

结合用户长短期偏好的对话推荐方法

冯兴杰, 张 贺⁺

(中国民航大学 计算机科学与技术学院, 天津 300300)

摘 要: 目前的对话推荐系统过于注重用户当前的偏好信息, 更多地被设计成为类似搜索引擎的工具, 缺乏个性化。针对该问题, 提出一种与序列推荐相结合的对话推荐方法, 应用序列推荐为对话推荐系统中项目评分模块提供更为符合用户长短期偏好的项目候选集, 对话推荐中的路径选取操作也可以为序列推荐提供当前用户的偏好信息, 二者都可以产生一个更好的推荐效果。通过在两个真实数据集 Yelp 和 LastFM 上进行实验比较与分析, 该方法的各项评价指标的性能优于现有对话推荐模型。

关键词: 对话推荐; 序列推荐; 交互式路径推理; 注意力机制; 推荐系统; 长短期偏好; 对话回合

中图法分类号: TP391 **文献标识号:** A **文章编号:** 1000-7024 (2023) 10-3160-10

doi: 10.16208/j.issn1000-7024.2023.10.037

Conversational recommendation method combining user's long and short term preferences

FENG Xing-jie, ZHANG He⁺

(School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

Abstract: Current conversational recommendation systems focus too much on users' current preference information and are designed more as search engine like tools, which lack personalization. To address this problem, a conversational recommendation method combined with sequence recommendation was proposed, in which sequence recommendation was applied to provide item scoring module in dialogue recommendation system with item candidates that better matched users' long- and short-term preferences. The path selection operation in the conversation recommendation also provided the current user preference information for the sequence recommendation, both of which produced a better recommendation effect. Through experimental comparison and analysis on two real datasets, Yelp and LastFM, the performance of the evaluation metrics of this method is better than that of the existing conversation recommendation model.

Key words: conversational recommendation; sequence recommendation; interactive path reasoning; attention mechanism; recommendation system; long and short term preferences; conversation turns

0 引 言

现有的对话推荐方法大多仅从对话数据中学习用户的有效偏好表示, 无法充分利用用户的历史交互数据, 进而导致推荐结果缺乏个性化。例如, 在给用户实时推荐感兴趣的项目时, 如果仅仅按照用户当前对于项目属性的拒绝或接受来选择下一步的交互策略, 很容易因为过于注重用户目前的偏好信息, 使得对话推荐系统的推荐结果与用户

采用搜索方式得到的结果并无较大差异, 缺乏推荐系统应有的个性化。

为了解决以上问题, 本文提出了一种结合用户长短期偏好的对话路径推理方法 (conversational path reasoning with long-term and short-term preferences, LSCPR)。在交互式路径推理的基础上, 采用注意力机制将用户的历史偏好作为选取当前项目候选集的影响因子, 从而能够在更少的对话回合中推荐尽可能准确的项目。本文的主要贡献可

收稿日期: 2022-03-21; 修订日期: 2023-09-26

基金项目: 国家自然科学基金青年基金项目 (61301245、61201414)

作者简介: 冯兴杰 (1969—), 男, 河北邢台人, 博士, 教授, 硕士生导师, 研究方向为推荐系统、数据库及数据仓库和智能信息处理理论与技术; ⁺通讯作者: 张贺 (1997—), 女, 河北邢台人, 硕士研究生, CCF 学生会员, 研究方向为深度学习和推荐系统。

E-mail: zhjiabei@yeah.net

总结如下:

(1) LSCPR 将历史交互数据单独充分利用到对话推荐系统中, 在对话推荐中融合用户历史交互的时序动态信息, 使得路径推理过程中选取的项目候选集更能符合用户个性化的实时偏好。

