

用逻辑条件丰富推荐模型

论文ID: 38

摘要

本文提出了RecLogic，这是一个用于提高机器学习(ML)推荐模型准确性的框架。它旨在增强现有的具有逻辑条件的ML模型，以减少假阳性和假阴性，而无需训练新模型。RecLogic的基础是(a)用TIEs表示的图上的一类预测规则，(b)学习TIEs的新方法，以及(c)使用TIEs进行推荐的新范例。TIEs可以嵌入ML推荐模型作为谓词；与先前的图规则相反，确定图是否满足一组TIEs是很容易的。为了丰富ML模型，RecLogic使用每轮反馈迭代训练生成器，以概率界学习TIEs。RecLogic还提供了一个PTIME并行算法，用于使用学习到的TIEs进行推荐。使用实际数据，我们经验验证了RecLogic在预测强度既不够大也不够小的区域平均提高了22.89%的ML预测精度，最高可达33.10%。

1 介绍

推荐模型在电商中已经发现普遍使用，向用户推荐物品[43]。的确，“Netflix上80%的电影观看来自推荐，60%的视频点击来自YouTube上的主页推荐”(cf.[104])。流行的推荐机器学习(ML)模型包括协同过滤(CF)和基于内容的(CB)模型。CF通过学习用户与物品之间的历史交互，例如用户以前的评分和浏览历史来识别用户偏好并进行推荐[6,42,52,53,61,69,88,97,101]。CB推荐主要基于对用户和项目内容的比较，如用户简介和项目特征[66,92,98]。

然而，单一的策略，无论是CF还是CB，在实践中往往是不够的。例如，当交互数据稀疏或推荐系统启动时，CF就不能很好地工作。为了纠正这些局限性，探索了混合方法来统一交互级相似度和内容级相似度[4,11,14,17,89,91,106,108]。然而，混合方法需要从头开始训练一个新的ML模型。此外，ML模型通常计算每个用户 x 和项目 y 之间的预测分数，即推荐强度，如果分数超过某些预定阈值，则将 y 推荐给 x 。因此，在预测得分接近阈值的“模糊区域”，即预测强度既不够大也不够小的区域，这些ML模型往往是“不确定的”。

示例1:考虑以下取自流行电商平台日常实践的现实例子。

(1)如果用户为女儿购买了K-12学生设计的玩具，那么根据购买历史，一些CF模型，例如[42]，会向她推荐相同品牌的K-12学生的新玩具，尽管她的孩子现在正在上大学(假阳性)。

(2)如果用户 x 已经结婚并为男性购买了香水，那么即使用户 x 有女性购物偏好，我们也可以向 x

推荐电动剃须刀。相比之下，CB模型[51]可能会做出假阴性预测，因为剃刀和女性偏好与强度阈值的内容相关性不强。

(3)虽然用户 x 偶尔会购买现场剧场的门票，但如果 x 预定了与演出地点不同的城市的航班，或者该演出与用户的购买历史不符，我们无法向 x 推荐即将到来的演出的门票 y 。混合模型[85]忽略了这样的情景感知特征，可能会以高概率做出假阳性推荐。

(4)如果用户 x 搬到了沿海城市，并且有一个朋友喜欢某个特定品牌的海滩配饰，那么即使 x 没有按照推荐强度阈值购买足够多的相似物品，我们也应该向 x 推荐该品牌的产品。[85]的混合模型没有考虑用户迁入和社交链接的特征，产生假阴性。2

这个例子提出了几个问题。是否有可能在不训练新的ML模型的情况下，简单地增强现有的CF、CB或混合模型，从而通过合并上下文感知特征和内容比较来减少假阳性(FPs)和假阴性(fn)?特别是，给定一个现有的推荐模型，我们能否在模型具有不确定预测强度的模糊区域改进它?我们能否通过突出重要特征来解释预测?我们能否在不使推荐问题变得更昂贵(例如难以处理)的情况下扩展模型?

RecLogic。实际的考虑促使我们提出RecLogic，一个推荐的逻辑框架。RecLogic的特点是一类预测规则，基于规则的新推荐范例，以及学习和应用规则的新方法。

(1) ML模型的逻辑规则(第2节)。我们引入了一类预测规则，用TIEs(预测规则)表示。TIE是根据用户和项目的两颗星的图形模式定义的。此外，TIEs可以将现有的ML模型作为谓词嵌入，无论模型是CF、CB还是混合模型。TIEs通过在上下文、内容和社会联系上加入额外的逻辑条件来丰富这些模型，以减少它们的FPs和FNs。这些编码了重要的特征，这些特征在决定是否推荐时具有很强的辨别能力(更多信息请参见第2节)。

我们证明了在多项式时间(PTime)内判断一个图是否满足给定的一组TIEs。相反，对于图函数依赖关系(GFDs)[26]和图关联规则(gar)[23]来说，问题是conp完备的，而对于时态关联规则(TACOs)[24]来说，问题是np完备的。我们之所以研究这个问题，是因为它表明了用TIEs提出建议的复杂性。

(2)一个新的范例(第3节)。RecLogic不是训练一个新的机器学习模型，而是通过应用TIEs来预测物品的潜在用户，并提高了现有机器学习模型的准确性，特别是在模糊区域 \mathcal{H} 中，机器学习推荐的强度既不够大也不够小。我们表明，TIEs通过过滤带有额外条件的错误推荐来减少(a)假阳性，以及(b)通过合并CB特征来减少假阴性

来推翻错误的CF决定。此外，TIEs的逻辑条件能够解释为什么向用户推荐一个项目。

(3)一种新的发现方法(第4节)。与传统的分层方法相反，RecLogic的发现模块提倡一种新的基于代的TIEs学习方法。它迭代地训练一个生成器，用从真阳性和真阴性中提取的特征创建TIEs，以丰富现有的ML模型。我们证明了给定一组需求(支持度和置信度)，RecLogic可以在概率范围内找到所有高质量的TIEs。

(4)并行推荐(Section 5)。RecLogic的推荐模块应用学习到的TIEs，预测用户在PTIME中可能喜欢的物品。它也是可并行扩展的[54]，即当使用更多处理器时，它保证减少运行时间。

(5)实验研究(第6部分):使用真实数据和合成数据，我们实验验证了RecLogic的准确性和可扩展性。(a) RecLogic有效降低了基于ml的方法的FPs和FNs。在现实生活的图上，在模糊区域 H 中，平均而言，它比四个流行的ML模型提高了22.89%(最高33.10%)的准确率，在整个区域上的改进为2.93%，低于预期的 H 。其Recall@K在top-K推荐的全领域提升12.16%。(b)基于代的TIE学习比分层搜索快48.31倍。学习10个模式节点的TIEs需要2924.88s，而level-wise不能在24小时内完成。(c) RecLogic在提出建议时，随逻辑核数n的变化而扩展良好:当n从4到20变化时，速度提高3.28倍。

独特的特性。RecLogic具有以下特性。

- 轻量化:它以非侵入的方式丰富了任意的ML模型，不需要调整其机制或参数。
- 准确:它减少了现有ML模型的假阳性和假阴性，用于CF, CB甚至混合模型。

- 可处理:它的特点是TIEs，一类图规则的验证是在PTIME，以决定一个图是否满足规则。可靠:它探索了一种新的基于代的方法来学习在概率范围内具有合理大模式的TIEs。
- 解释性特征:它突出了做出预测所基于的逻辑条件，而不是黑盒ML模型。

相关的工作。我们将相关工作分类如下。

推荐模型。各种各样的模型被开发出来。

(1)基于cf的模型利用用户-物品交互矩阵进行推荐。假设相似的用户对物品有相似的偏好，相似性是从交互矩阵中推断出来的。(a)基于记忆的CF使用预定义的函数测量相似性，如余弦相似性[47,61]、欧几里得距离[88]、Pearson相关性[8,95]和Jaccard指数[56]。(b)基于模型的CF通过使用SVD++[52]、NMF[69]和FISM[49]等方法，从通过交互矩阵训练的模型中推断评级。矩阵分解(MF)[53]将每个用户和物品映射到一个潜在向量空间中，并将评分预测为内积。

深度学习已被用于增强CF，以(a)通过对数据中的非线性建模来捕获用户-项目交互之间复杂和错综复杂的模式，以及(b)有效地学习用户和项目的有用表示和解释因素。这样的模型包括多层感知器(MLP)[15,19]，神经矩阵

分解[42]，自编码器[84,86,87]，CNN[41,90]，受限玻尔兹曼机(RBM)[32,64,81]，注意力网络[12,48,63]。此外，图卷积网络(GCN)也被应用于开发用户和项目之间的高阶连通性，并编码交互图之间的协作信号，如NGCF[97]、GC-MC[6]、PinSage[101]和LightGCN[40]。

(2)基于内容的系统使用辅助特征(如图像、文本、音频)来表示物品，并根据用户过去访问过的物品的特征对用户进行profile。然后对用户配置文件和项目表示进行比较和匹配，以进行推荐。基于记忆的方法使用预定义的相似度函数进行匹配，例如欧氏距离[102]和皮尔逊相关距离[7]，而基于模型的方法从数据[5,51,67,73,83,92,98]中学习内容相似度函数。

(3)混合模型结合不同的策略来提高推荐质量。CF的冷启动和数据稀疏性问题可以通过纳入项目/用户侧信息等部分解决[89]。物品侧信息包括物品评论[100,106]，属性(类别、品牌、标签)[55,85,105,108]和多媒体数据(图像[17]、视频[12]和音频[59]的特征)，以及文本描述[37]。用户侧信息包括人口统计学属性(如性别、年龄、兴趣和职业)[30,34,58,107]，以及社交链接[46,71]。此外，知识图[35,96,103]、时间信息[4,14,65,72,80,91]和空间信息[11,18,50,57,60]也被用来改进CF方法。

RecLogic与这些系统的不同之处如下。(1) RecLogic的目标不是开发另一个特定的推荐模型，而是提供一个通用的、灵活的框架，在这个框架中，任何当前的推荐模型都可以插入和改进。与混合策略相反，它不需要用户从头开始训练一个新模型。(2) RecLogic强调了做出预测决策的逻辑条件，这是当前基于ml的模型所不提供的。(3) RecLogic统一了图模式、ML和逻辑条件，同时保留了建议与规则的可追溯性(验证问题)。

关联规则。研究关联规则是为了识别图中实体之间的规律性。gpar[25]用模式定义关联规则。gar[23]使用前提条件和ML谓词扩展gpar。TACOs是用时间模式定义的，用于时间图中的事件预测[24]。gcr用星形模式定义，用于实体解析和冲突解析[20]。

相比之下，(1)TIEs是用星形模式和受限谓词定义的。与gcr一样，它的验证问题也存在于PTIME中，但与gpar、gar和TACOs的难处不同。(2)这项工作显示了(a)什么逻辑条件可以减少现有ML模型的FPs和FNs，以及(b)如何通过提出一个新的框架及其并行算法来改进ML模型

规则的发现。人们提出了各种算法来从图中发现规则[13,16,20-22,24,28,29,76,77,79]，分类如下:(1)通过迭代地向一组种子规则[20,22,28,29,77]添加约束来枚举候选规则的分层算法，特别是AMIE+ [28];(2)通过GAN和强化学习等直接生成候选规则的ML方法[13,16,24,76,79];以及(3)基于采样的方法[21]，它首先从一个大图中采样小图，然后从样本中挖掘规则。

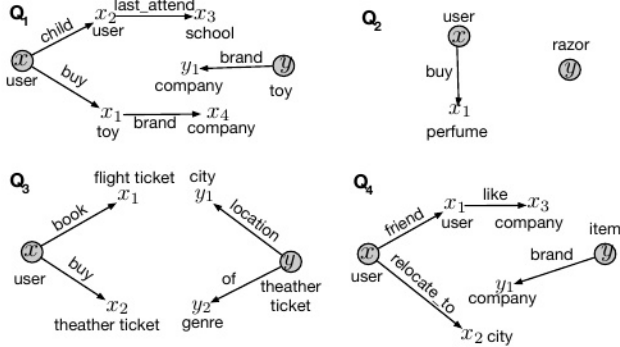


图1:示例双星模式Q1-Q 4

与[20,22,28,29]相比，我们发现了TIEs来提高推荐的准确性。为此，我们从知识图谱和社交网络等附加特征扩展图，并从过滤后的模糊区域中挖掘规则。此外，我们利用基于生成的方法来学习TIEs，这与现有的图规则挖掘算法不同[21,24,25]。

2 预测规则

在本节中，我们将介绍TIEs，即RecLogic的预测规则。我们首先定义TIEs(第2.1节)。然后，我们用TIEs解决了提出建议的复杂性(第2.2节)。

2.1 语法和语义

假设三个可数无限的符号集合，分别用 Γ 、 \boxtimes 和 U 表示标签、属性和常量。

初步的。考虑有向标记图 $G = (V, E, L, FA)$ ，其中(a) V 是节点的有限集合;(b) $E \subseteq V \times \Gamma \times V$ 是边的集合，其中 $E = (V, \tau, V)$ 表示一条从节点 V 到 V 的边，用 $\tau \in \Gamma$ 标记;(c)每个 $v \in V$ 的节点都有来自 Γ 的标签 $L(v)$;(d)每个节点 $v \in V$ 携带一个元组 $FA(v) = (A_1 = A_1, \dots)$ ，一个有限数的属性的一个 $= A_n$ ，其中 $A_i \in \boxtimes$ and $A_i \in U$ ，写成 $v.A_i = A_i, A_i, A_j$ if i, j ，表示属性。

一条路径 ρ 是一个列表 (v_1, \dots, v_l) 使得 (v_i, τ_i, v_{i+1}) 是 E 中的一条边($i \in [1, l-1]$)。如果每个节点在 ρ 上最多出现一次，则很简单。我们在续集中考虑简单路径，简称为路径。

