



 所谓 (/users/845813) 2020-10-26 09:16:40 47 阅读 0 收藏 (/articles/184911/mark/)

知识体系: 机器学习/深度学习 (/articles/?kid=941) 数据挖掘 (/articles/?kid=944) 搜索推荐 (/articles/?kid=945) 人工智能 (/articles/?kid=951) 数据处理 (/articles/?kid=955) 数据建模 (/articles/?kid=957) 数据管理 (/articles/?kid=960)  修改知识体系

文章标签: 算法 (/search?q=算法&type=ARTICLE)  修改标签  标签历史 (/articles/184911/tags/history)

附加属性: **作者原创**

多任务学习11:Jointly Learning Explainable Rules for Recommendation with Knowledge Graph(WWW19)

- 1 Jointly Learning Explainable Rules for Recommendation with Knowledge Graph(WWW19)
 - 1.1 简介
 - 1.2 背景知识
 - 1.2.1 问题定义
 - 1.2.1.1 商品推荐(Item Recommendation)
 - 1.2.1.2 知识图谱(Knowledge graph)
 - 1.2.1.3 知识图的归纳规则
 - 1.2.1.4 问题定义
 - 1.2.2 推荐基础模型
 - 1.2.2.1 BPRMF:贝叶斯个性化排序矩阵分解
 - 1.2.2.2 NCF(Neural Collaborative Filtering):神经网络协同过滤
 - 1.3 RULEREC框架
 - 1.3.1 异构图构建
 - 1.3.2 规则学习模块
 - 1.3.2.1 规则学习
 - 1.3.2.2 规则选择
 - 1.3.3 Item推荐模块
 - 1.3.4 多任务学习
 - 1.4 实验
 - 1.4.1 模型效果
 - 1.4.2 案例研究&效果比较
 - 1.5 小结
- 2 参考文献

Jointly Learning Explainable Rules for Recommendation with Knowledge Graph(WWW19)

简介

可解释性和有效性是构建推荐系统的两大关键成份,之前的工作主要关注通过引入辅助信息来获得更好的推荐效果。但这些方法会存在下面的两个问题:

- 基于神经网络的embedding方法很难解释和debug;
- 基于图的方法需要人工以及领域知识来定义模式和规则, 并且会忽略商品的相关类型;(可替代并且互补)

本文提出了一种新的联合学习框架, 我们将知识图中可解释规则的归纳与规则引导的神经推荐模型的构建相结合;该框架鼓励两个模块可以互相补并生成有效的可解释的推荐。

- 归纳规则：从**商品为中心**的知识图谱中挖掘, 总结出用于推断不同商品关联的常见多跳关系模式, 并为模型预测提供人类可理解的解释;
- 通过引入规则对推荐模块进行扩充, 使其在处理冷启动问题时具有更好的泛化能力;

所以本文是希望通过联合训练推荐和知识图谱来给出既准确又可解释的推荐。

背景知识

问题定义



商品推荐(Item Recommendation)

给定用户 U 和商品 I ,商品的推荐任务就是基于用户和商品的历史交互找出最适合每个用户的商品;用户会通过购买或者对某个商品打分的形式表述他的喜好, 这些交互可以通过矩阵进行表示, 最具代表的方法就是: 将用户和商品embed到低维度的潜在空间中, 然后使用用户-商品交互的结果来进行推荐。

知识图谱(Knowledge graph)

知识图谱是由实体(节点)和关系(边)构成的多关系图,我们可以使用多个三元组(头部实体 E_1 , 关系类型 r_1 , 尾部实体 E_2)来表示知识图谱的事实。

知识图的归纳规则

知识图谱中两个实体之间可能有许多路径, 一条路径由关系类型构成(例如 $P_k = E_1 r_1 E_2 r_2 E_3$ 就是两个实体 E_1 和 E_3 的路径); 规则 R 由两个实体的关系序列定义,例如 $R = r_1 r_2$, 路径和规则的区别在于规则更多关注关系类型而不是实体。

问题定义

给定用户 U ,商品 I ,用户和商品的交互,商品关联以及知识图谱,我们的联合训练框架目的在于:

- 基于商品的关系学习商品之间的规则;
- 学习推荐系统推荐对每个用户基于规则 R 以及他的交互历史 I_u 推荐商品;

