

利用知识图谱的推荐系统的综述

姓名：霍沛

学号：51184506062

摘要

基于知识图谱的推荐系统通过将用户与用户、用户与项目以及项目与项目间相连接起来，增强数据的语义信息以进一步提高推荐准确度，从而也能在一定程度上解决数据稀疏以及冷启动问题。本文主要对目前一些先进的利用知识图谱特征来提高推荐任务的准确性做一个综述报告，会以知识图谱与推荐任务的学习方式为主线，较为详细的介绍不同学习方式对应的不同的模型。

一. 背景介绍 (Introduction)

在信息爆炸的新媒体时代，各种娱乐、电商平台都迅速发展起来。移动互联网、人工智能等技术的迅速发展为人们的工作生活带来了很多便利，但是同时也带来了信息过载问题。在这其中推荐系统在这些网络在线服务中扮演越来越重要的角色，推荐系统可以兼顾个性化需求和解决信息过载问题。推荐系统是信息过滤系统的一个子集，目的在于根据用户的喜好、习惯、个性化需求以及商品的特性来预测用户对商品的喜好，为用户推荐最合适的商品，帮助用户快速地做出决策，提高用户满意度。

推荐系统可以用于各种互联网平台，可以为用户推荐商品，电影，新闻，书籍等一系列的信息。关于推荐方面的技术也是层出不穷，在各种推荐任务中由于用户和项目信息缺失，这就会带来数据稀疏的问题，缺少对用户与用户、项目与项目等之间的相互联系的考虑。基于知识图谱的推荐系统通过将用户与用户、用户与项目以及项目与项目之间相互连接起来，增强数据的语义信息以进一步提高推荐准确度，具有重要的研究意义和实用价值。将知识图谱作为辅助信息引入到推荐系统中可以有效地解决传统推荐系统存在的稀疏性和冷启动问题。近年来很多关于这方面的研究，因此本文主要是对如何应用知识图谱或者外部知识来完成推荐任务的综述。基于知识图谱的推荐系统主要是利用知识图谱对多源异构数据的整合性，得到更加粒度的用户和项目的特征信息，从而更精准的计算用户与用户、用户与项目以及项目与项目之间的相关性，最后为用户做出推荐目前把知识图谱的特征学习应用到推荐系统中主要通过三种方式，分别是依次学习、联合学

习还有交替学习。

依次学习 (one-by-one learning) 的通常做法是首先使用知识图谱特征学习得到实体向量和关系向量,学习的方法一般是 Trans 系列,然后将这些低维特征向量引入推荐系统,从而能够够好的学习推荐任务中的用户向量以及物品向量,典型的代表是[1]。这篇文章通过把新闻标题中出现的实体链接到知识图谱中,基于知识图谱学习实体的特征表示,然后应用于后续的新闻推荐任务中。联合学习区别于依次学习是将知识图谱的特征学习和推荐算法的目标函数结合,使用端到端的方法进行联合学习。以[2]和[3]为典型代表,其中[2] 是一个基于协同过滤和知识图谱特征学习的推荐系统,[3]模拟了用户兴趣在知识图谱上的传播过程,正如论文标题水波一样,以一个用户感兴趣的实体为中心,在知识图谱上向外层逐层扩散。交替学习是将知识图谱特征学习和推荐算法视为两个分离但又相关的任务,使用多任务学习 (multi-task learning) 的框架进行交替学习。典型的代表是[4],这篇文章采用多任务的学习框架,由于推荐系统中的物品和知识图谱中的实体存在着重合的部分,因此两个任务存在着相关性,两者的信息可以达到互补的作用,后续会对以上内容详细介绍。

二. 相关工作 (Related work)

知识图谱的提出[11]最早是为了优化搜索引擎,改善用户的搜索质量以及搜索体验。知识图谱旨在描述真实世界中存在的各种实体或概念及其关系,其构成一张巨大的语义网络图,节点表示实体或概念,边则由属性或关系构成。后来知识图谱已被广泛应用于智能搜索、智能问答、个性化推荐、内容分发等领域。本文主要研究将知识图谱应用于推荐技术中。想要在推荐任务中应用知识图谱很大程度上是为了获取他在知识图谱中的特征表示,那么如何学习知识图谱中的实体的特征表示是一个很大的研究点。近年来关于这方面的研究也是层出不穷,最典型的的就是 Trans 系列的文章了。最早的是 TransE[5],它的提出解决了多关系数据的处理问题,将每个三元组对中的关系看作是从实体 head 到实体 tail 的翻译过程,本质是一种向量相加的运算。TransE 更是作为知识库向量化的基础,衍生出来了很多变体。TransH[6]为了解决图谱中一对多,多对一,多对多建模的问题,所以为了平衡模型复杂度和建模效果,它将把关系的表示映射到一个超平面训练