双星模式。我们使用星型图案来降低复杂度。

以节点 v 为中心的星 S_v 由一组路径 ρ_1, \dots, ρ_k 从 v 开始，使得对于所有 i, j ， ρ_i 和 ρ_j 除了 v 之外没有共同的节点。我们将 v 称为恒星 S_v 的枢轴，并且将一个没有外向邻居的节点作为 S_v 的叶节点。

双星模式定义为 $Q[x, y, x'] = (VQ, EQ, LQ, \mu)$ 。它由两颗不相交的恒星组成，其中(1) VQ (p。EQ是一组模式节点。边)的两颗星;(2) LQ 为每个节点 $u \in VQ$ 分配一个标签 $LQ(u) \in \Gamma$;(3) x' 是一个不同变量的列表， μ 是 x' 到 VQ 的双射映射，即它将每个变量映射到 VQ 中的不同节点 v ；以及(4) x 和 y 是 x' 中两个指定的变量，映射到两个枢轴上，分别称为 Q 的用户节点和项目节点。对于变量 $z \in x$ ，我们可以互换地使用 $\mu(z)$ 和 z ，如果它从上下文中清楚的话。

直观地说，用户 x 的星型模式旨在通过路径从 x 链接收集其选择的属性;物品 y 的星型也是如此。

模式匹配。图 G 中对偶星图 $Q[x, y, x']$ 的匹配是一个从 Q 到 G 的同态映射 h ，使得(A)对于每个模式节点 $u \in VQ$ ， $LQ(u) = L(h(u))$;和(b)为每个模式

Notations	Definitions
$G, Q[x, y, \bar{x}]$	directed graph, dual star pattern
φ	TIE $\varphi = Q[x, y, \bar{x}](X \rightarrow (x, \text{likes}, y))$
$\delta, \delta_H, \delta_L$	strength thresholds of ML recommendations
\mathbb{H}	fuzzy area where \mathcal{M} is uncertain
$\mathcal{M}(x, y) \otimes \delta$	an ML predicate
\mathcal{T}	a set of training data
$\text{supp}(\varphi, G, \mathcal{T}), \text{conf}(\varphi, G, \mathcal{T})$	support and confidence of φ in G , resp.

Table 1: Notations

边 $e = (u, \tau, u')$ in $Q, e' = (h(u), \tau, h(u'))$ 是 G 中的一条边。

示例2:如图1所示，双星模式Q1-Q 4收集用户和要检查的物品的属性以进行推荐(参见示例1)。模式的轴点被标记为灰色。注意，在图案Q2中， y 既是枢轴，也是叶子。2

预测规则。我们首先定义两种形式的谓词。

谓词。一个模式 $Q[x, y, x']$ 的逻辑谓词具有这样的形式:

$$z_1.A \oplus z_2.B \mid z.A \oplus c,$$

其中 \oplus 是其中之一， $, , <, \leq, >, \geq$;变量 $z, z_1, z_2 \in x; c \in U$ 为常数;而 A, B 是 \boxtimes 中的属性。我们参考 z_1 。 $\oplus z_2.B$ 为变量谓词，而 $z.A \oplus c$ 为常量谓词。

模式 $Q[x, y, x']$ 的一个ML谓词的形式是:

$$\mathcal{M}(x, y) \otimes \delta,$$

其中 \otimes 为 \geq 或 $<$; $\mathcal{M}(x, y)$ 是一个现有的ML推荐模型，它返回一个在 $[0, 1]$ 范围内的数值，表示用户 x 喜欢物品 y 的程度;而 δ 是 $[0, 1]$ 范围内的阈值。

这里ML模型 $\mathcal{M}(x, y)$ 可以是(a) CF，例如NMF [69]，NCF[42]和LightGCN [40];(b) CB，例如LightGBM[51]和LSTUR [5];或(c)混合，如DeepFM[34]和xDeepFM[58]。必要时，我们将 $\mathcal{M}(x, y)$ 预测的分值进行归一化，以拟合到 $[0, 1]$ 范围内。

TIEs. A TIE (predicTion rule) φ has the following form:

$$Q[x, y, \bar{x}](X \rightarrow (x, \text{likes}, y)),$$

其中 $Q[x, y, x']$ 是一个双星图案， x 是 $Q[x, y, x']$ 的谓词的连词，最多包含一个ML模型。此外，(1)变量谓词 z_1 。 $\oplus z_2.B$ 只分别定义在 $Q[x, y, x']$ 中两颗星的叶子或支点 z_1 和 z_2 上，并且(2)每个叶子或支点最多可以携带一个变量谓词。

也就是说，变量谓词只用于“连接”两个不相交的星号，并比较相关联的叶子或枢轴的属性。稍后将在定理1和定理2中看到，这样的限制是为了在复杂性和表现力之间取得平衡。

注意，前置条件 X 可能包含多个属于同一个ML模型的ML谓词，并且可以表达形式为 $\mathcal{M}(X, y) \geq \delta \wedge \mathcal{M}(X, y) < \delta_h$ 的范围约束(见第3节)。

直观地说，我们分离模式 Q 和条件 X ，以便(a)通过 Q 可视化用户和项目的拓扑特征，(b)在 X 中嵌入ML模型，(c)通过利用模式匹配的局部性和双星型模式 Q 来加速评估(第5节)。

示例3:继续示例1和2，我们展示了减少ML模型的假阳性(FPs)和假阴性(FNs)的TIE。

(1) $\phi_1 = Q_1[x, y, x'](\mathcal{M}^{-1}(x, y) \geq 0.6 \wedge X_1 \rightarrow (x, \text{likes}, y))$ ，其中 Q_1 如图1所示， M_1 为[42]的CF模型， X_1 为 X_1 。目标是 $K-12$ 学生 $\wedge y$ 。目标是 $K-12$ 学生 $\wedge x_3$ 。型，大学 $\wedge x_4.name = y_1.name$ 。这个TIE过滤 M_1 的FPs。直观地说，虽然 M_1 预测 x 是物品 y 的目标用户(即 $M_1(x, y) \geq 0.6$)，但如果 x 的孩子目前正在上大学，那么条件 X_1 不满足， $andy$ 不推荐给 x ，换句话说，只有当 X_1 成立时才进行推荐。

(2) $\phi_2 = Q_2(x, y, x) (M^{-2}(x, y) < 0.6 \wedge X_2 \rightarrow (x, y))$, x_1, X_2 。类别=men \wedge x。地位=已婚 \wedge x。喜好=女性喜好。它是通过结合用户的婚姻状况和之前的交互来降低CB模型M2的FNs[51]。直观上, 虽然M2可能不会推荐tox(即 $M(x, y) < 0.6$, 因为x的女性购物偏好与y不匹配), 但如果满足条件X2, 则会进行推荐。

(3) $\phi_3 = Q_3[x, y, x] (M^{-3}(x, y) \geq 0.7 \wedge X_3 \rightarrow (x, \text{likes}, y))$, 其中 X_3 为 $x_1.\text{destination} = y_1.\text{name} \wedge x_2.$ 。体裁= $y_2.\text{name}$ 。它通过纳入关于航班和剧院的上下文特征来增强[85]的混合模型M3, 并在 X_3 中使用逻辑谓词过滤FPs。也就是说, 如果y不在x旅行的城市(同一季节), 或者y的类型与用户的购买历史不符, 那么M3的预测是FP, 我们不会向x推荐y。

(4) $\phi_4 = \text{第四季度}(x, y, x) (M^{-3}(x, y) < 0.7 \wedge X_4 \rightarrow (x, y))$, 在 X_4 $x_3.\text{name} = y_1.\text{name} \wedge x_2.$ 。Type = 沿海 \wedge y1。商业=海滩配件。它通过考虑M3忽略的位置和用户社会关系(模式Q4中指定)的变化, 降低了M3模型的FNs[85]。如果x已经搬到沿海城市, 并且他的朋友喜欢海滩配件, 那么它会覆盖M3的预测, 并将y推荐给x。

(5) 考虑另一种模式Q3 that用两条新边(x, fansOf, x_3)和($y, \text{主演}, y_3$)扩展Q3, 其中 x_3 和 y_3 都被标记为person。那么 $\phi_5 = Q_3[x, y, x] (x^{-2}.\text{price} = y.\text{price} \wedge x_1.\text{destination} = y_1.\text{name} \wedge x_3.\text{name} = y_3.\text{name}, \text{“教育中的剧场”} \rightarrow (x, \text{likes}, y))$ 就是上面 ϕ_3 的替代。它用逻辑条件替换了ML谓词M3, 即如果价格(resp.;主演)匹配x的购买历史(resp. 偏好), 如果不是教育界的剧场。 ϕ_5 中的逻辑条件提供了M3模型预测的一种解释[85]。2

的话。TIEs旨在捕捉重要特性以供推荐。如果一个特征已经被(经验)验证在决定是否进行推荐方面具有很强的辨别能力, 则被归类为重要特征。换句话说, 如果加入一个重要的功能, 更多的FPs和FNs可以不断降低。这包括(a)由双星模式收集的图拓扑特征, 例如基于cf的交互(Q1 - Q4的用户之星), 基于内容的信息(Q1 - Q4的用户之星, Q1、Q2和Q4的物品之星), 基于上下文的信息(Q3和Q4的用户之星), 社交网络连接(Q4的用户之星), 以及(b)属性特征, 例如示例3中所示的谓词。

语义。用 $h(x)^{-}$ 表示图G中双星模式Q的匹配, 用p表示 $Q[x, y, x]$ 的谓词。我们说 $h(x)^{-}$ 满足p, 用 $h(x)^{-} \models p$ 表示, 如果(a)当p是 z_1 时。 $\oplus z_2$ 。B, 属性A和B分别存在于节点 $h(z_1)$ 和 $h(z_2)$, 以及 $h(z_1)$ 。 $A \oplus h(z_2).B$;同理对于 $z.A \oplus c$;和(b)当p为 $M(x, y) < \delta$ 时, ML模型预测用户x如何喜欢物品y的强度低于阈值 δ ;类似地, 对于 $M(x, y) \geq \delta$ 。对于谓词的连词X, 我们写 $h(X)^{-} \models X$ 如果 $h(X)^{-} \models p$ 对于X中的所有谓词p来说。

图G满足TIE $\phi = Q[x, y, x] (x^{-} \rightarrow (x, \text{likes}, y))$, 记为 $G \models \phi$, 如果对于任意用户x和物品y, 只要在G中存在Q的匹配h使得 $h_1 = x$, 则也存在从x到y的标记为likes的链接(可能是隐藏的), 即物品y应该被推荐给用户x。设 Σ_{be} 为一组TIE。对于 Σ 中的每一个 ϕ , 我们写成 $G \models \Sigma$ if $G \models \phi$ 。

本文的表示法总结在表1中。

2.2 推荐的复杂性

接下来, 我们用TIEs来解决在图中识别用户-项目对的复杂性, 以便提出建议。我们展示了这个问题是可处理的。相反, 解除TIEs定义中的任何限制都会使问题变得棘手。

为了与之前的图关联规则的复杂性进行比较, 我们将验证问题表述如下。

◦ 输入: 一个图形G和一组 Σ of TIEs。◦

输出: $G \models \Sigma$?

直观地说, 对于每个TIE $\phi = Q[x, y, x] (x^{-} \rightarrow (x, \text{likes}, y))$, 验证是检查图G中模式Q的所有匹配h, 如果 $h(x)^{-} \models x$ 则将 $h(y)$ 项推荐给用户 $h(x)$ 。

与此相关的是推荐问题。即判定给定G中的用户u和物品v, 是否存在一个TIE $\phi \in \Sigma$ and与Q的匹配h使得 $h(x) = u, h(y) = v, h(x)^{-} \models x$, 即是否应该将v推荐给u。该问题与验证问题的补题具有相同的复杂度。

对GFDs[26]、GARs[23]和TACOs[24]等图规则的验证问题进行了研究, 结果表明该问题是难以解决的。相反, 我们表明这个问题对于TIEs是可处理的。在这里, 我们假设w.l.o.g.在TIE中使用预训练的ML模型M进行测试需要在图G中花费PTIME ($f(|G|)$), 这在实践中很常见。

Theorem 1: *The validation problem is in PTIME for TIEs.* ◻

在第5节中, 我们将为推荐问题提供一个 $O(|V| \Sigma |(\Sigma|E| + |V| f(|G|)))$ 时间算法。它的小修改适用于复杂度相同的验证分析。

其处理能力相当微妙。回想一下, 我们定义了一个具有双星模式的TIE, 其中每个叶子最多携带一个变量谓词。放松其中任何一个限制都会让问题变得棘手。即使我们允许树型模式, 也会发生这种情况, 即当用户(一个物品)被建模为树而不是星型时。

定理2: 如果两个TIEs中的一个允许叶节点携带多个变量谓词, 则验证问题变为comp完备;或者(2)TIEs用树形而不是星形来定义。2

的话。(1)通过对3SAT的补全进行约简, 验证了其难解性, 3SAT是NP-complete[31]。它是由模式的拓扑结构和变量谓词的组合引入的, 而不是由模式的大小引入的。实际上, 对于具有大型模式的TIEs, 如果我们适当地限制拓扑结构和谓词, 验证问题是可以处理的, 如定理1所示。

(2)定理1和定理2都是关于验证的综合复杂性。问题在于数据复杂性的PTIME。这是因为TIEs的大小根据数据复杂度是固定的[78];本例验证的PTIME算法如下:对于 Σ 中的每个TIE $Q[x, y, x] (x^{-} \rightarrow (x, \text{likes}, y))$, 枚举G中Q的所有匹配h, 并检查 $h \models x \rightarrow (x, \text{likes}, y)[22]$ 。