(2) LSCPR 采用注意力机制从历史交互数据的上下文中计算注意力权重, 并根据用户最近的兴趣增强相应的注意力权重。

(3) 在两个真实数据集上进行大量实验, 实验结果表明, LSCPR 方法能够在保证平均对话回合数的同时, 提供更为精确的推荐结果。

1 相关工作

1.1 对话推荐方法

已有的对话推荐研究, 大多聚焦于设计算法去输出最佳交互策略, 以快速降低用户兴趣偏好的不确定性, 并实现对当前用户的成功推荐。早期研究中利用基于人工设计的规则和逻辑来删除属性不满足的项目来减少候选集的大小^[1]。文献 [2] 将协同过滤的推荐算法思想引入到对话推荐系统, 以提升对话推荐结果的准确率, 同时协同过滤算法^[3]中存在的冷启动、可解释性较差的缺陷也可以通过对话推荐的多轮交互特征有效弥补。近期神经网络已经被广泛应用于对话推荐方法中, 除了常见的 CNN 和 RNN 被用来建模具有时序性的对话数据^[4], GNN 也可以用来建模图类型数据, 从而学习对话交互信息中的复杂关系^[5,6]。与此同时, 为了解决动态交互环境下复杂的长期规划问题, 深度强化学习被引入到对话推荐算法的设计中, 一方面基于知识增强的强化对话推荐算法^[7]可以应用开源的知识库 DBpedia 来增强项目和属性的向量表示, 学习到推荐项目和交互属性之间更多的语义信息, 提高推荐效果; 另一方面基于图推理的强化对话推荐算法^[8]将用户的交互过程建模为无向图中路径推理的过程, 通过强化学习来判断每行走一步所应采取的动作 (询问或者推荐), 最终到达用户的目标项目。然而以上各种方式的对话推荐方法很少甚至没有考虑用户的历史交互信息, 这就导致了建模过程中缺少了用户的一般偏好信息, 而且诸多的对话推荐系统以减少系统与用户的交互回合为目标, 可能会遗漏一些有用的属性选项。此外仅利用基于属性的偏好不足以筛选项目候选集, 所以需要更加充分地利用用户的历史交互信息, 从而在对话系统的效率和推荐结果的质量上达到更优。

1.2 序列推荐方法

推荐系统的实际应用中, 采用所有的数据信息来训练模型学习可以表示用户兴趣偏好的嵌入, 但同时用户的序列行为信息也是尤为重要的, 它代表着用户偏好的变化状态, 所以序列推荐的出现可以显式地建模用户的序列行为, 提升推荐系统的效果。常见的序列建模方法有 Pooling、

RNN、CNN^[9]、Attention、Transformer^[10]等, 其中 transformer 中自注意力机制的应用可以在用户的交互序列中评估每个项目的相关权重^[11], 以便更好地学习用户的兴趣偏好表示以及捕获用户偏好的时序性, 提高目前对话推荐系统的推荐质量。

2 准备工作

2.1 对话推荐流程描述

对话推荐系统 (下文中称 CRS) 可以定义为: 通过使用自然语言或槽定义的方式与用户进行实时多回合交互, 引出用户动态偏好并根据用户当前需求采取相应的行动。目标则是在最少的对话回合内向用户推荐准确的项目。图 1 是多回合对话推荐场景的流程图, 可以将整个流程划分为 3 个模块, 分别是用户界面、推荐模块和对话策略模块。其中用户界面可以对系统的询问进行反馈以及对系统推荐的项目进行接受或拒绝; 推荐模块用来学习和建模用户对项目和项目属性的偏好, 是得出询问用户的属性信息和提供推荐结果的基础; 对话策略模块是整个流程的核心模块, 通过决定询问能够消除项目不确定性的属性来辅助用户得到精确的推荐结果。