该框架输出一套规则 R 以及推荐商品的列表 I' 。

推荐基础模型

此处介绍两种基础的模型。

BPRMF:贝叶斯个性化排序矩阵分解

BPRMF的思想是: 将用户和商品分别表示为潜在特征向量, 使用 U_u 表示用户 u 的特征向量, I_i 表示商品 i 的特征向量。它们的维度是一样的, 在BPRMF中,

$$S_{u,i} = U_u^T \cdot I_i$$

BPRMF的目标函数为:

$$O_{BPRMF} = \sum_{u \in U} \sum_{p \in I_u, n \notin I_u} (S_{u,p} - S_{u,n})$$

其中 p 是和用户 u 交互的正商品, n 是从商品中随机采样的未交互的负样本。

NCF(Neural Collaborative Filtering):神经网络协同过滤

NCF的核心思想可以表示为:

$$\begin{aligned} & h_{u,i} = U_u^T \cdot I_i \\ & g_{u,i} = \phi_n(\dots \phi_2(\phi_1(z_1))), z_1 = \phi_0(U_u \oplus I_i) \\ & \phi_k(z_{k-1}) = \phi_k(W_k^T z_{k-1} + b_{k-1}) \end{aligned}$$

其中 n 为隐藏层的个数, W_l, b_l 个 z_k 是权重矩阵,bias向量以及每个层的输出。 \oplus 表示concatenation, $h_{u,i}$ 和 $g_{u,i}$ 都可以表示为用户商品的交互向量。

我们将NCF的目标函数改为:

$$\begin{aligned} & S_{u,i} = \phi(\alpha \cdot h_{u,i} \oplus (1 - \alpha) \cdot g_{u,i}) \\ & O_{NCF} = \sigma(\sum_{u \in U} \sum_{p \in I_u, n \notin I_u} (S_{u,p} - S_{u,n})) \end{aligned}$$

RULEREC框架

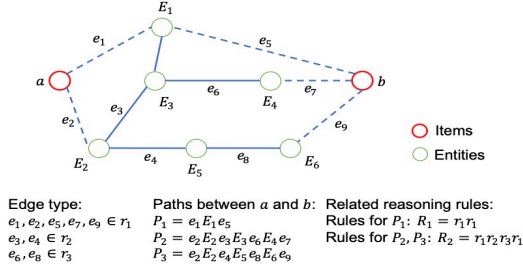
带有rule learning的推荐主要由下面两个子任务组成: (1). 基于商品的关系从知识图谱中进行规则学习; (2). 对每个用户基于他的购买历史 I_u 和得到的规则 R 推荐商品; 我们将其建模为多任务学习, 于是我们有:

$$\bullet \min_{V,W} O = \min_{V,W} \{O_r + \lambda O_l\}$$

其中 V 表示我们推荐模块的参数, W 表示规则学习和推荐模块的共享参数。 O_r 为推荐系统的目标, O_l 为规则学习模块的目标。

异构图构建

我们构建异对于推荐和知识图谱的商品的异构图, 具体如下:



规则学习模块

规则学习模块的目的是在异构图中找到与给定项目关联A相关联的可靠规则集 R_A 。

规则学习

对于在异构图中的任何商品对 (a, b) , 我们使用随机游走的算法计算在商品对的特定规则下的路径的概率。这样我们便可以得到商品对的特征向量。特征向量的每个规则是商品对之间规则的概率。

我们先定义商品对之间规则的概率, 给定一个规则: $R = r_1 \dots r_k$, 从 a 到 b 的概率可以定义为:

$$\bullet P(b|a, R) = \sum_{e \in N(a, R')} P(e|a, R') \cdot P(b|e, r_k)$$

其中 $R' = r_1 \dots r_{k-1}$, $P(b|e, r_k) = \frac{I(r_k(e, b))}{\sum_i I(r_k(e, i))}$ 表示在关系 r_k 下面从一步随机walk从节点 b 到节点 e 的概率。

如果在关系 r_k 下存在连接从 e 到 b , 那么我们就有 $I(r_k(e, b)) = 1$; 如果 b 和 e 是一样的, 那么我们就有 $P(b|e, r_k) = 1$

- 一个例子

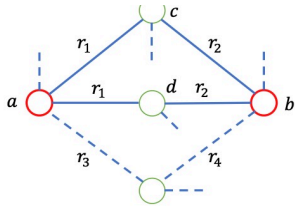


Figure 4: An example of a graph between items a and b . r represents a edge type or a relation type.