(3)我们发现在实践中, TIEs能够表达额外的逻辑条件来降低ML模型的FPs和fn。也就是说, TIEs在复杂性和表达能力之间取得了平衡。

3 一个推荐框架

在本节中, 我们将介绍一个推荐框架RecLogic, 并展示RecLogic如何丰富ML推荐模型。

概述。如图2所示，RecLogic由规则发现模块和推荐模块组成。人们可以插入任意现有的推荐模型M，RecLogic可以帮助减少M预测的假阳性(FPs)和假阴性(fn)。这里M可以用于预测CTR(点击率)或进行top-K推荐。

在预测CTR时，我们假设M设置了一个单一的强度阈值 δ ，当 $M(x,y) \geq \delta$ 时，它向用户x推荐项目。这个阈值可以通过不同的策略来确定，例如二分搜索，这样M的推荐精度就会最大化。相反，如果使用M进行top-K推荐，即将top-K排名为 $M(x,y)$ 的项目y推荐给用户x，则通过取第k个强度 $M(x,y)$ 的值来确定每个用户x的强度阈值 δ 。那么，如果 $M(x,y) \geq \delta$ ，M仍然向这个特定的用户x推荐y，而不同的用户与不同的阈值 δ 相关联。

发现和推荐模块的工作原理如下。

规则的发现。RecLogic的发现模块从现实生活中的图中学习一组 Σ of TIEs来丰富M。与之前的图规则挖掘算法[21,22,24]不同，RecLogic由模型M驱动来寻找降低其FPs和FNs的规则。提出了一种基于生成的方法来学习TIEs和对大图进行缩放。

RecLogic在图形上工作，扩展了知识图和社交网络等附加功能。

建议。学习完 Σ of TIEs集合后，RecLogic将TIEs应用到图G上进行推荐。考虑CTR预测。如果在 Σ and G 中存在一个TIE $\phi = Q[x, y, x](x \rightarrow (x, \text{likes}, y))$ ，并且 G 中 Q 的匹配 h 使得(a) $h(x)$ 为用户 u ， $h(y)$ 为物品 v ，且(b) $h \models x$ ，则RecLogic将 v 推荐给 u ，否则不进行推荐。当涉及到top-K推荐时，RecLogic将推荐强度 $M(u,v)$ 调整到最高值(resp. 如果上述条件(a)和(b)保持(resp. 不保持)。然后根据更新优势的先后顺序选择用户 u 的top-K推荐项目。请注意，如果将多个强度调整为相同的值，那么它们的相对顺序与模型 m 确定的顺序保持一致。我们开发了一个算法在PTIME中进行推荐，并将其并行化以在保证性能的情况下对大型图进行缩放(参见第5节)。

更具体地说，TIEs降低M的FNs和FPs如下。

(一)问 $[x, y, x](M(x, y) < \delta \wedge 1 \rightarrow (x, y))$ 与逻辑前提 $X1$ 。也就是说，虽然 $M(x,y)$ 预测用户x可能不喜欢物品y(低于阈值 δ)，但如果 $X1$ 成立，那么RecLogic要么覆盖M的预测并将y推荐给x，要么扩大推荐强度 $M(x,y)$ 。这样的TIEs被称为l模式；它们有助于降低FNs，例如例3的 $\phi2$ 和 $\phi4$ 。

可以在2.1节所述的规则中插入任何现有的ML模型 $M(x, y)$ ，CF, CB或混合。因此，当CF的交互数据稀疏时，可以在 $X1$ 中编码内容级相似度和CB模型来弥补，反之亦然。

(b) $Q[x, y, x](M(x, y) \geq \delta \wedge X1 \rightarrow (x, \text{likes}, y))$ 。即 $M(x,y)$ 建议向用户x推荐物品y(即 $M(x,y) \geq \delta$)，RecLogic检查附加条件 $X1$ 来过滤FPs，即如果 $X1$ 不满足，则不进行推荐，或者降低top-K推荐中的强度 $M(x,y)$ 。这个的TIEs

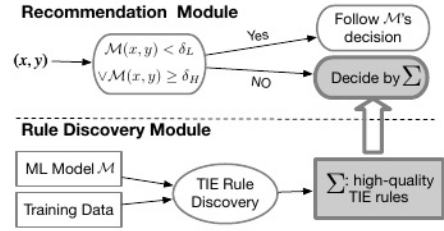


图2:推荐框架RecLogic

形式被称为h模，例如例3的 $\phi1$ 和 $\phi3$ 。

(c) $Q[x, y, x](x \rightarrow (x, \text{likes}, y))$ ，其中前提条件x不包含ML谓词，如例3的 $\phi5$ 。这种TIEs根据ML模型M所忽略的侧特征做出决策。

模糊区域。RecLogic在区域 H 中特别有效，其中M的建议可能是FPs或FNs，因为它们的强度对于推荐场景既不够大也不够小。以CTR预测为例。面积 H 由 δ_L 和 δ_H 参数定义， $\delta_L \leq \delta < \delta_H$ 。则 $H = \{(x, y) \mid \delta_L \leq M(x, y) < \delta_H\}$ 。设 H' 为 H 的补数，即如果 $M(x, y) < \delta_L$ 或 $M(x, y) \geq \delta_H$ ，则一个用户-物品对 $(x, y) \in H'$ 。另一方面，回想一下在top-K推荐中采用了不同的推荐阈值 δ 。因此在这种情况下，存在一个明显的模糊区域 H 和对应于每个用户x的补集 H' ，其中 H 和 H' 中的所有用户都固定为 x 。

对于用户-项目对 $(x, y) \in H$ ，RecLogic通过应用 Σ in l模式和h模式的TIEs来细化ML模型M的推荐，分别减少FNs和FPs。更精确地说，TIEs可以分别写成 $Q[x, y, x](M(x, y) < \delta \wedge M(x, y) \geq \delta_L \wedge 1 \rightarrow (x, \text{likes}, y))$ 和 $Q[x, y, x](M(x, y) \geq \delta \wedge M(x, y) < \delta_H \wedge X1 \rightarrow (x, \text{likes}, y))$ ，即M预测的分数在一定范围内。对于配对 $(x, y) \in H$ ，RecLogic简单地遵循M的决定，即如果 $M(x, y) \geq \delta_H$ ，则向用户x推荐y项，如果 $M(x, y) < \delta_L$ ，则不推荐。这些由 $Q[x, y, x](M(x, y) \geq \delta_H \rightarrow (x, \text{likes}, y))$ 形式的TIEs执行，其中Q仅由两个节点x和y组成，没有任何边。请注意，这里的阈值 δ 、 δ_L 和 δ_H 属于top-K推荐中的每个特定用户x，即具有不同 δ 的TIEs细化了针对不同用户的推荐。此外，更新推荐强度的重排序仅在H内执行。

RecLogic确定参数 δ_L 和 δ_H 如下。它生成 δ_L 的候选值(resp. δ_H)通过递归二分 $[0, \delta)$ (resp. $[\delta, 1]$)，其中范围被递归的中位数均匀划分，中位数保留为候选。二分过程继续进行，直到两个连续的候选值之间的差值小于0.01。然后它选择 δ_L (resp. δ_H)作为候选 δ_{cL} (resp. δ_{cH})导致最大FN值($p < 0.05$)。FP在M预测中的比值在 $[\delta_{cL}, \delta)$ 中具有优势。 $[\delta, \delta_{cH}]$ FN和FP比值是通过具有真假用户-物品交互(正样本和负样本)的训练数据应用M得到的。特别是在top-K推荐的情况下，对每个固定用户x的训练数据应用M。

实现RecLogic的发现和推荐模块的方法将分别在第4节和第5节中给出。在续集中，我们将重点关注CTR预测。尽管如此，所有的方法也可以很容易地处理top-K推荐。

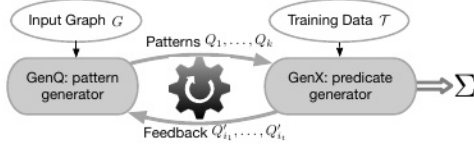


图3:TIE Learner工作流程

4 学习预测规则

在本节中，我们开发了TIE Learner，这是一种用于RecLogic的TIE发现方法。我们首先阐述了TIEs的发现(第4.1节)，并概述了TIE Learner(第4.2节)。然后我们提供TIE Learner的两个组件(第4.3-4.4节)。

4.1 发现问题

我们用支持和信心来衡量TIEs的质量。

标准。给定一个TIE $\phi = Q[x, y, x](x \rightarrow (x, \text{likes}, y))$ 和一个图G，用 $\text{Rec}(G, \phi)$ 表示 ϕ 预测的所有用户-物品对的集合。即 $(u, v) \in \text{Rec}(G, \phi)$ 如果(a) $u = h(x)$ 且 $v = h(y)$ 对于G中Q的某个匹配h，且(b) h满足x中的所有谓词。

支持。TIE ϕ 的支持度衡量的是 ϕ 应用的频率。它被评估为w.r.t. 训练数据T。这里T是图G内的真正和负真用户项目对的集合，它们是现实生活中的交互，可以作为基础真理。则 ϕ 在图G w.r.t T上的支撑力，用 $\text{supp}(\phi, G, T)$ 表示，定义为

$$\text{supp}(\phi, G, T) = |\text{Rec}(G, \phi) \cap \{(u, v) \mid (u, \text{likes}, v) \in T\}|.$$

这个度规是反单调的w.r.t. 在TIEs定义域上的偏序 \sqsubseteq 。给定两个TIEs $\phi = Q[x, y, x](x \rightarrow (x, \text{likes}, y))$ 和 $\phi' = Q'[x, y, x'](X \rightarrow (x, \text{likes}, y))$ ， $\phi \sqsubseteq \phi'$ if (a) Q是Q'的子图 and (b) x中的所有谓词也包含在x'中。若 $\phi \sqsubseteq \phi'$ ，则可验证 $\text{supp}(\phi, G, T) \leq \text{supp}(\phi', G, T)$ 。

信心。给定一个TIE ϕ ，其在图G w.r.t. 训练数据T上的置信度记为 $\text{conf}(\phi, G, T)$ ，定义为 $\text{conf}(\phi, G, T) = \text{supp}(\phi, G, T) / |\text{Rec}(G, \phi) \cap \{(u, v) \mid (u, \tau, v) \in T\}|$ ，其中 τ 为喜欢或不喜欢。置信度衡量 ϕ 的前提条件与 ϕ 的预测之间的关联强度，即 ϕ 的“精度”参考t中的ground truth。高置信度的TIE可以帮助我们减少FPs和FNs。

TIE发现。我们将TIE发现问题形式化如下。

◦ 输入: 图G, G中的训练数据T, 支持阈值 r_1 , 置信度阈值 r_2 和正整数 α 。

◦ 输出: 一个集 Σ of TIEs, 使得对于每个 $\phi \in \Sigma$, $\text{supp}(\phi, G, T) \geq r_1$, $\text{conf}(\phi, G, T) \geq r_2$, 且 ϕ 的模式在大多数 α 节点存在。我们使用 $\text{Req}(r_1, r_2, \alpha)$ 来表示TIE发现需求。问题的目的是找到符合 $\text{Req}(r_1, r_2, \alpha)$ 的一组TIEs。这里参数 α 控制着发现过程的成本。

4.2 发现方法

接下来，我们设计TIE Learner来支持RecLogic中的TIE发现。

概述。如图3所示，给定一个图G和一个训练集T，TIE Learner迭代学习TIEs，以轮次丰富ML模型M。每一轮都有两个步骤，通过模式生成器GenQ和谓词生成器GenX。在每一轮中，GenQ首先训练一个ML模型来生成少量TIE骨架，即双星模式Q和Q节点的属性。使用GenQ的属性，GenX然后用谓词将骨架扩展到TIE。

在每一轮结束时，GenX用T评估生成的TIEs的质量，保留高质量的，并收集反馈

Input: A graph G, a set of training data T, an iteration number I, ML models M, and a discovery requirement $\text{Req}(r_1, r_2, \alpha)$.

Output: A set Σ of TIEs from G s.t. each one conforms to $\text{Req}(r_1, r_2, \alpha)$.

