框架,第一个能够发现具有大图模式的高质量图规则;(4)基于时间局部性的划分时序图的策略;(5)保证并行可扩展性的时态事件预测算法;(6)基于发现TACOs和利用TACOs预测事件新方法的TASTE系统。

相关的工作。我们将相关工作分类如下。

时序事件预测。各种ML模型已经被探索用于时间事件预测,包括(1)回归[24,48,64,67];(2)点过程[73,99];(3)生存分析[16,94];以及(4)嵌入方式,例如张量分解[18,105]、循环神经网络(RNN)[36,56,92,93]、图神经网络(GNN)[11,50,58,71,81]、自编码器[29,30,78]和历时编码器[15,26,28,101]。特别是,作为一种特殊类型的事件预测,(时序)推荐吸引了大量的兴趣。例如,动态推荐系统(DRS)已经被开发出来以提高推荐质量。它

们可以使用时间依赖的邻域模型[23,60,89]、矩阵因子化[42,55,59,76]和张量模型[33,96,97]来纳入用户偏好的时间因素[75]。

时态关联规则已被开发出来,以指定关系ta-table中事务数据的时间依赖相关性[95,102]。此外,关联规则也被研究在图上[5,21,65]。GARs[21]是在静态图上定义的,用来指定实体之间的规律性。GTARs(图时态关联规则)[65]通过两个事件模式指定事件之间的连接,这两个事件模式共享一个共同的焦点节点和一个恒定的时间间隔。类似地,ger(图进化规则)[5]被定义为由单个模式分解成两个相连的子模式,以表示局部变化,例如重新标记[83]。

这项工作与之前的工作在以下方面有所不同。

(1)为了统一基于规则的事件预测方法和基于机器学习的事件预测方法,我们提出了通过嵌入事件预测机器学习模型作为谓词的TACOs。这使得我们不仅可以利用现有的ML模型,还可以用逻辑条件来完善ML模型。

(2)与现有的基于机器学习的预测策略(如DRS)采用无法解释的黑箱模型不同,我们的TACOs方法为机器学习对时间事件的预测提供了逻辑解释。

(3)TACOs比GARs、GTARs和ger更具表达性。正如将在第2节中看到的,这些规则都可以表示为TACOs。TACOs可以嵌入超越GTARs和ger的预测ML方法,并支持不同事件的各种时间条件,而GTARs和ger只能表达恒定的时间间隔。

(4)我们为时序图规则建立了经典问题的复杂性。虽然这些问题在静态图上对GARs进行了研究,但之前没有工作考虑过GTARs或ger的这些问题。

规则的学习者。先前的图规则发现方法可以分为以下几类:(1)分层数据挖掘方法,迭代枚举所有可能的候选规则,并以支持度/置信度[5,20,65]、头部覆盖和PCA置信度[25]等标准修剪无趣的规则;以及(2)规则学习器,通常枚举路径作为候选规则,并学习每个候选规则的权重以量化其质量。权重学习方法包括马尔可夫逻辑网[40,79]、路径排序[44]、关系依赖网络[66,68]、神经逻辑编程[80,82,103]和强化学习[14,52,62,74,87,100]。

与现有的规则学习器不同,(a)研究时序图中的规则发现,而之前的规则学习器侧重于静态图。(b)我们建议发现具有一般拓扑结构的TACOs,而不仅仅是路径。(c)我们发现可能携带ML谓词的TACOs。(d)我们提出了一种基于深度生成模型的方法来学习具有大图模式的图规则。相比之下,传统的图规则挖掘方法[20]很难发现具有7条边及以上模式的规则。

发电机模式。图生成模型已经被广泛研究来构建图模式。此类模型通常确保生成的模式维持某些属性,例如幂律节点度分布[1]、社区结构[84,98]、随机游走上的分布[7,110]、节点和边序列分布[51,54,104]、graphlets分布[35,106]以及时不变条件和时变条件[108]。

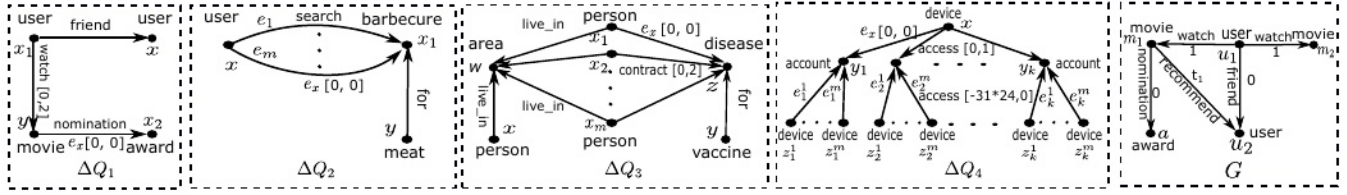


图1:  $\Delta$ -模式和时序图

模式生成是我们的creator-critic框架中的一个步骤。然而，我们的TACO发现方法与模式生成器不同，因为它合并了(a)将节点和边标签附加到生成的变化模式的语言模型，以及(b)用时间条件和ML谓词创建依赖关系的策略。在模式生成之前，这些都没有被研究过。

## 2 时序事件预测规则

本节将介绍TACOs。我们从基本符号(2.1节)开始，然后定义时间预测规则(2.2节)。

### 2.1 预赛

假设标签、常量和时间戳分别由三组可数无限的符号组成，分别用 $\Gamma$ 、 $\Sigma$ 和 $\Omega$ 表示。这里 $\Omega$ 是 lin-early ordered w.r.t. 一个离散阶 $\leq$ 。我们用 $\tau$ 表示一个时间窗口，对于从 $\Omega$ 中选择的时间戳 $t_1$ 和 $t_2$  ( $t_1 \leq t_2$ )，它是 $[t_1, t_2]$ 。

时序图。指定时间图 $G=(V, E, L, T, FA)$ ，其中(A)  $V$ 是节点的有限集合;(b)  $E \subseteq V \times \Gamma \times \Omega \times V$ 是边的集合，其中 $E=(V, l, t, V)$ 表示一条从 $V$ 到 $V$ 的边 that 用 $l \in \Gamma$ 标记，携带时间戳 $t=(t) \in \Omega$ ;(c) 每个 $v \in V$ 的 $node_v$ 都有来自 $\Gamma$ 的标签 $L(V)$ ;(d) 节点 $v$ 携带一个元组 $FA(v)=(a_1=a_1, \dots, a_n=A_n)$ 有限度属性的集合，其中 $A_i$ 表示一个属性， $A_i \in \gamma$ ，写成 $v.A_i = A_i, A_i, A_j$  if  $i, j$ 。

当 $G$ 从上下文中清晰可见时，我们称其为图。

直观地说， $G$ 是一个有向标记图，其中每条边都有一个时间戳记录它何时被添加到 $G$ 或何时被最后更新。注意，从一个 $node_v1$ 到 $v2$ ，可能存在多条边，可能具有相同的标签，但时间戳不同。为了简化演示，我们通过将时间戳自循环边附加到节点上来指定节点时间戳，以标记节点更新的时间。

$\Delta$  模式。 $\Delta$  模式为 $\Delta Q[x] = (VQ, EQ, LQ, TQ, \mu)$ ，其中(1)  $VQ$  (resp:  $EQ$  是一组模式节点。边);(2)  $LQ$  在 $\Gamma$  to 中为每个模式节点 $u \in VQ$  分配一个标签。edge  $e \in EQ$ ;(3)  $TQ$  为每个模式边缘分配一个时间窗 $\tau$ ;特别地，给定一个指定的边 $ex \tau=[0,0]$ ，它固定了当前时间;(4)  $x^-$  是一个不同变量的列表;(5)  $\mu$  是一个从 $x^-$  到 $vq$  的双射映射，即它将一个不同的变量分配给每个节点 $in VQ$ 。对于 $x \in x^-$ ，当从上下文清楚时，我们互换使用 $\mu(x)$ 和 $x_0$ 。

示例2:图1描述了示例1的情况下的 $\Delta$ 模式 $\Delta Q_1 - \Delta Q_4$ :(1)  $\Delta Q_1$  指定了电影与潜在用户一起获得奖项提名的变化，其中指定的边缘 $ex$ 标记为提名;(2)  $\Delta Q_2$  描述了一个用户提出 $m$ 个关于6月烧烤的查询，其中边 $ex$ 表示最后一个查询;(3)  $\Delta Q_3$  表示在 $w$ 地区发现的疾病 $z$ 的 $m$ 例的变化，以及 $z$ 的疫苗 $y$ ;这里 $ex$ 表示报告的第一个病例;(4)  $\Delta Q_4$  是账户访问的异常模式，其中 $ex$ 是设备 $x$ 到账户 $y1$ 的第一个访问边。2

时序模式匹配。图 $G$ 中 $\Delta$ -模式 $\Delta Q[x]^-$ 的匹配是 $\Delta Q[x]^-$ 到 $G$ 的同态 $h$ ，使得(A)对于每个模式节点 $u \in VQ$ ， $LQ(u) = L(h(u))$ ;和(b)对于每个模式边 $e=(u, l, \tau, u)$ 在 $\Delta Q[x]$ 中， $\tau_e=(h(u), l, t, h(u))$ 是 $G$ 中的一条边， $t-t^* \in \tau$ 。这里 $t^*$ 表示当前时间，即边 $in G$ 的时间戳，其中指定的边 $ex(\Delta Q[x]^-)$ 被映射到 $h$ 。

我们用 $\Delta Q(G)$ 表示图 $G$ 中 $\Delta Q[x]^-$ 的所有匹配的集合。

直观地说， $\Delta Q[x]^-$ 是一种变化模式。它监视图 $G$ 的更新。在 $G$ 中 $\Delta Q[x]^-$ 的匹配可以在 $\Delta G Q$ 中计算，其中包括时间戳在 $[tmin, tmax]$ 范围内的节点和边(更新)相对于当前时间 $t^*$ ，并且被称为相对于 $\Delta Q$ 的 $G$ 更新。这里的 $tmin$ 和 $tmax$ 分别表示 $\Delta Q[x]^-$ 中的最早和最近的时间戳。更新通常比 $g$ 小得多。因此，计算 $\Delta$ -模式的匹配通常比计算传统模式的匹配更有效。

### 2.2 规则的定义和语义

TACOs是根据 $\Delta$ -模式和谓词定义的。

谓词。一个 $\Delta$ -模式的谓词 $p \Delta Q[x]^-$ 具有如下形式:

$$x.A \mid l(x, y) \mid M(x, y, l, t) \mid x.A \oplus y.B \mid x.A \oplus c \mid e_1.t \oplus e_2.t \mid e.t \oplus c,$$

其中 $x, y \in x^-$ ， $e_1, e_2$ 和 $e$ 是 $\Delta Q[x]$ 的模式边， $x.A$ 表示 $x$ 的一个属性 $A$ ， $c$ 是一个常数， $l(x, y)$ 是一条从 $x$ 到 $y$ 的边，标记为 $l$ ， $e.t$ 是边的时间戳， $\oplus$ 为 $=, <, \leq, >, \geq$ 之一。如果 $e.t \geq t_1$ ， $e.t \leq t_2$ ，我们写成 $e.t \in [t_1, t_2]$ 。我们称 $e_1.t \oplus e_2.t$ 和 $e.t \oplus c$ 为时态谓词。

特别地， $M$ 是一个用于时序图事件预测的ML分类器:如果它预测边缘 $l(x, y)$ 所指示的事件将发生在时间 $t^* + t$ ，则返回true。它可以是一个动态推荐模型，例如SASRec[37]，也可以是一个时序图补全模型，例如RE-GCN[50]。我们将 $M$ 称为ML谓词。

