KGAT: Knowledge Graph Attention Network for Recommendation

Xiang Wang
National University of Singapore
xiangwang@u.nus.edu

Xiangnan He*
University of Science and Technology of China
xiangnanhe@gmail.com

Yixin Cao
National University of Singapore
caoyixin2011@gmail.com

Meng Liu
Shandong University
mengliu.sdu@gmail.com

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

Abstract

为了提供更准确、多样和可解释的推荐，必须超越对用户项交互的建模，并考虑边信息。传统的方法如因子分解机制（FM）将其转化为一个有监督的学习问题，该问题假设每个交互都是一个独立的实例，并对边信息进行编码。由于忽略了实例或项之间的关系（例如，电影的导演也是另一部电影的演员），这些方法不足以从用户的集体行为中提取协作信号。

在这项工作中，我们研究了知识图（KG）的效用，它通过将项目与其属性链接来打破独立交互假设。我们认为，在KG和用户项图的这种混合结构中，高阶关系（将两个项与一个或多个链接属性连接起来）是成功推荐的关键因素。我们提出了一种新的知识图注意网络（Konwledeg Graph Attention Network KGAT）方法，它以端到端的方式显式地对KG中的高阶连接进行建模。它递归地从节点的邻居（可以是users、items或attrs）传播嵌入，以细化节点的嵌入，并使用注意机制来区分邻居的重要性。我们的KGAT在概念上有利于现有的基于KG的推荐方法，这些方法要么通过提取路径来利用高阶关系，要么通过正则化隐式地对它们建模。三个公共基准的实证结果表明，KGAT显著优于Neural FM和RippleNet等最新方法。进一步的研究验证了嵌入传播在高阶关系建模中的有效性以及注意机制带来的可解释性好处。

1 INTRODUCTION

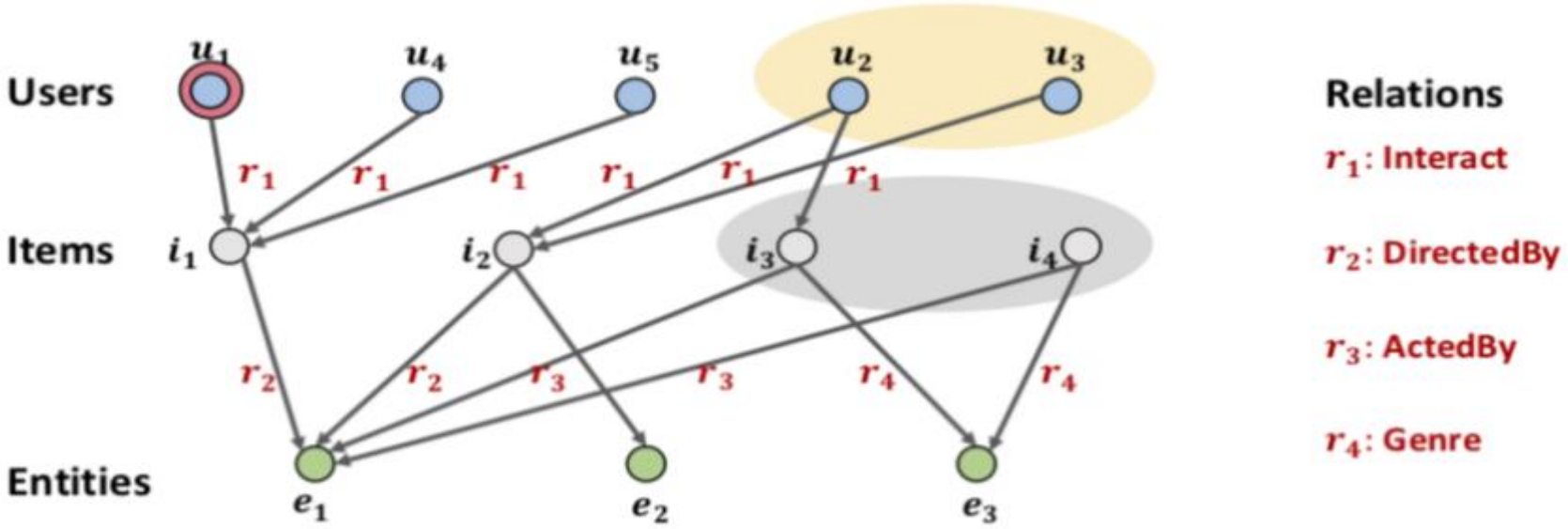
1 Introduction

推荐系统的成功使其在网络应用中大行其道，从搜索引擎、电子商务到社交媒体网站和新闻门户——毫不夸张地说，几乎所有向用户提供内容的服务都配备了推荐系统。为了从用户行为数据的关键（和广泛可用的）来源预测用户偏好，许多研究工作都致力于协同过滤(CF)。尽管CF方法具有有效性和通用性，但它无法对诸如项属性、用户配置文件和上下文之类的边信息进行建模，比如在用户和项很少交互的稀疏情况下性能很差。为了集成这些信息，一个常见的范例是将它们与用户id和项的id一起转换为一个通用的特征向量，并将它们输入到一个监督学习（SL）模型中以预测得分。这种推荐的SL范型已经在工业界广泛应用，一些有代表性的模型包括因子分解机（FM）、NFM（神经FM）、广度和深度和xdeepFM等。

尽管这些方法提供了很强的性能，但其缺点是它们将每个交互建模为一个独立的数据实例，而不考虑它们之间的关系。这使得它们不足以从用户的集体行为中提取基于属性的协作信号。

1 Introduction

如图1所示，用户u1和电影i1之间有一个交互，由person e1导演。CF方法关注同样观看了i1,i.e的相似用户的历史，即u4和u5；而SL方法强调与i1具有相似属性的项e1，i2也具有此属性，显然，这两类信息不仅是推荐信息的补充,同时也在目标用户和项之间形成一种高阶关系。然而，现有的SL方法不能统一起来，不能考虑高阶连通性，如黄圈内观看同一个人e1导演的其他电影的用户，或灰圈内与e1有其他共同关系的项。



黄圈和灰圈表示通过高阶关系发现但被传统方法忽略的重要用户和项。

1 Introduction

为了解决基于特征的SL模型的局限性，提出了一种基于Item边信息图的解决方案。知识图考虑了预测模型的构造。我们将知识图和用户项图的混合结构称为协同知识图（CKG）。如图1所示，成功推荐的关键是充分利用CKG中的高阶关系，例如，长范围连接。

- $u_1 \xrightarrow{r_1} i_1 \xrightarrow{-r_2} e_1 \xrightarrow{r_2} i_2 \xrightarrow{-r_1} \{u_2, u_3\}$,
- $u_1 \xrightarrow{r_1} i_1 \xrightarrow{-r_2} e_1 \xrightarrow{r_3} \{i_3, i_4\}$,

分别表示黄色和灰色圆圈的方式。

然而，要利用这样的高阶信息，挑战是不可忽视的：

1）与目标用户具有高阶关系的节点随着阶数的增加而急剧增加，这给模型带来了计算过载；

2）高阶关系对预测的贡献是不平等的，这就要求模型关心完全称量（或选择）它们。

1 Introduction

最近有几项努力试图利用CKG结构进行推荐，大致可分为两类：基于路径的和基于正则化的：

•基于路径的方法提取携带高阶信息的路径，并将其输入预测模型。为了处理两个节点之间的大量路径，它们要么应用路径选择算法来选择突出的路径，要么定义元路径模式来约束路径。这种两阶段方法的一个问题是，路径选择的第一阶段对最终性能有很大的影响，但并没有针对推荐目标进行优化。此外，定义有效的元路径需要领域知识，对于具有不同类型关系和实体的复杂KG来说，这可能是相当劳动密集的，因为必须定义许多元路径才能保持模型的保真度。

•基于正则化的方法设计额外的损失项，捕捉KG结构，以正则化推荐者模型学习。例如，KTUP和CFKG通过共享项嵌入共同训练推荐和KG完成这两个任务。这些方法没有直接将高阶关系插入到为推荐而优化的模型中，而是以隐式方式对它们进行编码。由于缺乏显式的建模，既不能保证捕捉到长程关联，也不能解释高阶建模的结果。

1 Introduction

考虑到现有解决方案的局限性，我们认为开发一个能够以高效、明确和端到端的方式利用KG中高阶信息的模型至关重要。为此，我们从图形神经网络的最新发展中得到了启发，这些网络具有实现目标的潜力，但是对于基于KG的推荐还没有进行太多的探索。具体地说，我们提出了一种新Knowledge Graph Attention Network（KGAT）方法，它具有两种设计来相应地解决高阶关系建模中的挑战：

1）递归嵌入传播，它基于邻域的嵌入更新节点的嵌入，并递归地执行这种嵌入。在线性时间复杂度下捕获高阶连接的运算；

2）基于注意的聚集，它利用神经注意机制来学习传播过程中每个邻居的权重，这样级联传播的注意权重可以揭示高阶连接的重要性。与基于路径的方法相比，KGAT避免了路径物化的繁琐过程，使用起来更加方便快捷；与基于正则化的方法相比，KGAT直接将高阶关系分解到预测模型中，从而得到了所有相关的预测模型。所有相关参数是为优化推荐目标而定制的。