1. pre-train the LSTM network with G; $\Sigma := \emptyset$; iter := 0;
2. **while** iter < I **do**
3. $\Sigma^0 := \text{GenQ}(\text{LSTM}, \alpha)$; $\Sigma' := \emptyset$;
4. **for each** $Q \in \Sigma^0$ **do** $\Sigma' := \Sigma' \cup \text{GenX}(Q, G, T, \text{Req}(r_1, r_2, \alpha))$;
5. refine the LSTM network with patterns extracted from Σ' ;
6. $\Sigma := \Sigma \cup \Sigma'$; iter := iter + 1;
7. **return** Σ ;

图4:发现算法TIE Learner

对于GenQ。反馈由高质量TIEs中的模式和属性组成，即降低M预测的FPs和FNs的重要特征。根据反馈，GenQ改进其ML模型，并在随后的回合中生成更多相关的模式。

模式发生器。GenQ生成双星模式Q来捕获用于推荐的特性。由于这样的Q可以表示为一组路径，因此GenQ采用了一种内射编码，将每个Q的路径映射到预定义字母表上的一个单词，并具有Q的节点的有用属性。它应用了长短期记忆(LSTM)网络[44]来创建这样的单词(参见第4.3节)。

谓词生成器。给定一个模式Q，GenX生成与Q相关的先决条件X，使得TIEs $Q[X, y, X](X \rightarrow (X, \text{likes}, y))$ 有助于减少FPs和FNs。GenX原则上采用分层搜索生成X。为了提高效率，它采用了两种优化: (a)模式匹配的紧凑存储及其增量维护;和(b)基尼指数[9]来挑选X的谓词。

算法。将这些放在一起，我们在图4中概述了TIE Learner。给定一个图G，训练数据T，ML模型M，一个正整数I(即整数的界)，以及一个发现要求 $\text{Req}(r_1, r_2, \alpha)$ ，TIE Learner在I轮中从G中学习符合要求的一组 Σ of TIEs。

在每一轮中，GenQ首先使用在图G(第1行)上预训练的LSTM网络生成一组 Σ^0 of 星型模式(第3行)。这里 α 是LSTM的超参数，使得 Σ^0 中的每个模式具有最多 α 节点。然后，对于每个生成的模式Q，TIE Learner应用GenX来查找与Q相关的谓词X，并生成TIEs $Q[X, y, X](X \rightarrow (X, \text{likes}, y))$; 它保留集合 Σ' (line 4中符合 $\text{Req}(r_1, r_2, \alpha)$ 的TIEs。TIE Learner接下来用反馈细化GenQ的LSTM(第5行)。它最终返回 Σ through 轮中所有TIEs的并集为 Σ (第6-7行)。

性能保证。TIE Learner保证在足够多的回合后，它能够以高概率学习所有符合给定要求的TIEs。这是因为GenQ可以探索所有可能的双星模式，而GenX能够根据 $\text{Req}(r_1, r_2, \alpha)$ 水平检查所有前提条件。

定理3:对于具有训练数据T的图G，一个要求Req和一个常数 $\lambda \in (0, 1)$ ，如果there存在一个符合 $|G| \alpha \lambda \log \lambda$ 与Req相符的TIEs集合 Σ^* ，那么在 $I = k(1-\omega) \frac{1}{1-\ln L} + \ln L (\ln L - 2)$ 轮之后，RecLogic发现 Σ^* 中所有TIEs的概率至少为 $1-\lambda$ 。

这里k和 ω 是GenQ的两个超参数。迭代上的界是根据切尔诺夫界[74]推导出来的。

之前的工作。TIE Learner与以前的挖掘方法[23 - 25, 79]在图关联规则方面有以下不同。

RNNLogic。TIELearner的灵感来自RNNLogic [79], mlogic是一种规则学习器, 它将逻辑规则视为知识图上的潜在变量, 同时训练规则生成器和推理预测器在RDF上学习简单形式的RDF规则。然而, TIELearner和RNNLogic在目标、逻辑规则和方法上有所不同。

(1) TIELearner旨在学习TIEs, 通过降低FPs和FNs来改进现有的ML推荐模型。相比之下, RNNLogic学习知识图推理的逻辑规则。

(2) 每个TIE采用双星模式, 并带有像 $x.A \oplus y$ 这样的谓词。B, x, $A \oplus c$ 或 $M(x,y) \otimes \delta$ 。相比之下, RNNLogic的每个规则都以路径作为其模式, 并且不携带谓词。这使得TIELearner中的规则发现比RNNLogic更具挑战性。

(3) TIELearner结合LSTM和优化的分层搜索生成tie, RNNLogic单独采用LSTM进行规则发现。此外, TIELearner在支持度和置信度两方面评估生成的tie, 而RNNLogic仅使用支持度。

采矿方法。与之前的采矿方法[23-25]相反,

(1) TIELearner旨在降低ML模型m的FPs和FNs, 它使用现实生活中正、负样本的训练集T来训练ML模型, 评估学习到的tie在T中的支持度和置信度, 并选择使用基尼指数来降低FPs和FNs的谓词。相比之下, 之前的方法不是由ML模型驱动的。

(2) TIELearner通过迭代GenQ和GenX学习tie。两步方法提高了准确率和效率。特别是, GenQ在每轮中生成少量模式, GenX将这些模式扩展到tie。因此, TIELearner能够在合理的时间内学习tie, 这是分层方法[23,25]无法做到的(参见第6节)。

(3) 虽然TACOs是通过创造者-批评家方法学习的[24], 但(a)它们针对时间图并采用时间分区, 这不适用于RecLogic;(b)学习方法有很大不同:[24]使用GAN生成时间模式和语言模型来识别标签, 而TIELearner通过LSTM来同时识别模式和属性, 并采用紧凑的模式匹配和基尼指数来加快过程, 降低FPs和FNs;这些对TACOs没用。(c) RecLogic在PTIME中提出建议, 而不是在EXPTIME中对TACOs进行预测。

4.3 模式发生器

现在我们介绍模式生成器GenQ, 从将模式映射到单词的编码开始。然后讨论了GenQ如何采用LSTM网络在编码的基础上生成模式。

编码模式。编码将每个双星模式映射成字母表上的一个单词 $\Gamma \cup \Sigma \cup \{ \#, \text{百科} \}$ 。这里 Γ 和 Σ 分别是标签和属性的集合(见2.1节); #和百科是两个在 $\Gamma \cup \Sigma$ 中没有出现的新符号, 分别用于分隔双星模式中的不同路径和星号。

设 $\leq L$ 为 $\Gamma \cup \Sigma$ 上的单词的词典顺序。对于一个具有标签L(u)且携带属性A1的模式节点 u, \dots, A_n , 当 $A_1 \leq L \dots \leq L A_n$ 时, 编码将u映射到序列 $L(u)A_1 \dots A_n$ 。另一方面, 每条边 (u, τ, u') 都由其标签 τ 编码, 以及一条路径 (u_1, \dots, u_l) 通过依次连接其节点 u_i 和边 (u_i, τ, u_{i+1}) 的表示进行编码。然后一个双星模式 $Q[x,y,x']$ 被编码为序列 $\beta_1 \# \beta$

$2 \dots \# \beta_s \# \text{百科} \xi_1 \# \xi_2 \dots \# \xi_t \# \text{百科}$, 其中(a) β_1, \dots, β_s (resp.) ξ_1, \dots, ξ_t 对从枢轴用户节点x开始的路径进行编码。(b)它们按照 $\leq L$ 的顺序进行排序, 即 $\beta_1 \leq L \dots \leq L \beta_s$ 和 $\xi_1 \leq L \dots \leq L \xi_t$ 。

模式的一代。通过编码, GenQ采用LSTM网络在 Γ 上生成每个单词 $u \in \{ \#, \text{百科} \}$, 直到它遇到第二个-符号。给定模式节点数 α 的上限, GenQ将单词长度限制为 $(\alpha-2)(4+2|\Sigma|)+2$ 。这指的是编码具有 α 节点的双模式Q的单词的最大长度。观察到, 当从一个支点出发的每条路径都由q中的一条边组成时, 单词达到其最大长度。此外, 对于一个超参数k, GenQ强制LSTM在每次调用后返回k个单词, 然后k个双星模式可以很容易地由GenQ从生成的单词中恢复。为了生成更多相关模式, LSTM网络使用从给定图G中采样的模式进行预训练, 如果模式Q在G中至少存在一个匹配, 则模式Q是相关的。

示例4:从示例3中回忆TIE ϕ_3 中的双星型模式Q3。它的编码是“用户, 预订, 机票, 目的地 # 用户, 购买, 剧场票, 类型 # 百科剧场票, 位置, 城市, 名字 # 希瑟票, of, 类型, 名字 # 百科”。需要注意的是, 除了标签, ϕ_3 中的重要属性, 例如目的地, 也被包含在内。2

均匀平滑。为了探索双星模式的完整搜索空间, 避免过度拟合LSTM模式的词生成分布, GenQ进行了均匀平滑。给定一个双星模式Q, 设 $p(Q)$ 为LSTM生成Q对应词的概率。然后GenQ将 $p(Q)$ 更新为 $\omega \times p(Q) + (1-\omega) \times |G|^{-1}$, 其中 ω 是度量训练可信度的超参数[68], $|G|^{-1}$ 是具有最多 α 节点的星型模式数量的上界。然后由GenQ以至少为 $|G|^{-1}$ 的概率生成每个最多 α 个节点的模式。这可以通过平滑 $(1-\omega)$ 来实现

向量 $\vec{v} = [(l_1, \boxtimes l_1), \dots, (l_M, \phi M)]$, LSTM从中选择生成单词的符号。这里, l_i 是 Γ 中的一个符号 $u \in \{ \#, \text{百科} \}$, $\boxtimes l_i$ 是LSTM模型生成符号 l_i 的概率, M是 $\Gamma \cup \Sigma \cup \{ \#, \text{百科} \}$ 的大小。

将反馈。每一轮之后, GenQ“增量”训练LSTM网络与来自GenX的反馈(第4.4节), 其中包括模式和属性, 已被验证用于捕获重要的属性进行推荐。它还将前几轮中生成的样本模式和从图G中采样的模式作为扰动纳入其中。这是为了避免灾难性遗忘[62], 并避免在训练过程中出现局部最小值, 即GenQ重复创建的一小组模式。

4.4 谓词发电机

我们继续讨论谓词生成器GenX。下面我们首先提出一个基于分层搜索的生成框架;然后我们引入两种策略来加速这个过程。

分层生成框架。对于GenQ返回的每个模式 n_q , GenX生成一个关于Q的先决条件集合 X_q , 并对所有 $X \in X_q$ 生成TIEs $Q[X, y, X]$ ($X \rightarrow (X, \text{likes}, y)$)的集合 ΣQ , 使得它们满足 $\text{Req}(r_1, r_2, \alpha)$ 的要求。回想一下, GenQ也产生q的属性, 用 Δ 表示的集合

用这些属性定义的逻辑谓词。然后用 Δ 中的谓词和(可能)ML谓词定义每个 $X \in \mathcal{X}_0$ 。

GenX以分层方式计算 \mathcal{X} 。我们定义一个前提条件 X 的大小为 X 中逻辑谓词的个数, 设 \mathcal{X}_i 为大小为 i 的前提条件集合。生成过程从 $\mathcal{X}_0 = \{X_0\}$ 开始, 其中 X_0 includes ML谓词 $M(X,y) < \delta \wedge M(X,y) \geq \delta$ 或 $M(X,y) \geq \delta \wedge M(X,y) < \delta$ 。这里的阈值 δ 对应于ML模型采用的阈值 δ , 不同的 δ 将用于top-K推荐中的不同用户(参见第3节)。对于 \mathcal{X}_i 中的每个 X , GenX从 Δ 中选取一个谓词 p , 并将 X 扩展到 $X \wedge p$; 检查推导出的TIE $\phi = Q[x,y,x](x \wedge p \rightarrow (x, \text{likes}, y))$ w.r.t. $\text{Req}(r1, r2, \alpha)$ 。若 $\text{supp}(\phi, G, T) \geq r1$, 且 $\text{conf}(\phi, G, T) \geq r2$, 则GenX将 $X \wedge p$ 加到 \mathcal{X} 上。若只有 $\text{supp}(\phi, G, T) \geq r1$ 成立, 则GenX将 $X \wedge p$ 加到 \mathcal{X}_{i+1} 上继续探索。注意, 当 $\text{supp}(\phi, G, T) < r1$ 时, 由于支撑的反单调性, $X \wedge p$ 不被选中。当没有 \mathcal{X}_i 可以进一步扩展时, 这个过程就会终止。

谓词集。接下来我们讨论如何构建谓词集 Δ , 其中既包括变量谓词, 也包括常量谓词。

(1) 由于变量谓词只定义在叶子或枢轴上, for 对于 Q 中用户 x 的叶子的每个属性 $z.A$, 我们将 $z.A \oplus z.B$ 添加到 Δ , 其中 $z.B$ 是叶子的一个相容属性或从 y 枢轴出发。我们确保每个叶子或枢轴最多有一个变量谓词。

(2) 基于属性类型构建常量谓词。如果 $z.A$ 是非数值的, 我们在 G 中收集 $z.A$ 的所有值, 并将 $z.A = c$ 和 $z.A, c$ 添加到每个收集到的值 c 。如果 $z.A$ 是数值的, 我们确定若干分裂点 $ss1, \dots, [93]$, 并在 Δ 中包含形式为 $z.A \in [ss, i]$ 的谓词。

GenQ的反馈。对于符合 $\text{Req}(r1, r2, \alpha)$ 的TIEs, GenX将 Q 和 \mathcal{X} 中的属性编码成一个序列, 反馈给GenQ, 以细化LSTM网络。

GenX的挑战。在最坏的情况下, GenX需要 $O(2^{|\Delta|} |G|^{|\mathcal{Q}|})$ 时间。这是因为(a)最多有 $2^{|\Delta|}$ 前置候选项;(b)为了验证一个前提条件, GenX可能需要检查图 G 中 Q 的指数次匹配(例如, $|G|^{|\mathcal{Q}|}$)。为了提高GenX的效率, 下面我们引入两个优化。

(1) 高效验证。给定一个前提条件 X , 我们用 $Q(u,v,X)$ 表示 G 中映射 X (resp. X)的对星模式 $Q[X,y,X]$ 的匹配集。 Y 到 u (resp. 观察到, 为了验证由 X 和 Q 导出的TIE的支持度和置信度, 对于每个集合 $Q(u,v,X)$ 检查 (u, τ, v) 在训练数据 T 中的隶属度就足够了, 其中 τ 为喜欢或不喜欢。为了加快这一过程, 我们引入了 $Q(u,v,X)$ 的紧凑表示。我们还提出了一种高效的增量算法来计算它。

