社交网络期末课程报告

与知识图谱的交互学习下的推荐系统

刘　威 21210180060
王笛合 21210180067
肖儒峰 21210980120

2022 年 1 月 14 日

1 概述

传统的推荐系统只使用用户和物品的历史交互信息作为输入，这会带来两个问题：
一，在实际场景中，用户和物品的交互信息往往是非常稀疏的。例如，一个电影类 APP 可能包含了上万部电影，然而一个用户打过分的电影可能平均只有几十部。使用少量的已观测数据来预测大量的未知信息，会极大地增加算法的过拟合风险；
二，对于新加入的用户或者物品，由于系统没有其历史交互信息，因此无法进行准确地建模和推荐。在实际场景中做用户推荐经常遇见冷启动的问题，冷启动包括完全没有信息的新用户或新物品和有相关信息的新用户或新物品。当我们遇到了有相关信息的新物品造成的数据稀疏性问题，那我们就能借助知识图谱一定程度上解决针对于数据稀疏性的冷启动问题。

在本报告中，我们使用了 MKR(**M**ulti-task feature learning approach for **K**nowledge graph enhanced **R**ecommendation) 模型 [1] 的思想在豆瓣数据集上进行实验。此外，我们还提出并实现了一个改进：使用语言模型对实体（entity）进行嵌入再输入模型训练。

2 base 模型介绍

MKR 由三个主要部分组成：推荐模块、知识图谱嵌入模块和交叉压缩单元。整体结构如下图所示：

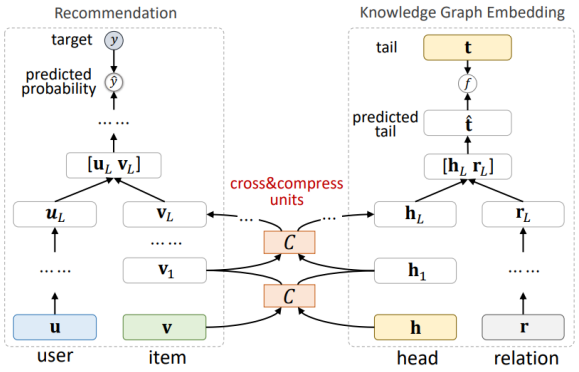

图 1: MKR 的基本框架

左侧的推荐模块在低层将用户和物品作为输入，使用多层感知器和交叉压缩单元分别提取用户特征和产品特征，将两个特征送入高层的另一个多层感知器，最终通过 sigmoid 函数得到介于 0-1 之间的小数作为推荐系数，推荐系数越大，表示用户喜欢该产品的可能性越大。

右侧的知识图谱嵌入模块在低层将知识三元组的头部和关系作为输入，头部经过交叉压缩单元提取特征，关系部分使用同层数的感知器提取，最后将头部和关系部分的特征合并作为高层输入，最终得到维数同尾部的特征向量。

交叉压缩单元是推荐模块和知识图谱嵌入模块的衔接。交叉压缩单元连通两个任务并交换和分享特征，能够学习推荐系统中的产品和知识图谱中的实体的特征交互关系。由于产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。由于主任务是推荐而不是知识图谱，因此对于交互压缩的系数更新并不是一比一，而是多对一，即多次训练推荐系统模块的系数更新对应一次知识图谱的系数更新。

2.1 交叉压缩单元

对于交叉压缩单元，交叉操作是指将产品 $\mathbf{v}_l \in \mathbb{R}^{d\times 1}$ 和它的一个相关实体 $\mathbf{e}_l \in \mathbb{R}^{d\times 1}$ 做向量乘法形成交叉矩阵 C_l（在本报告实现中，我们只有电影作为头部的三元组，所以产品 v 的相关实体就是其本身），

$$C_l = \mathbf{v}_l \cdot \mathbf{e}_l^T \in \mathbb{R}^{d\times d}$$

压缩操作是指将交叉特征矩阵投射到潜在表示空间，作为下一层的产品和实体的输入向量 $\mathbf{v}_{l+1}, \mathbf{e}_{l+1,i}$，过程如图2所示。形式化计算公式如下：

$$\mathbf{v}_{l+1} = C_l w_l^{vv} + C_l^T w_l^{ev} + b_l^v,$$

$$\mathbf{e}_{l+1} = C_l w_l^{ee} + C_l^T w_l^{ee} + b_l^e.$$

其中 $w_{l\cdot} \in \mathbb{R}^d, b_l \in \mathbb{R}^d$ 分别为训练的权重和偏置。应该注意的是，交叉压缩单元只存在于 MKR 的浅层特征提取的工作，在高层中特征的可转移性有显著下降。在 MKR 的高层中，将产品特征与用户特征合并，实体特征与关系特征合并，而混合特征由于缺乏关联性不再适用于共享 [4]。