目标和 TransE 基本一致。TransR[7]把实体映射到一个空间，关系映射到另外一个空间来解决多对多的问题，但是引来一个参数多大的问题，因此就有了 TransD[8]通过动态矩阵相乘的操作极大的减少了训练参数量，同时也获得了更好的知识图谱的特征表示。

Niaraki [9]较早地提出基于本体的推荐系统以用户模型和上下文模型为基础构建本体知识库，基于本体中概念的层级关系,来完成以用户为中心的推荐任务。深度学习在不同的应用领域取得了很大的成功，它已经被证明是人工智能领域中一项改变游戏规则的技术，其中将其用于解决推荐问题。Bellini[10]利用知识感知的自编码器完成评级预测的任务，取得了很好的效果。[12]提出了一种新的知识感知模型，通过组合实体和关系的语义来生成路径表示，利用 LSTM 来建模路径中顺序依赖关系，把推荐学习任务视为二元分类问题。Hongwei 等[1]提出的利用深度学习的知识感知的推荐系统，通过卷积网络来融合文本的语义层面、实体层面上的异构表示，使用了注意力机制对用户的兴趣进行动态提取，解决了新闻推荐任务中实时性和较多实体的问题。Zhang[3]使用知识化结构学习，文本知识学习，以及图像知识学习，将三种知识学习的目标函数与推荐系统中的协同过滤结合。Wang[3]提出了 Ripple Network 模拟了用户兴趣在知识图谱上的传播过程，整个过程类似于水波的传播。[4]这篇文章采用多任务的学习框架，由于推荐系统中的物品和知识图谱中的实体存在着重合的部分，因此两个任务存在着相关性。

在知识图谱中，实体中包含着丰富的描述性知识，而这些知识恰好可以为推荐结果提供精确性、多样性与可解释性。利用知识图谱进行推荐面临一些挑战：如何有效利用多源异构的知识图谱数据为用户偏好精准化的建模，如何把深度学习和知识表示进行融合来为用户更好的推荐。知识图谱中，用户之间、用户与物品之间信息互动产生大规模的数据集；将这些数据集融入知识图谱后会急剧增加计算复杂度。如何将用户之间产生的数据动态、快速地融入知识图谱并提高推荐系统的可扩展性，是要解决的难点。后续第三节主要以各种学习方式为主线依次介绍涉及到的模型细节，以及对这些模型的对比介绍，第四节是对整个文章的总结以及对未来工作的看法。

三、 代表模型介绍（Model Details）

本节主要以知识图谱和推荐任务相结合的三种学习方式（依次学习，联合学习，交替学习）为主线，对应各种学习方式，依次介绍当前效果先进的几种模型，

看这些模型是如何解决推荐任务中存在的问题。依次学习简单来说就是先在知识图谱中学习推荐任务中涉及到的物品的特征表示，然后将这种表示引入到后续的推荐任务中，在 3.1 小节中将介绍这方面的论文；3.2 中介绍的模型是一种联合学习的思想，这种方式将知识图谱的特征学习和推荐算法的目标函数相结合，利用端到端的技术来完成联合学习的任务；交替学习则是将知识图谱特征学习和推荐算法视为两个分离但又相关的任务，使用多任务学习（multi-task learning）的框架进行交替学习，在 3.3 中将介绍这方面的模型。

3.1 Deep Knowledge-Aware Network for News Recommendation

[1]这篇文章的提出是为了解决在新闻推荐任务中实时推荐以及新闻内容中出现很多的实体名词，作者提出的 DKN 模型把在知识图谱中学到的表示融合进去，来完成点击预测的任务。整个模型结构如图 1 所示：模型的输入是一系列作者曾经点击过的新闻标题，和一系列的候选新闻，模型的输出则是作者点击该新闻的概率。作者利用 KCNN 网络得到每条新闻的特征表示，利用然后利用 attention 网络去计算当前候选新闻标题和历史标题的相关权重，然后对历史新闻标题进行加权拼接，得到用户最终的向量表示，最终把用户的向量表示和当前候选新闻的向量拼接输入到 MLP 中，得到最终的概率。