紧凑的表示。假设模式 $Q[x,y,x]$ hasm 节点 $x1, \dots, Q(u,v,x)$ 的紧表示是一个 m 元组 $(C(x1), \dots, C(xm))$, 其中 $C(xi) = \{w \in V \mid \exists h (h(x)=u \wedge h(y)=v \wedge h(x \ i)=w \wedge h(x)=x) \}$ 对于 $i \in [1,m]$, 即 x_i 的可满足匹配。我们说 $Q(u,v,X)$ 非空, 如果每个 $C(xi)$ 非空。设 ϕ 为由 Q 和 X 导出的TIE, 则 $\text{supp}(\phi, G, T)$ 为 $(u, \text{likes}, v) \in T$ 的 $Q(u,v,X)$ 的非空实例的个数, $\text{conf}(\phi, G, T)$ 为 $\text{supp}(\phi, G, T)$ 与 $(u, \tau, v) \in T$ 的非空 $Q(u,v,X)$ 的个数之比。

给定节点 u, v 和模式 Q , 可以通过应用类似于[20,38

]的DAF算法来计算 $Q(u,v,X)$ in $O(|Q||G|)$ 时间。简而言之, 该算法首先将 Q 分解为一组路径; 然后它以自上而下的方式找到这些路径的匹配项, 即从枢轴 u 和 v 到叶子; 最后通过聚合匹配来计算 $Q(u,v,X)$ (参见第5节)。

动态维护。我们开发了一种增量算法IncM来保持紧凑匹配。它从 $Q(u,v,X)$ 计算 $Q(u,v,X \wedge p)$ 作为对一个新的谓词 p 的响应, GenX试图用它来扩展 X 。其基本原理是, 一个新的 p 只影响放置在连接那些涉及 p 的节点到枢轴的路径上的模式节点的匹配。因此, GenX只是沿着这些路径动态地将更改传播到存储在 $Q(u,v,X)$ 中的 $C(z)$ 。

更具体地说, 如果 p 是一个变量谓词 $z.a \oplus z.B$, IncM首先根据定义更新 $\text{esc}(z)$ 和 $c(z)$ 。然后它沿着无向路径 $\rho(z,x) = (z1, \dots, zl)$ 从 z 到枢轴 x 。在每一步中, 它检查同态条件是否仍然保持w.r.t. 每个节点 $w \in C(z_{i+1})$ 和更新的 $C(z_i)$, 并相应地调整 $C(z_{i+1})$ 。它还以同样的方式将变化从 z 传播到枢轴 y 。处理一个常量谓词 $z.a \oplus c$ 的情况与此类似, 不同之处在于, 变化传播是从 z 到主节点和叶节点, 并在另一个恒星的相应叶节点受到影响时继续进行。

分析。IncM需要 $O(|Q| |Q(u,v,X)|^2)$ 时间。直观地说, 在变化传播的每一步中, IncM通过验证另一个 $C(x_j)$ 中的每个元素的边连接w.r.t. 来检查当前 $C(x_i)$ 是否可以被精炼, 最多花费 $O(|Q(u,v,X)|^2)$ 时间。这里 $|Q(u,v,X)|$ 表示对于 $i \in [1,m]$, $|C(x_i)|$ 的和, 其中 $|C(x_i)|$ 是 x_i 的可满足匹配的个数。由于这种检查最多进行 $O(|Q|)$ 次, 因此复杂度是这样的。

(2) 基于基尼系数的剪枝。接下来, 我们引入了一种基于基尼系数的修剪策略, 以减少分层搜索空间, 并选择用于减少ML模型的FPs和FNs的谓词。基尼系数是衡量一种条件将一组数据划分为不同类别的有效标准[9,36]。

给定训练数据 T 的子集 D , 基尼值基尼(D)衡量 D 的纯度, 定义为 $1 - f1 - 2 - f2 - 2$, 其中 $f1$ 和 $f2$ 分别是 D 中标记为喜欢和不喜欢的实例的比率。基尼系数(D)越小, 表示 D 的纯度越高。定义前提条件 X 的基尼系数为

$$\text{Gini_index}(X, p) = |D_1| \text{Gini}(D_1) / |D| + |D_2| \text{Gini}(D_2) / |D|,$$

其中 D 表示 T 中 (u, τ, v) 与非空 $Q(u,v,X)$ 的集合, $D1$ (p). $D2$)是 D 中 (u, τ, v) 的集合, 具有非空 (resp. 空) $Q(u,v,X \wedge p)$ 。直观地, $\text{Gini_index}(X, p)$ 表示 p 将 D 分成带有喜欢和不喜欢标签的实例的程度。因此, 具有较小 $\text{Gini_index}(X, p)$ 的谓词 p 可以有效地区分真实的推荐案例和相反的推荐案例。

在此基础上, 每次GenX选择导致相对较小的基尼系数的谓词(重要特征)来扩展 x 。这有助于我们在几轮内发现降低FPs和FNs的tie, 并且可能具有高置信度。此外, 它可能导致使用少量谓词的tie, 因此易于解释。

例5: 继续例4, GenX首先根据GenQ生成的属性构造谓词集 Δ 。集合 Δ 包括所有的变量谓词, 如 $x1.destination =$

$y1, name, x1, destination, y1, name, x2$ 。类型= $y2.name, x2$ 。类型, $y2.name$ 。它还包括所有的常量谓词, 如 $x1.destination \oplus LA$ 和 $y1$ 。类型 $\oplus History$, 其中 LA 和 $History$ 是出现在 G 中的 $Q3$ 属性值。

GenX接下来生成先决条件集合 X 。假设 $X: x1.destination = y1$ 。name是GenX目前探索过的一个前提条件, 其中省略ML谓词, TIE $Q3[X, y, X](X \rightarrow (X, likes, y))$ 的置信度小于 $r2$ 。然后, GenX在 Δ 中选择一个谓词, 进一步扩展前提条件 x 。p的候选者包括变量谓词 $x2$ 。类型= $y2.name, x2$ 。类型, $y2.name$ 和 Δ 中的所有常量谓词。GenX对p的基尼指数进行评估, 如上图所示, 选择值相对较小的, 如 $x2$ 。类型= $y2.name$ 。如果对应的TIE满足发现要求, 则GenX添加“ $x1.destination = y1.name \wedge x2$ ”。作为结果, 我们得到一个TIE $\phi3$, 如例3所示。2

5 平行的建议

在本节中, 我们为RecLogic的推荐模块提出了一种并行算法ParRec。它计算每个TIE ϕ 的 $Rec(G, \phi)$ 。我们从一个PTIME序列SeqRec开始。然后我们展示了ParRec, 并展示了它相对于SeqRec是并行可扩展的。

这些算法可以很容易地扩展到一组 Σ of TIEs。

算法SeqRec。给定一个图 G 和一个TIE $\phi = Q[x, y, x](x \rightarrow (x, likes, y))$, SeqRec分两个阶段计算 $Rec(G, \phi)$ 。第一个对 ϕ 进行预处理。它通过在 q 中编码额外的边来消除变量谓词。第二步通过路径匹配推导出 $Rec(G, \phi)$ 。

(1)预处理。在这个阶段, SeqRec从给定的TIE ϕ 构造了一个新的TIE $\phi \approx Q[x, y, \approx x](\approx x \rightarrow (x, likes, y))$, 使得 $\phi \approx$ 在 $x \sim$ 中不包含变量谓词, 对于每个变量谓词 $p = z1 \oplus z2$ 。B在 ϕ 的条件 X 中, SeqRec从 X 中移除 p , 并在 q 中包含一条新边 $(z1, \tau p, z2)$ 。这里 τp 是表示 p 的新标签。这允许我们将谓词检查转换为路径匹配。

(2)路径匹配。接下来, SeqRec计算 $Rec(G, \phi)$, 方法是在应用 ϕ 时, 推导出 G 中要推荐给给定用户(节点 u)的项目的 $Rec(u)$ 集。然后, $Rec(G, \phi)$ 可以很容易地通过对主轴 x 的匹配候选点 u 进行测距来计算。

SeqRec算法通过对根为 u 的 G 中 $Q \sim$ 的所有匹配来识别 $Rec(u)$ 。由于模式 $Q \sim$ 是从双星模式转换而来, SeqRec将 $Q \sim$ 分解为不同的无向路径, 并独立验证这些路径的匹配, 如下所示。

(a)用 p 表示在修正后的模式 q 中从枢轴 x 开始或以枢轴 y 结束的无向路径的集合。

(b)对于每条路径 $p = (z1, \dots$ 在 p 中, SeqRec导出节点 w_i 的sets $C_p(z_i)$, 使得存在一条路径 $(w1, \dots)$, w_i), 与子路径 $(z1, \dots$ 子)。该过程从任一 $C_p(x) = \{u\}$ 或 $C_p(z1) = V_{u_i}$ 开始, 其中 $z1 \sim x$ 和 V_{u_i} 包含了 u_i 满足其常量谓词 $z1$ 的所有候选匹配项。A = c。它利用动态规划[75]计算所有 $i \in [1, l]$ 的 $C_p(z_i)$ 。

与[75]不同, $Q \sim$ 有表示变量谓词 $p = z1$ 的边 $(z1, \tau p, z2)$ 。 $\oplus z2$ 。b。为了处理这个问题, 它根据它们的A属性和B属性对 G 中的节点进行排序, 并按照这个顺序推导出 $C_p(z2)$, 例如, 如果 p 是 $z1 \leq z2$ 。B, 我们首先识别节点在

Input: A graph G and a TIE $\phi = Q[x, y, \bar{x}](X \rightarrow (x, likes, y))$.

Output: A set $Rec(G, \phi)$ of recommendations predicated by ϕ .

1. pre-process Q and eliminates variable predicates in X ;
2. $V_x := \{u \in V \mid L(u) = L_Q(x) \wedge u \text{ satisfies constant predicates of } x\}$;
3. for each $u \in V_x$ do /* run on n processors in parallel */
4. $Rec(u) := \{v \in V \mid L(v) = L_Q(y) \wedge v \text{ satisfies } y \wedge M(u, v) \otimes \delta\}$;
5. for each path $p = (z1, \dots, z_l)$ in \mathbb{P} do
6. compute $C_p(z1), \dots, C_p(z_l)$ recursively;
7. $Rec(u) := Rec(u) \cap C_p(y)$;
8. return $\{(u, v) \mid u \in V_x \wedge v \in Rec(u)\}$;

图5:并行推荐算法ParRec

具有最小A属性的 $C_p(z1)$, 然后将 $C_p(z2)$ 定义为B属性大于此值的节点集合。

(c)最后, 通过计算 P 中所有路径匹配的交集得出 $Rec(u)$, 即 $Rec(u) = \bigcap_{p \in P} \rho(y)$ 。

例6:考虑例3的TIE $\phi3$ 。在预处理过程中, SeqRec包括模式 $Q3$ 中的两条边, 这两条边连接 $x1$ 和 $y1$ 。 $X2$ 和 $y2$ 表示 $x1.destination = y1.name$ (resp. $x2.genre = y2.name$)。这导致从枢轴 x 到 y 的两条路径的集合 p 。然后SeqRec在这些路径上执行动态规划来推断匹配, 例如, $x1$ 和 $x2$ 的匹配都是通过检查 x 的候选匹配及其相邻边标记为book和buy来决定的。特别是, 在验证 $x1.destination = y1.name$ 时, 它(1)首先将目的地和名称值按字典序排序, (2)然后通过引用这个顺序和 $x1$ 匹配项的目的地值来确定 $y1$ 的候选匹配项, 而不是检查图 g 的边。 $\phi3$ 中的其他变量谓词也进行类似的处理。2

分析。SeqRec运行时间为 $O(|V|(|\phi||E| + |V||\phi||G|))$ 。(1)其预处理成本为 $\ln O(|\phi|)$ 。(2)路径匹配是 $\ln O(|\phi||V||E|)$ 时间, 因为(a)找到 x 个点的候选匹配需要 $O(|\phi||V|)$ 时间, (b)计算每个点 $\in V$ 的 $Rec(u)$ 需要 $O(|\phi||E|)$ 时间。(3)对某多项式函数 $f(\cdot)$ 应用ML模型 M 需要 $f(|G|)$ 时间。因此它在PTIME;相比之下, 其对应的GFDs、GARs和TACOs[23, 24, 26]需要指数时间。

此外, 该算法还表明, 双星模式的图模式匹配可以通过动态规划在PTIME中完成, 而对于一般的图模式则难以处理(cf. [31])。

SeqRec的一个小修订通过其与推荐的连接进行验证(第2.2节); 如果存在 $\phi \in \Sigma$ such且 $G \in \phi$, 则返回false, 否则返回true, 且复杂度相同。这为定理1提供了构造性证明。

平行的可伸缩性。算法SeqRec仍然是昂贵的, 当应用于大型图。为了加快速度, 我们提出了并行算法ParRec。为了衡量其有效性, 我们采用了并行可扩展性[54]。

考虑图 G 上的问题 P , 我们用 $t(|I_P|, |G|)$ 表示处理问题 P/G 的实例 I_P 的顺序算法a的最坏情况复杂度。对于问题 P 的并行算法 A_p , 我们用 $\tau_p(|I_P|, |G|, n)$ 表示它在 G 上使用 n 个处理器处理问题实例 I_P 所花费的时间。我们说算法 A_p 相对于 A 是并行可伸缩的