毫升模型。我们要求模型 $M$ 在“转导式设置”[10]中工作，它用观察到的标签推断节点之间的信息。一旦 $M$ 的训练和TACO的发现完成，观察到的标签的嵌入是固定的，并且不会引入可能违反发现的TACOs的新嵌入。这使得可以通过TACOs中的逻辑谓词解释 $M$ 。

规则。时间事件预测规则(TACO)  $\phi$  定义为

$$\Delta Q[\bar{x}](X \rightarrow (p_0, \tau)),$$

其中， $\Delta Q[x]^-$ 是一个 $\Delta$ 模式， $x$ 是 $\Delta Q[x]$ 谓词的连词， $p_0$ 是 $\Delta Q[x]$ 谓词， $\tau$ 和 $\tau$ 是一个时间窗口。当 $p_0 = l(x, y)$ 时，用 $p_0$ 表示的边被预测，并且不一定已经在 $\Delta Q[x]$ 中。我们称 $\Delta Q[x]$ 、 $x^-$ 、 $p_0$ 和 $x \rightarrow (p_0, \tau)$ 分别为 $\phi$ 的模式、先决条件、事件和依赖关系。

直观地说，TACO表示，如果先决条件 $X$ 在变化模式 $\Delta Q[X]$ ，匹配的实体上成立，那么由 $p_0$ 指定的事件将在时间窗口 $\tau$ 内发生。

例3:例1中的规则可以表示为TACOs。

(1)  $\phi_1 = \Delta Q_1[x](x \rightarrow (\text{recommend}(y, x), [0, t_1]))$ , 其中  $X_1 = \emptyset$  and  $t_1$  是以周为单位的时间戳。TACO表示, 如果用户  $x_1$  在  $y$  提名  $z$  奖项后2周内观看了电影  $y$ , 并且如果  $x$  是  $x_1$  的朋友, 则在颁奖典礼的日期  $t^* + t_1$  之前向  $x$  推荐。注意, 用户  $x_1$  在2周内观看电影的条件由函数  $tq$  在  $\Delta Q_1$  中指定(见图1)。

(2)  $\phi_2 = \Delta Q_2[x](x \rightarrow (\text{推荐}(y, x), [0, 60]))$ , 其中  $x_2(O_i \in (1, m-1) \text{ ei.t} < +1. t) \wedge (\text{我} \in (1, m) \text{ ei.t} \in (t_{\text{june}}, t_{\text{june}} + 1))$ 。在这里

$\text{Ei.t} < \text{Ei} + 1$ 。T确保搜索  $e_i$  和  $e_{i+1}$  是不同的,  $t_{\text{june}}$  (resp.  $t_{\text{june}}'$ ) 表示6月1日的日期(resp. 6月30日)。它说, 如果  $x$  在6月搜索烧烤至少  $m$  次(在  $X_2$  中指定), 那么在未来两个月(60天)向  $x$  推荐肉类。

(3)  $\phi_3 = \Delta Q_3[x](x \rightarrow (\text{offer}(y, x), [2, 6]))$ 。前置条件  $X_3$  是  $\bigwedge_{i,j \in [1,m], i \neq j} (x_i.\text{id}, X_j.\text{id})$ , 以强制  $x_i$  和  $x_j$  的情况是不同的。观察到在  $\Delta$ -模式  $\Delta Q_3$  中, 时间窗  $[0, 2]$  表明  $x_i$  在当前时间  $t^*$  之后的2周内感染疾病  $z$ , 即最早确认的病例。它说, 如果在2周内  $w$  地区至少确诊  $m$  例  $z$  疾病, 如果  $x$  人生活在  $w$  地区, 则在一个月内向  $x$  提供  $z$  疾病疫苗。

(4)  $\phi_4 = \Delta Q_4[x](x \rightarrow (x. \text{状态} = \text{欺诈}, [0]))$ ,  $X_4$  是  $O_i \in (1, k) ((O_j \text{ l} \in (1, m) j \text{ l} j.\text{id} = z \text{ li}.\text{id} \wedge \text{ei} j.\text{t}, \text{ei} 1.\text{t}) \wedge M(x, \text{嗅}, \text{攻击}, \text{ei}.\text{t}))$ 。这里  $M$  是一个ML模型, 怀疑  $x$  在时间  $\text{ei}.\text{t}$  攻击了  $\text{accounty}_i$ 。  $\Delta Q_4$  的时间窗以小时为单位。TACO说, 如果设备  $x$  访问  $y_1, \dots, y_k$  在1小时内,  $M$  怀疑它们是攻击行为, 而如果每个  $y_i$  在过去的一个月内都被其他设备访问过  $M$  次, 那么  $x$  很可能是诈骗。2

的话。(1)  $\text{gar}[21]$ 、 $\text{GARs}[65]$  和  $\text{GERs}[5]$  是TACOs的特殊情况。(a) 当所有时间窗为  $[-\infty, \infty]$  且前提条件  $X$  不含时间谓词时,  $\text{gar}$  为TACOs。为了简化讨论, 我们对每个TACO中的事件采用单个谓词  $p_0$ ; 与  $\text{gar}$  相比, 这并不会失去表达能力[20]。(b) GTAR可以表示为多个TACOs, 其中每个TACO  $(i)$  具有一个模式, 该模式扩展GTAR的前一个模式  $P_1$ , 其节点在其后一个模式  $P_2$  中;(ii) 将  $P_2$  中的单个边编码为事件  $p_0$ ,  $\tau = [t\alpha, t\alpha]$ , 其中  $t\alpha$  是GTAR中指定的恒定时间间隔。(c) 类似地, GER可以被编码为一组TACOs, 每个更新由其具有最大时间戳的模式表示[5]。

(2) 如例3的  $\phi_4$  所示, 人们可以在前置条件  $X$  中插入一个训练良好的ML预测模型  $M$ , 并用时序和逻辑条件丰富  $M$ 。形式为  $\Delta Q[x](x \rightarrow (M(x, y, l, t), \tau))$  的TACO用逻辑谓词解释ML预测。

(3) 可以将TACOs与预测事件的概率联系起来。然而, 这将招致更高的复杂度(#P-hard), 如概率逻辑编程[41]所示。

(4) TACOs可以很容易地处理间隔时间戳图G[4]。唯一不同的是, 当TACO的模式边  $e$  与  $G$  中的边  $e'$  匹配时, 我们要求  $e$  的时间范围与至少一个时间间隔  $e'$  重叠, 而不是覆盖单个时间戳。

语义。用  $h(x)^-$  在图形  $G$  中表示  $\Delta Q[x]$  的匹配,  $p$  是  $\Delta Q[x]$  的谓词。- 我们说匹配  $h(x)^-$  满足一个谓词  $p$ , 用  $h(x)^- \models p$  表示, 如果满足以下条件:(a) 如果  $p$  满足

表1:符号

Notations	Definitions
$G, \Delta Q[\bar{x}]$	graph and $\Delta$ -pattern, respectively
$\Delta Q$	updates to $G$ relative to pattern $\Delta Q$
$\varphi$	TACO $\Delta Q[\bar{x}](X \rightarrow (p_0, \tau))$
$\text{supp}(\varphi, G)$	the support of TACO $\varphi$ in graph $G$
$\text{conf}(\varphi, G)$	the confidence of TACO $\varphi$ in graph $G$
$\alpha, \beta, \gamma, \delta$	thresholds (node, support, confidence, time window)

$x.A$ , 则节点  $h(x)$  携带属性  $A$ ;(b) 如果  $p$  是  $l(x, y)$ , 那么有一条从  $h(x)$  到  $h(y)$  的边, 标记为  $l$ ;(c) 如果  $p$  是  $M(x, y, l, t)$ , 那么ML分类器  $M$  预测  $h(x)$  和  $h(y)$  在时间  $t$  与标签  $l$  之间的关联  $^* + t$ ;(d) 如果  $p$  是  $x.A \oplus y.B$ , 则属性  $A$  和  $B$  分别存在于  $h(x)$  和  $h(y)$ , 以及  $h(x). \oplus h(y)$ 。b; 同理,  $x.A \oplus c$ ; 和(e) 如果  $p$  为  $e_1.t \oplus e_2.t$ , 那么匹配  $e_1$  和  $e_2$  的边的时间戳具有  $\oplus$  关系; 同样对于  $e.t \oplus c$ 。

对于谓词的合词  $X$ , 如果  $h(X)^- \models p$  对于  $X$  中的所有  $p$ , 我们写成  $h(X)^- \models X$ 。对于TACO  $\varphi = \Delta Q[X](X \rightarrow (p_0, \tau))$ , 如果  $h(X)^- \models X$  意味着  $h(X)^- \models p_0$  并且  $p_0$  发生在时间窗  $\tau$  内, 我们写成  $h(X)^- \models \varphi$ 。

我们说图  $G$  满足TACO  $\varphi = \Delta Q[x](x \rightarrow (p_0, \tau))$ , 用  $G \models \varphi$  表示, 如果对于  $G$  中  $\Delta Q[x]^-$  的所有匹配  $h(x)^-$ ,  $h(x)^- \models \varphi$ 。我们说  $G$  满足一个集合  $\Sigma$  of TACOs, 表示为  $G \models \Sigma$ , 如果对于所有TACOs  $\varphi \in \Sigma$ ,  $G \models \varphi$ , 即  $G$  满足  $\Sigma$  中的所有TACOs。

示例4: 考虑图1中所示的时序图  $G$ , 其中每条边都标注了一个时间戳。则  $G$  满足例3的TACO  $\phi_1$ 。观察到  $G$  中只存在一个单体匹配  $h(x)^-$  of  $\Delta Q_1$ :  $x_7 \rightarrow u_2$ ,  $x_1 \rightarrow u_1$ ,  $x_2 \rightarrow a$  和  $y_7 \rightarrow m_1$ 。由于  $u_2$  通过  $G$  中时间戳  $t_1$  的推荐边与  $m_1$  相连, 我们知道  $h(x)^- \models (\text{recommend}(y, x), [0, t_1])$ ,  $G \models \phi_1$  紧随其后。2

论文的符号总结在表1中。

### 3 关于时态规则的推理

在本节中, 我们研究了TACOs的基本问题, 包括可满足性、蕴涵和预测问题。我们表明, 尽管TACOs比  $\text{gar}$  更具表达性[21], 但TACOs的这些问题并不比  $\text{gar}$  难多少。

可满足性。可满足性问题如下。

- 输入: 一组  $\Sigma$  of TACOs。
- 问题: 是否存在这样的图  $G$ : 对于每个TACO  $\Delta Q[x](x \rightarrow (p_0, \tau)) \in \Sigma$ ,  $\Delta Q$  在  $G$  中有匹配?

直观地说, 这是为了确保发现的TACOs没有冲突, 并且可以同时应用于图。

这个问题对于GARs是  $\text{conp}$  完备的[21], 对于图拒绝约束(gdc)是  $\Sigma_2^P$  完备的[22]。我们展示了以下内容。

定理1: TACOs的可满足性问题是  $\Sigma_2^P$  完全的。2

这里  $\Sigma_2^P$  表示可以使用NP oracle在NP中检查的决策问题的类别, 即  $\Sigma_2^P = \text{NPNP}$ 。类似地,

定义类  $\Pi_2^P = \text{coNPNP}$  (详见[70])。

我们假设w.l.o.g. TACOs中的ML谓词如下:(a) 机器学习预测(即使用预训练的  $M$  进行测试)需要多项式时间(PTIME), 并且(b) 存在一个“小”范围的值, 使得对于任何节点对  $(x, y)$ , 它们的属性值可以由小范围内的值编码, 并为  $M$  产生相同的真值。这是TACOs在实践中使用的机器学习模型的情况。

证明草图:我们首先证明了一个小的模型性质:如果一个集合 $\Sigma$  of TACOs是可满足的, 则存在一个时间图 $G \models \Sigma$  such, 其 $|G| \leq 4|\Sigma| + 3$  且  $G \models \Sigma$ 。在此基础上, 我们开发了一种 $\Sigma_2P$ 算法来检验给定的一组 $\Sigma$  of TACOs是否可满足。

下界是由gdc的可满足性问题 $\Sigma_2P$ -完备性得出的[22], 因为很容易验证gdc也是TACOs的一个特例。2

暗示。一个集合 $\Sigma$  of TACOs隐含一个TACO  $\phi$ , 表示为 $\Sigma \models \phi$ , 如果对于所有图 $G$ , 如果 $G \models \Sigma$  then  $G \models \phi$ 。

蕴涵问题表述如下。◦ 输入: 一组 $\Sigma$  of

TACO和一个TACO  $\phi$ 。

◦ 问题:  $\Sigma \models \phi$  吗?