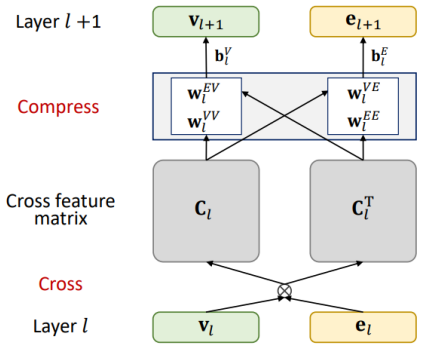

图 2: 交叉压缩单元结构图

2.2 推荐模块

在推荐模块，向量 $\mathbf{u}$ 和 $\mathbf{v}$ 分别表示用户 u 和产品 v，$\mathbf{u}$ 和 $\mathbf{v}$ 在原模型中使用 one-hot 编码，这是我们之后进行改动的地方。我们给定用户 u 的原始特征向量 $\mathbf{u}$, 使用 L 层的多层感知器提取用户的潜在特征：

$$\mathbf{u}_L = \mathcal{M}(\mathcal{M}(\cdots \mathcal{M}(\mathbf{u}))) = \mathcal{M}^L(\mathbf{u})$$

其中 M 是全连接神经网络层:

$$\mathcal{M}(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

而对于产品 v 使用 L 个交叉压缩单元提取他的特征，推荐系统模块是点击率预估模型，在最终得到用户的特征向量和产品的特征向量后，通过多层感知机可计算用户 u 喜欢产品 v 的可能性 $\hat{P}_{uv}$:

$$\hat{P}_{uv} = sigmoid \circ f_{RS}(\mathbf{u}_L, \mathbf{v}_L)$$

2.3 知识图谱嵌入模块

知识图谱嵌入 (Knowledge Graph Embedding, 下简称 KGE) 模块。KGE 是将实体和关系映射到连续的低维向量空间同时保留他们原来的空间结构。我们首先对数据集预处理得到一个三元组（实体，关系，实体），比如（叶问前传, directors, 邱礼涛）表示叶问前传的导演是邱礼涛。

对于给定的知识三元组 $(\mathrm{h}, \mathrm{r}, \mathrm{t})$, 利用交叉压缩单元和多层感知器从原始的头部 h 和关系 r 提取特征。将 h 和 r 对应的特征向量拼接传入多层神经网络, 得到尾部 t 对应向量的预估值 $\hat{\mathbf{t}}$。模块希望预测得到的尾部向量和真实的尾部向量相近: $(\hat{\mathbf{t}}, \mathrm{t})$ 的分数由相似度函数 f_{KG} 计算得到, f_{KG} 函数在本报告中是将 t 和 $\hat{\mathbf{t}}$ 的内积之后在用 sigmoid 函数处理:

$$\mathrm{score}(h, r, t) = f_{KG}(\mathbf{t}, \hat{\mathbf{t}}) = sigmoid(\mathrm{t} \cdot \hat{\mathbf{t}}).$$

2.4 模型优势

推荐系统和知识图谱特征学习的交替学习类似于多任务学习的框架。我们注意到推荐系统中的物品和知识图谱中的实体存在重合，因此两个任务之间存在相关性。将推荐系统和知识图谱特征学习视为两个分离但是相关的任务，采用多任务学习的框架，可以有如下优势：两者的可用信息可以互补；KGE 任务：帮助推荐模块摆脱局部极小值；防止推荐系统过拟合，提高推荐系统的泛化能力。