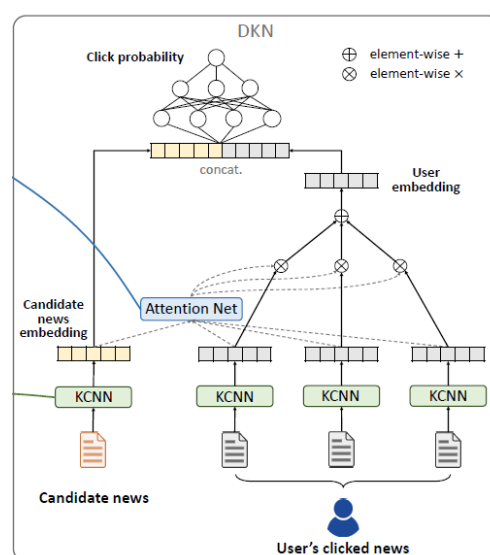


图 1

本文的关键核心在于 KCNN 网络的设计，以及如何把知识图谱的特征向量融入任

务中，在介绍 KCNN 之前，首先要介绍作者提出的知识蒸馏的技术。知识蒸馏的过程是：首先通过实体链接的技术来找到知识库中对应的实体，然后选取一跳范围内的所有边和节点构建子图，接下来利用知识图谱中向量表示方法来获得每个实体的表示，然后找到周围该实体周围实体的表示通过取平均的方式获得该实体的上下文表示 \bar{e} ，可参加公式（1），（2）。

$$\text{context}(e) = \{e_i | (e, r, e_i) \in \mathcal{G} \text{ or } (e_i, r, e) \in \mathcal{G}\} \quad (1)$$

$$\bar{e} = \frac{1}{|\text{context}(e)|} \sum_{e_i \in \text{context}(e)} e_i \quad (2)$$

作者提出 KCNN（Knowledge-aware CNN），把知识表示和句子表示融合起来学到句子级别的向量表示，KCNN 结构图如图 2 所示。假定有一个长度为 n 的新闻标题， $\mathbf{w}_{1:n} = [\mathbf{w}_1 \mathbf{w}_2 \dots \mathbf{w}_n] \in \mathbb{R}^{d \times n}$ ，经过知识蒸馏后得到的这个句子中实体词的向量表示 $e_i \in \mathbb{R}^{k \times 1}$ ，以及对应的实体上下文表示 $\bar{e}_i \in \mathbb{R}^{k \times 1}$ ，之后利用了一种多通道的思想，利用公式（3）分别得到句子中每个词语的实体向量表示，和上下文向量表示，如果不是实体词就用 $\mathbf{0}$ 向量代替即可，最终得到句子的另外两种表示，见公式（4），（5）。

$$g(e) = \tanh(Me + b) \quad (3)$$

$$g(\mathbf{e}_{1:n}) = [g(e_1)g(e_2) \dots g(e_n)] \quad (4)$$

$$g(\bar{\mathbf{e}}_{1:n}) = [g(\bar{e}_1)g(\bar{e}_2) \dots g(\bar{e}_n)] \quad (5)$$

KCNN 的输入就是句子三个通道的表示（6），借鉴学习图像特征的学习过程，利用 CNN 网络来得到，最终整个句子的表示。

$$\mathbf{W} = [[w_1 g(e_1)g(\bar{e}_1)][w_2 g(e_2)g(\bar{e}_2)] \dots [w_n g(e_n)g(\bar{e}_n)]] \in \mathbb{R}^{d \times n \times 3} \quad (6)$$