蕴涵分析帮助我们去除冗余规则。

隐含问题对于GARs是 $NP$ 完全的[21], 对于gdc是 $\Pi_2 P$ 完全的[22]。对于TACOs来说, 这并不比gdc更难。

定理2: 隐含问题是 $\Pi_2$ ——对于TACOs来说是完整的。p 2  
证明草图:我们证明了问题的一个小模型性质:如果 $\Sigma \models \phi$ , 那么存在一个时间图 $G \models \phi$ 使得 $|G| \leq 8(|\phi| + |\Sigma|) + 2$ ,  $G \models \Sigma$ 但是 $G \not\models \phi$ 。在此基础上, 我们设计了一个 $\Sigma_2P$ 算法来检验 $\Sigma \models \phi$ 是否成立。下界直接遵循了GDCs[22]的 $\Pi_2$ -完全蕴涵问题。

2

预测。我们也研究了预测问题。

◦ 输入: 一个时间图 $G$ , 一组 $\Sigma$  of TACOs, 一个时间窗口 $\tau$ , 一个标签 $l$ , 以及 $G$ 中的两个节点 $u$ 和 $v$ 。

◦ 问题: 是否存在边缘标记 $l$ 从 $u$ 到 $v$ 在 $\tau$ 由 $\Sigma$ ? 本文旨在研究TACOs时间预测的复杂性。

一个类似的问题(演绎问题)被证明是GARs的 $NP$ 完全问题[21]。用TACOs预测事件并不难。

定理3: TACOs的预测问题是 $NP$ 完全的。2

证明概要: 对于上界, 我们定义了一个证明树的概念, 并表明 $(\Box)$  预测 $(p_0, \tau)$ 可以在 $G$ 中推导当且仅当存在一个证明树 $T$ , 具有 $|T| \leq 7|G| + 2(|G| + |\tau|) + 3$  见证预测。基于(), 我们设计了一种 $NP$ 算法: 猜测一棵树 $T$ , 使得 $|T| \leq 7|G| + 2(|G| + |\tau|) + 3$ , 并检查它是否是 $P$ TIME中预测 $(p_0, \tau)$ 的证明树。

我们通过对图同态问题的约简给出了下界, 该问题是 $NP$ 完全的(参见[27])。后者是决定, 给定两个无向图 $G_1$ 和 $G_2$ , 是否存在从 $G_1$ 到 $G_2$ 的同态 $h$ 。我们用TACO  $\phi$ 编码 $G_1$ , 用时间图 $G$ 编码 $G_2$ , 用 $(p_0, \tau)$ 表示 $h$ 。对于间隔时间戳图[4], 其复杂度界限保持不变。对于预测问题, 我们只需要检查一个时间间隔是否与另一个时间间隔重叠, 这需要 $P$ TIME。对于另外两个问题, 小的模型属性保持不变, 复杂度也保持不变。此外, 当从密集有序的集合中选择时间戳时, 结果也保持不变。

## 4 时间系统

在本节中, 我们将介绍TASTE (Temporal SysTEm), 这是一个用于发现TACOs并使用TACOs预测事件的系统。我们首先陈述了相应的发现和预测问题(4.1节)。然后我们介绍TASTE的体系结构(第4.2节)。

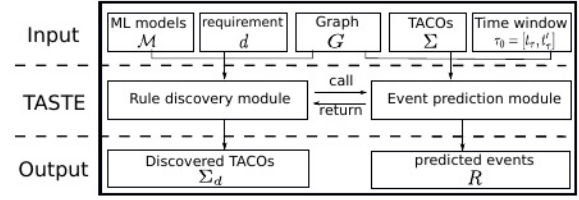


图2: TASTE的体系结构、输入和输出

### 4.1 TACO发现和事件预测

我们首先定义TACOs的质量度量。

质量指标。我们使用支持度和置信度来量化TACOs的质量, 分别为频率和可靠性。

支持。图 $G$ 中TACO  $\phi = \Delta Q[x](x \rightarrow (p_0, \tau))$ 的支持度为 $\text{supp}(\phi, G) = |\Delta Q(e x, \phi, G)|$ ,

其中,  $\Delta Q(e x, \phi, G)$ 表示这些匹配中的边 $h(ex)$ 的集合 $h(x) \in \Delta Q(G)$ , 使得 $h(x) \models x$ ,  $h(x) \models p_0$ , 并且 $p_0$ 出现在 $\tau$ 中。

直观地说, 支持计算关于指定边 $ex$ 的不同“可满足”匹配的数量。我们可以证明这个测度是反单调的, w.r.t. TACOs上的偏阶 $\Box$ , 即对于任意 $G$ , 当 $\phi \Box \phi'$ 时,  $\text{supp}(\phi, G) \geq \text{supp}(\phi', G)$ 。

的信心。我们定义图 $G$ 中TACO  $\phi = \Delta Q[x](x \rightarrow (p_0, \tau))$ 的置信度, 用 $\text{conf}(\phi, G)$ 表示, 如下:

$$\text{conf}(\phi, G) = \frac{\text{supp}(\phi, G)}{|\Delta Q(e x, X, G)|}.$$

这里 $\Delta Q(e x, x, G)$ 是满足 $x$ 的所有匹配 $h \in \Delta Q(G)$ 中的边 $h(ex)$ 的集合。比率量化了事件 $p_0$ 在 $\tau$ 中发生的匹配满足预测 $x$ 的可能性, 即可靠性。

TACO发现。TACOs的发现问题如下。

◦ 输入: 一个时间图 $G$ , 一个正整数 $\alpha$ , 一个支持阈值 $\beta > 0$ , 一个置信度阈值 $\gamma \in [0, 1]$ , 和一个时间窗长度的界限 $\delta > 0$ 。

◦ 输出: 一个 $\Sigma$  of TACOs集合, 使得对于每个 $\phi \in \Sigma$ ,  $\phi$ 中每个时间窗口 $[t_1, t_2]$ 的 $\text{supp}(\phi, G) \geq \beta$ ,  $\text{conf}(\phi, G) \geq \gamma$ ,  $t_2 - t_1 \leq \delta$ , 且 $\phi$ 模式中最多存在 $\alpha$ 节点。

我们将元组 $(\alpha, \beta, \gamma, \delta)$ 称为发现需求 $d$ 。问题的目的是发现符合 $d$ 的所有TACOs的集合, 并且具有以 $\alpha$ 为界的期望模式节点数。

时序事件预测。问题表述如下。

◦ 输入: 一个图 $G$ , 一组 $\Sigma$  of TACOs和一个时间窗 $\tau_0$ 。◦ 输出:  $\Sigma$  such 预测 $G$ 的边缘集合 $R$ ,  $R$ 中每个边缘编码的事件的发生在 $\tau_0$ 内。

为了简化讨论, 我们认为事件 $p_0$ 由边缘 $l(x, y)$ 表示, 在实践中常见。

### 4.2 TASTE概述

TASTE的体系结构如图2所示。TASTE支持TACO发现和事件预测两个模块, 如下所示。

(1) 规则发现。发现模块负责从时间图 $G$ 中挖掘一组 $\Sigma_d$ 符合给定需求 $d$ 的TACOs, 称为高质量规则。除了 $d$ 之外, 它还将预训练的ML模型 $M$ (第2节)作为额外的输入, 并在发现 $\Sigma_d$ 时将它们嵌入到TACOs中。



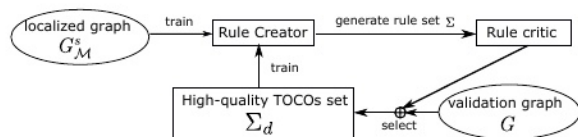


图3:迭代TACO发现的数据流。

(2)事件预测。预测模块使用 $\Sigma_d$ 中发现的TACOs, 通过运行并行算法来预测所有事件(或特定事件)。应该提到的是, 如果可用, 该模块还可以接受领域专家提供的TACOs。

我们将在第5节和第6节分别介绍发现和预测模块背后的算法。

## 5 基于代的TACO发现

在本节中, 我们提出了一种创造者-评论家发现方法, 通过CCD来发现高质量的TACOs  $\Sigma_d$ 。我们为味觉的规则发现模块开发了这样一种算法。

下面我们首先介绍基于生成的方法, 并从概率分布学习的角度证明该方法。然后我们介绍CCD及其功能和模型。最后对CCD的性能保证进行了形式化证明。

生成的发现。有人可能会尝试扩展现有的基于层次搜索的方法, 例如[5,20], 以列举候选TACOs并确定所需的taco。然而, 这些方法需要在格子中搜索, 并需要指数时间, 尽管有剪枝策略来降低成本[5,20,65]。

鉴于此, 我们提出CCD, 它通过使用生成式对抗网络(GAN), 并在规则创建者和规则批评者之间使用弱监督的交互式学习来发现高质量的TACOs。这种基于GAN的框架受到药物发现和化合物设计的最新进展的启发[6,88], 其中GAN被训练以生成有希望的候选物, 而无需在大搜索空间中列举所有可能的组合。这样的生成式方法大大加快了药物发现过程, 并产生了更有希望的化合物。

如图3所示, 在每次迭代中, CCD的创建者首先使用ML模型来生成一组 $\Sigma$ of候选TACOs。然后, 评价者从 $\Sigma$ and中选择高质量的规则添加到 $\Sigma_d$ 中。将 $\Sigma_d$ 中的规则作为评论家的反馈, 创造者再次训练自己, 以提高在下次迭代中生成高质量TACOs的概率。当迭代次数达到用户指定的界 $I$ 时, CCD返回 $\Sigma_d$ 。

从概率论的角度来看, 每个TACO  $\phi$ 可以看作一个事件, 所有TACOs的并构成一个离散的可数无限样本空间 $\mathcal{a}$ 。给定需求 $d$ , 发现的目的是找到一个概率质量函数 $P: \mathcal{a} \rightarrow \mathbb{R}$ (实数)所描述的概率分布 $P$ , 其中

$$\begin{cases} P(\phi) > 0, \text{ if } \phi \text{ is a high-quality TACO w.r.t. } d, \\ P(\phi) = 0, \text{ if } \phi \text{ is not a high-quality TACO w.r.t. } d. \end{cases} \quad (1)$$

为了找到分布, 传统的水平方法建立一个搜索格来枚举 $\mathcal{a}$ 中的每一个 $\phi$ 规则, 并检查它是否满足要求 $d$ 。相比之下, CCD在creator中交互训练深度生成模型[63,110]来近似目标分布 $P$ , 避免在指数级大的样本空间中进行代价高昂的搜索。

### Algorithm 1: Creator-Critic Discovery (CCD)

**Input:** A temporal graph  $G$ , discovery requirement  $d=(\alpha, \beta, \gamma, \delta)$ , ML models  $\mathcal{M}$ , sample size  $N$ , and iteration number  $I$ .