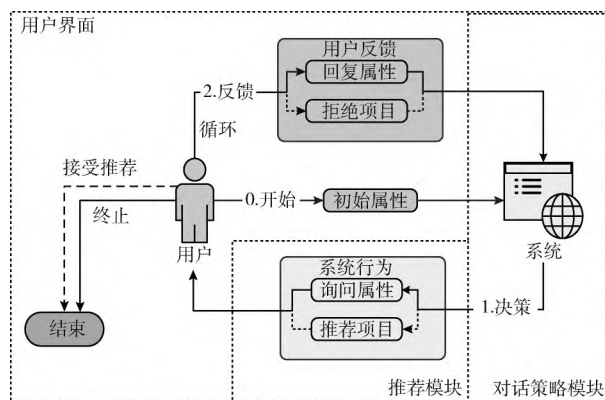


图 1 多回合对话推荐场景的流程

如图 1 所示, 对话推荐系统的详细描述为: 每一个对话 session 从用户侧开始, 用户侧有一个初始的属性 p_0 , CRS 会询问用户对于某一个属性的偏好 (从属性候选集 P_{cand} 中选取) 或者直接推荐项目 (从项目候选集 V_{cand} 中选取)。此时用户会有以下几种可能: ① 如果给出的是项目, 且用户接受了该项目, session 结束; ② 如果给出的是项目, 且用户拒绝了该项目, 则 CRS 把这个项目从 V_{cand} 中移除; ③ 如果给出的是属性, 且用户接受了该属性, 则把这个属性放进用户已选择的属性集 P_u 中, 并把其从 P_{cand} 中除去, 更新 $V_{cand} = V_{cand} \cap V_p$; ④ 如果给出的是属性, 且用户拒绝了该属性, 则 CRS 把这个属性从 P_{cand} 中移除。直到用户接受了 CRS 推荐的项目或者达到最大交互回合数, 则整个 session 结束。表 1 给出了本文常用的符号及其含义。

表 1 常见符号及其含义

符号	含义
u, v, p	用户, 项目, 属性
P	图中的属性路径
aa_i	属性 P_i 的一个邻接属性
AA_i	属性 P_i 的邻接属性集
P_u	在一个会话中, 用户 u 确定的属性集
P_{cand}	属性候选集
V_p	包含属性 p 的项目集
V_{cand}	项目候选集
a	采取的动作 (a_{ask} 或 a_{rec})

2.2 问题定义

给定 3 个分别包含用户历史交互数据、用户偏好信息、项目属性信息的数据集, 其中用户历史交互数据集中每个样本 (u, V_u) 表示用户 u 和与之交互过的项目集 V_u ; 用户偏好信息数据集中每个样本 (u, F_u, L_u) 表示用户 u 、与其具有双向好友关系的用户集 F_u , 以及用户 u 喜好的项目属性集 L_u ; 项目属性信息数据集中每个样本 (i, P_i) 表示项目 i 和其所属性集 P_i 。任务则是利用上述 3 个数据集中的信息构建包含用户、项目和属性 3 个实体的无向图,

从用户节点开始, 通过对话交互确定下一个要移动的节点, 目标是到达用户接受推荐的项目节点。由此将与用户的实时交互反馈过程结合用户历史偏好信息体现在无向图的路径推理中, 能够在最少的对话回合下给出准确的推荐项目, 同时推荐的可解释性也可以通过推理路径得以体现。

图 2 是一个对话推荐系统的过程示例。右图中的顶点表示用户、项目和项目的属性, 两个顶点之间的边表示它们之间的关系, 例如, 用户与项目之间的边表示用户已与该项目进行交互, 用户与属性节点之间的边则表示用户已在对话中确认该属性, 项目与属性节点之间的边表示项目拥有该属性等。图中首先从带有初始属性“唱跳”的用户“李明”开始进入对话, 所以回合 1 为从“李明”移动到“唱跳”节点。回合 2 系统进行邻接属性的搜索, 选择询问用户属性节点“摇滚”, 用户接受该属性, 则路径移动至“摇滚”节点。回合 3 与回合 2 相似, 路径移动至“流行”属性节点。回合 4 中系统已经掌握足够的用户偏好信息, 搜索同时拥有“唱跳”、“摇滚”、“流行”属性的项目, 选择“迈克尔”项目节点进行推荐, 用户接受推荐, 该对话结束。但需要注意的是, 如果用户拒绝询问的属性或推荐的项目, 系统将停留在原节点, 搜索其它符合用户偏好的属性重新进行询问, 多次重复该循环, 直至用户接受推荐的项目或者达到最大交互回合, 对话自动结束。