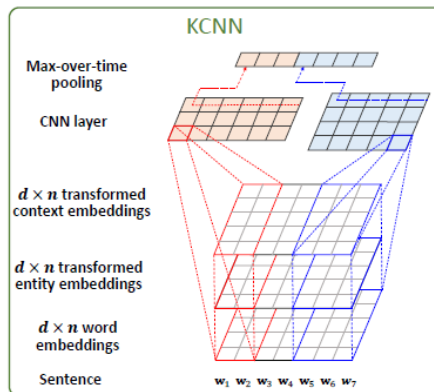


图 2

在得到一系列用户点击过的历史新闻标题后如何与当前候选的新闻标题建立联系呢，这里作者使用了 **attention** 网络，如图 3 所示。从图中也能看出作者把通过 **KCNN** 网络得到的候选新闻的向量和历史新闻向量拼接输入一个简单的全连接网络，得到这两者之间的权重，这也是一个用户兴趣点特征提取的过程，见公式 (7)，(8)，其中 \mathcal{H} 代表 **attention** 网络。

$$s_{t_k^i, t_j} = \text{softmax}(\mathcal{H}(\mathbf{e}(t_k^i), \mathbf{e}(t_j))) = \frac{\exp(\mathcal{H}(\mathbf{e}(t_k^i), \mathbf{e}(t_j)))}{\sum_{k=1}^{N_i} \exp(\mathcal{H}(\mathbf{e}(t_k^i), \mathbf{e}(t_j)))} \quad (7)$$

$$\mathbf{e}(i) = \sum_{k=1}^{N_i} s_{t_k^i, t_j} \mathbf{e}(t_k^i) \quad (8)$$

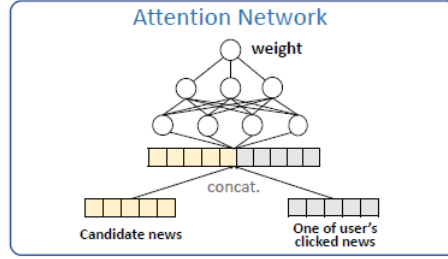


图 3

作者作了一系列的实验证明上述的模型取得了很好的效果。

3.2 Collaborative Knowledge Base Embedding for Recommender Systems

[2]这篇文章在推荐系统中引入了结构信息，文本数据，图像数据等知识库中的信息来提升了推荐系统的质量其中，结构信息采用 **TransR** 来得到实体的向量特征，文本数据与图像数据分别使用栈式降噪自编码 (**Stacked Denoising Auto-encoders, SDAE**) 与栈式卷积自编码 (**Stacked Convolutional Auto-encoders**) 来提取出向量特征。

协同过滤是一种典型的推荐算法，但是存在稀疏问题，无法对用户推荐新的项目。考虑到用户对于 **item** 的选择也会受到与 **item** 相关的其他信息的影响。因此，混合了协同过滤与其他信息的推荐系统通常可以更好地解决这些问题，从而得到更好的结果。知识库中通常包含了大量的信息，因此有部分研究工作将知识库作为其他信息来与协同过滤进行融合。这方面的工作存在一些问题：仅仅使用单一的网络结构信息，从知识库中特征提取依赖于传统的特征工程，为解决这些

问题，作者提出了 CKE（Collaborative Knowledge Base Embedding）推荐系统框架。主要任务是训练一个模型，给定具有结构知识，文本及图片的知识库还有用户的隐式反馈，返回用户最可能感兴趣的项目列表。这边考虑结构信息的原因是作者考虑到知识图谱结构中结构信息暗示了实体或项目之间的相似性，而文本知识指的是比如对于知识库中的书籍或者电影这样的条目，可以使用文本摘要来表示知识，文本知识通常给出这本书或者某部电影的主要主题，同样的道理除了文本描述外某些实体在知识库中还会有对应的图像，比如书籍的封面或者是电影的海报知识。用户的隐式反馈和结构知识是项目的结构特征而文本知识和视觉知识则是项目的内容特征，可见图 4 形象展示。

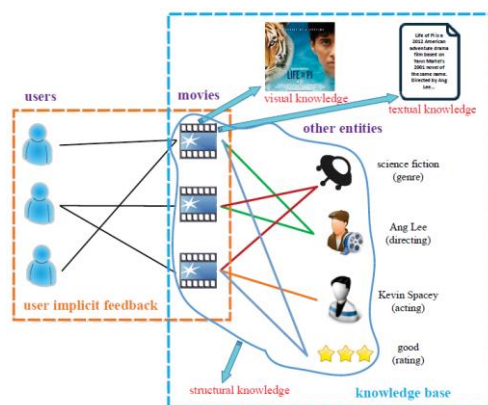


图 4

模型框架如图 5 所示，首先模型分别针对于知识库中存储的结构知识（structural knowledge），文本知识（textual knowledge），图片知识（visual knowledge）进行向量化表示。embedding 的方式分别为 Bayesian TransR[], Bayesian SDAE[14], Bayesian SCAE 模型主要分为两大部分：知识向量学习和协同联合学习（这也是和 3.1 介绍的模型的最大的不同之处）。