**Output:** A set  $\Sigma_d$  of TACOs mined from  $G$  such that every rule in  $\Sigma_d$  conforms to the requirement  $d$ .

```

1   $\varepsilon \leftarrow 0$ ;  $\Sigma_d \leftarrow \emptyset$ ;  $G_M \leftarrow \text{MLExp}(G, \mathcal{M})$ ;
2   $G_M^s \leftarrow \text{LocalizedSample}(G_M, \alpha, N)$ ;
3   $\text{TrainCreator}(G_M^s, \alpha, \delta)$ ; /* Pretrain the rule creator */
4   $\Sigma \leftarrow \text{GenerateRule}(\alpha, \delta, \Sigma_d)$ ;
5   $\Sigma_d \leftarrow \text{SelectRule}(\Sigma, \beta, \gamma, G)$ ;
   /* Generate TACOs via interactive training */
6  while  $\varepsilon < I$  do
7     $G_M^s \leftarrow \text{LocalizedSample}(G_M, \alpha, N)$ ;
8     $\text{TrainCreator}(G_M^s \cup \Sigma_d, \alpha, \delta)$ ;
9     $\Sigma \leftarrow \text{GenerateRule}(\alpha, \delta, \Sigma_d)$ ;
10    $\Sigma' \leftarrow \text{SelectRule}(\Sigma, \beta, \gamma, G)$ ;
11    $\Sigma_d \leftarrow \Sigma_d \cup \Sigma'$ ;  $\varepsilon \leftarrow \varepsilon + 1$ ;
12 return  $\Sigma_d$ ;
```

这种方法是可行的, 因为深度生成模型, 如GAN, 能够近似各种分布[13]。此外, 实证研究验证了在有限的迭代次数[31]内训练这样的模型来近似一个目标分布 $P$ 是可行的, 在这里模型的训练损失收敛。CCD还以数据驱动的方式优化规则发现, 因为规则生成器中的生成模型能够从训练数据中生成拓扑结构和语义相似的规则[31,47,110]。同时, 生成器可以使用评论家选择的高质量规则作为反馈来重新训练其生成模型, 以提高近似的质量。

算法。CCD采用创建者-批评家方法, 分三个阶段发现高质量的TACOs, 如算法1所示。

(1)首先调用函数MLExp从 $G$ 中准备图数据 $G_M$ , 便于发现带有ML谓词 $M$ 的TACOs。

(2)然后使用TrainCreator对规则创建者进行预训练, 使用LocalizedSample采样的本地化图 $G_M^s$ (第2-3行), 使其开始生成可能达到输入需求 $d$ 的支持度和置信度阈值的规则。创建者通过GenerateRule生成一组 $\Sigma$ of候选TACOs;评论家接下来评估 $\Sigma$ on时间图 $G$ , 并通过函数SelectRule(第4-5行)将高质量的TACOs保存在一个集合 $\Sigma_d$ 中。

(3)第三阶段, CCD算法迭代发现高质量的TACOs, 并通过 $I$ 次迭代的交互训练过程(第6-11行)将更多的高质量TACOs规则添加到 $\Sigma_d$ 。每个迭代都类似于第二阶段, 除了 $G_M^s$ 和 $\Sigma_d$ 都用于训练和改进创建者(第8行)。这里 $\Sigma_d$ 中的TACOs是来自评论家的反馈。最后 $\Sigma_d$ 被返回(第12行)。

参数。除了四个阈值的发现要求 $d$ 和ML模型 $M$ 之外, CCD还需要另外两个输入参数:样本量 $N$ 和迭代次数 $i$ 。这里 $N$ 在生成高质量规则的概率和每次迭代的工作量之间取得了平衡; $I$ 决定了 $\Sigma_d$ 的大小和模型训练的成本, 其中较大的 $I$ 通常有助于生成更多高质量的TACOs并更好地训练模型。

接下来我们介绍CCD中采用的功能和模型。

帮助功能。我们从两个函数MLExp和LocalizedSample开始，CCD调用它们为模型训练准备图形数据。最初，MLExp通过忠实地添加输入ML模型M预测的边来将图G扩展到G<sub>M</sub>，这允许创建者将M合并为ML谓词并加速TACO发现，因为在发现过程中不需要重复应用M。

此外，LocalizedSample通过从G<sub>M</sub>中采样局部化图结构，收集一组G<sub>sM</sub>的N Δ-模式，作为每次迭代的创建者的训练数据(第2行和第7行)。在推导模式ΔQ<sub>i</sub>时，LocalizedSample对样本时间路径应用随机选择源节点v的时间随机行走[69]，其中时间路径上所有边的时间戳落在给定的时间窗口内。它在v周围找到顶部α-1个频繁采样的节点，然后ΔQ<sub>i</sub>由这些节点和连接它们的边形成。在Δ模式的指导下inG<sub>sM</sub>在预训练期间(第3行)，规则创建者可以学习生成更有可能在图中找到匹配的模式。这是因为每个ΔQ<sub>i</sub>必须有匹配，因为它是在G<sub>M</sub>中随机游走获得的，并且生成模型学习生成在语义和结构上类似于ΔQ<sub>i</sub>的“新”模式。如果没有预训练，在创建者中随机初始化的生成模型可能会产生无意义的TACOs。

事实上，更大的样品量N允许更多的采样轨迹，以更高的概率生成良好的模式，并通过批量优化进行更高效的ML训练[63,110]。如果N很小，CCD需要运行更多的迭代来找到足够数量的TACOs。

规则的创造者。当生成一个TACO φ时，创建者首先生成它的Δ-模式，然后是依赖性(第4行和第9行)。

Δ生成模式。创建者分两步生成模式:结构生成和语义标签生成，分别采用时序图GAN和LSTM网络。

(1)在第一步中，创建者将来自G<sub>sM</sub>和Σd的每个Δ-模式ΔQ<sub>i</sub>作为输入:它采用TagGen[110]，一个端到端的时间图深度生成框架，来推断候选的Δ-模式ΔQ<sub>i</sub><sub>d</sub>。由于TagGen中的GAN模块[110]，每个生成的ΔQ<sub>i</sub>与输入的ΔQ<sub>i</sub>具有相同的节点数(在d处大多数α)，并承受相似的拓扑结构和时间约束。由于TagGen不支持标签，此步骤没有生成标签。

(2)创建者接下来使用LSTM语言模型ML为每个ΔQ<sub>i</sub><sub>d</sub>生成标签。我们采用LSTM网络，因为它可以模拟知识图中路径上标签的丰富语义[47,52,53]。更具体地说，它首先在由perplexity[63]驱动的语料库C上训练ML，其中每个单词都是一对L(e)，L(v)的边标签和节点标签，名为“标签对”，v是e的目标节点。语料库C由时间路径的标签对序列组成，这些序列是通过在每个ΔQ<sub>i</sub>上应用时间随机游走而派生的。训练后，对于d中的每两个nodesu和v

ΔQ<sub>i</sub><sub>d</sub>with最短时间路径ρ从mu到v，ML生成一个带有随机种子的标签对序列，并将这个标签序列附加到ρ。只考虑最短路径，因为它们具有更强的关联[2,39]。最后，创建者从ΔQ<sub>i</sub><sub>d</sub>by建立一个Δ模式ΔQ<sub>i</sub>'，保持最频繁的标签。

请注意，创建器使用最新的Σd重新训练自己，以增加生成高质量Δ模式的概率

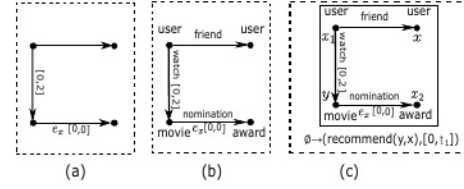


图4:TACO生成过程。

(第8行)。这背后的基本原理是(1)在训练数据中，Σd中高质量TACO的Δ模式ΔQ<sub>i</sub>具有多个匹配;(2) GAN和LSTM网络使创建者能够生成新的Δ模式——ΔQ<sub>i</sub> that在拓扑结构、时间约束和标签语义方面与ΔQ<sub>i</sub>相似[47,52,110]。因此，生成的ΔQ<sub>i</sub>'is也很可能找到足够的匹配。LocalizedSample返回的样本G<sub>sM</sub>也用于再训练，向创建者引入干扰，从而防止ML生成模型在重复生成陈滥调模式的局部最小值处收敛。

依赖的一代。给定一个生成的Δ-模式ΔQ，创建者适应[20]的水平扩展过程来构建一组有效的依赖关系X→(p<sub>0</sub>, τ)。更具体地说，对于每个可能的事件(p<sub>0</sub>, τ)，它从X=∅开始，并迭代地扩展ΔQ的前置条件X。但与[20]在每个候选TACO φ=ΔQ[x](x→(p<sub>0</sub>,τ))生成后直接验证其有效性不同，创建者首先检查φ是否冗余，即φ是否由前几次迭代中发现的TACOs集合Σd隐含。如果φ不冗余，即Σd⊭φ，则继续进行评价者对φ的验证。我们发现提前检查隐含有助于降低发现成本，因为隐含是在Σd上进行的，比图g小得多。此外，我们在此阶段对Σand中新发现的候选规则中的TACOs进行可满足性检查，以避免进一步验证g中不一致的TACOs。当无法添加更多谓词时，该过程终止。终止是保证的，因为X中的谓词定义在ΔQ中的(a)节点上，以及更新ΔGQ中的(b)属性和值(包括时间戳)上;因此，所有可能谓词的数量都以ΔQ和ΔGQ的大小为界限。

规则的评论家。对于每个生成的候选TACO φ在Σthat中通过了隐含性和可满足性检查，规则评论家计算supp(φ, G)和conf(φ, G)，并选择高质量的TACOs，其支持度和置信度高于阈值(第5行和第10行中的SelectRule)。这些规则被添加到Σd中，并提供给规则创建者，以便在下次迭代中改进生成模型。计算supp(φ, G)和conf(φ, G)是高效的，因为这些过程可以与时间模式匹配所特有的优化并行化(参见第6节)，并且我们使用现有的高效子图匹配方法，即DAF[32]及其提出的CS结构来减少冗余计算。

例5:继续例4，图4显示了从时间图g生成TACO φ1的过程。首先，创建者调用TagGen来创建一个有时间约束的候选Δ-模式(图4(a))。然后，如图4(b)所示，利用LSTM模型ML为节点和边缘添加具有语义的标签。创建者通过构建与变量的依赖关系来完成生成(图4(c))。经过G中的规则评论家的验证，生成的候选规则保存在Σd中，因为它支持和

置信度都为1。注意，如果标记为提名的边从 $\phi$ 中去掉，置信度将降低到0.5。2

性能保证。当迭代次数 $I$ 足够大时，CCD可以以高概率地返回满足要求 $d$ 的所有TACOs。回想一下，当为生成模型积累每轮训练数据时，CCD通过函数LocalizedSample对采样子图进行时序随机游走。基于此，我们有以下内容。

定理4:给定图 $G$ ，要求 $d$ 和常数 $\lambda \in (0,1)$ ， $|G| \alpha \beta \lambda \beta \lambda \lambda \lambda$ 经过 $N \beta 1 - \ln |G| \alpha + \ln |G| \alpha (\ln |G| \alpha - 2)$ 迭代，CCD能够以至少 $1 - \lambda$ 的概率发现满足 $d$ 的所有TACOs。2

证明草图:由于[20]的水平扩展过程被适应于构建依赖关系，它足以表明CCD可以在上文给出的充分迭代之后，对满足概率 $1 - \lambda$ 的所有 $\Delta$ -模式进行采样。使用Chernoff-Hoeffding界[17]和bonferroni不等式[9]，我们通过分析在整个迭代过程中挖掘的期望 $\Delta$ 模式数量来证明这一点。2