这项工作的贡献总结如下：

- 我们强调在协作知识图中显式建模高阶关系的重要性，以提供更好的项的边信息推荐。
- 我们开发了一种新的方法KGAT，它在图形神经网络框架下，以一种明确的、端到端的方式实现高阶关系建模。
- 我们在三个公共基准上进行了广泛的实验，证明了KGAT的有效性及其在理解高阶关系重要性方面的可解释性。

2 TASK FORMULATON

2 Task Formulation

首先介绍了CKG的概念，重点介绍了节点间的高阶连通性以及节点间的组成关系。

•用户项二分图：在推荐场景中，我们通常有历史用户项交互（例如，购买和点击）。这里我们将交互数据表示为用户项二分图G1，它被定义为{（u，y_{ui}，i）|u∈U，i∈I）}，其中u和i分别表示用户和项集，并且链接y_{ui}=1表示用户u和项i之间存在观察到的交互；否则y_{ui}=0。

•知识图：除了交互，我们还有项目的边信息（例如，项目属性和外部知识）。通常，这些辅助数据由现实世界中的实体和它们之间的关系组成，以描述一个项它们之间的关系来分析一个item。例如，电影可以由导演、演员和类型来描述。我们以知识图G2的形式组织边信息，G2是由主客体三重事实组成的有向图。形式上，它表示为$\{(h,r,t)|h,t \in \mathcal{E}, r \in \mathcal{R}\}$ 其中每个三元组描述从头部实体h到尾部实体t之间存在关系r。例如，（hugh jackman，actor of，logan）陈述了休jackman是电影logan的演员的事实。注意R包含规范方向（例如actor of）和逆方向（例如acted by）的关系。此外，我们建立了一组项目实体对应的集合 $\mathcal{A} = \{(i,e)|i \in \mathcal{I}, e \in \mathcal{E}\}$ 其中（i，e）表示项i可以与KG中的实体e对齐。

Task Formulation

•协作知识图。这里我们定义了CKG的概念，它将用户行为和项目知识编码为一个统一的关系图。我们首先将每个用户行为表示为三元组（u，interact，i），其中y_{ui}=1表示为用户u和项目i之间的附加关系interact，然后基于项目实体对齐集，用户项目图可以与KG无缝集成为统一图 $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{E}', r \in \mathcal{R}'\}$, $\mathcal{E}' = \mathcal{E} \cup \mathcal{U}$ $\mathcal{R}' = \mathcal{R} \cup \{Interact\}$

任务描述：我们现在制定了本文要解决的推荐任务：

•输入：协作知识图G：包括用户项二部图G1和知识图G2。

•输出：预测用户u采用项目i的概率$\hat{y}_{ui}$ 的预测函数。

高阶连接性。利用高阶连通性是实现高质量推荐的关键。形式上，我们将节点间的L阶连通性定义为一个多跳关系路径：$e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_L} e_L,\ e_l \in \mathcal{E}'\ \ r_l \in \mathcal{R}'$;（$e_{l-1}$，$r_l$，$e_l$）是l阶的三元组，L是序列的长度。为了推断用户偏好，CF方法建立在用户之间的行为相似性基础上——更具体地说，相似的用户在项目上会表现出相似的偏好。这种直觉可以表现为 $u_1 \xrightarrow{r_1} i_1 \xrightarrow{-r_1} u_2 \xrightarrow{r_1} i_2$,这表明u1可能会对i2表现出偏好，因为她的类似用户u2之前采用了i2。与cf方法不同，fm和nfm等SL模型侧重于基于属性的连接，假设用户倾向于采用共享相似属性的项。例如 $u_1 \xrightarrow{r_1} i_1 \xrightarrow{r_2} e_1 \xrightarrow{-r_2} i_2$ 表示u1可能会采用i2因为i2与i1之前喜欢的i1有相同的导演e1。但是，FM和NFM将实体视为单个特征字段的值，无法揭示字段和相关实例之间的关系。例如，很难建模 $u_1 \xrightarrow{r_1} i_1 \xrightarrow{r_2} e_1 \xrightarrow{-r_3} i_2$,尽管e1被认为是连接导演和演员的桥梁。因此，我们认为这些方法并没有充分探索高阶连接性，也没有触及组成的高阶关系。

3 METHODOLOGY

3 Methodology

我们现在提出了建议的KGAT模型，它以端到端的方式利用高阶关系。

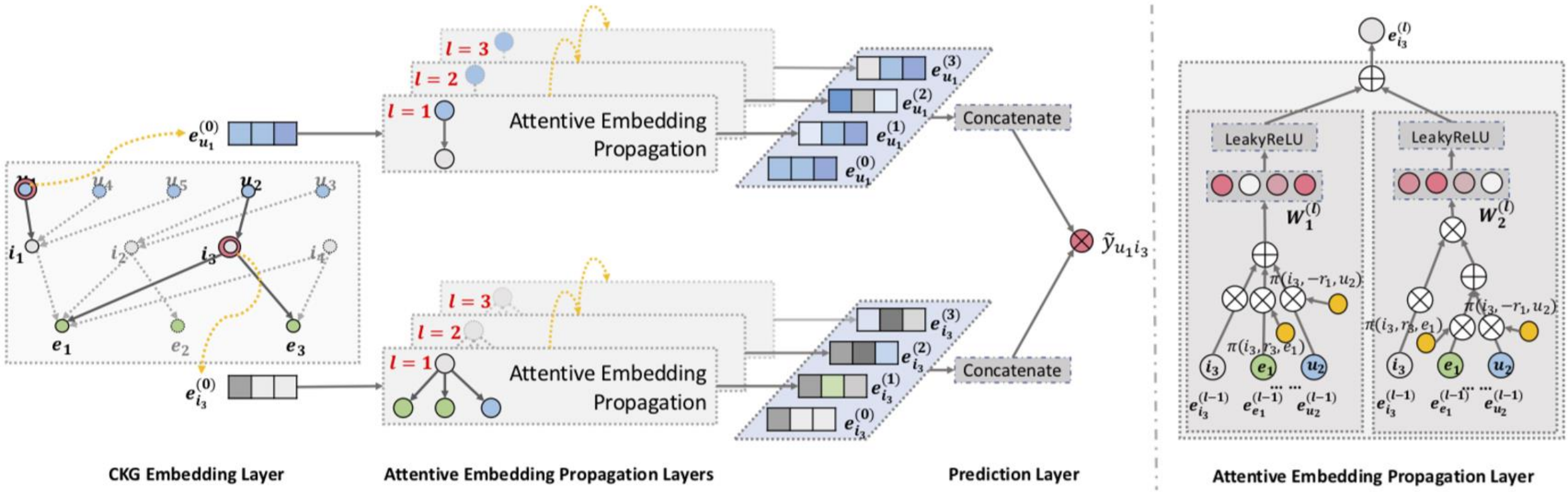
图2显示了模型框架，它由三个主要组件组成：

1）嵌入层，通过保留CKG的结构将每个节点参数化为向量；

2）注意嵌入传播层，递归地从节点的邻居传播嵌入以更新其表示，并使用知识感知注意在传播过程中学习每个邻居的权重的机制；

3）预测层，它从所有传播层聚合用户和项目的表示，并输出预测的匹配得分。

3 Methodology



3.1 嵌 入 层

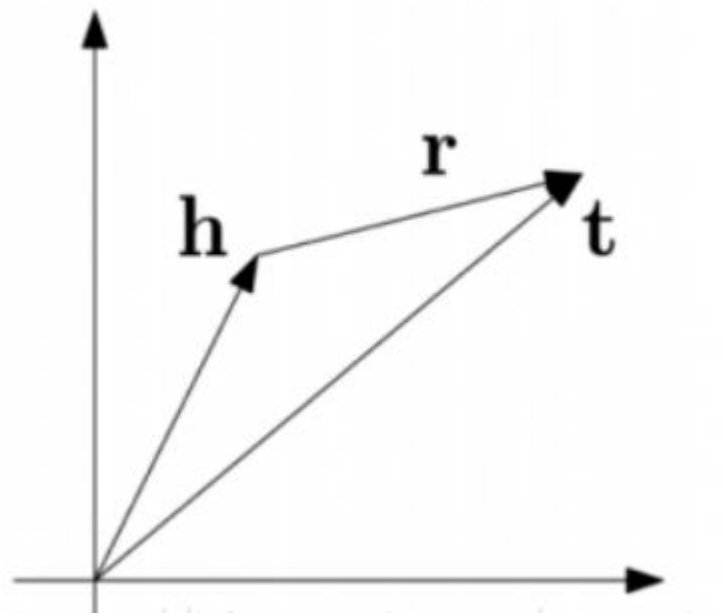
知识图嵌入是一种在保持图结构不变的情况下，将实体和关系参数化为向量表示的有效方法。在这里，我们使用TransR，对CKG是一个广泛使用的方法，更具体地说，如果图中存在一个三元组（h，r，t），它通过优化平移原理 $\mathbf{e}_h^r + \mathbf{e}_r \approx \mathbf{e}_t^r$ 来学习嵌入每个实体和关系。$\mathbf{e}_h, \mathbf{e}_t \in \mathbb{R}^d$ and $\mathbf{e}_r \in \mathbb{R}^k$ 分别是h，t和r的嵌入；$\mathbf{e}_h^r, \mathbf{e}_t^r$ 是关系r空间中eh和et的投影表示。因此，对于给定的三元组（h，r，t），其可信性得分（或能量得分）公式如下

$$g(h,r,t) = \|\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r\mathbf{e}_t\|_2^2,$$