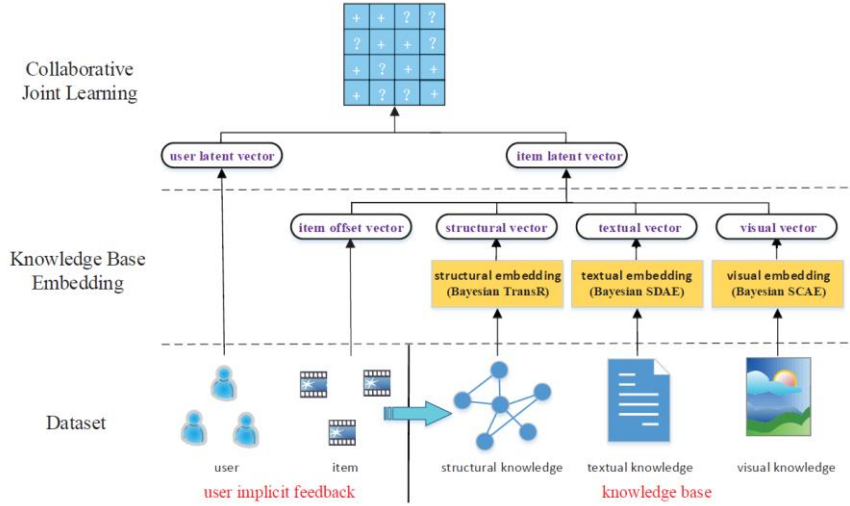


图 4

为了把从知识库中提取的 item embedding 融合到协同过滤中去。项目的潜在向量表示如下公式 (9)

$$\mathbf{e}_j = \boldsymbol{\eta}_j + \mathbf{v}_j + \mathbf{X}_{\frac{L_t}{2}, j*} + \mathbf{Z}_{\frac{L_v}{2}, j*} \quad (9)$$

其中 η 代表协同过滤中的物品的隐含向量。其他的三个分别代表从知识库中提取出来的特征。然后使用 pair-wise 的形式

$$p(j > j'; i | \theta) = \sigma(\mathbf{u}_i^T \mathbf{e}_j - \mathbf{u}_i^T \mathbf{e}_{j'}) \quad (10)$$

这个式子的意思是，对于用户 i 来说，相比 j' 来说，更喜欢 j 物品。 θ 表示模型参数。

3.3 RippleNet

在 3.1 节中 DKN 中，我们需要首先学习到 entity 的向量和 relation 的向量，但是学习到的向量，其目的是为了还原知识图谱中的三元组关系，而并非是为了我们的推荐任务而学习的，这个模型和上一个模型都是一种联合训练的网络结构。

Ripple 中文意思是波纹，相当于向水里投了一枚石头，石头就是波纹的中心，放在推荐系统中中心就是用户的历史点击矩阵，被激起的波纹就对应了知识图谱中一个又一个的实体。隐身含义就是由用户的历史点击记录激起的水波，就是用户潜在感兴趣的项目。除此之外，波纹还有一个特点，它会随着层数的变大而逐渐衰减，反映在项目中就是用户的兴趣随着跳数的增加逐渐递减，反映在下图 5 颜色的深度逐渐变浅。