$$\bullet P(b|a, R) = P(c|a, r_1) \cdot P(b|c, r_2) + P(d|a, r_1) \cdot P(b|d, r_2)$$

同时, 此处定义了一个商品对的特征向量, 给定一堆规则, 那么对于每个商品对 (a, b) , 我们将其定义为:

$$\bullet x_{(a,b)} = [P(b|a, R_1), \dots, P(b|a, R_n)]^T$$

在特征向量 $x_{(a,b)}$ 表示在 a 和 b 之间的规则 R_i .

规则选择

1. Hard selection method: 硬选择方法通过设置一个超参数来决定我们首先要选择多少条规则。
2. Soft selection method: 利用基于学习的目标函数的方法是将每个规则的权重作为对推荐模块中规则权重的约束。这样就不会从规则集中删除任何规则, 也不会引入额外的超参数。

Item推荐模块

我们提出了一个由共享权重向量 w 参数化的函数 f_w , 以将候选项商品 i 和用户交互 (如购买) 项在规则集 R 下的规则特征相结合。

$$\bullet S'_{u,i} = f_w(S_{u,i}, \sum_{k \in I_u} F(i, k|R))$$

商品对 (a, b) 在规则 R 下的特征向量表示为 $F(a, b|R)$, 注意: $F(a, b|R) = \sum_{e \in N(a, R')} P(e|a, R') \cdot I(b|e, r_k)$

如果 b 和 e 之间在关系类型 r_k 上存在一条边, 我们就有 $I(b|e, r_k) = 1$, 也就是说我们可以通过一个点 e , 从 a 到 b 。这么做我们可以令模型更具有解释性。

为了考虑候选商品 i 以及商品集 I_u 的全局商品相关性, 我们增加 i 和在 I_u 中的每个商品 I_k 的规则特征。为了方便, 新的特征向量被命名为: $F(i, I_u|R)$, 所以上面又可以被重写为:

$$\bullet S'_{u,i} = f_w(S_{u,i}, F(i, I_u|R))$$

我们将推荐模块的目标函数定义为:

$$\begin{aligned} O_r &= \sum_{u \in U} \sum_{p \in I_u, n \notin I_u} (S'_{u,p} - S'_{u,n}) \\ &= \sum_{u \in U} \sum_{p \in I_u, n \notin I_u} (f_w(S_{u,p}, F(p, I_u|R)) - f_w(S_{u,n}, F(n, I_u|R))) \end{aligned}$$

其中 p 是正向商品 ($\in I_u$), n 是随机采样的负样本 ($\notin I_u$).

注意, 规则权重向量 w 给出了推荐模块中规则对商品对的解释。如果一个候选商品 i 的得分高于其他候选项, 则可以使用该规则为 i 贡献最高分数的规则以及用户购买的相应项来解释算法为什么向用户推荐 i

这种组合方法灵活, 易于将规则特征引入到已有的推荐模型中 (利用算法的预测函数计算 $S_{u,i}$)。

多任务学习

此处我们采用多任务目标函数进行优化:

$$\bullet O = O_r + \lambda O_l$$

其中 O_l 和 O_r 是规则学习和推荐学习模块的目标函数, 注意目标函数中共享了 w 。

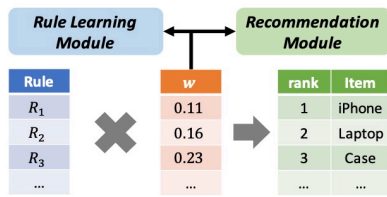


Figure 5: Multi-task learning of the rule learning module and the recommendation module. These two modules share the parameter w .

实验

模型效果

Table 5: Performance Comparison between RuleRec and Other Methods in Different Domains. RuleRec_{two} and RuleRec_{multi} are our proposed models. RuleRec_{two} is a two-step rule-based model and RuleRec_{multi} is a multi-task model. These models use BPRMF or NCF as a recommendation model. * indicates statistical significance at $p < 0.01$ compared to the best baseline model.