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

例如, I_{p_0} 。也就是说, 并行算法 A_p 能够“线性”降低尺度算法a的顺序代价。

Method	$F_{FP, FN}$	F1	AUC
MovieLens			
RecLogic _{NCF}	0.6199 (\uparrow 17.67%)	0.7364 (\uparrow 30.29%)	0.6243 (\uparrow 16.98%)
RecLogic _{LightGCN}	0.7057 (\uparrow 17.26%)	0.4541 (\uparrow 22.07%)	0.5763 (\uparrow 6.17%)
RecLogic _{DeepFM}	0.6580 (\uparrow 25.98%)	0.6745 (\uparrow 25.89%)	0.5861 (\uparrow 11.13%)
RecLogic _{xDeepFM}	0.6446 (\uparrow 16.33%)	0.7105 (\uparrow 10.58%)	0.6225 (\uparrow 13.57%)
CiaoDVD			
RecLogic _{NCF}	0.7038 (\uparrow 32.05%)	0.6269 (\uparrow 15.84%)	0.7062 (\uparrow 29.20%)
RecLogic _{LightGCN}	0.6660 (\uparrow 30.26%)	0.5512 (\uparrow 21.52%)	0.6662 (\uparrow 22.01%)
RecLogic _{DeepFM}	0.5658 (\uparrow 13.11%)	0.6114 (\uparrow 19.76%)	0.5615 (\uparrow 9.20%)
RecLogic _{xDeepFM}	0.6028 (\uparrow 20.66%)	0.6147 (\uparrow 17.87%)	0.5694 (\uparrow 9.46%)
Yelp			
RecLogic _{NCF}	0.6648 (\uparrow 16.20%)	0.7303 (\uparrow 27.25%)	0.7210 (\uparrow 16.52%)
RecLogic _{LightGCN}	0.8202 (\uparrow 20.53%)	0.8752 (\uparrow 24.60%)	0.8576 (\uparrow 17.27%)
RecLogic _{DeepFM}	0.7130 (\uparrow 33.10%)	0.7318 (\uparrow 26.98%)	0.6144 (\uparrow 13.69%)
RecLogic _{xDeepFM}	0.7230 (\uparrow 31.53%)	0.7411 (\uparrow 28.86%)	0.7251 (\uparrow 29.32%)

表2: 丰富CTR预测的有效性

并行算法。我们在图5中概述了并行算法ParRec。给定一个图和一个TIE ϕ , ParRec并行计算Rec(G, ϕ)。这里我们假设G在每台机器上都被复制。该算法可以很容易地扩展到跨机器分割的G。

它首先像SeqRec一样并行预处理Q(第1行)。它计算枢轴x的候选匹配V_x(第2行)。对于V_x中的每个u, ParRec并行计算Rec(u)(第4-7行)。负载w.r.t.不同的u被均匀地分配到处理器上。最后, 它合并并返回所有推荐对(第8行)。

Theorem 4: ParRec is parallelly scalable relative to SeqRec. \square

证明: 我们证明了ParRec在n个处理器下运行的时间为 $O(|V|(|\phi| + |V|f(|G|))/n)$ 。这是因为(1)并联预处理成本为 $O(|\phi|)$, (2)并链路匹配和ML测试需要 $O(|V|/n \times (|\phi||E| + |V|f(|G|)))$ 时间, $n \ll |V|$ 。2

6 实验研究

使用真实数据集和合成数据集, 我们评估了(1)RecLogic在丰富ML推荐方面的有效性, (2)其规则发现模块的质量和(3)效率, 以及(4)推荐模块的(并行)可扩展性。我们还提出(5)一个用于解释推荐的案例研究。

实验设置。我们从实验环境开始。

数据集。我们使用了三个现实生活中的图表。(1) 20M版本的MovieLens[39], 这是一个推荐基准, 拥有138K用户, 27K电影和20M评分(边)。它被从知识图谱Freebase中提取的特征丰富了, 增加了4百万节点和7.4百万边。(2) CiaoDVD[1], 一个拥有17K用户、16K电影和65K评分的二分用户-电影评分网络。它还包括用户之间的信任关系和用户评论的评分, 有85K个节点和1.8M条边。(3) Yelp[2], 一个由46K用户、46K商家和120万条边组成的用户-商家点评网络。我们用知识图谱和社交网络的特征丰富了它, 增加了1.8K节点和1.3M边。

为了评估可扩展性, 我们实现了一个图生成器来生成十亿规模的更大的合成图。它们由节点数(最多0.2B)和边数(最多0.8B)控制。

算法和模型。RecLogic是用c++和Python实现的。模式生成器基于Pytorch[3]的LSTM包构建, 具有默认的训练配置和一个256宽的层, 而谓词生成器和推荐模块则用c++从头开始实现。

RecLogic增强了各种推荐ML模型M, de-

Method	Recall@K	NDCG@K	$F_{FP, FN}$
MovieLens			
RecLogic _{NCF}	0.4728 (\uparrow 5.56%)	0.8825 (\uparrow 2.34%)	0.7427 (\uparrow 8.55%)
RecLogic _{LightGCN}	0.4917 (\uparrow 6.52%)	0.8992 (\uparrow 3.13%)	0.7189 (\uparrow 4.17%)
RecLogic _{DeepFM}	0.4181 (\uparrow 7.79%)	0.8269 (\uparrow 4.25%)	0.7371 (\uparrow 5.89%)
RecLogic _{xDeepFM}	0.4127 (\uparrow 5.12%)	0.8339 (\uparrow 5.54%)	0.7363 (\uparrow 5.47%)
CiaoDVD			
RecLogic _{NCF}	0.4532 (\uparrow 8.03%)	0.4029 (\uparrow 5.11%)	0.8279 (\uparrow 4.92%)
RecLogic _{LightGCN}	0.4768 (\uparrow 5.79%)	0.4313 (\uparrow 2.89%)	0.8311 (\uparrow 4.63%)
RecLogic _{DeepFM}	0.4415 (\uparrow 10.15%)	0.3869 (\uparrow 5.19%)	0.8391 (\uparrow 6.22%)
RecLogic _{xDeepFM}	0.4749 (\uparrow 9.73%)	0.4192 (\uparrow 5.26%)	0.8288 (\uparrow 5.07%)
Yelp			
RecLogic _{NCF}	0.5680 (\uparrow 25.25%)	0.6174 (\uparrow 6.06%)	0.8616 (\uparrow 11.13%)
RecLogic _{LightGCN}	0.6761 (\uparrow 10.11%)	0.7398 (\uparrow 3.40%)	0.8497 (\uparrow 8.66%)
RecLogic _{DeepFM}	0.5411 (\uparrow 31.81%)	0.5820 (\uparrow 6.99%)	0.8688 (\uparrow 12.19%)
RecLogic _{xDeepFM}	0.5713 (\uparrow 20.05%)	0.6295 (\uparrow 5.44%)	0.8542 (\uparrow 10.12%)

表3: 充实top-K推荐的有效性

记为RecLogic_M。这里M是(a)基于神经网络或图卷积网络的流行CF模型NCF[42]和LightGCN[40]中的一个, 以及(b) DeepFM[34]和xDeepFM[58]中的混合模型, 它们同时探索交互和内容特征。ML模型M也在丰富的数据集上进行了训练。也就是说, M可能会忽略某些有用的特征, 例如社会链接, 即使它被提供了丰富的数据。

通过修正[21], 实现了一种基于水平搜索的LTIE算法, 用于挖掘tie, 作为发现基线。并与挖掘Horn规则进行链路预测的AMIE+[28]算法进行了比较。

之后[40,42,97], 在学习tie和训练上述模型时, 我们从图中随机选择80%的用户-物品交互(每个用户)作为训练集的正样本(真正)。其余的用户-项目交互是测试集中的积极样本, 以评估挖掘的tie和训练的模型。在CTR预测中, 对于正样本中的每个用户, 我们在训练集和测试集中都选择了与用户没有交互的相同数量的项目作为负样本。在top-K推荐中, 我们在生成测试集时为每个阳性样本随机选择99个阴性项目。

环境。实验是在一台由Intel Xeon处理器驱动的机器上进行的; 它有1.5T内存, 2个cpu(每个有52个逻辑核), 8个NVIDIA GeForce RTX 3090 gpu。每个实验运行了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。默认情况下, 我们在学习tie时设置 $\alpha=10$, $|X|=20$, $r_1=50$, $r_2=0.5$, $l=10$, 并使用 $n=4$ 个逻辑核。正如即将显示的, RecLogic对这些发现参数表现出不同的敏感性;因此, 它们的默认值是根据同时实现相对较高的准确性和效率的设置来确定的。这可以作为一种指导, 来决定具有相似规模和语义的其他图的参数。为简单起见, 除非另有说明, 否则我们在CTR预测中也设置 $\delta=0.5$, 在top-K推荐中设置 $K=10$ 。由于空间不足, 我们报告了一些数据集的结果;在其他数据集上的结果是一致的。

Exp-1: 有效性。我们首先测试了RecLogic在改进推荐模型方面的有效性。在CTR预测中, 我们采用了三个精度指标: (1) $F_{FP, FN}$, 真阳性和真阴性与推荐方法所做的所有决策(阳性或阴性)的比率, 即总体信度; (2)传统的f1分数; (3)AUC, 该方法对随机阳性样本返回较高分数的概率

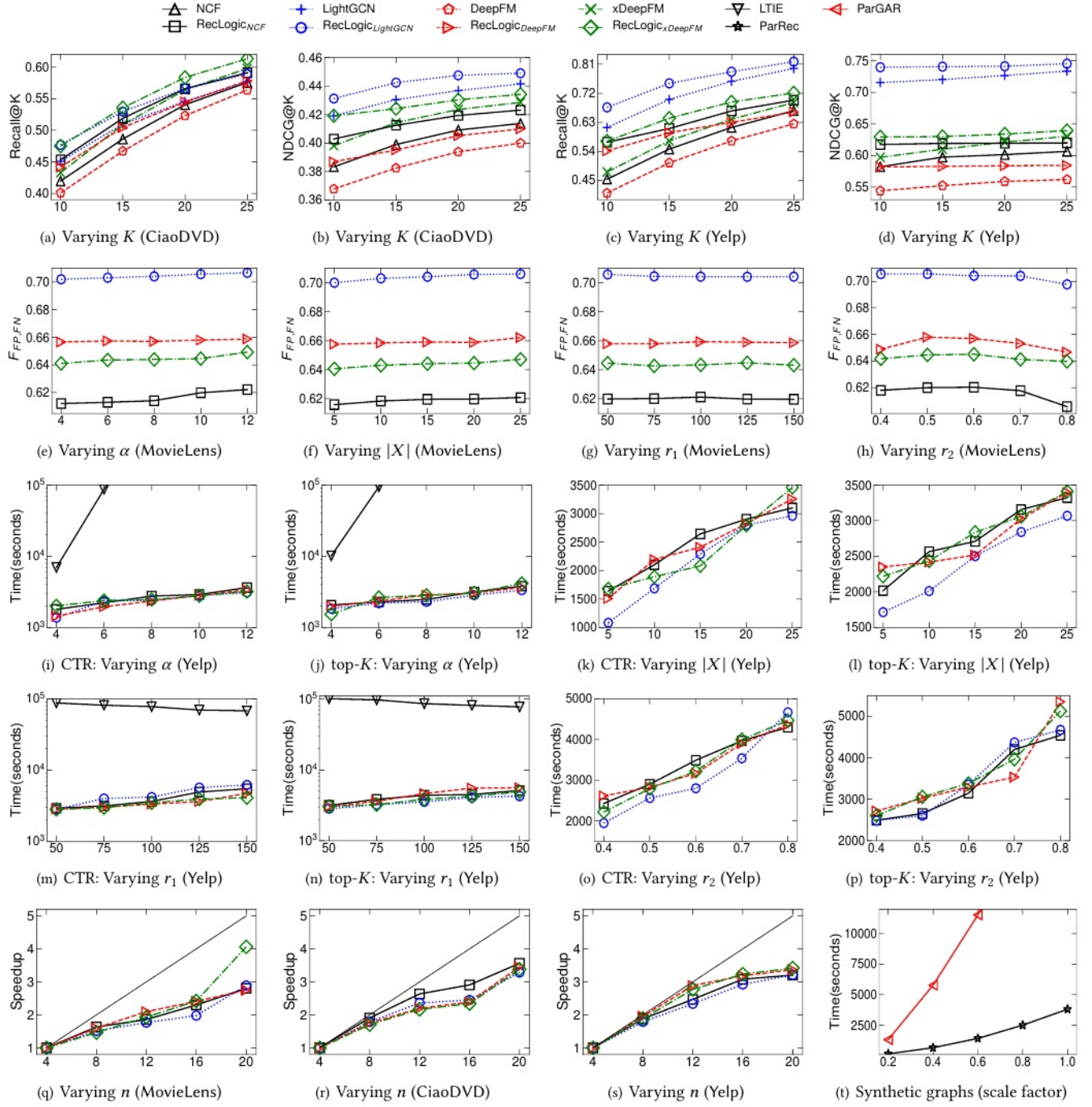


图6:绩效评价

对于随机负样本[34];它是CTR预测中常用的指标。在top-K推荐中,除了 $F_{FP, FN}$,我们还选择了(4)Recall@K和(5)NDCG@K。Recall@K测量在top-K推荐中检索到的正面条目的比例。NDCG@K根据相关性和位置访问top-K项的排名质量。大多数现有的top-K推荐系统都应用这两个指标来检查它们检索相关项目的的能力,并对它们进行适当的排名。

请注意,我们如上所述定义 $F_{FP, FN}$,因为它以平衡的方式测量降低FPs或FNs的效果。相比之下,f1分数倾向于减少FNs,即减少相同数量的FNs产生的f1分数高于减少FPs。

精度提高。给定一个ML模型M,我们从训练集中确定模糊区域 H (见第3节)。然后我们应用tie来改进 H 中M的推荐并检查有效性。这些tie是在 $I \leq 10$ 轮之后学习的。

I	1	3	5	8	10
$F_{FP, FN}$	0.6374	0.6487	0.6522	0.6580	0.6574
F1	0.6587	0.6637	0.6777	0.6745	0.6710
AUC	0.5296	0.5660	0.5786	0.5861	0.5825

表4:发现质量(MovieLens, 丰富DeepFM)