成本分析。要看CCD的成本，请观察以下。函数LocalizedSample应用时间随机游走需要 $O(|G|)$ 时间[69]。TrainCreator中的模型训练成本与训练样本数量呈线性关系，即 $O(N + |\Sigma d|)$ 。GenerateRule(creator)需要一个大小为 $|G|$ 的时间多项式来生成 $\Delta$ -模式和依赖关系。而SelectRule(即critic)由于时序匹配中的图同态，计算每个候选TACO的支持度和置信度需要 $O(|G| \alpha)$ 时间;尽管如此，我们还是利用并行性(参见第6节)和[32]中的辅助结构来加快计算速度。

的话。(1)虽然理论迭代次数 $I$ 可能很大，但在实践中，当样本大小 $N$ 默认设置为250时，生成模型在25次迭代内收敛(即 $I=25$ )，其中额外的迭代会给 $\Sigma d$ 增加很少的新TACOs(参见第7节)。因此，我们使用一个较小的实用值来设置 $I$ 。这有助于我们降低总体成本，并证明深度生成模型的使用是合理的。

(2)用户可以选择检查每轮生成的TACOs，选择自己感兴趣的规则，并添加到 $\Sigma d$ 。这就将用户的兴趣融入到发现中。当他们对 $\Sigma d$ 中的TACOs满意时，他们也可以手动终止迭代过程。

## 6 平行事件预测

在本节中，我们开发了一个并行算法来支持味觉的事件预测模块。该算法还用于计算算法CCD(第5节)的规则评价器的支持度和置信度。我们提出了时序图的划分策略，并表明该算法保证了并行可扩展性。

我们从TACOs的顺序预测方法开始。

序贯算法。给定一个时间图 $G$ 、一组 $\Sigma$ of TACOs和一个时间窗 $\tau_0$ ，一个记为SeqEP的顺序预测算法可以找到 $\Sigma$ in  $G$ 预测的所有边(事件) $R$ ，如下所示。对于每个TACO  $\phi = \Delta Q[x](\Sigma \text{with } \tau = [t_1, t_2])$  中的  $x \rightarrow (p_0, \tau)$ ，SeqEP(1)通过图同胚性找到 $\Delta G$   $Q$ 中的 $\Delta Q$ 的所有匹配;和(2)对于每个这样的匹配 $h(x)$ ，它检查 $h(x) = x$ 和时间窗口 $[t^* + t_1, t^* + t_2]$ 推导出的时间窗口是

$\tau_0$ 的子区间;这里 $t^*$ 是当前时间(第2.1节);如果是这样，SeqEP将连接 $p_0$ 中匹配变量的节点的边添加到 $R$ 中。

注意，SeqEP应用于 $\Sigma$ 中 $\Delta Q$ 的子图 $\Delta gq(G)$ ，它通常比整个图 $G$ 小得多。在后续中，我们将这些 $\Delta gq$ 的并集简称为asG。尽管如此，时序模式匹配(步骤(1))中图同态[27]的指数代价促使我们并行化SeqEP。

平行的可伸缩性。为了衡量并行化的有效性，我们将[43]引入的标准应用到图计算中。考虑图 $G$ 上的问题 $P$ ，我们用 $t(|P|, |G|)$ 表示处理问题 $P/G$ 的实例 $I_P$ 的顺序算法 $F$ 的最坏情况复杂度。对于并行算法 $F_p$ ，我们用 $T_p(|P|, |G|, k)$ 表示它在 $G$ 上使用 $k$ 个处理器处理问题实例 $I_P$ 所花费的时间。我们说算法 $F_p$ 相对于 $F$ 是并行可伸缩的

$$T_p(|I_P|, |G|, k) = O\left(\frac{T_s(|I_P|, |G|)}{k}\right)$$

例如 $I_P$ 。也就是说，并行算法 $F_p$ 在尺度算法 $F$ 的连续运行时间上实现了“线性”减少，允许我们通过添加资源来处理大型图。

并行化事件预测。并行化顺序图计算的一个典型策略是首先将图划分为 $k$ 个小片段，然后在 $k$ 个处理器上通过必要的消息传递并行地进行片段上的计算，例如使用GARs[21]的演绎算法。遵循这一范式，我们可以通过现有的图分割方法来分割时序图，例如，边切或顶点切[3,77]。然而，这很容易在随后的并行预测中引起大量的通信。这是因为之前的大多数划分方法旨在最小化切割边或顶点的数量，但忽略了时间戳，而时间戳对时间模式匹配至关重要;一个 $\Delta$ 模式匹配中的边往往被分割成不同的片段，并需要通信。

为了解决这个问题，我们基于时间局部性划分了一个时间图 $G$ ，这是与TACOs匹配的时间模式的独特性质。也就是说，每个匹配中的边的时间戳 $h(x)$  of  $\Delta$ -pattern  $\Delta Q$ 都在局部时间跨度 $[t^* + t_{\min}, t^* + t_{\max}]$ 的范围内，其中 $t_{\min}$  (resp.  $T_{\max}$ )是最小值 (resp. 最大)时间戳在 $\Delta Q$ 中，如2.1节所述。

时间分区。我们提出了一种时间划分策略，利用该策略可以实现无通信的并行事件预测。直观地说，它将一个时间区间划分为 $k$ 个子区间，并保证寻找匹配 $h(x)$ 的每个特定时间戳范围完全被一个子区间覆盖。在由此产生的子区间的指导下，时序图 $G$ 被划分为 $k$ 个片段 $F_1, \dots, F_k$ ，使得每个 $F_i$ 由时间戳在一个子区间内的边组成。通过时间局部性，可以在不通信的情况下在这些 $F_i$ 上并行进行时间模式匹配和事件预测。

并行事件预测的代价由片段的最大大小 $\max_{i \in [1, k]} |F_i|$ 决定。因此，我们需要找到一个很好的时间区间划分，使 $\max_{i \in [1, k]} |F_i|$ 最小。为了做到这一点，我们开发了函数BTPart，如图2算法所示。它以一个候选时间间隔 $(t_0, t_k)$ 作为输入，该候选时间间隔是从TACOs中推导出来的 $\Sigma$ and时间窗 $\tau_0$ 来匹配指定的



### Algorithm 2: Parallel Event Prediction (ParEP)

**Input:** The number  $k$  of processors, a  $k$ -way randomly partitioned temporal graph  $G$ , a set  $\Sigma$  of TACOs, a time window  $\tau_0$ .

**Output:** A set  $R$  of edges predicted by  $\Sigma$  with events occur within  $\tau_0$ .

```

1   $[t_0, t_k] \leftarrow \text{RefTime}(\Sigma, \tau_0); \quad (t_{\min}, t_{\max}) \leftarrow \text{ExtractTS}(\Sigma);$ 
2  collect the size  $|G_t|$  of  $t$ -graph  $G_t$  for  $t \in [t_0, t_k];$ 
3   $\{t_1, \dots, t_{k-1}\} \leftarrow \text{BTPart}(\{[t_0, t_k], \{|G_t| \mid t \in [t_0, t_k]\}\}, k, t_{\min}, t_{\max});$ 
4   $F_i \leftarrow G_{[t_{i-1}+t_{\min}, t_i+t_{\max}]}$  for each  $i \in [1, k];$ 
5   $\text{RBalance}(\{F_i \mid i \in [1, k]\});$ 
6  run  $\text{SeqEP}(F_i, \Sigma, \tau_0)$  at each fragment  $F_i$  to get  $R_i$  for  $i \in [1, k];$ 
7  return  $\bigcup_{i \in [1, k]} R_i;$ 

Function  $\text{BTPart}(\{[t_0, t_k], \{|G_t| \mid t \in [t_0, t_k]\}\}, k, t_{\min}, t_{\max}):$ 
1  foreach  $t \in [t_0, t_k]$  do
2     $S[t][1] \leftarrow |G_{[t_0+t_{\min}, t+t_{\max}]}|;$ 
3  foreach  $i \in [2, k]$  do
4    foreach  $t' \in [t_0, t_k]$  do
5       $S[t'][i] \leftarrow \min_{t \in [t_0, t']} \max(S[t][i-1], |G_{[t+t_{\min}, t'+t_{\max}]}|);$ 
6  foreach  $i \in [2, k]$  in descending order do
7     $t_{i-1} \leftarrow \arg \min_{t \in [t_0, t_i]} \max(S[t][i-1], |G_{[t+t_{\min}, t_i+t_{\max}]}|);$ 
8  return  $\{t_1, \dots, t_{k-1}\};$ 

```

事件预测中的边(见下文),  $t$ 图大小的集合 $\{|G_t| \mid t \in [t_0, t_k]\}$ , 片段(处理器)的数量 $k$ , 以及最大(resp. 最小值)时间戳 $t_{\max}$  (resp.)  $t_{\min}$ 在 $\Sigma$ 。这里 $G_t$ 指的是由时间戳为 $t$ 的 $G$ 中的所有边组成的 $t$ 图。BTPart计算一个集合 $\{t_1, \dots, t_{k-1}\}$ 对于区间 $[t_0, t_k]$ 的 $k-1$ 个分区点, 使得对于 $i \in [1, k]$ , 片段 $F_i$ 变成 $G_{[t_{i-1}+t_{\min}, t_i+t_{\max}]}$ , 其中包括时间戳在 $[t_{i-1}+t_{\min}, t_i+t_{\max}]$ 范围内的 $G$ 的边。这里的 $t_{\min}$ 和 $t_{\max}$ 是为了保证如上所述的整个覆盖范围。

BTPart采用动态规划。它维护一个二维数组 $S$ , 其中 $S[t][i]$ 记录了通过将 $G_{[t_0+t_{\min}, t+t_{\max}]}$ 划分为 $i$ 个子区间得到的最大片段的最小大小。因此 $S[t_k][k]$ 是我们的目标值。BTPart首先处理基本情况 $i=1$  (第1-2行), 其中 $S[t][1] = |G_{[t_0+t_{\min}, t+t_{\max}]}|$ 因为只有一个片段w.r.t. 单个子区间。这里的大小是从输入 $t$ -图的大小派生出来的。对于 $i > 1$ 的情况,  $S[t][i]$ 是通过检查关于 $t < t'$ 及其 $i-1$ 子区间的所有可能范围 $[t_0, t]$ 的值来确定的(行3-5)。  $S[t_k][k]$ 计算完成后, 就可以识别出对应的 $k-1$ 个分区点, 并作为结果返回(第6-8行)。

通过对迭代的归纳, 可以验证返回的分区点产生最小的 $\max_{i \in [1, k]} |F_i|$  (即 $S[t_k][k]$ )。

并行算法。利用时间划分, 我们开发了一种并行预测算法, 用ParEP表示。如算法2所示, 最初ParEP使用函数RefTime来推断 $G$ 中可能与TACOs  $\Sigma$ 中指定的边缘匹配的边缘的时间间隔 $[t_0, t_k]$  (第1行)。由于我们只预测时间窗口 $\tau_0$ 中的边缘, 因此时间标记 $t$ 在 $[t_0, t_k]$ 中当且仅当 $t$ 和 $\tau_0$ 之间的间隙小于 $\Sigma$ 中某些 $(p_0, \tau)$ 之间的间隙时。它还通过ExtractTS从 $\Sigma$ 提取最大和最小时间戳(第1行)。然后ParEP执行时间分区以获得 $k$ 个片段 $F_1$ 到 $F_k$ , 每个 $F_i$ 被分配给一个不同的处理器(第2-4行)。它指定一个处理器作为协调器来收集 $t$ 图的大小并应用函数BTPart。