$\mathbf{W}_r \in \mathbb{R}^{k\times d}$ 是关系r的变换矩阵，它将实体从d维实体空间投影到k维关系空间。更低的分数g（h，r，t）表明三元组更可能是真的。

(Trans系列）的知识表示学习

TransE: 多元关系数据嵌入



利用了词向量的平移不变现象。将每个三元组实例 (head, relation, tail) 中的关系 relation 看做从实体 head 到实体 tail 的翻译，通过不断调整h、r和t (head、relation 和 tail 的向量)，使 (h + r) 尽可能与 t 相等，即 h + r ≈ t。

数学上表示就是通过约束 $d(h+l,t) = ||(h+r)-t||_2^2 \approx 0$

来对实体和关系建模，将它们映射到相同的向量空间中。

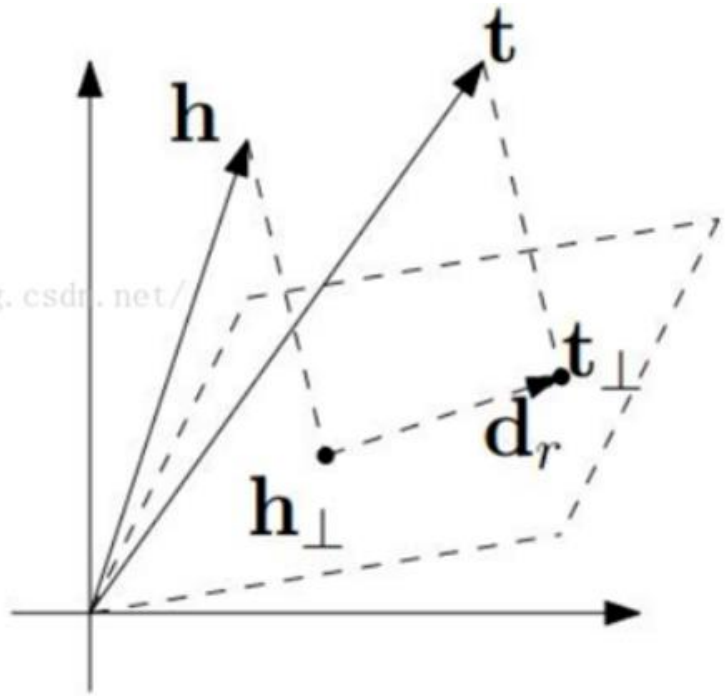
损失函数表示为：

$$\mathcal{L} = \sum_{(h,l,t)\in S} \sum_{(h',l',t')\in S'_{(h,l,t)}} [\gamma + d(h+l,t) - d(h'+l',t')]_+$$

其中，$[x]_+$表示x的正数部分，γ表示margin，$S'_{h,l,t} = \{(h',l,t|h' \in E)\} \cup \{(h,l,t'|t' \in E)\}$

（Trans系列）的知识表示学习

TransH: 将知识嵌入到超平面



TransE中三元组(h,r,t)需要满足 $d(h+r,t) = ||(h+r)-t||_2^2 \approx 0,$

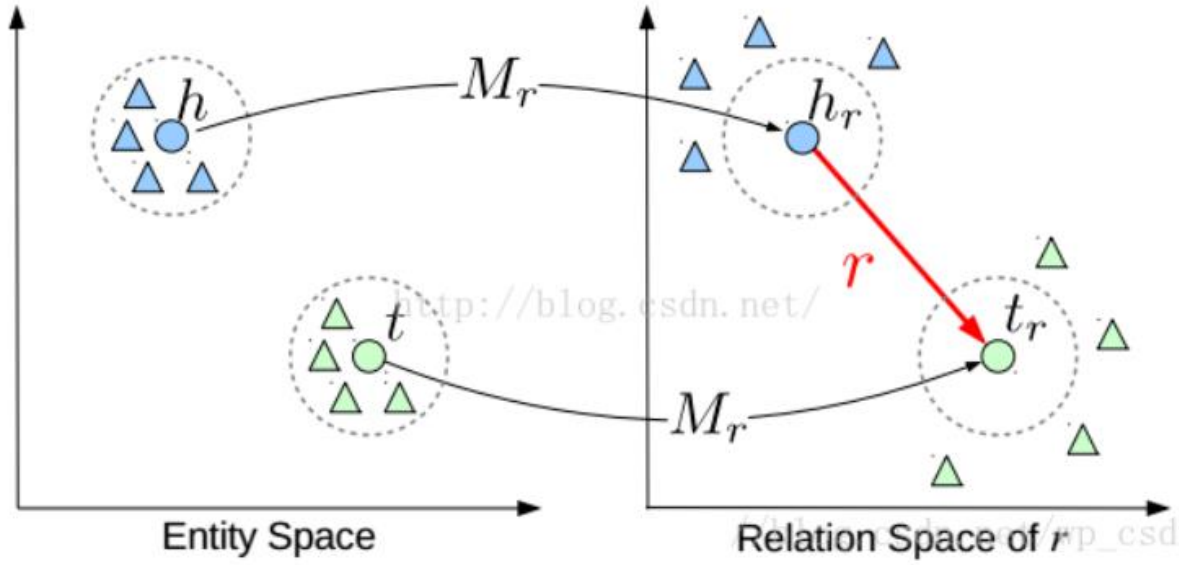
而TransH中三元组(h,r,t) 则需要满足

$$d(h+r,t) = ||(h - w_r^T h w_r) + d_r - (t - w_r^T t w_r)||_2^2 \approx 0,$$

其中$w_r, d_r \in R^k$表示关系。

（Trans系列）的知识表示学习

TransR: 实体和关系分开嵌入

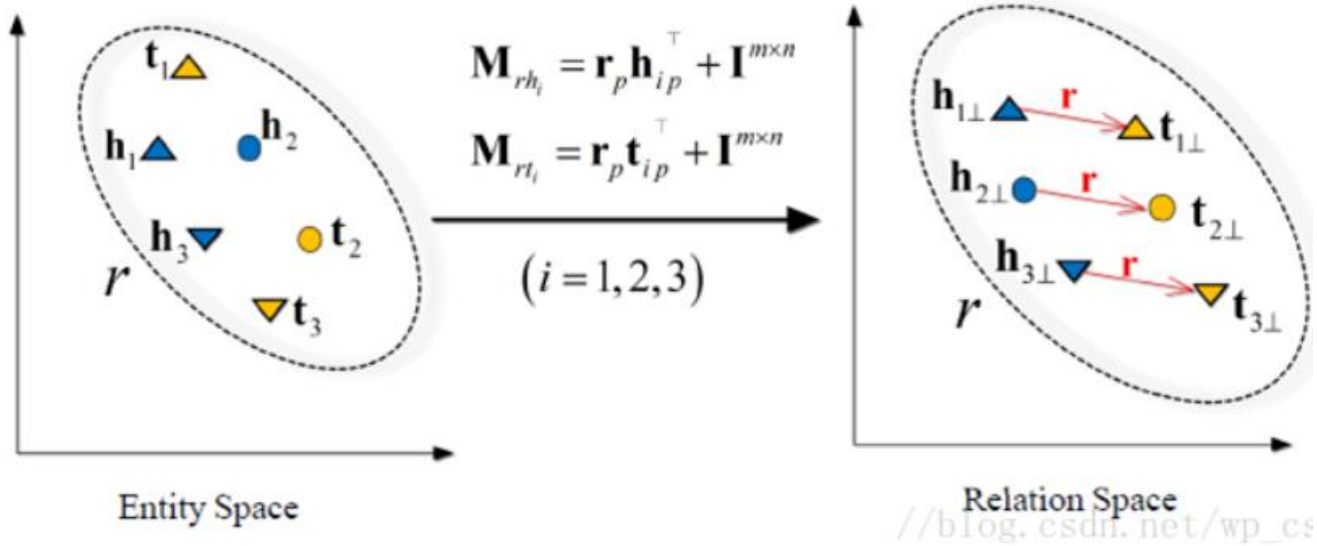


TransR在TranE的基础上的改进，在数学上的描述看起来会更加直观：对于每一类关系，不光有一个向量r来描述它自身，还有一个映射矩阵Mr来描述这个关系所处的关系空间，即对于一个三元组(h，r，t)，需要满足：