(a) CTR预测。表2显示了当对CTR进行预测时, RecLogic_M在模糊区域_H中获得的最佳 $F_{FP, FN}$ 、f1分数和AUC改进;这里 δ 是为每个M和每个数据集确定的, 以最大化M的精度, 在[0.3,0.7]的范围内。

我们发现如下。(1) RecLogic始终使所有四种ML模型在所有现实生活中的图形上更加准确。(2)平均提高NCF_{FP, FN}($p < 0.05$)。LightGCN, DeepFM, xDeepFM)的增长率为21.97%。22.68%, 24.06%, 22.84%), f1分数分别为24.46% ($p < 0.05$)。22.73%, 24.21%, 19.10%), AUC为20.90% ($p < 0.05$)。分别为15.15%, 11.34%, 17.45%。(3)其 $F_{FP, FN}$ (分别为15.15%、11.34%、17.45%)。f1分数, AUC)最高可达0.8202 ($p < 0.05$)。0.8752, 0.8576)。这些验证了RecLogic在增强现有ML推荐模型方面的有效性。

在不限于_H的整个Yelp中, 我们应用了所有学习到的TIEs, 不仅包括带有ML谓词的TIEs, 还包括 $Q[x, y, x](x \rightarrow (x, \text{likes}, y))$ 形式的TIEs, 其中前提条件 x 只包含逻辑谓词而没有嵌入ML模型(第3节), 例如例3的 ϕ_5 。AUC平均得分($p < 0.05$)。 $F_{FP, FN}$ 和F1-score)达到0.9219 (resp.:0.9033和0.8961), 增产2.62% (p. 平均增产2.93%和3.16%。改进低于_H内, 因为M在_H内的预测强度接近阈值;其在_H中的预测接近随机选择, 对应的AUC在0.5左右。请注意, 在不训练新模型的情况下, 改进已经可以在整个区域内媲美最先进的推荐模型-els[27, 45, 70, 82, 94, 99]。

(b) Top-K推荐。如表3所示, 在进行top-K推荐时, RecLogic平均提高了NCF的Recall@K ($p < 0.05$)。LightGCN, DeepFM, xDeepFM)增长12.95%。7.47%, 16.58%, 11.63%), NDCG@K为4.50% ($p < 0.05$)。3.14%, 5.48%, 5.41%)。这里我们设 $K = 10$;将学习到的具有不同阈值 δ 的TIEs应用于_H中不同用户的推荐(Section 3)。

请注意, 表3中的Recall@K和NDCG@K的值是通过检查所有项目的预测分数来计算的, 包括但不限于模糊区域_H所覆盖的项目。因此, 尽管我们只调整了top-K推荐中_H内项目的分数, 但其有效性反映为对这两个全局指标在整个区域的改进。与CTR预测相反, 这两个指标不能单独定义_H。

我们还将K从10变化到25, 以检查其对准确率的影响。如图6(a) -6 (d)所示, 所有方法的Recall@K都随着K的增加而增长, 因为K越大, 检索到的阳性项越多。然而, Recall@K的改进在K增大时略有下降, 因为ML模型遗漏的阳性项较少。相比之下, NDCG@K的改善是一致的。

对参数的敏感性。在MovieLens上, 我们还评估了其他发现参数对准确性的影响。迭代次数I将在示例2中进行分析。这里我们报告CTR预测;top-K推荐的结果是一致的(未显示)。(1)如图6(e)所示, 当学习到的TIEs在模式节点上的界值 α 较大时, RecLogic_M对M的改进效果更好。这是预期的, 因为更有效的TIEs可以有助于减少

Datasets	len. of rules	RecLogic	AMIE+	# of rules
MovieLens	3	143.05s	2h 31min	14
CiaoDVD	3	17.08s	28.61s	10
	4	89.19s	6min 9s	90
Yelp	3	53.82s	20min 41s	5

表5:RecLogic与AMIE+的比较

FPs和FNs返回的 α 值较大。同样, 在谓词数上, 当 $|X|$ 的边界较大时, 它的表现也会更好(图6(f))。

(2)图6(g)至图6(h)显示, 分别使用较大的支持度阈值 r_1 和置信度阈值 r_2 学习TIEs, RecLogic_M对M的改善效果更好。这种TIEs可以频繁应用, 具有更高的“精度”。也就是说, 当 r_1 或 r_2 太大时, 一些有用的TIEs可能会被忽略, 这在 r_2 中更为明显。

示例2:学习TIEs的质量。我们评估了发现模块TIELearner学习到的TIEs的质量。表4报告了在MovieLens的区域_H中, 使用学习到的TIEs w.r.t.模型DeepFM, 将回合数I从1到10改变后的精度改进情况。

如图所示, 当I增加时, 推荐精度在几轮迭代中收敛, 例如, 当 $I = 8$ 时 $F_{FP, FN} = 0.6580$ 。也就是说, 我们的发现方法能够在几轮内以高置信度找到有效的TIEs来降低FPs和FNs。

观察一下, 当我从8进一步增加到10时, 准确率会下降一点。这是因为(a)在前8轮中学习的TIEs已经足以识别大多数FPs和FNs, 并且(b)当I在第8轮后继续增加时, 被TIEs忽略的错误决策多于被发现的错误决策。

Exp-3:效率。接下来, 我们报告了不同参数对RecLogic_M中用于CTR预测和Yelp top-K推荐的TIE发现效率的影响。我们学习了固定数量的500个TIEs, 它们携带符合Req(r_1, r_2, α)的ML谓词, 因为LTIE在查找整个规则集时成本太高。RecLogic_M配备了相同的TIELearner, 但它的行为可能不同, 因为(a)不同M的面积_H不同, (b)降低FPs和FNs的TIEs可能不同。

不同的 α 。我们将TIEs中模式节点的 α 数从4个改变为12个。如图6(i) -6 (j)所示, (1)正如预期的那样, 当 α 变大时, 所有的发现方法都需要更长的时间。(2)RecLogic_M中的TIELearner比同级LTIE更快。 α 越大, TIELearner优于LTIE, 因为LTIE的成本呈指数增长。(3)RecLogic_M能够找到具有合理大模式的TIEs。在Yelp上, 当 $\alpha = 10$ 时, RecLogic_M平均需要2924.88秒来挖掘TIEs, 而当 $\alpha = 6$ 时, LTIE需要24.3小时来挖掘TIEs。

变化 $|X|$ 。我们将TIEs的前提条件中的谓词数量从5个改变为25个。如图6(k) -6 (l)所示, (1)所有的方法都需要更长的时间, 且 $|X|$ 更大。(2)当 $|X| = 5$ 时, RecLogic_M平均优于LTIE至少48.31倍;此外, LTIE不能在24小时内完成(未显示)。这些验证了我们的优化策略在挖掘谓词方面的有效性。

不同的 r_1 。图6(m) -6 (n)显示, 当支持阈值 r_1 变大时, TIELearner所需的时间略长。虽然它的模式生成器对 r_1 不敏感, 但它需要更多的轮来学习具有更高支持的预期TIEs数。对于较大的 r_1 , 水平LTIE变得更快, 因为它通过支持的反单调性来修剪模式。然而, 它经常需要超过18个小时。

不同的 r_2 。如图6(o) -6 (p)所示, 对于置信阈值 r_2 较大的情况, RecLogic_M学习TIEs所需的时间也更长

Tasks	RecLogic		xDeepFM	
	rule discovery	recomm.	training	testing
CTR prediction	2793.76s	221.11s	2937.55s	6.26s
Top-K recommendation	3058.71s	309.52s	2937.55s	192.69s

表6:RecLogic花费的时间(Yelp, 丰富xDeepFM)

和上面给出的理由一样。LTIE对 r_2 增加的反应类似，但在所有情况下仍不能在18小时内终止。

与其他基线的比较。使用32个逻辑内核，我们比较了RecLogic和AMIE+的规则发现模块在整个图上挖掘相同长度(模式大小)的规则。在这里，我们应用AMIE+来挖掘支持度高于1000的Horn规则，RecLogic被设置为找到相同数量的TIEs。如表5所示，当规则长度为3时，RecLogic的平均速度比AMIE+快29.36倍。事实上，如果长度达到4,AMIE+需要更长的时间，特别是在挖掘带有常量的规则时(未显示)。RecLogic之所以更快，是因为:(1)它只考虑星型模式，而不像AMIE+那样考虑循环模式，后者的匹配成本更高;(2)使用基于生成的方法来查找模式，避免了AMIE+的大搜索空间;(3)高效的验证和基于基尼指数的修剪(章节4.4)有助于降低谓词生成和检查的成本。

如表6所示，我们还测试了RecLogic中用于改进xDeepFM而不是Yelp的不同阶段所花费的时间比例。我们可以看到，规则发现部分主导了整体成本。然而，这与训练xDeepFM的成本相当。另一方面，由于TIEs的应用，推荐时间比xDeepFM的测试时间要长。请注意，我们对CTR预测和top-K推荐有相同的训练时间，因为通过抽样选择了相同的训练集。

如第3节所述，RecLogic采用不同的策略来识别这两个任务之间的模糊区域。在top-K推荐中，每个用户都有一个不同的模糊区域，而CTR预测只需要一个全局模糊区域，而且更简单。因此，RecLogic中的规则挖掘和推荐需要花费更长的时间进行top-K推荐。

Exp-4:可伸缩性。接下来，我们在RecLogic_M中测试了推荐模块ParRec的(并行)可伸缩性。我们应用了RecLogic_M中发现模块学习到的40个TIEs。

平行的可伸缩性。改变逻辑核的数量 n ，图6(q) - 6 (s)报告了使用4个逻辑核时每种方法相对于设置的加速比。请注意，细实线表示线性加速比，即完美的缩放性能。如图所示，RecLogic_M的推荐模块与 n 的伸缩性很好，例如，当 n 从4到20变化时，在MovieLens、CiaoDVD和Yelp上平均分别快3.07倍、3.42倍和3.35倍。

可伸缩性。固定 $n = 20$ ，我们测试了ParRec在合成图G上的可扩展性，通过使用从0.2到1的比例因子改变G的大小 $|G| = |V| + |E|$ 。我们对比了采用图关联规则进行关联分析的ParGAR[23]。如图6(t)所示，当 $|G|$ 变大时，两种方法所需的时间都更长，正如预期的那样。使用 $|G| = 1B$ ，ParRec在3813.42秒内找到用于推荐的所有用户项对。而当 $|G| = 0.8B$ 时，ParGAR不能在5小时内终止。这验证了为具有双星模式的TIEs量身定制的优化的有效性。

Exp-5:案例研究。我们试图通过使用TIEs的一个变体来解释ML推荐。如例3的 ϕ_5 所示，当一个ML模型M向 x 推荐物品 y 时，我们找到的TIEs使

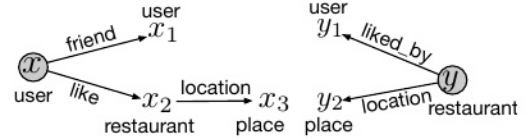


图7:现实生活中领带的图案

同样的决策和carry逻辑前提条件只有 X (没有ML模型)。此外，我们考虑 $\phi = Q[x, y, x](x \rightarrow M(x, y))$ 与这样的 x ，其中 $M(x, y)$ 缩写为 $M(x, y) \geq \delta$ 与预定义的阈值 δ ，并被视为 ϕ 的“结果”。给定 $M(x, y)$ ，发现图模式 Q 和逻辑前提 x 。总之， Q 确定了模型 M 检查的 x 的相关特征， x 揭示了这些特征之间的相关性、相互作用和依赖关系。 Q 和 X 共同提供了 M 为什么做出预测的解释。当 M 基于 gnn 时，这是可能的，因为基于 gnn 的模型并不比带有计数谓词 \geq^k_z 的2变量逻辑更具表现力[10,33]。

例如，我们从Yelp学习这样一个规则 $Q[x, y, x](x \rightarrow M(x, y))$ ，其中 M 是模型KGAT [96]， Q 是图7中的模式， x 是 x_2 。星星 $\leq 2.5 \wedge y. \geq 4.5 \wedge x_1$ 的恒星。 $Id = y_1$ 。 $Id \wedge x_3.zip = y_2.zip \wedge x_2$ 。类别=海鲜和好孩子 \wedge 类别=海鲜和好孩子。它解释了为什么 M 推荐餐厅用户 x, y ，因为(a) y 是高级(4.5星以上),(b) y 是位于靠近餐厅 x_2 相同(邮政编码),和 x 喜欢 x_2 虽然 x_2 评级较低(低于2.5),(c) y 和 x_2 在同一个类别, y (d)是由一个朋友喜欢 x 的 x_1 。 Q 和 x 揭示环境敏感特性(明星、类别和位置)和社会(朋友的偏好)的链接。按照例3的 ϕ_5 ，规则可以写成TIE $Q[x, y, x](x \rightarrow (x, 点赞, y))$ 。

总结。我们发现了以下内容。(1)通过结合逻辑条件，RecLogic不断提高ML推荐模型的准确性。在3个现实生活图上， $F_{FP, FN}$ 、 $F1$ -score和AUC分别在模糊区域 H 的4 ML模型上平均提高了22.89%、22.63%和16.21%，最高达到33.10%、30.29%和29.32%。在top-K推荐的整个区域中，Recall@K和NDCG@K分别提升了12.16%和4.63%。(2)其发现模块比levelwise方法的性能至少提高48.31倍。(3)经过前8轮后，RecLogic学习到的TIEs已经可以将DeepFM的 $F_{FP, FN}$ (从MovieLens的 H 的0.5223)提高到0.6580。(4) RecLogic推荐模块具有并行扩展性;当 n 从4到20变化时，加速比为3.28X。