Methods / Dataset	Cellphone				Electronic			
	Recall@5	Recall@10	NDCG@10	MRR@10	Recall@5	Recall@10	NDCG@10	MRR@10
BPRMF [27]	0.3238	0.4491	0.2639	0.2058	0.1886	0.2763	0.1571	0.1207
GMF [13]	0.3379	0.4666	0.2789	0.2223	0.1988	0.2835	0.1657	0.1298
MLP [5]	0.3374	0.4779	0.2790	0.2182	0.2000	0.2883	0.1681	0.1315
NCF [14]	0.3388	0.4751	0.2761	0.2151	0.2005	0.2916	0.1679	0.1300
Hec _{sl} [31]	0.2436	0.3481	0.2040	0.1600	0.1870	0.2851	0.1534	0.1135
Hec _{pl} [31]	0.2511	0.3564	0.2090	0.1641	0.1948	0.2851	0.1628	0.1256
RippleNet [34]	0.2834	0.4042	0.2219	0.1780	0.1965	0.2865	0.1638	0.1265
RuleRec _{two} (BPRMF)	0.3495*	0.4768	0.2813*	0.2201*	0.2050*	0.2932	0.1707*	0.1334*
RuleRec _{multi} (BPRMF)	0.3568*	0.4829*	0.2864*	0.2246*	0.2071*	0.2946*	0.1718*	0.1341*
RuleRec _{two} (NCF)	0.3538*	0.4876*	0.2902*	0.2296*	0.2049*	0.2947*	0.1681	0.1296
RuleRec _{multi} (NCF)	0.3569*	0.4894*	0.2902*	0.2290*	0.2074*	0.2917	0.1702*	0.1330

- 在加权的训练中结合推荐损失和规则选择损失可以提高推荐效果;
- 从知识图谱中抽取得到的规则对于商品对的特征向量学习是非常有价值的, 学习得到的向量可以加强多个基本的推荐模型;

- 多任务学习的方式对于推荐和规则选择比规则加权学习贡献更多;因为提出框架的灵活性,提出的规则能够和其它推荐算法结合进一步提升模型的效果。

案例研究&效果比较

Table 6: Performance Comparison on MovieLens dataset.

Model	AUC
Hec_{sl} [31]	0.894
Hec_{pl} [31]	0.895
RippleNet [34]	0.921
RuleRec _{two} (BPRMF)	0.907



- 提出的模型比HERec好, 但比RippleNet差 > 有两个原因: 1) 关系类型在这个数据集中非常有限 (只有7个), 因此RuleRec中推荐规则选择的能力在这种情况下是有限的。2) MovieLens-1M不同于的真实知识图数据集 (它是通过将项目链接到Freebase来构建的), 它的连接覆盖非常完美, RippleNet从中受益匪浅。结果表明, 所提出的算法能够在紧凑异构图中取得显著的性能;

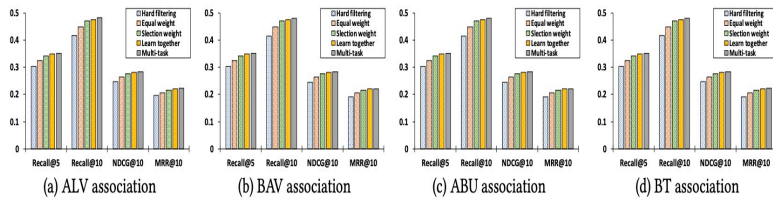


Figure 6: Performance comparison under different score function settings in the Cellphone dataset. We select the top 50 rules in each association by chi-square method. Among all score function settings, the multi-task learning method performs the best.

- 多任务学习方法在所有指标上都取得了最好的性能, 并且改进效果显著, 这表明多任务学习在规则权重学习中非常有助于获得更好的推荐结果。
- 由于硬过滤方法可能同时忽略负项目和正项目 (从表3中, 我们可以看到, 有时正项目和项目采购历史 I_u 之间没有规则)。
- 虽然选择权重有助于推荐 (更好同样的重量), 它仍然比一起学习模式更糟糕。

Table 7: The results of using a single association vs. all associations. BPRMF is used as a recommendation model. Our proposed model shows the best performances when using all kinds of associations.