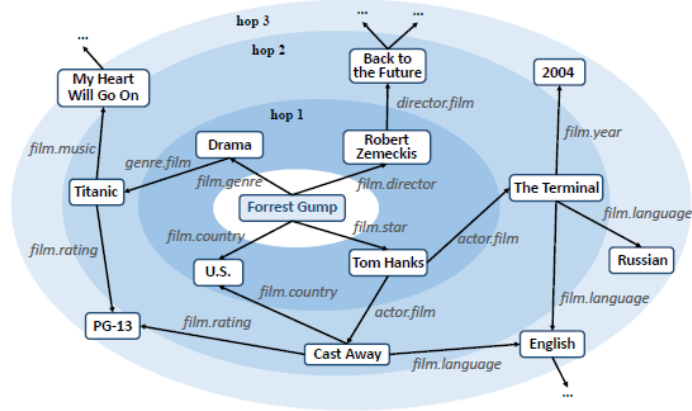


图 5

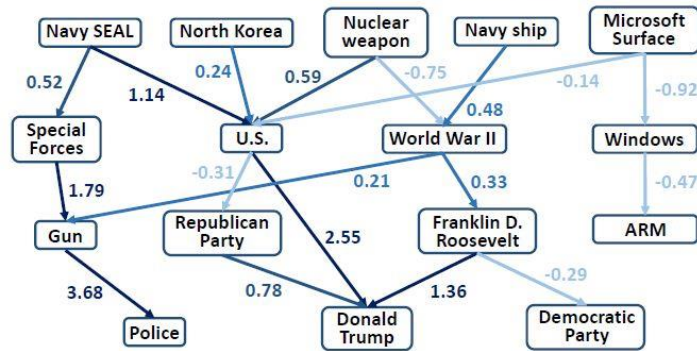
整个模型的输入是用户历史点击记录和一个项目，输出是用户点击该项目的概率，就是一个预测矩阵。最终想要得到预测矩阵，其实就是要比较 **user embedding**(特征)与 **item embedding** 的相似程度，可以用 $u^T v$ ，其中 u 是用户矩阵, v 是项目矩阵，那么现在的关键如何得到这两矩阵。通过 **embedding** 层可以直接得到项目矩阵，而用户矩阵就比较复杂了，需要通过用户历史点击记录，知识图谱以及 **item embedding matrix** 共同参与得到，这也是联合学习的体现。首先构建与用户 u 相关的前 k 跳的 **item** 的集合（知识图谱中以初始的 **item** 向外扩展），这些 **item** 实质上用户兴趣偏好的一种体现。然后根据 **embedding** 向量内积，计算候选 **item** i 和每一层 **hop** 上的 **head item** 的归一化相似度见公式(11)；然后根据相似度，对尾实体加权求和，作为这一层 **hop** 的输出（本质上，属于 **Attention**），重复上述过程 k 次，这里的 k 就是知识图谱最大波及的跳数。

$$p_i = \text{softmax}(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i) = \frac{\exp(\mathbf{v}^T \mathbf{R}_i \mathbf{h}_i)}{\sum_{(h,r,t) \in S_u^1} \exp(\mathbf{v}^T \mathbf{R} \mathbf{h})} \quad (11)$$

最后将所有 k **hop** 的输出向量相加，作为 **user** 的 **Embedding**，与 **item** 的 **Embedding** 内积计算最终的相似度。因此可解释性非常强，模型的构建也充满技巧，比如采样等。模型没有对用户直接使用向量进行刻画，而是利用用户点击过的物品的向量集合作为其特征。**Ripple Network** 在电影、图书和新闻的点击率预测上取得了非常好的效果，将其结果可视化如图 6 所示可以看出，知识图谱连接了用户的历史兴趣和推荐结果，其中的若干条高分值的路径可以视为对推荐结果的解释。

Click history:

1. Family of **Navy SEAL** Trainee Who Died During Pool Exercise Plans to Take Legal Action
2. **North Korea** Vows to Strengthen **Nuclear Weapons**
3. **North Korea** Threatens 'Toughest Counteraction' After **U.S.** Moves **Navy Ships**
4. Consumer Reports Pulls Recommendation for **Microsoft Surface** Laptops



Candidate news: **Trump** Announces Gunman Dead, Credits 'Heroic Actions' of **Police**

图 6

3.4 Multi-task Learning for KG enhanced Recommendation

对比之前的模型，3.1 小节中的模型提出 DKN，利用一个 CNN 网络把实体向量和词向量进行了融合，但是前提是要在知识图谱中学习实体向量这个过程导致整个模型不能端到端的训练，还有一点就是 DKN 不能很好的将处于文本之外的边的信息融合到特征嵌入中；3.3 小节的 Ripple Net 类似于一个用户兴趣的记忆网络，图谱中的关系很少一部分才能被特征化，由于关系矩阵 R 很难被训练通过形式： $v^T R h$ 。这篇文章提出的模型 MKR[4]（Multi-task feature learning approach for Knowledge graph enhanced Recommendation）主要分为三个部分：推荐模块，知识图谱嵌入模块（KGE），以及压缩和交叉单元。其中压缩和交叉单元是前两个模块连接的桥梁。