$$d(h,r,t) = ||h_r + r - t_r||_2^2 = ||hM_r + r - tM_r||_2^2 \approx 0。$$

（Trans系列）的知识表示学习

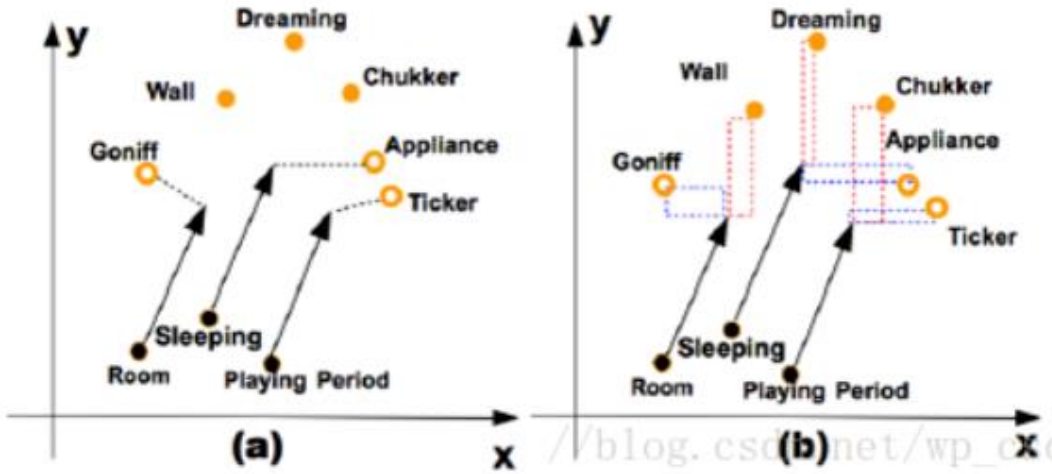
TransD: 通过动态映射矩阵嵌入



TransD在TransR的基础上，将关系的映射矩阵简化为两个向量的积，图中 $M_{rh} = r_p h_p + I^{m*n}$ 与 $M_{rt} = r_v t_v + I^{m*n}$ 表示实体h与实体r映射到关系空间的矩阵，那么对于三元组(h,r,t)，需要满足 $d(h,r,t) = ||M_{rh}h + r - M_{rt}t||_2^2 \approx 0$。

（Trans系列）的知识表示学习

TransA: 自适应的度量函数



TransA模型在TransE的基础上的改进也非常小，简单地说就是给实体/关系的每一个维度都加上了一个权重，增加模型的表示能力。TransE模型的一般形式为：

$$d(h+l,t) = ||(h+r)-t||_2^2 = (h+r-t)^T(h+r-t)$$

TransA对于每一类关系，给实体/向量空间加上了一个权重矩阵Wr，然后可以对权重向量做矩阵分解

$W_r = L_r^T D_r L_r$，最后TransA的数学形式为：

$$d(h+l,t) = (h+r-t)^T W_r (h+r-t) = (L_r|h+r-t|)^T D_r (L_r|h+r-t|))$$

3.1 嵌 入 层

TransR的训练考虑了有效三元组和破损三元组之间的相对顺序，并通过两两排序的损失提高了他们的辨别能力：

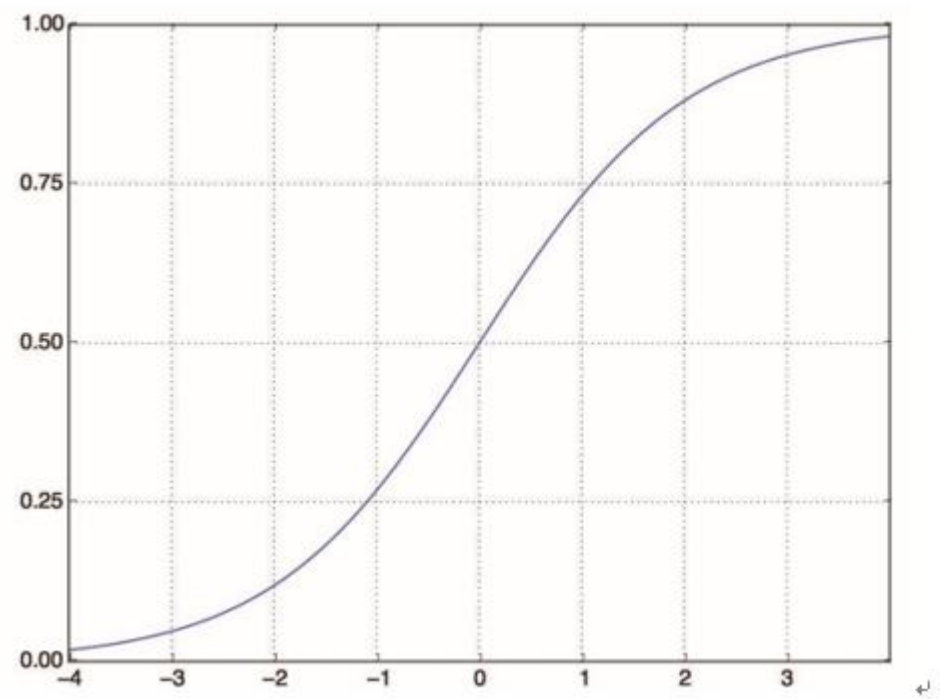
$$\mathcal{L}_{\mathrm{KG}} = \sum_{(h,r,t,t')\in\mathcal{T}} -\ln\sigma\big(g(h,r,t') - g(h,r,t)\big),$$

$$\mathcal{T} = \{(h,r,t,t')|(h,r,t)\in\mathcal{G},(h,r,t')\notin\mathcal{G}\},$$

（h，r，t’）是通过随机替换有效三元组中的一个实体而构造的破三元组；σ（•）是sigmoid函数。该层为实体建模以及三元组的粒度关系，作为正则化子并将直接连接注入到表示中，从而提高模型表示能力（证据见第4.4.3节）。

3.1 嵌 入 层

sigmoid: 将任意值“压缩”到 [0, 1] 区间内，其输出值可以看作概率值。



3.2 注 意 嵌 入 传 播 层

接下来，我们在图卷积网络的基础上，沿着高阶连通性递归地传播嵌入；此外，通过利用图关注网络的思想，我们生成级联传播的注意权重，以揭示这种连通性的重要性。在这里，我们首先描述一个由三个部分组成的单层：信息传播、知识感知注意和信息聚合，然后讨论如何将其推广到多层。

信息传播：一个实体可以参与多个三元组，充当连接两个三元组和传播信息的桥梁，以 $e_1 \xrightarrow{r_2} i_2 \xrightarrow{-r_1} u_2$ 和 $e_2 \xrightarrow{r_3} i_2 \xrightarrow{-r_1} u_2$ 为例：项i2将属性e1和e2作为输入，以丰富其自身特性，然后贡献用户u2的偏好，这可以通过从e1到u2传播信息来模拟。我们基于这种直觉在一个实体和它的邻居之间进行信息传播.

考虑到一个实体h，我们使用Nh={(h，r，t)|(h，r，t)∈G}来表示三元组,其中h是头实体，称为ego-network。为了刻画实体h的一阶连通结构，我们计算了h的ego-network的线性组合：

$$\mathbf{e}_{\mathcal{N}_h} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t,$$

其中π(h，r，t)控制边(h，r，t)上每次传播的衰减因子，指示从t到h传播的信息与关系r的条件.

3.2 注 意 嵌 入 传 播 层

ego Network：又称自我中心网络，网络节点由唯一的一个中心节点（ego），以及这个节点的邻居（alters）组成，边只包括ego与alter之间，以及alter与alter之间的边。

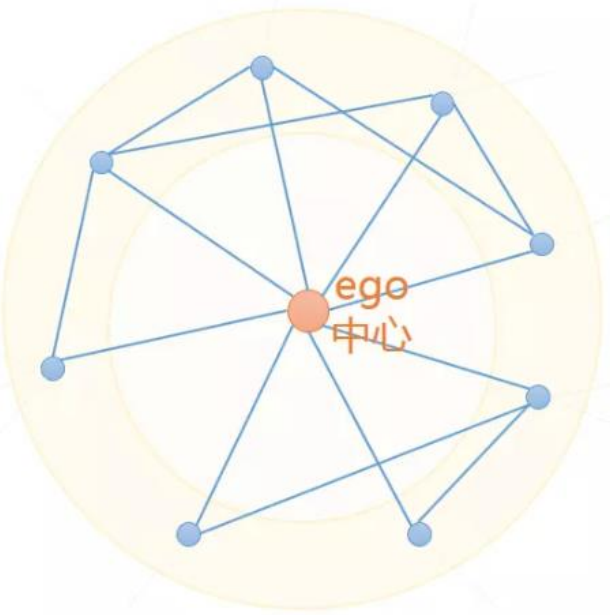


图1：Ego Network示例。橙色节点为中心节点，是ego network的中心，周边的蓝色节点都与此节点直接相连；图中所有节点与他们之间的相互连边（蓝色）便构成了橙色节点的ego network。

3.2 注 意 嵌 入 传 播 层

知识感知注意：通过相关的注意机制实现π（h，r，t），公式如下：

$$\pi(h,r,t) = (\mathbf{W}_r\mathbf{e}_t)^\top \tanh\left((\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r)\right)$$

我们选择tanh作为非线性激活函数。这使得注意力得分依赖于关系r空间中的两个eh和et之间的距离，例如，为更接近的实体传播更多的信息。请注意，为了简单起见，我们只在这些表示上使用内积，并将注意力模块的进一步探索留作以后的工作。

此后，我们通过采用softmax函数对与h相连的所有三元组的系数进行规范化：

$$\pi(h,r,t) = \frac{\exp(\pi(h,r,t))}{\sum_{(h,r',t')\in\mathcal{N}_h}\exp(\pi(h,r',t'))}.$$