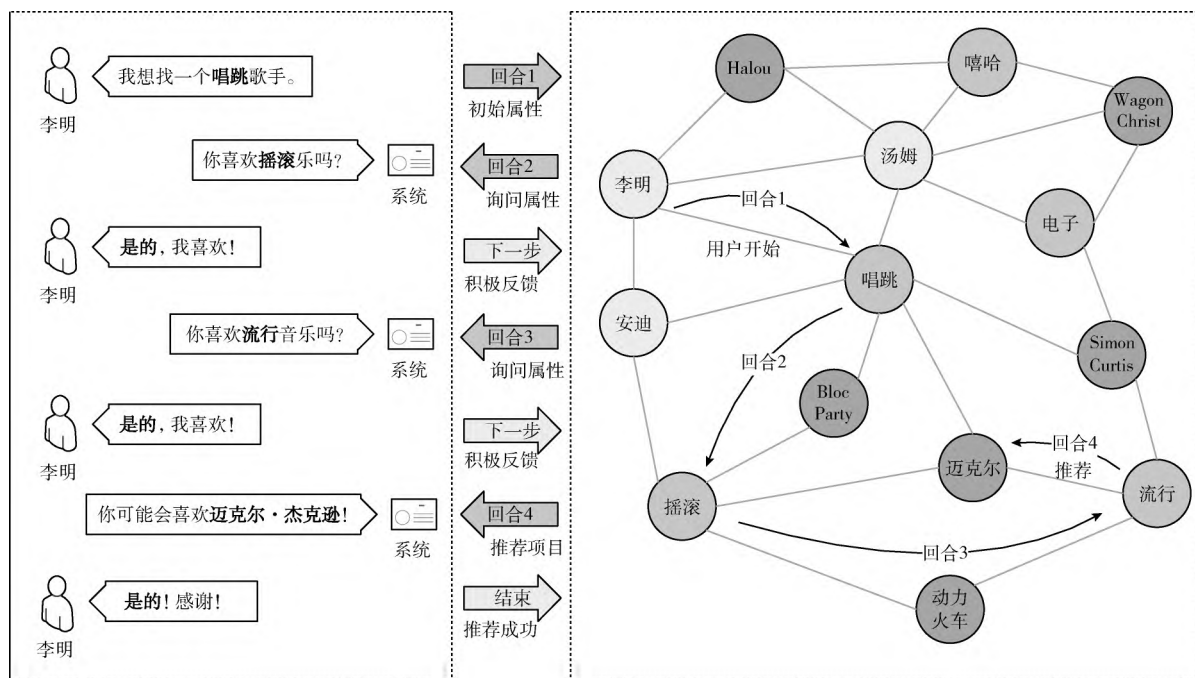


图 2 对话推荐系统过程示例

此外, 目前并没有成熟的对话推荐场景的数据集, 所以为优化对话推荐的策略, 大多研究采用用户行为数据集进行模拟对话交互训练数据的构建。将用户历史交互数据集以 7:1.5:1.5 的比例进行划分, 最大部分用来构造实体

无向图, 而剩余两部分跟随模型^[8]和模型^[12]中用户模拟数据集的设置, 生成用来训练和测试的对话交互数据。

2.2.1 实体无向图的构建

LSCPR 旨在通过图的路径推理来实现为用户的成功推

荐, 并为用户长短期偏好的结合提供先决条件。将数据集中用户、项目和属性实体构建成为无向图的过程见下述算法 1。

算法 1: 实体无向图构建

Input: 用户历史交互数据字典 $user_item$, 用户信息数据字典 $user_dict$, 项目信息数据字典 $item_dict$, 属性信息数据字典 $attribute_dict$ 。

Output: 实体-关系-实体无向图 G_{rec}

```
(1) entities = load_entities(user_dict, item_dict, attribute_dict) # 获取用户、项目、属性实体
(2) relations = load_relations(user_dict, item_dict, user_item, attribute_dict) # 获取实体间相应关系
(3) for relation r in relations do
(4)   for entity e1 in entities do
(5)     set = get_entity(r, e1) # 获取头实体 e1 的 r 关系的所有尾实体集
(6)     for entity e2 in set do
(7)       Grec[e1][r].append(e2) # 添加实体间的关系边
(8)     end for
(9)   end for
(10) end for
(11) clean(Grec) # 删除重复的边
(12) return Grec
```