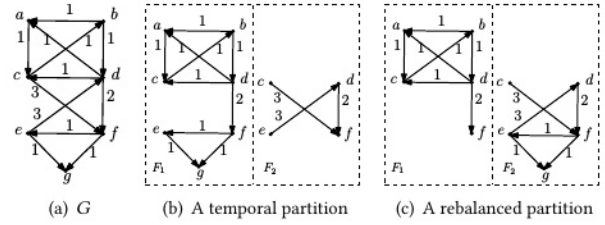


图5:时间分区和再平衡

为了进一步平衡并行预测的工作量, ParEP接下来通过函数RBalance(第5行)将数据重新分配为 $|F_i| > |G|/k$ 的大片段 $F_i$ 。更具体地说, 对于每个大的 $F_i$ , 它将 $F_i$ 中能够匹配 $\Sigma$ 中指定边缘的边缘均匀分区;而在 $F_i$ 中对 $\Sigma$ 中其他模式边的候选匹配集在所有处理器上复制。这里的候选匹配是通过标签比较来识别的。然后所有处理器在没有通信的情况下并行调用SeqEP来获得预测的边, 并最终返回它们的union(第6-7行)。

示例6:考虑将图5(a)所示的时序图 $G$ 划分为2个片段, 其中仅标记每条边的时间戳, 而省略标签以简化讨论。假设 $(t_{\min}, t_{\max}) = (0, 1)$ 对于给定集合 $\Sigma$  of TACOs和从 $\Sigma$ 推导出的候选时间间隔 $[t_0, t_k]$ , 输入时间窗口 $\tau_0$ 为 $[1, 4]$ , 我们可以看到如下。

(1) BTPart生成的时间分区如图-所示

保证5 (b)。这里候选区间分为 $[1, 2]$ 和 $[2, 4]$ ;

因此片段 $F_1$ 和 $F_2$ 具有时间戳在之内的边分别为 $[1, 2+1]$ 和 $[2, 4+1]$ 。注意边 $(d, f)$ 与

时间戳2在两个片段上都被复制。这确保了所有 $\Sigma$  can中的所有模式都在 $F_1$ 或 $F_2$ 上进行本地匹配。

(2)虽然图5(b)是一个最优的时间分区, 但它是倾斜的, 即 $F_1$ 中时间戳为1的边占了 $G$ 的很大一部分通过函数将边 $(f, e)$ 、 $(e, d)$ 和 $(f, d)$ 移动到片段 $F_2$ 重新平衡, 分区变得平衡(图5(c))。

2

ParEP的并行可扩展性保证如下, 其中顺序SeqEP在最坏情况下花费 $O(|\Sigma| |G| |\Sigma|)$ 时间, ParEP的成本以 $O(|\Sigma| |G| |\Sigma| k)$ 为界。

定理5:相对于SeqEP, ParEP具有并行可扩展性。2

的话。(1)通过允许用户设置范围 $\tau_0$ 并在 $\Sigma$ 中应用相关TACOs, ParEP可以很容易地适应预测特定事件 $p_0$ 是否会发生以及何时发生。(2)利用ParEP计算TACOs(即CCD中的规则批评家)的支持度和置信度。为此, 将输入时间窗口 $\tau_0$ 定义为覆盖 $G$ 中所有时间戳的最小范围。

## 7 实验研究

利用现实生活和合成图, 我们实验评估了(1)创造者-评论家规则发现方法的效率和(2)质量, (3)TASTE用于事件预测和动态推荐的准确性, 以及(4)ParEP的(并行)可扩展性。

实验设置。我们从实验环境开始。数据集。我们使用了六个现实生活中的时序图基准数据集, 如表2所示, 并分为三个不同的类别:

(1)event-

表2:数据集

Dataset	V	E	Relation type	Timestamp interval
ICEWS18	23K	469K	256	1 day
GDELT	8K	2.2M	240	15 minutes
YAGO	11K	201K	10	1 year
WIKI	13K	670K	24	1 year
MovieLens	80K	10M	10	1 day
Amazon	12.2M	30.3M	5	1 day

基于时间知识图:来自综合危机预警系统[8]的ICEWS18[36]和来自全球事件、语言和语气数据库[46]的GDELT [36];(2)具有时间关联事实的知识图谱:YAGO[57]和WIKI[45], 例如, YAGO记录了足球运动员为俱乐部效力的时间;以及(3)动态推荐数据集:电影评分的MovieLens[34]和产品评分的Amazon[61], 其中原始时间戳被重新组织, 使得两个相邻时间戳之间的时间粒度为一天。

如[36,50]所设计的, 每个数据集被分为训练集、验证集和测试集, 按时间戳的比例分别为80%、10%和10%。训练集和验证集用于模型训练和规则发现, 而测试集用于精度测试。每个数据集都包括事件(时间边缘)预测结果的真实值[36,50], 测试集实际上构成了预测时间事件的“查询”w.r.t.集。

我们还设计了一个图生成器来创建更大的合成数据集, 用于评估可扩展性。合成图在10000个时间戳范围内有多达10M个节点和1B条边, 标签、属性和值来自200个符号。

算法。规则发现模块中的creator和critic分别用Pytorch和c++实现, 而时间事件预测模块用c++实现。我们将TASTE与五种事件预测的基线方法进行了比较:(1)AGER, 它应用ger[5]捕捉时间图中的局部变化进行事件预测;(2) SACN[86], 一种基于卷积的知识图补全嵌入方法;(3)基于图卷积网络的知识图推理方法REGCN [50];(4)基于卷积神经网络的序列推荐算法Caser [91];(5) SASRec[37], 一个基于变压器的顺序推荐系统。我们采用默认配置的SACN [85]、REGCN[49]、Caser[90]和SASRec[38]开源代码, 用c++语言实现AGER。我们还实现了两种基于分层搜索的规则挖掘方法, 用于在c++中发现ger和TACOs, 分别表示为GERMine和TACOMine, 作为发现基线。

通过PyTorch对基线SACN、REGCN、Caser和SASRec进行多线程并行化。对于AGER、GERMine和TACOMine, 我们采用与ParEP相同的并行化方法(第6节)来计算相应的匹配, 这决定了它们的成本。为了进行公平的比较, 并行化中的线程数等于TASTE使用的核数k。

毫升模型。TACOs中的ML谓词分别采用REGCN和SASRec进行时态图补全和动态推荐。对于CCD creator中的模型, 我们使用了作者提供的TagGen的代码, 默认配置-

表3:ICEWS18上创作者-评论家发现的质量

$I \backslash N$	50	100	150	200	250	300
15	20.00%	23.81%	28.57%	39.05%	48.57%	65.71%
20	23.81%	25.71%	40.95%	55.24%	66.67%	73.33%
25	30.47%	41.90%	44.76%	71.43%	84.76%	96.19%
30	39.05%	44.28%	48.94%	76.15%	87.23%	97.14%

urations[111], 并将LSTM模型实现为[63], 具有默认训练配置和两个650宽的层。

我们在多达72个Intel Xeon 3.1 GHz处理核的集群上进行了实验, 集群上的两台机器通过10Gbps链路连接, 内存为256GB。默认情况下, 我们为发现需求设置 $\alpha=5$ ,  $\beta=100$ ,  $\gamma=0.8$ 和 $\delta=20$ ; 对于发现模块CCD, 迭代次数 $I=25$ , 样本大小 $N=250$ ; 预测方法ParEP的核数 $k=32$ , 除非另有说明。所有实验重复5次。这里报告平均值。

实验结果。我们接下来报告我们的发现。在每个实验中, 都显示了三个类别中每个类别中至少一个图表的结果。我们将其他结果推迟到完整版本。

Exp-1:效率。我们首先比较了CCD、GERMine和TACOMine的规则发现效率。由于分层方法发现具有大模式的规则的成本非常高, 为了在可承受的运行时间内比较效率, 我们设定了发现100条高质量规则的目标作为基准。也就是说, 当发现100条规则时, 每个发现过程就会终止。

(1)变化 $\alpha$ 。为了研究模式节点数对ICEWS18、WIKI和MovieLens发现方法的影响, 我们将 $\alpha$ 值从3变化到11。从图6(a)到图6(c), 当 $\alpha \leq 5$ 时, CCD的平均速度分别是GERMine和TACOMine的9.1倍和14.3倍。随着 $\alpha$ 的增大, 基于水平搜索的方法的计算时间呈指数增长, 当 $\alpha > 5$ 时, GERMine和TACOMine都不能在1.2天内终止, 而CCD的成本增加幅度较小。这是因为更大数量的模式节点会为水平挖掘带来指数级的更大搜索空间, 但ML生成模型的额外成本很少。特别是, CCD发现1639年的TACOs具有超过20条边的图案。

(2)变化的 $\beta$ 。为了研究支持阈值对ICEWS18、YAGO和亚马逊的影响, 我们将 $\beta$ 值从50变化到150。如图6(d)至图6(f)所示, CCD的运行时变化不大, 因为无论 $\beta$ 的值如何, 创建者总是需要多次迭代才能生成高质量的规则。相比之下, GERMine和TACOMine在大多数情况下用更大的 $\beta$ 花费更少的时间, 因为支持度的上界减少了更多的候选, 减少了搜索空间。

(3)改变 $\gamma$ 。将置信度 $\gamma$ 从0.7变化到0.9, 我们分别在图6(g)到6(i)中报告GDELT、WIKI和亚马逊的结果。结果表明, 随着 $\gamma$ 的增加, 所有的发现算法需要更长的时间。然而, 基于生成的CCD对 $\gamma$ 不太敏感。这是因为水平方法必须以指数级规模扩展其搜索空间, 以获得所需数量的具有更高置信度的高质量规则。

(4)变化 $\delta$ 。我们把 $\delta$ 从10变到30。这里 $\delta$ 统计所需时间窗口内离散时间戳的数量。如ICEWS18、YAGO和MovieLens上的图6(j)至6(l)所示:

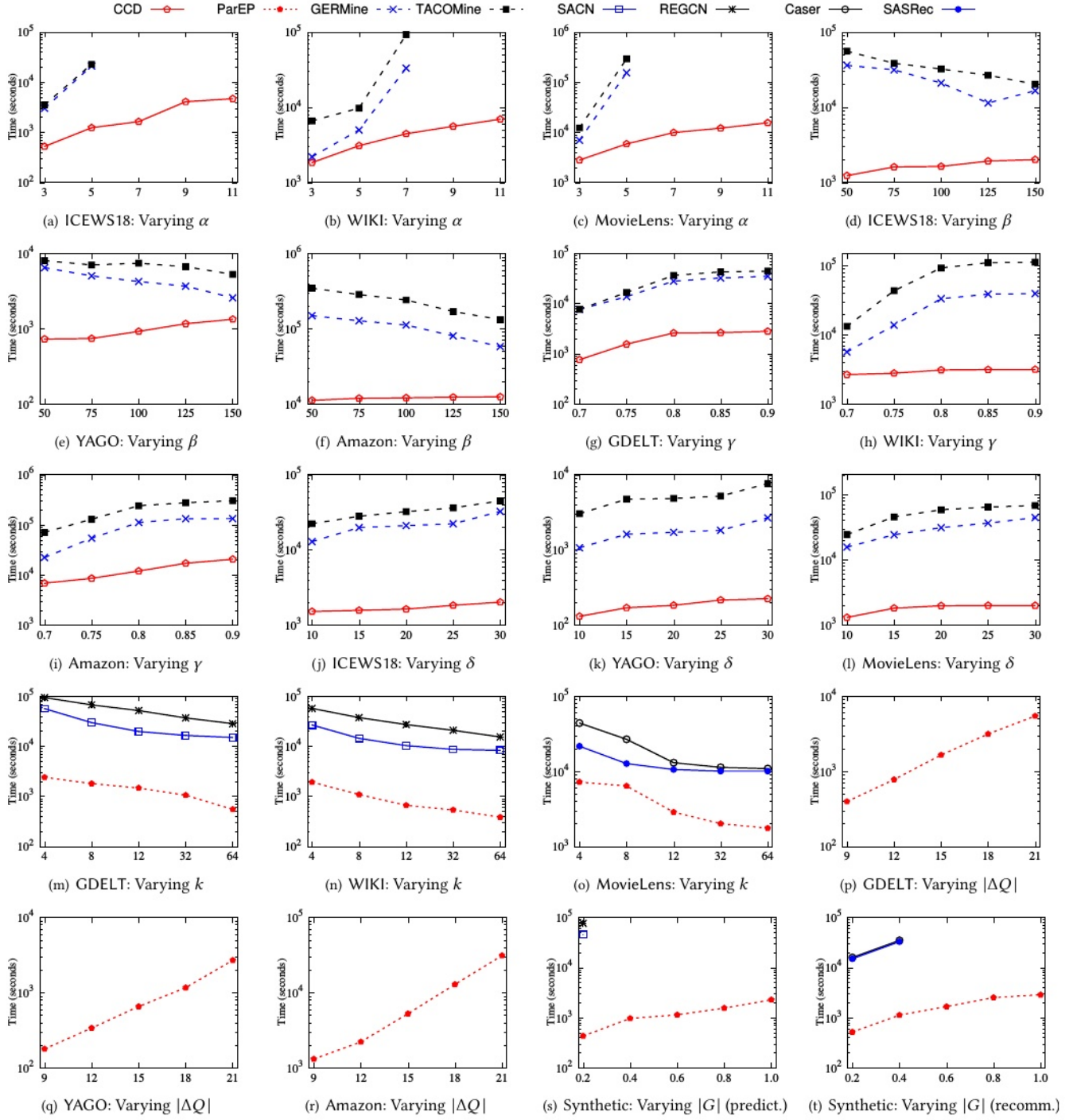


图6:性能评估

这三种算法的运行时间都随着 $\delta$ 的增长而增加，因为较长的时间窗口边界会在创建者为TagGen提供更多的生成工作量，并扩展GERMine和TACOMine的搜索空间。