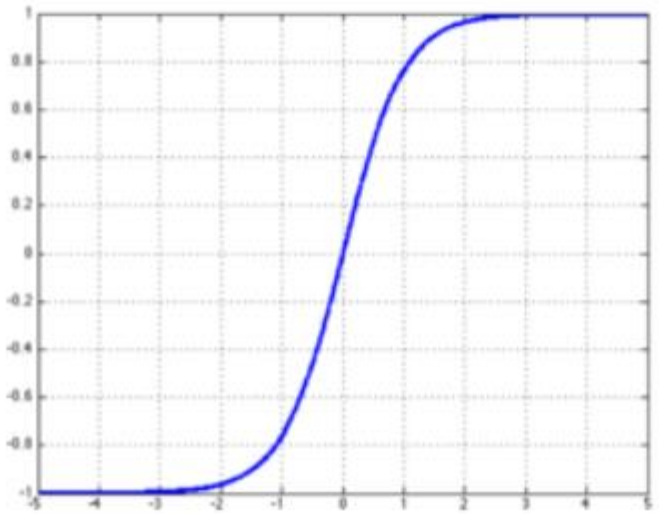
结果，最后的注意得分能够提示哪些邻居节点应该给予更多的注意来捕获协作信号。当执行向前传播时，注意流建议关注部分数据，这可以视为推荐背后的解释。

不同于GCN中的信息传播和GraphSage将两个节点之间的折扣因子设$1/\sqrt{|\mathcal{N}_h||\mathcal{N}_t|}$或者 $1/|\mathcal{N}_t|$，我们的模型不仅利用了图的邻近结构，而且还指定了邻域的不同重要性。此外，与只以节点表示为输入的图注意网络不同，我们在eh和et之间建立了关系er，在传播过程中编码更多信息。实验验证了注意机制的有效性，并分别在第4.4.3节和第4.5节中可视化了注意流。

3.2 注 意 嵌 入 传 播 层

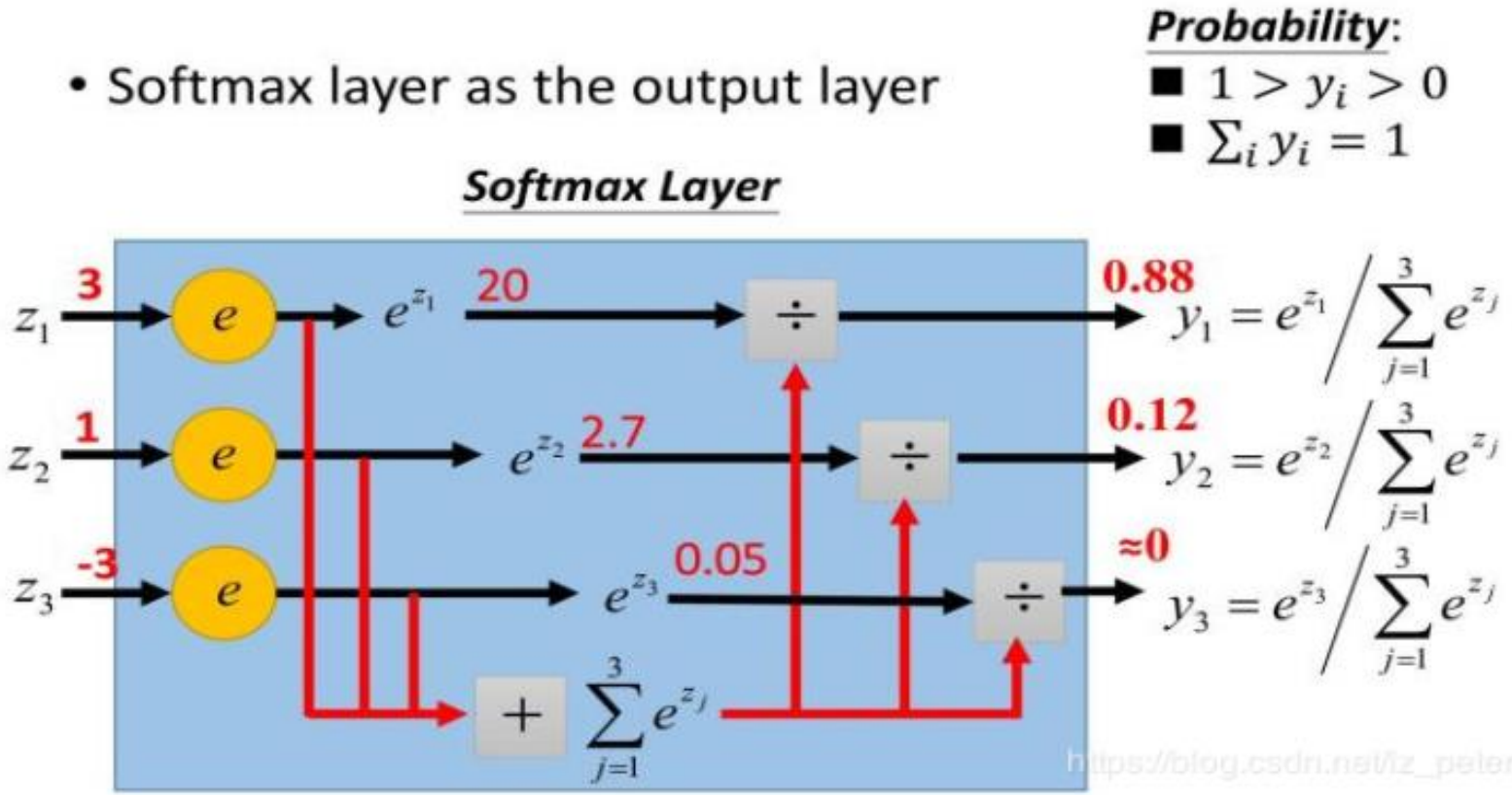
$$tanhx = \frac{sinhx}{coshx} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

其曲线如下图所示：



3.2 注 意 嵌 入 传 播 层

- Softmax layer as the output layer



3.2 注 意 嵌 入 传 播 层

信息聚合：最终阶段是将实体e_h表示和它的ego-network表示e_{Nh}聚合为实体h的新表示-更正式地说 $e_h^{(1)} = f(e_h, e_{\mathcal{N}_h})$

我们使用以下聚合器的三种类型实现f(•)。

- GCN Aggregator:总结了两种表示并应用非线性变换,如下所示：

$$f_{\text{GCN}} = \text{LeakyReLU}\big(\mathbf{W}(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\big),$$

其中，我们将激活函数集设为LeakyReLU；$\mathbf{W} \in \mathbb{R}^{d' \times d}$ 是可训练权重矩阵，以提取有用的传播信息，d’是变换的大小。

- GraphSage Aggregator:连接两个表示，遵循非线性转换。

$$f_{\text{GraphSage}} = \text{LeakyReLU}\big(\mathbf{W}(\mathbf{e}_h || \mathbf{e}_{\mathcal{N}_h})\big),$$

其中||是串联操作。

- Bi-Interaction Aggregator:是由我们精心设计，以考虑e_h和e_{Nh}之间的两种特征交互，如下所示：

$$f_{\text{Bi-Interaction}} = \text{LeakyReLU}\big(\mathbf{W}_1(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\big) + \text{LeakyReLU}\big(\mathbf{W}_2(\mathbf{e}_h \odot \mathbf{e}_{\mathcal{N}_h})\big),$$

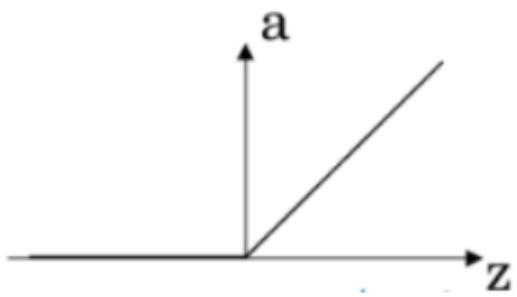
其中，$\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d' \times d}$ 是可训练权重矩阵，且⊙表示按元素划分的乘积。与GCN和GraphSage聚合器不同，我们还对e_h和e_{Nh}之间的特征交互进行编码。这一术语使传播的信息对e_h 和e_{Nh}之间的亲和力敏感，例如，从相似的实体传递更多的信息。总之，嵌入传播层的优点在于显式地利用一阶连接信息来关联用户、项和知识实体表示。我们在第4.4.2节中对三个聚合器进行了经验比较。

3.2 注 意 嵌 入 传 播 层

Relu:

数学表达式：a = max(0, z)

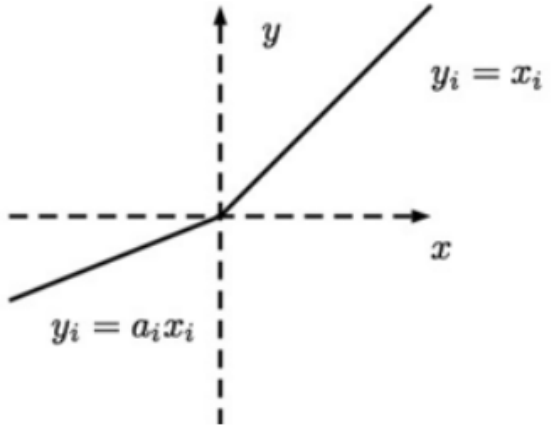
函数图像为：



leakyRelu:

数学表达式：y = max(0, x) + leak*min(0,x)　(leak是一个很小的常数，这样保留了一些负轴的值，使得负轴的信息不会全部丢失)

leakyRelu的图像：



3.2 注 意 嵌 入 传 播 层

高阶传播：我们可以进一步堆叠更多的传播层来探索高阶连接性信息，收集从更高跳邻居传播的信息。更正式地说，在第l步中，我们递归地将实体的表示形式表示为

$$\mathbf{e}_h^{(l)} = f\left(\mathbf{e}_h^{(l-1)}, \mathbf{e}_{\mathcal{N}_h}^{(l-1)}\right),$$

其中，在l阶ego-network中为实体h传播的信息定义如下，

$$\mathbf{e}_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t^{(l-1)},$$

$\mathbf{e}_t^{(l-1)}$ 表示从先前的信息传播步骤生成的实体t，存储来自其(l-1)跳邻居的信息；$\mathbf{e}_h^{(0)}$ 表示为eh的初始信息传播迭代。它进一步有助于实体h在层l上的表示。结果，在嵌入传播过程中可以捕获高阶连接性，如 $u_2 \xrightarrow{r_1} i_2 \xrightarrow{-r_2} e_1 \xrightarrow{r_2} i_1 \xrightarrow{-r_1} u_1$ 。此外，来自u2的信息被显式编码为 $\mathbf{e}_{u_1}^{(3)}$。显然，高阶嵌入传播将基于属性的协作信号无缝地注入到表示学习过程中。

3.3 模 型 预 测

在执行l层之后，我们获得了用户节点u的多个表示，即 $\{\mathbf{e}_u^{(1)}, \cdots, \mathbf{e}_u^{(L)}\}$ ；项节点i与之类似，得到了 $\{\mathbf{e}_i^{(1)}, \cdots, \mathbf{e}_i^{(L)}\}$ 。由于第l层的输出是图l所示的根在u（或i）处的l层深度的树结构消息聚合，不同层的输出强调不同阶的连接信息。因此，我们采用层聚合机制将每一步的表示连接成一个单独的向量，如下所示：

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \| \cdots \| \mathbf{e}_i^{(L)}$$

其中||是串联操作。通过这样做，我们不仅可以通过执行嵌入传播操作来丰富初始嵌入，还可以通过调整L来控制传播强度。

最后，我们对用户和项目表示进行内积，以预测它们的匹配得分：

$$\hat{y}(u, i) = \mathbf{e}_u^{*\top} \mathbf{e}_i^*.$$

3.4 优　化

为了优化推荐模型，我们选择BPR损失。具体地说，它假设观察到的交互（表示更多的用户偏好）应该被分配比未观察到的更高的预测值：

$$\mathcal{L}_{\mathrm{CF}} = \sum_{(u,i,j)\in O} -\ln \sigma\left(\hat{y}(u,i) - \hat{y}(u,j)\right)$$

其中 $O = \{(u,i,j)|(u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$表示训练集，R+表示用户u和项j之间观察到的（正）交互作用，而R-是采样未观察到的（负）交互作用集；σ（•）是sigmoid函数。

最后，我们用目标函数来共同学习方程，如下所示：

$$\mathcal{L}_{\mathrm{KGAT}} = \mathcal{L}_{\mathrm{KG}} + \mathcal{L}_{\mathrm{CF}} + \lambda \|\Theta\|_2^2$$

其中，$\Theta = \{\mathbf{E}, \mathbf{W}_r, \forall l \in \mathcal{R}, \mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}, \forall l \in \{1, \cdots, L\}\}$是模型参数集，E是所有实体和关系的嵌入表；通过在Θ上进行了由λ参数化的L2正则化来防止过拟合。值得指出的是，就模型尺寸而言，大多数模型参数来自实体嵌入（例如，在实验的Amazon数据集上有650万个），这几乎与FM相同；传播层的权重是轻量的（例如，对于三层的塔结构，即64-32-16-8，在亚马逊数据集上）。

3.4.1 训　　练

我们采用最小批量Adam对嵌入损失和预测损失对Lkg和Lcf分别进行优化。Adam是一种应用广泛的优化算法，它能够自适应地控制学习率w.r.t.梯度的绝对值。特别地，对于一批随机抽样的（h，r，t，t’），我们更新所有节点的嵌入；此后，我们随机抽样一批（u，i，j），在传播L步后检索它们的表示，然后使用预测损失的梯度更新模型参数。

3.4.1 训　练

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

第一行是更新step，

第二行是计算梯度，

第三行计算一阶矩的估计，即mean均值

第四行计算二阶距的估计，即variance，和方差类似，都是二阶距的一种。

第五、六行则是对mean和var进行校正，因为mean和var的初始值为0，所以它们会向0偏置，这样处理后会减少这种偏置影响。

第七行是梯度下降。注意alpha后的梯度是用一阶距和二阶距估计的。

3.4.2 时 间 复 杂 度 分 析

由于采用了交替优化策略，时间成本主要来自两部分。对于知识图嵌入（参见方程（2）），平移原理的计算复杂度为 $O(|\mathcal{G}_2|d^2)$。对于注意嵌入传播部分，第1层的矩阵乘法的计算复杂度为 $O(|\mathcal{G}|d_l d_{l-1})$；并且$d_l$和$d_{l-1}$是当前和先前的变换大小。对于最终预测层，只进行内积，整个训练周期的时间代价为 $O(\sum_{l=1}^{L}|\mathcal{G}|d_l)$。最后，KGAT的总体训练复杂度为 $O(|\mathcal{G}_2|d^2 + \sum_{l=1}^{L}|\mathcal{G}|d_l d_{l-1} + |\mathcal{G}|d_l)$.

由于在线服务通常需要实时推荐，推理过程中的计算代价比训练阶段的计算代价更为重要。根据经验，FM、NFM、CFKG、CKE、GCMC、KGAT、MCRec和RippleNet在亚马逊图书数据集上的所有测试实例花费的成本分别为700s、780s、800s、420s、500s、560s、20小时和2小时。如我们所见，KGAT的计算复杂度与SL模型（FM和NFM）和基于正则化的方法（CFKG和CKE）相当，比基于路径的方法（MCRec和RippleNet）效率更高。

4 EXPERIMENTS

4 实　验

我们评估了我们提出的方法，特别是嵌入传播层，在三个真实世界的数据集。我们旨在回答以下研究问题：

- RQ1：与最先进的知识意识推荐方法相比，KGAT的表现如何？
- RQ2：不同的成分（即知识图嵌入、注意机制和聚合器选择）如何影响KGAT？
- RQ3：KGAT能否就用户对物品的偏好提供合理的解释？

4.1 数 据 集 描 述

为了评估KGAT的有效性，我们使用了三个基准数据集：Amazon-book、Last-FM和Yelp2018，这些数据集是可以公开访问的，并且在域、大小和稀疏性方面有所不同。

Amazon-book：亚马逊评论是一个广泛使用的产品推荐数据集。我们从这次收集中选择了Amazon-book。为了确保数据集的质量，我们使用10个核心设置，即保留至少有10个交互的用户和项。

Last-FM：这是从Last.fm在线音乐系统收集的音乐收听数据集。其中，轨迹被视为项。特别是，我们取数据集的子集，其中时间戳是从2015年1月到2015年6月。我们使用相同的10核设置，以确保数据质量。

Yelp2018：该数据集取自Yelp挑战的2018版。在这里，我们将当地的餐馆和酒吧等企业视为项。类似地，我们使用10核设置来确保每个用户和项至少有10个交互。

4.1 数 据 集 描 述

Table 1: Statistics of the datasets.

		Amazon-book	Last-FM	Yelp2018
User-Item Interaction	#Users	70, 679	23, 566	45, 919
	#Items	24, 915	48, 123	45, 538
	#Interactions	847, 733	3, 034, 796	1, 185, 068
Knowledge Graph	#Entities	88, 572	58, 266	90, 961
	#Relations	39	9	42
	#Triplets	2, 557, 746	464, 567	1, 853, 704

4.1 数 据 集 描 述

除了用户-项交互之外，我们还需要为每个数据集构造项的知识。对于Amazon-book和Last-FM，如果有可用的映射，我们通过标题匹配将项映射到Freebase实体中。特别是，我们考虑与与项对齐的实体直接相关的三元组，不管它充当哪个角色（即主体或对象）。与现有只提供项的一跳实体的知识感知数据集不同，我们还考虑了包含项的两跳邻居实体的三元组。对于Yelp2018，我们从本地业务信息网络（如类别、位置和属性）中提取项的知识作为KG数据。为了保证KG的质量，我们通过过滤掉不经常出现的实体（即两个数据集中低于10个）并保留至少50个三元组中出现的关系，对这三个KG部分进行预处理。

我们总结了表1中三个数据集的统计数据，并
https://github.com/xiangwang1223/knowledge_graph_attention_network
上发布了我们的数据集。

对于每个数据集，我们随机选择每个用户80%的交互历史来构成训练集，并将剩余的作为测试集。从训练集中，我们随机选择10%的交互作为验证集来优化超参数。对于每个观察到的用户项交互，我们将其视为一个正实例，然后执行负抽样策略，将其与用户以前未消费的一个负项配对。

4.2 实 验 设 置

4.2.1 评估指标

对于测试集中的每个用户，我们将用户未与之交互的所有项视为负项。然后每种方法输出用户对所有项的偏好得分，除了训练集中的正的项目。为了评估top-K推荐和偏好排序的有效性，我们采用了两个广泛使用的评估协议：recall@k和ndcg@k。默认情况下，我们设置K=20。我们报告测试集中所有用户的平均度量。