根据用户、项目、属性和历史交互的数据提取相应的实体和关系, 将 G_{rec} 无向图的表示定义为包含头实体、关系、尾实体的嵌套字典。

2.2.2 模拟对话交互训练数据的构建

对话推荐是一个动态的过程, 并且现阶段没有成熟的数据集可以利用, 所以遵循 EAR^[12] 方法和 CPR^[8] 方法中用户模拟器的设定来构建对话交互训练数据, 完成对话推荐系统的训练和评估。正如前文中所提到的, 采用真实的用户项目的交互数据来实现这种模拟。根据模型训练的需要, 生成 $(user_id, item_id, neg_item, cand_neg_item, prefer_attributes)$ 结构的数据元组。详细来说, 给定一个用户 u 和拥有属性集 P_v 的项目 v , 不妨假设 $P_v = \{p_0, p_1, p_2, p_3, p_4\}$, 当前时刻用户确认的属性集为 $P_u = \{p_0, p_1, p_2\}$, 那么项目候选集 V_{cand} 为满足 P_u 中所有属性的项目集合。因此, 元组中元素 $user_id$ 和 $item_id$ 分别为 u 和 v , 第三个元素 neg_item 可以直接从用户 u 的非交互项目集中随机抽取, 第四个元素 $cand_neg_item$ 则是从项目候选集 V_{cand} 未与用户交互的项目中随机抽取, 第 5 个元素为当前回合中用户的偏好属性 P_u , 即 $[p_0, p_1, p_2]$ 。

3 模型

本文模型是一种结合用户的长短期偏好的图推理路径

模型, 其中用户的短期偏好可以在实时与用户交互过程中进行获取, 而长期偏好可以根据对话之前的用户历史交互数据进行提取, 从而辅助对话过程中的项目候选集的更新与选取, 最终为用户提供符合其长短期偏好的项目。LSCPR 模型结构如图 3 所示。首先提取用户、项目和属性相关数据集的实体和关系构建实体无向图, 详细过程见章节 2.2.1。然后通过用户与系统之间的实时交互进行图上路径的推理, 根据用户的反馈信息, 更新相应的项目和属性候选集, 同时策略模块判断系统下一步动作是询问属性还是推荐项目, 如果动作为询问属性, 则选取属性预测模块中得分最高的属性进行询问; 如果动作为推荐项目, 则选取项目预测模块中前 10 个项目进行推荐。以此循环, 直至推中用户心仪的项目或是达到最大交互回合数, 对话结束。为更好地解释 LSCPR 模型的路径推理过程, 现假设当前推理路径为 $P = p_0, p_1, \dots, p_t$, 系统保持在 p_t 节点位置, 并尝试寻找下一个属性或项目节点的位置。该过程可以通过预测、决策、更新 3 个阶段完成, 分为以下 4 个模块实现。

3.1 预测阶段

该阶段是一个回合的开始, 当一个属性被用户预定义或者确认时执行, 需要通过预测用户最可能接受的项目和属性来为决策阶段提供相应的选择空间。另外由于属性和项目的预测是相互依赖的, 所以可以采用交替优化策略来使其产生更好的预测结果。

3.1.1 项目预测模块

当前时刻用户 u 的属性偏好集 P_u 由此时推理路径 P 中的所有属性节点组成, 即 $P_u = \{p_0, p_1, \dots, p_t\}$, 所以项目候选集的选取是满足当前属性偏好集中各属性的项目集合。图 3 中无向图和交互部分为 $t=1$ 时的简要对应展示, 用户的首选属性为 p_0 , 那么 $P_u = \{p_0\}$, 所以项目候选集即需要进行预测的项目集为 $\{v_0, v_1, v_2\}$ 。但随着推理路径变长, 相应的项目候选集也会变大, 导致计算效率降低, CPR 模型直接选取所得项目候选集的前 1000 个项目进行后续预测^[8], 为减少该设置方法的随机性, 本文遂采用基于注意力机制的序列推荐方法对未交互的项目进行排序, 再选取其排序完成后的前 1000 个项目作为项目候选集。该处理逻辑可以充分利用用户的历史交互数据, 将其与现阶段的属性偏好信息相结合, 得到更具个性化的项目候选集合。