7 结论

我们提出了一种不需要训练新的ML模型就能提高推荐质量的方法。该工作的新颖性包括:(1)RecLogic，一个丰富现有推荐模型的框架;(2) TIEs预测规则，可以减少具有逻辑条件的ML模型的假阳性和假阴性，并允许PTIME推荐;(3)基于生成的基于概率界的TIEs学习方法;(4)一种PTIME并行可扩展算法，用于使用学习到的TIEs进行推荐。我们的实验研究验证了RecLogic在实践中的应用前景。

未来工作的一个主题是研究近似TIEs和扩展RecLogic以概率预测销售。另一个主题是通过将 M 视为TIEs的结果并发现这些规则的前提条件来解释ML模型 M 的预测。

参考文献

- [1] 2017. CiaoDVD movie ratings. http://konect.cc/networks/librec-ciaodvd-movie_ratings/.
- [2] 2021. Yelp dataset. <https://www.yelp.com/dataset/>.
- [3] 2022. Pytorch. <https://github.com/pytorch/pytorch/tree/v1.8.1>.
- [4] Sajad Ahmadian, Nima Joorabloo, Mahdi Jalili, Majid Meghdadi, Mohsen Afsharchi, and Yongli Ren. 2018. A temporal clustering approach for social recommender systems. In *ASONAM*. IEEE, 1139–1144.
- [5] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.
- [6] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
- [7] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. 2013. Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management* 49, 1 (2013), 13–33.
- [8] John S Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *CoRR arXiv:1301.7363* (2013).
- [9] L Breiman, JH Friedman, R Olshen, and CJ Stone. 1984. *Classification and Regression Trees*. Wadsworth.
- [10] Jin-yi Cai, Martin Fürer, and Neil Immerman. 1992. An optimal lower bound on the number of variables for graph identification. *Comb.* 12, 4 (1992), 389–410.
- [11] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning points and routes to recommend trajectories. In *CIKM*. 2227–2232.
- [12] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *SIGIR*. 335–344.
- [13] Lihan Chen, Sihang Jiang, Jingping Liu, Chao Wang, Sheng Zhang, Chenhao Xie, Jiaqing Liang, Yanghua Xiao, and Rui Song. 2022. Rule mining over knowledge graphs via reinforcement learning. *Knowl. Based Syst.* 242 (2022).
- [14] Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic explainable recommendation based on neural attentive models. In *AAAI*, Vol. 33. 53–60.
- [15] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishu Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *DLRS*. 7–10.
- [16] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. RLogic: Recursive Logical Rule Learning from Knowledge Graphs. In *KDD*. 179–189.
- [17] Wei-Ta Chu and Ya-Lun Tsai. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web* 20, 6 (2017), 1313–1331.
- [18] Amine Dadoun, Raphaël Troncy, Olivier Ratier, and Riccardo Petitti. 2019. Location embeddings for next trip recommendation. In *WWW*. 896–903.
- [19] Gintare Karolina Dziugaite and Daniel M Roy. 2015. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443* (2015).
- [20] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Muyang Liu, Ping Lu, and Chao Tian. 2023. Making It Tractable to Catch Duplicates and Conflicts in Graphs. In *SIGMOD*.
- [21] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Ping Lu, and Chao Tian. 2022. Discovering Association Rules from Big Graphs. *PVLDB* 15, 7 (2022), 1479–1492.
- [22] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. 2018. Discovering Graph Functional Dependencies. In *SIGMOD*. 427–439.
- [23] Wenfei Fan, Ruochun Jin, Muyang Liu, Ping Lu, Chao Tian, and Jingren Zhou. 2020. Capturing Associations in Graphs. *PVLDB* 13, 11 (2020), 1863–1876.
- [24] Wenfei Fan, Ruochun Jin, Ping Lu, Chao Tian, and Ruiqi Xu. 2022. Towards Event Prediction in Temporal Graphs. *PVLDB* 15, 9 (2022), 1861–1874.
- [25] Wenfei Fan, Xin Wang, Yinghui Wu, and Jingbo Xu. 2015. Association Rules with Graph Patterns. *PVLDB* 8, 12 (2015), 1502–1513.
- [26] Wenfei Fan, Yinghui Wu, and Jingbo Xu. 2016. Functional dependencies for graphs. In *SIGMOD*. 1843–1857.
- [27] Yufei Feng, Fuyu Lv, Binbin Hu, Fei Sun, Kun Kuang, Yang Liu, Qingwen Liu, and Wenwu Ou. 2020. MTBRN: Multiplex Target-Behavior Relation Enhanced Network for Click-Through Rate Prediction. In *CIKM*.
- [28] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *Vldb J.* 24, 6 (2015), 707–730.
- [29] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*.
- [30] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*. IEEE, 176–185.
- [31] Michael Garey and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.
- [32] Kostadin Georgiev and Preslav Nakov. 2013. A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines. In *ICML*. PMLR, 1148–1156.
- [33] Martin Grohe. 2020. word2vec, node2vec, graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data. In *PODS*. ACM, 1–16.
- [34] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
- [35] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *CoRR abs/2003.00911* (2020). <https://arxiv.org/abs/2003.00911>
- [36] Mandlenkosi Victor Gwetu, Jules-Raymond Tapamo, and Serestina Viriri. 2019. Random Forests with a Steep Gini-Index Split Function and Feature Coherence Injection. In *MLN*. 255–272.
- [37] Jiayu Han, Lei Zheng, Yuanbo Xu, Bangzuo Zhang, Fuzhen Zhuang, S Yu Philip, and Wanli Zuo. 2019. Adaptive deep modeling of users and items using side information for recommendation. *IEEE TNNLS* 31, 3 (2019), 737–748.
- [38] 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*. 1429–1446.
- [39] 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.
- [40] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.
- [41] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912* (2018).
- [42] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
- [43] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñero Candela. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–9.
- [44] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [45] Wenxing Hong, Ziang Xiong, Jinjie You, Xiaolin Wu, and Min Xia. 2021. CPIN: Comprehensive present-interest network for CTR prediction. *Expert Syst. Appl.* 168 (2021), 114469.
- [46] Mohsen Jamali and Martin Ester. 2009. TrustWalker: A random walk model for combining trust-based and item-based recommendation. In *SIGKDD*. 397–406.
- [47] Buhwan Jeong, Jaewook Lee, and Hyunbo Cho. 2010. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Information Sciences* 180, 5 (2010), 602–612.
- [48] Yogesh Jhamb, Travis Ebesu, and Yi Fang. 2018. Attentive contextual denoising autoencoder for recommendation. In *SIGIR*. 27–34.
- [49] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored item similarity models for top-n recommender systems. In *SIGKDD*. 659–667.
- [50] Saroj Kaushik, Sunita Tiwari, Chhavi Agarwal, and Aakash Goel. 2016. Ubiquitous Crowdsourcing Model for Location Recommender System. *Journal of Computers* 11, 6 (2016), 463–471.
- [51] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *NIPS*, Vol. 30. 3146–3154.
- [52] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *SIGKDD*. 426–434.
- [53] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [54] Clyde P. Kruskal, Larry Rudolph, and Marc Snir. 1990. A Complexity Theory of Efficient Parallel Algorithms. *Theor. Comput. Sci.* 71, 1 (1990), 95–132.
- [55] Maciej Kula. 2015. Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439* (2015).
- [56] Soojung Lee. 2017. Improving jaccard index for measuring similarity in collaborative filtering. In *ICISA*. Springer, 799–806.
- [57] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. 2012. Lars: A location-aware recommender system. In *ICDE*. IEEE, 450–461.
- [58] Jianxun Lian, Xiaohuan Zhou, Fuzhen Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *SIGKDD*. 1754–1763.
- [59] Dawen Liang, Minshu Zhan, and Daniel PW Ellis. 2015. Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks. In *ISMIR*. 295–301.
- [60] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems* 54, 2 (2018), 375–406.
- [61] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.
- [62] Huihui Liu, Yiding Yang, and Xinchao Wang. 2021. Overcoming Catastrophic Forgetting in Graph Neural Networks. In *AAAI*.

- [63] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In SIGKDD. 1831–1839.
- [64] Xiaomeng Liu, Yuanxin Ouyang, Wenge Rong, and Zhang Xiong. 2015. Item category aware conditional restricted boltzmann machine based recommendation. In ICONIP. Springer, 609–616.
- [65] Yung-Yin Lo, Wanjiun Liao, Cheng-Shang Chang, and Ying-Chin Lee. 2017. Temporal matrix factorization for tracking concept drift in individual user preferences. *IEEE Transactions on Computational Social Systems* 5, 1 (2017), 156–168.
- [66] Beth Logan and Ariel Salomon. 2001. A content-based music similarity function. Cambridge Research Labs-Tech Report (2001).
- [67] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook* (2011), 73–105.
- [68] Dave Lucy, RG Aykroyd, and AM Pollard. 2002. Nonparametric calibration for age estimation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 51, 2 (2002).
- [69] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans. Industr. Inform.* 10, 2 (2014), 1273–1284.
- [70] Rui Ma, Ning Liu, Jingsong Yuan, Huaifeng Yang, and Jiandong Zhang. 2022. CAEN: A Hierarchically Attentive Evolution Network for Item-Attribute-Change-Aware Recommendation in the Growing E-commerce Environment. In *RecSys*.
- [71] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *RecSys*. 17–24.
- [72] Paweł Matuszyk, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama. 2015. Forgetting methods for incremental matrix factorization in recommender systems. In *SAC*. 947–953.
- [73] Brian McFee, Luke Barrington, and Gert Lanckriet. 2012. Learning content similarity for music recommendation. *IEEE transactions on audio, speech, and language processing* 20, 8 (2012), 2207–2218.
- [74] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press.
- [75] Karolina Okrasa and Paweł Rżazewski. 2021. Fine-Grained Complexity of the Graph Homomorphism Problem for Bounded-Treewidth Graphs. *SIAM J. Comput.* 50, 2 (2021), 487–508.
- [76] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. 2021. An Embedding-Based Approach to Rule Learning in Knowledge Graphs. *TKDE* 33, 4 (2021), 1348–1359.
- [77] Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. 2018. Robust Discovery of Positive and Negative Rules in Knowledge Bases. In *ICDE*. 1168–1179.
- [78] Christos H. Papadimitriou and Mihalis Yannakakis. 1999. On the Complexity of Database Queries. *JCSS* 58, 3 (1999), 407–427.
- [79] Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2020. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. *CoRR abs/2010.04029* (2020). <https://arxiv.org/abs/2010.04029>
- [80] Dimitrios Rafailidis and Alexandros Nanopoulos. 2015. Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46, 6 (2015), 782–792.
- [81] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *ICML*. 791–798.
- [82] Arindam Sarkar, Dipankar Das, Vivek Sembium, and Prakash Mandayam Comar. 2022. Dual Attentional Higher Order Factorization Machines. In *RecSys*.
- [83] Jan Schluter and Christian Osendorfer. 2011. Music similarity estimation with the mean-covariance restricted Boltzmann machine. In *ICMLA*, Vol. 2. IEEE, 118–123.
- [84] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. 111–112.
- [85] Shilad Sen, Jesse Vig, and John Riedl. 2009. Tagomenders: Connecting users to items through tags. In *WWW*. 671–680.
- [86] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *DLRS*. 11–16.
- [87] Florian Strub and Jeremie Mary. 2015. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *NIPS workshop on machine learning for eCommerce*.
- [88] Huifeng Sun, Yong Peng, Junliang Chen, Chuanchang Liu, and Yuzhuo Sun. 2011. A New Similarity Measure Based on Adjusted Euclidean Distance for Memory-based Collaborative Filtering. *JSW* 6, 6 (2011), 993–1000.
- [89] Zhu Sun, Qing Guo, Jie Yang, Hui Fang, Guibing Guo, Jie Zhang, and Robin Burke. 2019. Research commentary on recommendations with side information: A survey and research directions. *Electronic Commerce Research and Applications* 37 (2019).
- [90] Jiayi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.
- [91] Farman Ullah, Ghulam Sarwar, Sung Chang Lee, Yun Kyung Park, Kyeong Deok Moon, and Jin Tae Kim. 2012. Hybrid recommender system with temporal information. In *ICOIN*. IEEE, 421–425.
- [92] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *NIPS*, Vol. 26. Neural Information Processing Systems Foundation (NIPS).
- [93] Sizzo Verwer and Yingqian Zhang. 2019. Learning Optimal Classification Trees Using a Binary Linear Program Formulation. In *AAAI*. 1625–1632.
- [94] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *WWW*.
- [95] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*. 501–508.
- [96] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *SIGKDD*. 950–958.
- [97] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.
- [98] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *ACM Multimedia*. 627–636.
- [99] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In *SIGIR*.
- [100] Yuanbo Xu, Yongjian Yang, Jiayu Han, En Wang, Fuzhen Zhuang, and Hui Xiong. 2018. Exploiting the sentimental bias between ratings and reviews for enhancing recommendation. In *ICDM*. IEEE, 1356–1361.
- [101] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for Web-scale recommender systems. In *SIGKDD*. 974–983.
- [102] Bingjun Zhang, Jialie Shen, Qiaoliang Xiang, and Ye Wang. 2009. Compositemap: A novel framework for music similarity measure. In *SIGIR*. 403–410.
- [103] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*. 353–362.
- [104] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep Learning based Recommender System: A Survey and New Perspectives. *CoRR abs/1707.07435* (2017).
- [105] Yi Zhen, Wu-Jun Li, and Dit-Yan Yeung. 2009. TagiCoFi: Tag informed collaborative filtering. In *RecSys*. 69–76.
- [106] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. 425–434.
- [107] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *SIGKDD*. 1059–1068.
- [108] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *CIKM*. 406–415.