4.2.2 基　　线

为了证明其有效性，我们将我们提出的KGAT与SL（FM和NFM）、基于正则化（CFKG和CKE）、基于路径（MCRec和RippleNet）和基于图形神经网络（GC-MC）的方法进行了比较，如下所示：

FM：这是一个贝克马克因子分解模型，其中考虑了输入之间的二阶特征交互作用。在这里，我们将用户、项及其知识（即与之相连的实体）的id作为输入特性。

NFM：该方法是一种最新的因子分解模型，它将FM包含在神经网络中。特别地，我们在输入特征上使用了一个隐藏层。

CKE：这是一种典型的基于正则化的方法，它利用TransR的语义嵌入来增强矩阵分解。

CFKG：该模型在包括用户、项、实体和关系的统一图上应用TransE，将推荐任务作为(u，interact，i)三元组的可信度预测。

MCRec：这是一个基于路径的模型，它提取合格的元路径作为用户和项之间的连接。

RippleNet：这种模型结合了正则化和基于路径的方法，通过在每个用户的根路径中添加项来丰富用户表示。

GC-MC：该模型设计为对图结构数据使用GCN编码器，特别是对用户项二部图。在这里，我们将其应用于用户项知识图。特别地，我们使用了一个图卷积层，其中隐藏维数被设置为嵌入大小。

4.2.3 参 数 设 置

我们在tensorflow中实现了我们的KGAT模型。所有模型的嵌入大小都固定为64，除了RippleNet为16，因为它的计算成本很高。我们使用Adam优化器优化所有模型，其中批大小固定为1024。用默认Xavier初始值设定项来初始化模型参数。我们对超参数应用网格搜索：学习率在{0.05，0.01，0.005，0.001}之间调整，L2正则化的系数在{10-5,10-4,…,101,102}中搜索，对于NFM，GC-MC和KGAT，DROPOUT率在{0.0，0.1，・・・，0.8}中调整。此外，我们还对GC-MC和KGAT采用了节点技术，在{0.0，0.1，・・・，0.8}中搜索比率。对于MCRec，我们手动定义了几种类型的user-item-attribute-item元路径，例如Amazon-book数据集的用户user-book-author-user和用户user-book-genre-user；我们设置了隐藏层，这是一个512、256、128、64维的塔式结构。对于RippleNet，我们分别将跃点数和内存大小设置为2和8。此外，执行早期停止策略，即，如果验证集中的recall@20在50个连续的时间段内没有增加，则过早停止。为了建立三阶连接性模型，我们将KGAT L的深度分别设置为3，隐藏维度分别为64、32和16；我们还在第4.4.1节中报告了层深度的影响。对于每一层，我们进行双交互聚合。

4.3 性 能 比 较

我们首先报告所有方法的性能，然后研究高阶连接性建模如何缓解稀疏性问题。

Table 2: Overall Performance Comparison.

	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
FM	0.1345	0.0886	0.0778	0.1181	0.0627	0.0768
NFM	**0.1366**	0.0913	**0.0829**	0.1214	0.0660	0.0810
CKE	0.1343	0.0885	0.0736	0.1184	0.0657	0.0805
CFKG	0.1142	0.0770	0.0723	0.1143	0.0522	0.0644
MCRec	0.1113	0.0783	-	-	-	-
RippleNet	0.1336	0.0910	0.0791	0.1238	**0.0664**	**0.0822**
GC-MC	0.1316	0.0874	0.0818	**0.1253**	0.0659	0.0790
KGAT	**0.1489***	**0.1006***	**0.0870***	**0.1325***	**0.0712***	**0.0867***
%Improv.	8.95%	10.05%	4.93%	5.77%	7.18%	5.54%

4.3.1 总 体 比 较

性能比较结果见表2。

- 在所有数据集上，KGAT始终获得最佳性能。特别是，KGAT在亚马逊图书、LastFM和Yelp2018中的召回率分别比最强的基线W.R.T.提高了8.95%、4.93%和7.18%。通过叠加多个注意嵌入传播层，KGAT能够以显式的方式探索高阶连通性，从而有效地捕获协同信号。这验证了捕获协同信号传递知识的重要性。此外，与GC-MC相比，KGAT证明了注意机制的有效性，它指定了注意权重w.r.t.组成语义关系，而不是GC-MC中使用的固定权重。
- SL方法（即FM和NFM）在大多数情况下比CFKG和CKE获得更好的性能，这表明基于规则化的方法可能无法充分利用项目知识。特别是，为了丰富一个项的表示，FM和NFM利用其连接实体的嵌入，而CFKG和CKE只使用其对齐实体的嵌入。此外，FM和NFM中的交叉特性实际上充当了用户和实体之间的二阶连接，而CFKG和CKE则在三重粒度上建立了连接模型，使得高阶连接不受影响。

4.3.1 总 体 比 较

- 与FM相比，RippleNet的性能验证了合并两跳相邻项对于丰富用户表示的重要性。因此，它指出了建模高阶连通性或邻域的积极作用。然而，RippleNet在AmazonBook和Last FM中的表现稍逊于NFM，而在Yelp2018中表现更好。一个可能的原因是NFM具有更强的表现力，因为隐藏层允许NFM捕捉用户、项和实体嵌入之间的非线性和复杂的特征交互。
- RippleNet在AmazonBook中的表现远远超过MCRec。一个可能的原因是MCRec在很大程度上依赖于元路径的质量，这需要大量的领域知识来定义。观察结果与[29]一致。
- GC-MC在LastFM和Yelp2018数据集中实现了与RippleNet相当的性能。GC-MC在将高阶连通性引入用户和项目表示时，放弃了节点间的语义关系；而RippleNet则利用关系来指导用户偏好的探索。

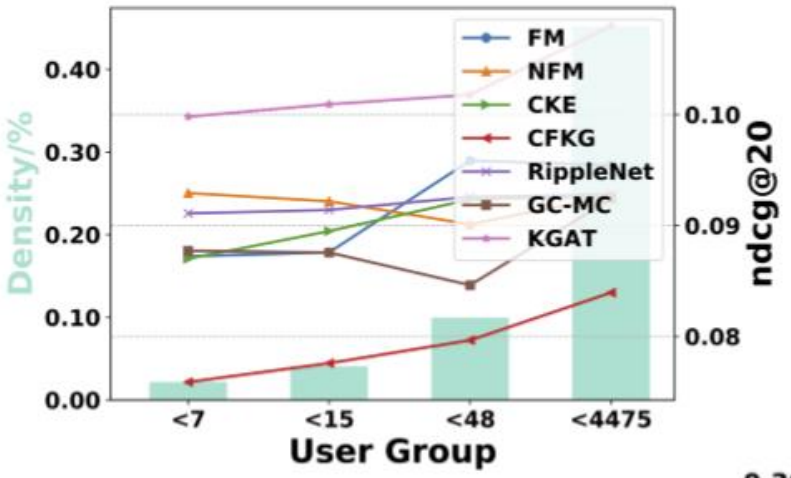
4.3.2 性 能 比 较

性能比较w.r.t.交互稀疏度水平。利用KG的一个动机是缓解稀疏性问题，这通常限制了推荐系统的表达能力。对于交互较少的非活动用户，很难建立最佳表示。在此，我们将研究利用连接信息是否有助于缓解此问题。

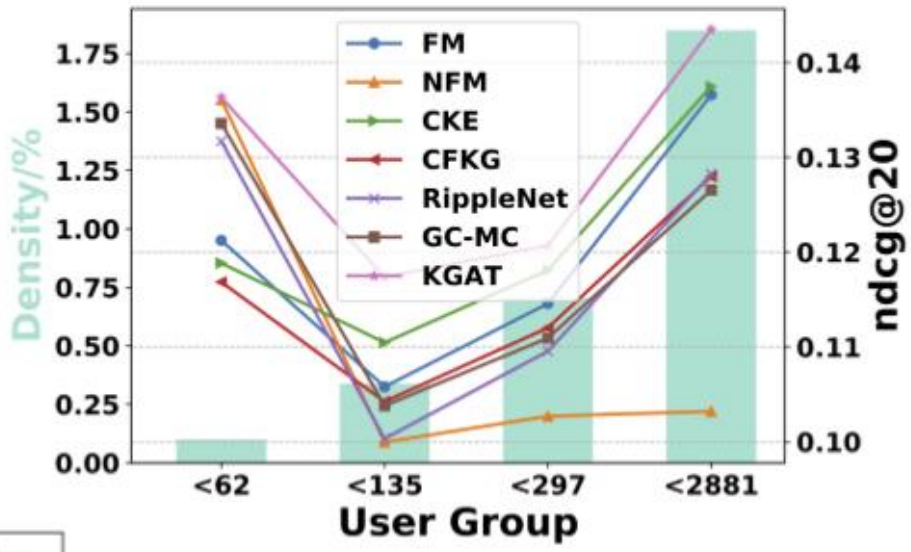
为此，我们对不同稀疏级别的用户组进行了实验。特别是，我们根据每个用户的交互次数将测试集分为四组，同时尝试保持不同的组具有相同的总交互。以amazon图书数据集为例，每个用户的交互次数分别小于7、15、48和4475。图3显示了亚马逊Book、Last FM和Yelp2018中不同用户组的w.r.t.ndcg@20结果。我们可以看到：