首先我们介绍交叉和压缩单元，见图 7，这个模块的设计主要是为了推荐项目和实体的交互建模，这里区别一下这两个概念，项目指的是推荐项目可以是一条新闻，或者是一部电影，而实体则是指和该推荐项目相关联的实体，比如一则新闻标题中出现多个人名，地名这些就都是相关联的实体。

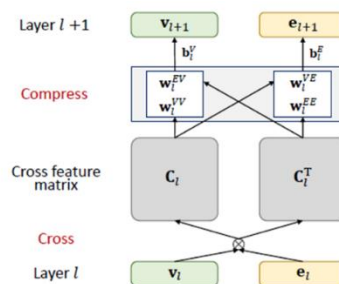


图 7

具体的对于 item i 和相关联的实体 e 构建一个 $d \times d$ 的点潜在的交互矩阵, 其中 $v_l \in \mathbb{R}^d$ 并且 $e_l \in \mathbb{R}^d$, 这里的 l 指的第 l 层, 公式 (12) 称为交叉操作。

$$C_l = v_l e_l^T = \begin{bmatrix} v_l^{(1)} e_l^{(1)} & \dots & v_l^{(1)} e_l^{(d)} \\ \dots & \dots & \dots \\ v_l^{(d)} e_l^{(1)} & \dots & v_l^{(d)} e_l^{(d)} \end{bmatrix} \quad (12)$$

然后通过交叉特征矩阵投影到下一层的 item 和实体的潜在表示空间中, 输出他们的特征向量, 见公式 (13), (14):

$$v_{l+1} = C_l w_l^{VV} + C_l^T w_l^{EV} + b_l^V = v_l e_l^T w_l^{VV} + e_l v_l^T w_l^{EV} + b_l^V \quad (13)$$

$$e_{l+1} = C_l w_l^{VE} + C_l^T w_l^{EE} + b_l^E = v_l e_l^T w_l^{VE} + e_l v_l^T w_l^{EE} + b_l^E \quad (14)$$

这里的 w 和 b 都是对交叉特征矩阵从不同的方向实现降维, 称为压缩操作。

图 8 左边的推荐模块以用户和项目为输入, 使用多层感知器 (MLP) 和交叉压缩单元分别为用户和项目提取短而密集的特征表示, 然后将提取的特征一起输入到另外一个 MLP 中, 输出最终预测的概率值。右侧的 KGE 模块也使用 MLP 来提取三元组的特征, 然后在有监督的情况下输出预测的尾实体的特征表示。由于推荐系统中的物品和知识图谱中的实体存在重合, 所以两个任务并非相互独立。我们在两个任务中设计了交叉特征共享单元作为两者的连接纽带, 这个单元一个可以让两个任务交换信息的模块。由于物品向量和实体向量实际上是对同一个对象的两种描述, 他们之间的信息交叉共享可以让两者都获得来自对方的额外信息, 从而弥补了自身的信息稀疏性的不足。在实际操作中, 我们采用交替训练的方式: 固定推荐系统模块的参数, 训练知识图谱特征学习模块的参数; 然后固定知识图谱特征学习模块的参数, 训练推荐系统模块的参数。

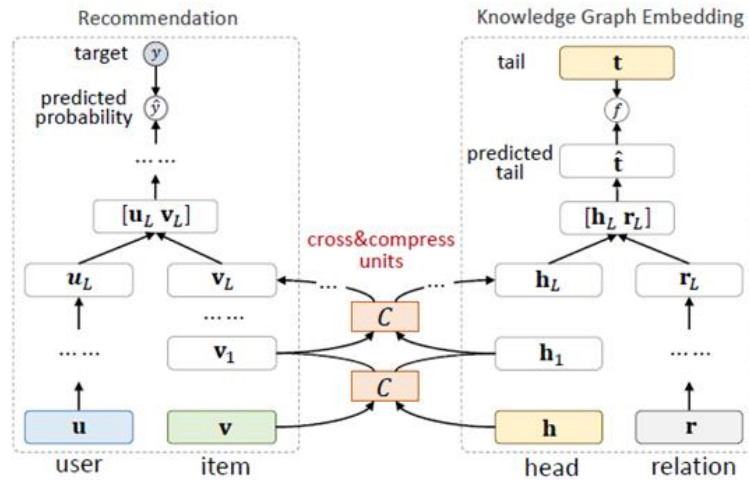


图 8

这样的设计使得整个模型可以是一种深层的端到端的框架，推荐和知识嵌入模块都采用的非线性层来提取输入的潜在特征。学习方式使用的交替学习也是一种较为创新和前沿的思路，其中如何设计两个相关的任务以及两个任务如何关联起来都是值得研究的方向。从实际运用和时间开销上来说，交替学习是介于依次学习和联合学习中间的：训练好的知识图谱特征学习模块可以在下一次训练的时候继续使用（不像联合学习需要从零开始），但是依然要参与到训练过程中来（不像依次学习中可以直接使用实体向量）。