将 $X = \{x_1, x_2, \dots, x_t\}$ 表示历史交互项目序列的嵌入向量, 其中 x_t 表示最近的交互项目的嵌入, m_s 表示用户当前的一般兴趣, 可以由平均历史交互项目的嵌入而得到, 表示如下

$$m_s = \frac{1}{t} \sum_{i=1}^t x_i \quad (1)$$

用户的长期偏好通过注意力机制获取, 采用前馈神经网络生成每个交互项目嵌入向量的注意力权值 a_i , 表示当前项目交互序列中各嵌入向量的注意力系数, 注意力系数

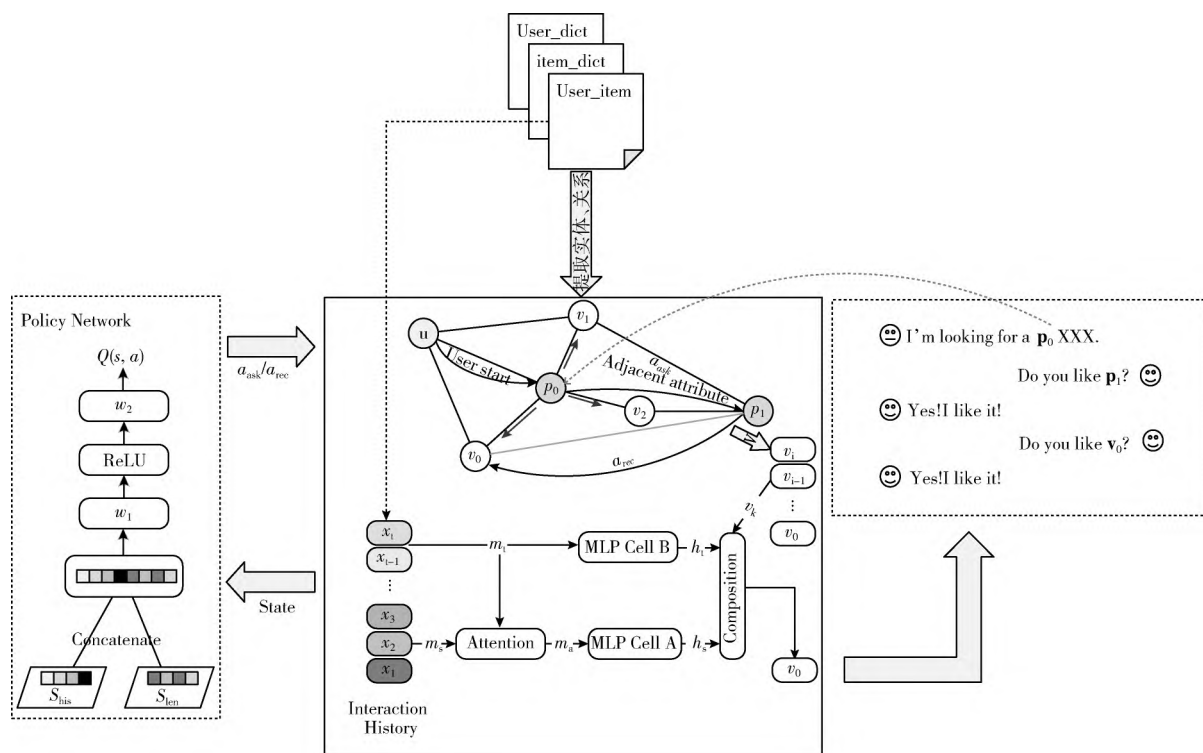


图3 LSCPR 模型结构

向量 $a = (a_1, a_2, \dots, a_t)$ 定义如下

$$a_i = W_0 \sigma(W_1 x_i + W_2 x_t + W_3 m_s + b_a), i \in [1, t] \quad (2)$$

其中, x_i 表示第 i 个项目的嵌入向量, W_0 是一个加权向量, W_1, W_2, W_3 是相应的加权矩阵, b_a 为偏置部分。在得到注意力系数向量 a 之后, 基于注意力的用户长期偏好表示如下

$$m_a = \sum_{i=1}^t a_i x_i \quad (3)$$

同时 $m_t = x_t$, 将 m_a 与 m_t 分别通过相应的多层感知机结构得到 h_s 与 h_t , 因此可以通过余弦相似度计算未交互项目集中各个项目 v_i 的得分, 从而经过得分排序获得相应的项目候选集。项目评分过程函数可以表示为

$$\hat{z}_i = \sigma(\langle h_s, h_t, v_i \rangle) \quad (4)$$

其中, $\sigma(\cdot)$ 表示 sigmoid 函数, 为方便 $\langle \cdot \rangle$ 的表述, 将变量 h_s, h_t, v_i 替换为维度为 d 的 a, b, c 这 3 个向量, 通过以下公式进行计算

$$\langle a, b, c \rangle = \sum_{i=1}^d a_i b_i c_i = a^T (b \odot c) \quad (5)$$