- KGAT在大多数情况下都优于其他机型，尤其是在Amazon Book和Yelp2018中最差的两个用户组上。再次验证了高阶连通性建模的意义：1）包含基线中使用的低阶连通性；2）通过递归嵌入传播丰富了非活动用户的表示。
- 值得指出的是，在最密集的用户组（例如Yelp2018的<2057组）中，KGAT稍微优于一些基线。一个可能的原因是，交互过多的用户的偏好过于笼统，无法捕捉。高阶连通性会在用户偏好中引入更多的噪声，从而导致负效应。

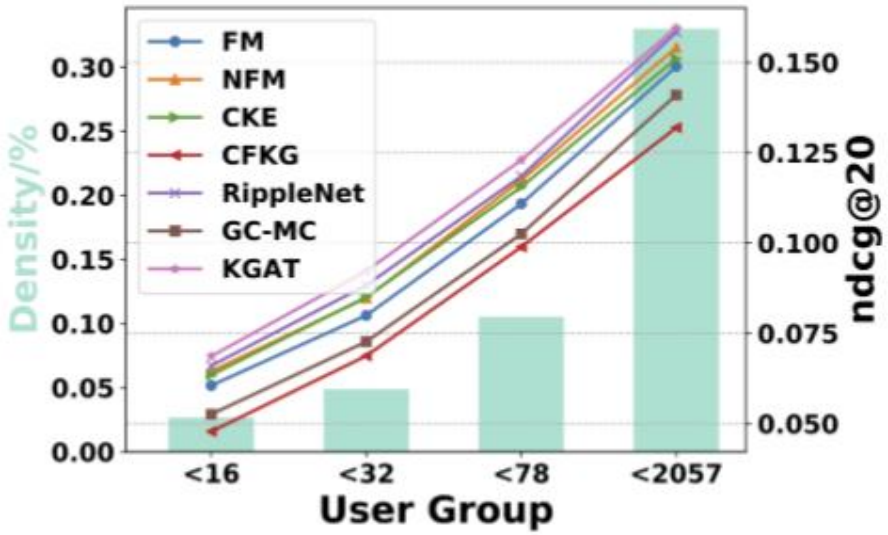
4.3.2 性 能 比 较



(a) ndcg on Amazon-Book



(b) ndcg on Last-FM



(c) ndcg on Yelp2018

4.4 KGAT（RQ2）的研究

为了深入了解KGAT的传播层，我们研究了它的影响。我们首先研究了层数的影响。接下来，我们将探讨不同的聚合器如何影响性能。然后，我们研究了知识图嵌入和注意机制的影响。

Table 3: Effect of embedding propagation layer numbers (L).

	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
KGAT-1	0.1393	0.0948	0.0834	0.1286	0.0693	0.0848
KGAT-2	0.1464	0.1002	0.0863	0.1318	0.0714	**0.0872**
KGAT-3	0.1489	0.1006	0.0870	0.1325	0.0712	0.0867
KGAT-4	**0.1503**	**0.1015**	**0.0871**	**0.1329**	**0.0722**	0.0871

4.4.1 模型深度的影响

我们改变KGAT的深度（例如，L）来研究多个嵌入传播层的使用效率。特别是，层号在{1，2，3，4}的范围内搜索；我们使用KGAT-1来表示具有一个层的模型，并为其他层使用类似的符号。我们总结了表3中的结果，并得出以下结论：

- 增加KGAT的深度能够显著提高性能。显然，KGAT-2和KGAT-3在所有方面都比KGAT-1取得了一致的改进。我们将这些改进归因于用户、项目和实体之间的高阶关系的有效建模，这些关系分别由二阶和三阶连接所承载。
- 在KGAT-3上再叠加一层，我们观察到KGAT-4只取得了边际改善。它表明，考虑实体间的三阶关系就足以捕获协作信号。
- 联合分析表2和表3，KGAT-1在大多数情况下始终优于其他基线。它再次验证了注意嵌入传播的有效性，实证表明它能更好地模拟一阶关系。

4.4.2 聚 合 器 的 作 用

为了探索聚合器的影响，我们考虑使用不同设置的KGAT-1的变体-更具体地说，GCN、GraphSage和Bi-Interaction（参见第3.1节），分别称为$\text{KGAT-1}_{\text{GCN}}$、$\text{KGAT-1}_{\text{GraphSage}}$和$\text{KGAT-1}_{\text{Bi}}$。表4总结了实验结果。我们有以下发现：

- $\text{KGAT-1}_{\text{GCN}}$始终优于$\text{KGAT-1}_{\text{GraphSage}}$。一个可能的原因是GraphSage放弃了实体表示和它的自我网络表示之间的交互作用，从而说明了特征交互在执行信息聚合和传播时的重要性。
- 与$\text{KGAT-1}_{\text{GCN}}$相比，$\text{KGAT-1}_{\text{Bi}}$的性能验证了加入额外的特征交互可以改善表征学习。再次说明了Bi-Interaction聚合器的合理性和有效性。

4.4.2 聚 合 器 的 作 用

Table 4: Effect of aggregators.

Aggregator	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
GCN	0.1381	0.0931	0.0824	0.1278	0.0688	0.0847
GraphSage	0.1372	0.0929	0.0822	0.1268	0.0666	0.0831
Bi-Interaction	**0.1393**	**0.0948**	**0.0834**	**0.1286**	**0.0693**	**0.0848**

4.4.3 知识图嵌入效果和注意力机制

为了验证知识嵌入和注意机制的影响，我们考虑了KGAT-1的三个变体。特别地，我们禁用了KGAT的TransR嵌入成分（参见方程2），我们禁用了注意机制（参见方程4）并且设置$\Pi(h, r, t)$为$1/|N_h|$，称为$KGAT\text{-}1_{w/o}$，同时，我们通过移除两个组件获得另一个变体，称为$KGAT\text{-}1_{w/o\ K\&A}$，我们将实验结果总结在表5中，并得出以下结论：

- 去除知识图嵌入和注意成分会降低模型的性能，这是有意义的，$KGAT\text{-}1_{w/o\ K\&A}$的表现一贯低于$KGAT\text{-}1_{w/o\ KGE}$和$KGAT\text{-}1_{w/o\ Att}$，因为$KGAT\text{-}1_{w/o\ K\&A}$未能在三元组的粒度上显式地建模表示关系。
- 相对于$KGAT\text{-}1_{w/o\ Att}$来说，$KGAT\text{-}1_{w/o\ KGE}$在大部分实例上表现更好。一个可能的原因是，平等地对待所有邻居（即$KGAT\text{-}1_{w/o\ Att}$）可能会引入噪声并误导嵌入传播过程。验证了图形注意机制的实质性影响。

4.4.3 知识图嵌入效果和注意力机制

Table 5: Effect of knowledge graph embedding and attention mechanism.

	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
w/o K&A	0.1367	0.0928	0.0819	0.1252	0.0654	0.0808
w/o KGE	0.1380	0.0933	0.0826	0.1273	0.0664	0.0824
w/o Att	0.1377	0.0930	0.0826	0.1270	0.0657	0.0815

4.5 案 例 研 究

受益于注意机制，我们可以推理高阶连接，以推断用户对目标的偏好，提供解释。对于这一点，我们从Amazon-book中选择了一个用户u208和一个相关项i4293（测试中，未设置训练阶段），基于注意得分，提取连接用户项目对的基于行为和基于属性的高阶连接。图4显示了高阶连接的可视化效果。有两个关键观察结果：

- KGAT捕获基于行为和基于属性的高阶连接，它们在推断用户偏好方面起着关键作用。检索到的路径可以看作是项目满足用户首选项的证据。如我们所见，连接 $u_{208} \xrightarrow{r_0} \textit{Old Man's War} \xrightarrow{r_{14}} \textit{John Scalzi} \xrightarrow{-r_{14}} i_{4293}$ 的注意力得分最高，标记为左侧子图形中的实线。因此，我们可以生成解释，因为您看过老人《战争》是同一作者约翰·斯卡齐写的。
- 项的知识的质量至关重要。如我们所见，实体英语与关系原语包含在同一条路径中，它提供了高质量的解释，这激发了人们对未来工作中的信息过滤问题的关注。

Figure 4: Real Example from Amazon-Book.

5 CONCLUSION AND FUTURE WORK

5 结 论 与 未 来 的 工 作

在这项工作中，我们探讨了知识感知推荐中CKG中语义关系的高阶连接性。我们设计了一个新的框架KGAT，它以端到端的方式显式地模拟CKG中的高阶连接。其核心是关注嵌入传播层，它自适应地从节点的邻居传播嵌入内容，以更新节点的表示。在三个真实数据集上的大量实验证明了KGAT的合理性和有效性。

本文探讨了图神经网络在推荐中的应用潜力，并提出了利用信息传播机制开发结构知识的初步尝试。除了知识图之外，许多其他的结构信息确实存在于现实世界的场景中，例如社交网络和项目上下文。例如，通过将社交网络与CKG整合，我们可以研究社交影响对推荐的影响。另一个令人兴奋的方向是信息传播和决策过程的集成，这为可解释推荐的研究开辟了可能性。

感谢