3 New 模型介绍

3.1 设计灵感

在 base 模型中，由于我们以 onehot 编码的形式储存实体值，原来的模型在输入时没有考虑到实体语意之间的关系，而仅仅是将不同的实体记作不同的序号进行输入，我们将某一实体名转换为新的实体名的话并不会对训练过程本身产生影响。例如：我们把一个电影的负面评价转换为正面评价，我们推荐的结果并不会产生变化。于是我们想寻找办法对实体本身角度发现规律，我们从语义角度，使用用语义功能十分强大的 Bert[2] 模型对实体进行 Embedding 代替 onehot 编码后的 Embedding。于是我们先对实体用 Bert 模型进行一次转换，再将高维的句向量当作输入给到模型。这就有了我们的第二个模型，称之为 New 模型。

3.2 模型优势

基于语义模型对实体本身进行 Embedding 能够提升模型在训练集外的实体（entity）上的表现，将语义蕴含在输入向量中，能够在第一时间判断出同属性的关系。换言之，在新的实体出现时，使用 Bert 给出的 Embedding 已是与其出现在训练集中同义词语的相近词向量，使用 Bert 进行 Embedding 后，的两个词向量的位置与 KGE 中该实体与该实体同义词的实际位置关系相似。

在训练上，可以认为是模型训练的一种初始化方案，而这个初始化方案就已经能够直接判断文本含义的相对关系，更利于损失函数达到全局极小值点。

3.3 模型设计

我们直接使用 huggingface 中的 Bert-base 模型，注意到该模型的输出向量是 768 维，为了简化模型我们使用了其前 k 维向量作为实体的 Embedding 向量，因为 Bert 的前 k 维向量可作为 Bert 模型简化的降维版本，并且是有效的信息保留方式 [5]，相较于添加一个全连接层的方案，直接选取前 k 维也能够减少模型的总参数量。由于我们在 Embedding 后的模型训练时，冻结 Bert 原模型的参数，因此也可以认为这是对 Bert 的迁移学习。

4 实验设计思路

4.1 数据集处理

我们将原有的豆瓣电影数据集转换为两部分：

1. 打分记录：将用户打分行为记录转化为（用户-电影-是否喜欢），其中是否喜欢根据用户的评分是否大于等于 4 进行判断；
2. 电影三元组：（头部-关系-尾部），其中：头部均为电影编号；关系取自（title，writer，director，...）；

打分记录用于构建推荐模块输入空间。电影三元组用于 KGE 模块的训练。

4.2 数据可视化

我们可以将三元组进行可视化，能够直观的看出电影之间存在的联系，如图3我们用有“烂片”为标签（relation: tags）的可视化结果。

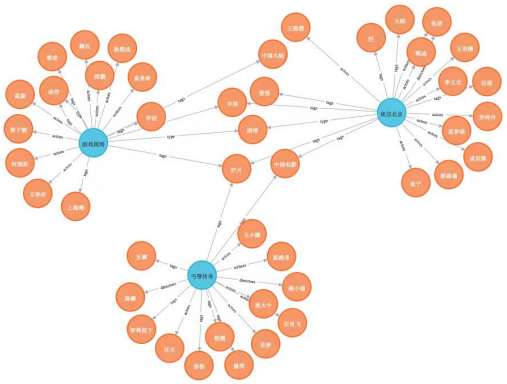

图 3: 三元组关系可视化

4.3 base 模型实验思路

我们使用原有的 MKR 模型对豆瓣电影数据集进行训练并推荐。称之为 base 模型实验，目标是利用 KG 来协助推荐。

- base 模型训练由两个阶段组成：更新推荐模块参数和更新 KGE 模块。由于我们更希望提高推荐性能，因此在交替学习中，对推荐模块参数更新 t 次（t 是一个超参数，通常 t>1），再对 KGE 模块的系数更新一次。

- 超参数根据模型在验证集上的 AUC 指标进行选择。
- 数据集的划分：训练、验证和测试集的比例为 6：2：2。实验重复 3 次，并取平均值。
- 模型评估方法：
  1. 在点击率（CTR）预测中，从训练集（包括训练集和验证集）和测试集的交集中取出用户，通过模型计算出各个产品的预测点击率。使用 AUC 和准确率评估模型的效果；
  2. 计算 TopK 的精确率与召回率，对测试集中的每个用户计算出前 K 个预测点击概率最高的产品，计算精确率和召回率进行评估。