Type	Recall@5	Recall@10	NDCG@10	MRR@10
None	0.3238	0.4491	0.2639	0.2058
ALV	0.3527	0.4815	0.2844	0.2225
BAV	0.3513	0.4812	0.2840	0.2222
ABU	0.3511	0.4811	0.2838	0.2218
BT	0.3514	0.4810	0.2841	0.2222
ALL	0.3568	0.4829	0.2864	0.2246

- 对于由四个关联中的任何一个导出的规则, 使用BPRMF的RuleRec_{multi}明显优于BPRMF算法。
- 使用不同的关联可以得到相似的结果, 但它们对于挖掘项目关系以提高推荐结果都是有价值的。

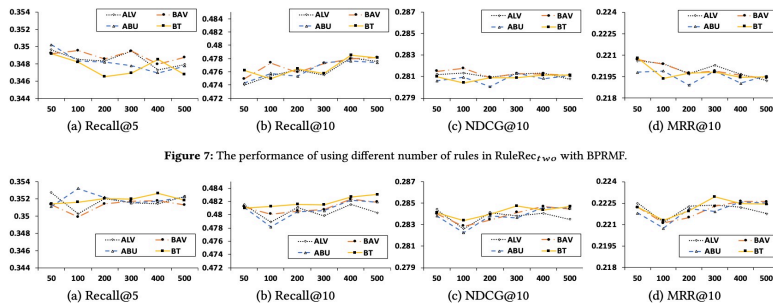


Figure 7: The performance of using different number of rules in RuleRec_{two} with BPRMF.

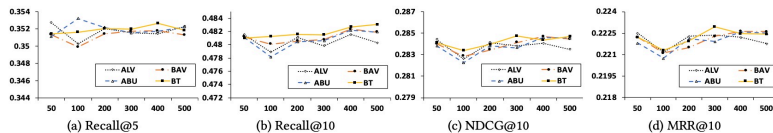


Figure 8: The performance of using different number of rules in RuleRec_{multi} with BPRMF.

- 在推荐中当我们使用更多规则的时候,整体效果没有变得更好;(可能随着规则的个数增加,很多不好的规则也被带入到模型中);
- 由于在多任务学习算法中考虑了规则的选择, 我们发现规则集的性能与随着规则数的增加, BPRMF算法表现出更好的性能;
- 使用BPRMF的RuleRec_{multi}的性能明显优于RuleRec_{two}和BPRMF (对不同规则计数的实验结果进行配对双样本t检验)的性能。p<0.01)。结果表明, 基于多任务学习的算法能够处理大量的规则, 即使存在一些无用的规则;

小结

本文提出了一种新的、有效的联合优化框架，用于从包含商品的知识图中归纳出规则，并基于归纳出的规则进行推荐。该框架由两个模块组成：**规则学习模块和推荐模块**。规则学习模块能够在具有不同类型商品关联的知识图中导出有用的规则，推荐模块将这些规则引入到推荐模型中以获得更好的性能。此外，有两种方法来实现这个框架：两步学习和联合学习。

本文研究使用大型知识图Freebase进行规则学习。该框架能够灵活地支持不同的推荐算法。我们修改了两种推荐算法，一种是经典的矩阵分解算法（BPRMF）和一种基于神经网络的推荐算法（NCF）。所提出的四种规则增强推荐算法在多个领域都取得了显著的效果，并优于所有的基线模型，表明了本文框架的有效性。此外，推导出的规则还能够解释我们为什么要向用户推荐这个项目，同时也提高了推荐模型的可解释性。进一步的分析表明，我们基于多任务学习的组合方法（带有BPRMF的 $RuleRec_{multi}$ 和NCF的 $RuleRec_{two}$ ）在不同规则数下的性能优于两步法。不同关联规则的组合有助于获得更好的推荐结果。



参考文献

1. Jointly Learning Explainable Rules for Recommendation with Knowledge Graph:<https://arxiv.org/pdf/1903.03714.pdf> (<https://arxiv.org/pdf/1903.03714.pdf>)
2. https://blog.csdn.net/qq_41621342/article/details/104188843 (https://blog.csdn.net/qq_41621342/article/details/104188843)
3. <https://github.com/THUIR/RuleRec> (<https://github.com/THUIR/RuleRec>)

评论文章 (0) 1 (/articles/184911/voteup) 0

他们赞过该文章

所谓 (/users/845813)

上一篇：多任务学习10:DBMTL_Deep Bayesian Multi-Ta... 下一篇：多任务学习12:Why I like it_Multi-task Learning ...

写下你的评论...

评论