(5)  $N$ 和 $I$ 的影响。我们还测试了样本量 $N$ 和迭代次数 $I$ 对CCD效率的影响。随着 $N$ 或 $I$ 的增加，它显示出适度的运行时增长(未显示)。

这是意料之中的，因为更大的 $N$  (resp.  $I$ )给生成器带来更多的工作。整个计算的轮数更多。

Exp-2:发现的质量。回想一下，基于生成的CCD获得的TACOs是分层搜索算法返回的taco的一个子集。因此我们检查了com-中有多少条规则

表4:事件预测/推荐准确率

Dataset	AGER	SACN	REGCN	Caser	SASRec	TASTE
ICEWS18	59.32%	63.28%	68.58%	-	-	75.24%
GDELT	54.60%	62.32%	66.31%	-	-	73.41%
YAGO	61.32%	64.03%	74.73%	-	-	84.14%
WIKI	58.56%	63.58%	71.32%	-	-	82.53%
MovieLens	65.80%	-	-	75.83%	80.45%	87.30%
Amazon	59.30%	-	-	63.50%	73.10%	73.10%

在所有数据集中, CCD都能发现层次法TACOMine发现的完备集, 即完整TACOs的覆盖范围。如表3所示, 随着样本数量N的增大或训练迭代次数I的增多, 覆盖率会越来越高。这是因为增大N或I可以增加近似目标分布的可能性, 从而生成高质量的规则, 如第5节所述。我们还可以看到, 较小的N和I足以获得大部分高质量的TACOs, 例如, 当N = 250, I = 25时, 覆盖率达到84.76%, 而TACOMine的时间同时减少了18.5倍(见Exp-1)。

除了采用GAN模型外, 我们还在CCD中测试了使用经典图形生成模型的性能, 即Erdős-Rényi (ER)[72]和Barabási-Albert (BA)[1]模型。我们发现, 当N = 300和I = 30时, ER和BA的TACOs覆盖率分别仅为33.07%和50.89%, 远低于GAN的覆盖率。这是因为这些经典的模型不能从图数据中迭代学习, 并以自适应的方式生成高质量的模式。

我们还手动检查了发现的TACOs。除了典型的只包括逻辑和时间谓词之外, 一些TACOs还可以帮助丰富或解释ML预测。

示例3:准确性。如表4所示, 我们使用TACOs和基线在时间事件预测和动态推荐两项任务上评估TASTE的准确性[37,50]。请注意, SACN和REGCN没有结果。Caser和SASRec)关于动态推荐的研究。时态事件预测)的数据集, 因为它们不是为任务而设计的。TASTE (ParEP)应用发现的置信度大于0.9的TACOs, 其中图模式最多有9个节点。我们发现, 很少有超过9个模式节点的TACOs具有高置信度和高支持度, 这与频繁模式挖掘的发现相似[19]。我们采用Hit Rate@10, 即ground-truth项目在前10个项目中所占的次数[37,50], 来评估准确率。

对于ICEWS18、GDELT、YAGO和WIKI上的事件预测, TASTE分别比AGER、SACN和REGCN平均高出34.9%、24.5%和12.2%。对于MovieLens和亚马逊的动态推荐, TASTE比AGER、Caser和SASRec分别高出35.8%、22.5%和10.6%。这些表明, 通过结合规则和ML模型, TASTE在这两个任务上的准确性都超过了最先进的基于深度学习的REGCN和SASRec, 而REGCN和SASRec都不能同时工作。

Exp-4:可伸缩性。我们最后评估了(1)TASTE系统中预测模块ParEP通过改变处理器数量k的并行可扩展性, (2)模式大小对其效率的影响, 以及(3)ParEP在更大的合成图上的可扩展性。由于ger是TACOs的特殊情况, 因此ParEP也适用于ger, 未显示AGER的运行时。

(1)并行可扩展性。k从4到64变化, 图6(m)-6(o)分别报告了在与示例3相同的设置下, 在GDELT和WIKI上进行事件预测的结果, 以及在MovieLens上进行推荐的结果。如图所示, (a) ParEP是并行可扩展的。当k从4增加到32时, 在三个图上平均快3.2倍。这验证了ParEP在数据分区并行性下的有效性。(b)除精度较高外, ParEP在效率上优于SACN和REGCN。Caser和SASRec), 例如, 当k = 64时, 它的平均速度分别是SACN、REGCN、Caser和SASRec的24.2、45.4、6.2和5.7倍。

(2)模式大小的影响。改变TACOs中 $\Delta$ 模式的大小 $|\Delta Q|$ , 即以每个 $\Delta Q$ 中的模式节点数和边缘数的总和来衡量, 我们在图6(p)至6(r)中报告了不同方法的性能。结果表明, ParEP随 $|\Delta Q|$ 的增大而变慢, 与预期一致。尽管如此, 当处理相对较大的 $\Delta$ 模式时, 仍然是高效的。例如, 当 $|\Delta Q| = 15$ 时, GDELT需要1645秒, 优于SACN计算的16380秒(见图6(m))。

(3)可伸缩性。固定k = 32, 我们使用0.2到1.0的比例因子改变合成图G的大小 $|G| = |V| + |E|$ , 并评估所有方法, 其中ParEP应用了100个TACOs。如图6(s)和图6(t)所示, 预测和推荐的ParEP在所有情况下都优于基线。当 $|G| = 810M$ 时, 平均需要1403秒, 而其他都不能在1天内完成。

总结。我们发现了以下内容。(1)平均而言, CCD的生成式ML方法在效率上比水平算法高出31倍以上。它能够从时间图中发现1639年20条边的TACOs, 而水平方法无法在1.2天内完成。(2) CCD能够发现levelwise方法推导出的高达84.76%的完整规则, 使用250个样本和25次训练尝试。(3)通过将规则和ML模型相结合, 发现的TACOs在事件预测和动态推荐方面的准确率平均分别提高了23.8%和23.0%。(4)我们的算法ParEP具有良好的并行可扩展性和数据集可扩展性, 在使用32个处理器的810M节点和边的图上, 算法耗时不到143s。

## 8 结论

我们提出了一种新的事件预测方法, 从基础(规则和复杂性)到系统(由可扩展算法支持), 新新性在第1节中总结。我们已经通过实验验证了TASTE在事件预测方面的前景。

未来工作的一个主题是评估TASTE预测其他类型事件的能力, 例如金融危机。另一个主题是通过增量地发现规则和预测事件来响应时间图的更新, 从而使TASTE“实时”。

## 致谢

研究得到了ERC 652976、英国皇家学会沃尔夫森研究优秀奖WRM/R1/180014和国家自然科学基金委员会61902274的资助。徐得到了新加坡国家研究基金会战略能力研究中心资助计划的支持。本材料中表达的任何观点、发现、结论或建议都是作者的观点, 并不反映新加坡国家研究基金会的观点