5 实验结果与分析

5.1 base 模型实验结果

由图4我们可以看到，通过少量 epoch，AUC 和 ACC 可以达到相对高值，并且相对稳定。而在精确率、召回率和 F1 值上，我们使用 Early Stop 的方法选择出 AUC 相对较高的结果，并测得该模型下的 TopK 的精确率、召回率和 F1 值，见表2。

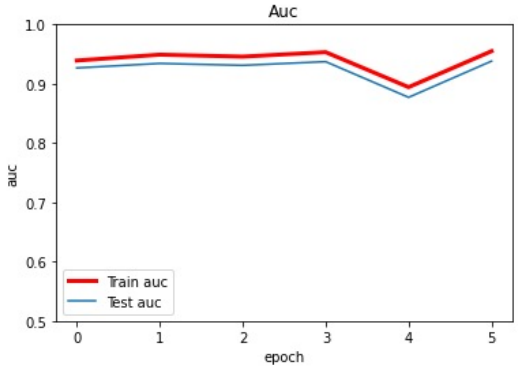

图 4: 训练集测试集上的 AUC 和 ACC

表 1: 最优 base 模型的指标

Epoch	train AUC	train acc	eval AUC	eval acc	test AUC	test acc
7	0.9566	0.8831	0.9385	0.8739	0.9380	0.8738

表 2: base 模型的精确率、召回率和 F1 值

	Top1	Top2	Top5	Top10	Top20	Top50	Top100
precision	0.0600	0.0600	0.0700	0.0680	0.0510	0.0426	0.0363
recall	0.0018	0.0049	0.0147	0.0294	0.0456	0.1013	0.1704
F1	0.0034	0.0091	0.0243	0.0411	0.0481	0.0600	0.0599

5.2 New 模型实验结果和分析

模型指标如 AUC 表现稍差于 base 模型，少部分指标如 ACC 显著少于 base 模型。分析原因，可能有以下几方面：

表 3: New 模型在 10% 数据集表现

	train AUC	eval AUC	test AUC
base model	0.9977	0.9826	0.9815
new model	0.7852	0.7806	0.7895

1. 高维下情形下，模型本身性能下降。
   通过原论文中的讨论，我们可以看到当 Embedding 的维数为 16 维时，模型本身在 AUC 和准确率上会有所不足。但作为添加了文本语义信息的 New 模型，若想得到较好的结果理应需要维数的扩充；

2. 知识图谱中的知识单一。
   由于在已有的数据集上，我们仅有（电影名（如警察故事），关系（如 actor），实体（如成龙））的三元组，其中的 head 仅有影片名，而影片名本身跟也并非能够完全描述电影本身，因此知识图谱中的关系相对单一，且不同的 tail 之间的关系不清晰；

3. Bert 对人名的理解有限。
   由于该问题中，Bert 对于人名实体的认别，理论性不足，由于缺乏外部知识，人名的 Embedding 也相对困难，而人名在该场景中是出现最多的实体（entity）。

该模型未来的可行性分析：

1. 在较大数据集上或许能够有比较好的效果；

2. 推荐模块和 KGE 模块的模型层数增加，或许会有比较好的效果。

6 小组分工

刘　威：数据集预处理，原模型学习，新提出的模型建立与实现；
肖儒峰：数据集预处理，原模型学习，Bert 服务搭建与测试；
王笛合：数据可视化。

参考文献

[1] Wang H, Zhang F, Zhao M, et al. , *Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation*, The World Wide Web Conference. 2019: 2000-2010.

[2] Devlin J, Chang M W, Lee K, et al. *Bert: Pre-training of deep bidirectional transformers for language understanding*,arXiv preprint arXiv:1810.04805, 2018.

[3] Vaswani A, Shazeer N, Parmar N, et al. *Attention is all you need*,Advances in neural information processing systems. 2017: 5998-6008.

[4] Yosinski J, Clune J, Bengio Y, et al. *How transferable are features in deep neural networks?*, In Advances in Neural Information Processing Systems. 3320–3328.

[5] [Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fern, a Viégas, Martin Wattenberg] *Visualizing and Measuring the Geometry of BERT*. arXiv preprint arXiv:1906.02715, 2019.