四. 总结和未来工作（Conclusion and future work）

知识图谱作为推荐系统的一种新兴的辅助信息，近年来得到了研究人员的广泛关注。本文主要以知识图谱和推荐算法的学习方式为主线，介绍了依次学习，联合学习，交替学习这三种学习方式下具有代表性并且取得了很好的效果的一些模型，还介绍现如今知识嵌入的一些方法，详细可参见对应的论文，这些模型没有绝对的好坏之分，更重要的是根据目前所要解决的推荐问题而定，在合适的场景下用恰当的模型，才会发挥出最大的效果。

尽管深度学习以及知识图谱，包括介绍的最后一个模型多任务学习的引入，使得推荐效果有了很大的改善。但是我个人认为这方面的研究还有很大的改进的空间并且需要进一步的探究和深思，比如：当前利用知识图谱更多的可以挖掘出来的是一种相关关系，但是实际上推荐系统如何要提高准确率更多的是需要获取一种准确的因果关系，比如我在某电商平台上买了泳裤，我下一步买的泳镜或者泳帽的可能性要比买毛巾的可能性大很多。还有一些其他的问题需要去解决，典型的问题包括多源信息的融合、知识图谱中用户特征的挖掘及潜在需求的分析、多维度推荐、与时间空间推理的结合、用户隐私保护，另外还有由于数据生成速度快，而且多源数据融合过程中引入了更多的噪声和冗余，如何在线使得这些模型能够实时的完成推荐任务也是一个很重要的研究点，因为很多学术论文的研究都是基于干净的数据集，并且在小批量的数据集上完成实验。

未来，知识图谱和时序模型的结合、知识图谱和基于强化学习的推荐系统的结合、推荐系统的可拓展性，以及知识图谱和其它辅助信息（比如社交网络信息的充分利用，用户不再是单一的用户需要进一步挖掘彼此之间的联系以发现更多隐含信息）在推荐系统中的结合等相关问题仍然值得更多的研究。

五. 参考文献(References)

- [1] Wang, Hongwei, et al. "DKN: Deep knowledge-aware network for news recommendation." Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2018.
- [2] Zhang, Fuzheng, et al. "Collaborative knowledge base embedding for recommender systems." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016.
- [3] Wang, Hongwei, et al. "RippleNet: Propagating user preferences on the knowledge graph for recommender systems." Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 2018.
- [4] Wang, Hongwei, et al. "Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation." arXiv preprint arXiv:1901.08907 (2019).
- [5] Bordes, Antoine, et al. "Translating embeddings for modeling multi-relational data." Advances in neural information processing systems. 2013.
- [6] Wang, Zhen, et al. "Knowledge graph embedding by translating on hyperplanes." Twenty-Eighth AAAI conference on artificial intelligence. 2014.
- [7] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In AAAI.
- [8] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In ACL. 687–696.
- [9] NIARAKI A S, KIM K. Ontology based personalized route planning system using a multi-criteria decision making approach[J]. Expert systems with applications, 2009, 36(2): 2250–2259.
- [10] Bellini, Vito, et al. "Computing recommendations via a Knowledge Graph-aware Autoencoder." arXiv preprint arXiv:1807.05006 (2018).
- [11] <https://googleblog.blogspot.it/2012/05/introducing-knowledge-graph-things-not.html>
- [12] Wang, Xiang, et al. "Explainable Reasoning over Knowledge Graphs for Recommendation." arXiv preprint arXiv:1811.04540 (2018).
- [13] Wang, Xiang, et al. "KGAT: Knowledge Graph Attention Network for Recommendation." arXiv preprint arXiv:1905.07854 (2019).
- [14] Wang, H., Wang, N., and Yeung, D.-Y. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, ACM (New York, NY, USA, 2015), 1235–1244