## 参考文献

- [1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.
- [2] Boanerges Aleman-Meza, Christian Halaschek-Wiener, Ismailcem Budak Arpinar, and Amit P. Sheth. 2003. Context-Aware Semantic Association Ranking. In *SWDB*.
- [3] Konstantin Andreev and Harald Räcke. 2006. Balanced Graph Partitioning. *Theory Comput. Syst.* 39, 6 (2006), 929–939.
- [4] Marcelo Arenas, Pedro Bahamondes, and Julia Stoyanovich. 2021. Temporal Regular Path Queries: Syntax, Semantics, and Complexity. *CoRR abs/2107.01241* (2021).
- [5] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. 2009. Mining Graph Evolution Rules. In *ECML/PKDD*.
- [6] Yumin Bian and Xiang-Qun Xie. 2021. Generative chemistry: Drug discovery with deep learning generative models. *Journal of Molecular Modeling* (2021).
- [7] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. 2018. NetGAN: Generating Graphs via Random Walks. In *ICML*.
- [8] Elizabeth Boschee, Jennifer Lautenschlager, Sean O’ Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS coded event data. *Harvard Dataverse* 12 (2015).
- [9] George Casella and Roger Berger. 2001. *Statistical Inference*. Duxbury Resource Center.
- [10] Ines Chami, Sami Abu-El-Hajja, Bryan Perozzi, Christopher Ré, and Kevin Murphy. 2020. Machine learning on graphs: A model and comprehensive taxonomy. *CoRR abs/2005.03675* (2020).
- [11] Jinyin Chen, Xuanheng Xu, Yangyang Wu, and Haibin Zheng. 2018. GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Link Prediction. *CoRR abs/1812.04206* (2018).
- [12] Alvaro Cortés-Calabuig and Jan Paredaens. 2012. Semantics of Constraints in RDFS. In *AMW*.
- [13] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* 35, 1 (2018), 53–65.
- [14] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *ICLR*.
- [15] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. 2018. HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In *EMNLP*.
- [16] Walter H. Dempsey, Alexander Moreno, Christy K. Scott, Michael L. Dennis, David H. Gustafson, Susan A. Murphy, and James M. Rehg. 2017. iSurvive: An Interpretable, Event-time Prediction Model for mHealth. In *ICML*.
- [17] Devdatt P. Dubhashi and Alessandro Panconesi. 2009. Concentration of Measure for the Analysis of Randomized Algorithms. Cambridge University Press.
- [18] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. 2011. Temporal Link Prediction Using Matrix and Tensor Factorizations. *ACM Trans. Knowl. Discov. Data* 5, 2 (2011), 10:1–10:27.
- [19] Mohammed Elseidy, Ehab Abdelhamid, Spiros Skiadopoulos, and Panos Kalnis. 2014. GRAMI: Frequent Subgraph and Pattern Mining in a Single Large Graph. *PVLDB* 7, 7 (2014), 517–528.
- [20] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. 2020. Discovering Graph Functional Dependencies. *ACM Trans. Database Syst.* 45, 3 (2020), 15:1–15:42.
- [21] Wenfei Fan, Ruochun Jin, Muyang Liu, Ping Lu, Chao Tian, and Jingren Zhou. 2020. Capturing Associations in Graphs. *PVLDB* 13, 11 (2020), 1863–1876.
- [22] Wenfei Fan and Ping Lu. 2019. Dependencies for Graphs. *ACM Trans. Database Syst.* 44, 2 (2019), 5:1–5:40.
- [23] Sébastien Frémal and Fabian Lecron. 2017. Weighting strategies for a recommender system using item clustering based on genres. *Expert Syst. Appl.* 77 (2017), 105–113.
- [24] Kaiqun Fu, Taoran Ji, Liang Zhao, and Chang-Tien Lu. 2019. TITAN: A Spatiotemporal Feature Learning Framework for Traffic Incident Duration Prediction. In *SIGSPATIAL/GIS*.
- [25] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*.
- [26] Alberto García-Durán, Sebastian Dumancic, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.
- [27] Michael Garey and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- [28] Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic Embedding for Temporal Knowledge Graph Completion. In *AAAI*.
- [29] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowl. Based Syst.* 187 (2020).
- [30] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. 2018. DynGEM: Deep Embedding Method for Dynamic Graphs. *CoRR abs/1805.11273* (2018).
- [31] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. 2020. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *CoRR abs/2001.06937* (2020).
- [32] Myoungji Han, Hyunjoon Kim, Geonmo Gu, Kunsoo Park, and Wook-Shin Han. 2019. Efficient Subgraph Matching: Harmonizing Dynamic Programming, Adaptive Matching Order, and Failing Set Together. In *SIGMOD*.
- [33] Negar Hariri, Bamshad Mobasher, and Robin D. Burke. 2012. Context-aware music recommendation based on latent topic sequential patterns. In *RecSys*.
- [34] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.
- [35] Yuriy Hulovatyy, Huili Chen, and Tijana Milenkovic. 2015. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinform.* 31, 12 (2015), 171–180.
- [36] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. In *EMNLP*.
- [37] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*.
- [38] Wang-Cheng Kang and Julian J. McAuley. 2019. SASRec implementation. <https://github.com/kang205/SASRec>.
- [39] Yoed N Kenett, Effi Levi, David Anaki, and Miriam Faust. 2017. The semantic distance task: Quantifying semantic distance with semantic network path length. *Journal of Experimental Psychology: Learning, Memory, and Cognition* (2017).
- [40] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude W. Shavlik. 2011. Learning Markov Logic Networks via Functional Gradient Boosting. In *ICDM*.
- [41] Angelika Kimmig, Bart Demoen, Luc De Raedt, Vítor Santos Costa, and Ricardo Rocha. 2011. On the implementation of the probabilistic logic programming language ProbLog. *Theory Pract. Log. Program.* 11, 2-3 (2011), 235–262.
- [42] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *KDD*.
- [43] Clyde P. Kruskal, Larry Rudolph, and Marc Snir. 1990. A Complexity Theory of Efficient Parallel Algorithms. *Theor. Comput. Sci.* 71, 1 (1990), 95–132.
- [44] Ni Lao, Tom M. Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *EMNLP*.
- [45] Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving Validity Time in Knowledge Graph. In *WWW*.
- [46] Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*.
- [47] Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare R. Voss. 2020. Connecting the Dots: Event Graph Schema Induction with Path Language Modeling. In *EMNLP*.
- [48] Yang Li, Nan Du, and Samy Bengio. 2018. Time-Dependent Representation for Neural Event Sequence Prediction. In *ICLR*.
- [49] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhao Wang, and Xueqi Cheng. 2021. REGCN implementation. <https://github.com/Lee-zix/RE-GCN>.
- [50] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhao Wang, and Xueqi Cheng. 2021. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning. In *SIGIR*.
- [51] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. 2019. Efficient Graph Generation with Graph Recurrent Attention Networks. In *NeurIPS*.
- [52] Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-Hop Knowledge Graph Reasoning with Reward Shaping. In *EMNLP*.
- [53] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *EMNLP*.
- [54] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. 2018. Constrained Graph Variational Autoencoders for Molecule Design. In *NeurIPS*.
- [55] Yung-Yin Lo, Wanjiun Liao, Cheng-Shang Chang, and Ying-Chin Lee. 2018. Temporal Matrix Factorization for Tracking Concept Drift in Individual User Preferences. *IEEE Trans. Comput. Soc. Syst.* 5, 1 (2018), 156–168.
- [56] Yao Ma, Ziyi Guo, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2020. Streaming Graph Neural Networks. In *SIGIR*.
- [57] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*.
- [58] Franco Manessi, Alessandro Rozza, and Mario Manzo. 2020. Dynamic graph convolutional networks. *Pattern Recognit.* 97 (2020).
- [59] Paweł Matuszyk, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama. 2015. Forgetting methods for incremental matrix factorization in recommender systems. In *SAC*.
- [60] Julian J. McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW*.
- [61] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*.
- [62] Christian Meilicke, Melisachew Wudage Chekol, Manuel Fink, and Heiner Stuckenschmidt. 2020. Reinforced Anytime Bottom Up Rule Learning for Knowledge Graph Completion. *CoRR abs/2004.04412* (2020).
- [63] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and Optimizing LSTM Language Models. In *ICLR*.

- [64] Bryan David Minor and Diane J. Cook. 2017. Forecasting occurrences of activities. *Pervasive Mob. Comput.* 38 (2017), 77–91.
- [65] Mohammad Hossein Namaki, Yinghui Wu, Qi Song, Peng Lin, and Tingjian Ge. 2017. Discovering Graph Temporal Association Rules. In *CIKM*.
- [66] Sriraam Natarajan, Tushar Khot, Kristian Kersting, Bernd Gutmann, and Jude Shavlik. 2010. Boosting relational dependency networks. In *ILP*.
- [67] Lukás Neumann, Andrew Zisserman, and Andrea Vedaldi. 2019. Future Event Prediction: If and When. In *CVPR Workshops*.
- [68] Jennifer Neville and David D. Jensen. 2007. Relational Dependency Networks. *J. Mach. Learn. Res.* 8 (2007), 653–692.
- [69] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunye Koh, and Sungchul Kim. 2018. Continuous-Time Dynamic Network Embeddings. In *WWW*.
- [70] Christos H. Papadimitriou. 1994. Computational complexity. Addison-Wesley.
- [71] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *AAAI*.
- [72] Erdős Paul and Rényi Alfréd. 1959. On random graphs I. *Publicationes Mathematicae (Debrecen)* 6 (1959), 290–297.
- [73] Zhi Qiao, Shiwan Zhao, Cao Xiao, Xiang Li, Yong Qin, and Fei Wang. 2018. Pairwise-Ranking based Collaborative Recurrent Neural Networks for Clinical Event Prediction. In *IJCAI*.
- [74] Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In *ICLR*.
- [75] Idris Rabi, Naomie Salim, Amin Da’u, and Akram Osman. 2020. Recommender System Based on Temporal Models: A Systematic Review. *Applied Sciences* 10, 7 (2020), 2204.
- [76] Dimitrios Rafailidis and Alexandros Nanopoulos. 2016. Modeling Users Preference Dynamics and Side Information in Recommender Systems. *IEEE Trans. Syst. Man Cybern. Syst.* 46, 6 (2016), 782–792.
- [77] Fatemeh Rahimian, Amir H. Payberah, Sarunas Girdzijauskas, and Seif Haridi. 2014. Distributed Vertex-Cut Partitioning. In *DAIS*.
- [78] Mahmudur Rahman and Mohammad Al Hasan. 2016. Link Prediction in Dynamic Networks Using Graphlet. In *ECML/PKDD*.
- [79] Matthew Richardson and Pedro M. Domingos. 2006. Markov logic networks. *Mach. Learn.* 62, 1-2 (2006), 107–136.
- [80] Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end Differentiable Proving. In *NIPS*.
- [81] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *CoRR abs/2006.10637* (2020).
- [82] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In *NeurIPS*.
- [83] Erik Scharwächter, Emmanuel Müller, Jonathan Donges, Marwan Hassani, and Thomas Seidl. 2016. Detecting change processes in dynamic networks by frequent graph evolution rule mining. In *ICDM*.
- [84] C. Seshadhri, Tamara G. Kolda, and Ali Pinar. 2011. Community structure and scale-free collections of Erdős-Rényi graphs. *CoRR abs/1112.3644* (2011).
- [85] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. ConvTransE implementation. <https://github.com/JD-AI-Research-Silicon-Valley/SACN>.
- [86] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*.
- [87] Yelong Shen, Jianshu Chen, Po-Sen Huang, Yuqing Guo, and Jianfeng Gao. 2018. M-Walk: Learning to Walk over Graphs using Monte Carlo Tree Search. In *NeurIPS*.
- [88] Tiago Sousa, João Correia, Vítor Pereira, and Miguel Rocha. 2021. Generative deep learning for targeted compound design. *Journal of Chemical Information and Modeling* 61, 11 (2021), 5343–5361.
- [89] BaoShan Sun and Lingyu Dong. 2017. Dynamic Model Adaptive to User Interest Drift Based on Cluster and Nearest Neighbors. *IEEE Access* 5 (2017), 1682–1691.
- [90] Jiaxi Tang and Ke Wang. 2018. Caser implementation. [https://github.com/graytowne/caser\\_pytorch](https://github.com/graytowne/caser_pytorch).
- [91] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*.
- [92] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs. In *ICML*.
- [93] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. DyRep: Learning Representations over Dynamic Graphs. In *ICLR*.
- [94] Amin Vahedian, Xun Zhou, Ling Tong, W. Nick Street, and Yanhua Li. 2019. Predicting Urban Dispersal Events: A Two-Stage Framework through Deep Survival Analysis on Mobility Data. In *AAAI*.
- [95] Ricardo Vilalta and Sheng Ma. 2002. Predicting Rare Events In Temporal Domains. In *ICDM*.
- [96] João Vinagre and Alípio Mário Jorge. 2012. Forgetting mechanisms for scalable collaborative filtering. *J. Braz. Comput. Soc.* 18, 4 (2012), 271–282.
- [97] Keqiang Wang, Yuanyuan Jin, Haofen Wang, Hongwei Peng, and Xiaoling Wang. 2018. Personalized Time-Aware Tag Recommendation. In *AAAI*.
- [98] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of ‘small-world’ networks. *Nature* 393, 6684 (1998), 440–442.
- [99] Jeremy C. Weiss and David Page. 2013. Forest-Based Point Process for Event Prediction from Electronic Health Records. In *ECML/PKDD*.
- [100] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *EMNLP*.
- [101] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Jens Lehmann, and Hamed Shariat Yazdi. 2019. Temporal Knowledge Graph Embedding Model based on Additive Time Series Decomposition. *CoRR abs/1911.07893* (2019).
- [102] Qiang Yang, Hui Wang, and Wei Zhang. 2002. Web-log Mining for Quantitative Temporal-Event Prediction. *IEEE Intell. Informatics Bull.* 1, 1 (2002), 10–18.
- [103] Yuan Yang and Le Song. 2020. Learn to Explain Efficiently via Neural Logic Inductive Learning. In *ICLR*.
- [104] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *ICML*.
- [105] Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Haifeng Chen, and Wei Wang. 2017. Link Prediction with Spatial and Temporal Consistency in Dynamic Networks. In *IJCAI*.
- [106] Giselle Zeno, Timothy La Fond, and Jennifer Neville. 2021. DYMond: Dynamic Motif-NoDes Network Generative Model. In *WWW*.
- [107] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38.
- [108] Wenbin Zhang, Liming Zhang, Dieter Pfoser, and Liang Zhao. 2021. Disentangled Dynamic Graph Deep Generation. In *SDM*.
- [109] Liang Zhao. 2021. Event Prediction in the Big Data Era: A Systematic Survey. *ACM Comput. Surv.* 54, 5 (2021), 94:1–94:37.
- [110] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. 2020. A Data-Driven Graph Generative Model for Temporal Interaction Networks. In *KDD*.
- [111] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. 2020. TagGen implementation. <https://github.com/davidchouzd/TagGen>.