其中, \odot 表示哈达玛积, 即向量 b 和 c 之间的元素乘积。

然后, 在当前对话状态下用户 u 对给定项目候选集中某项目 v 的喜欢程度可以通过 FM^[13] 来进行预测, 用户-项目对交互的可能性如下

$$f(u, v, P_u) = u^T v + \sum_{p_i \in P_u} v^T p_i \quad (6)$$

其中, P_u 是当前推理路径中所有属性节点的集合, u, v 和 p_i 分别表示用户、项目和属性的嵌入, 第一项建模了从用户到项目的信息, 第二项建模了从用户的偏好属性到项目的信息。

3.1.2 属性预测模块

首先, 根据图结构的自然约束性, 仅将当前用户所有偏好属性的邻接属性节点作为属性候选集, 其中邻接属性节点的定义为: 如果图中属性 p_i 和属性 p_a 之间的最短路径不包含任何其它属性, 那么属性 p_a 就是属性 p_i 的邻接属性。例如, 在图 3 中, p_1 为 p_0 的邻接属性。因此, 当前时刻的属性候选集为删掉用户接受或拒绝的属性的 p_i 邻接属性集。需要注意的是, 属性预测模块的作用不同于项目预测模块, 需要根据当前系统状态来决定哪些属性是值得询问的, 期望找到一个可以更好地消除项目不确定性的属性。所以采用了可以有效评估不确定性的信息熵的方法, 属性的不确定性越大, 对应的信息熵越大, 信息熵计算表示如下

$$h(u, p, V_{cand}, P_u) = -p(p) \cdot \log_2(p(p)) \quad (7)$$

$$p(p) = \frac{\sum_{v \in V_{cand} \cap V_p} \sigma(f(u, v, P_u))}{\sum_{v \in V_{cand}} \sigma(f(u, v, P_u))} \quad (8)$$

其中, $\sigma(\cdot)$ 是 sigmoid 归一化函数, V_{cand} 表示项目候选集, V_p 表示包含属性 p 的项目集, $f(u, v, P_u)$ 为章节 3.1.1 中介绍的项目预测公式。

3.1.3 模型学习

为了训练模型, 需要提供正样本和负样本, 并采用成对损失来进行优化。在项目预测模块, 将交互过的项目视为正样本, 负样本包含两类, 一类是用户未交互过的项目 V_u^- , 另一类是在项目候选集中用户未交互过的项目 \hat{V}_u^- 。模型的目标函数定义如下

$$L_{item} = \sum_{(u,v,v') \in S_1} -\ln \sigma(f(u,v,P_u) - f(u,v',P_u)) + \sum_{(u,v,v') \in S_2} -\ln \sigma(f(u,v,P_u) - f(u,v',P_u)) + \lambda_{\Theta} \|\Theta\|^2 \quad (9)$$

$$S_1 := \{(u,v,v') \mid v' \in V_u^+\}, V_u^- := V \setminus V_u^+ \quad (10)$$

$$S_2 := \{(u,v,v') \mid v' \in \hat{V}_u^-\}, \hat{V}_u^- := V_{cand} \setminus V_u^+ \quad (11)$$

S_1 中数据用来建模用户的一般偏好, S_2 中数据根据用户交互过程中更新的项目候选集 V_{cand} 获得, 建模了用户的实时偏好。 V 为包含所有项目的项目集合, V_u^+ 为用户交互过的项目集合, 采用 Adam 作为优化器进行优化。

3.2 决策阶段-基于 RL 的策略模块

完成项目和属性预测之后便进入决策阶段, 该阶段的目的是决定下一步动作为询问属性 a_{ask} 还是推荐项目 a_{rec} , 使用户在最少的对话回合内获得准确的推荐。具体实现过程遵循文献 [8], 将其作为一个强化学习问题来处理, 强化学习的策略函数基于全局对话状态来进行决策, 其中可以包括对话历史、候选项目信息等。策略函数根据相应的输出回报设置得到包含 a_{ask} 和 a_{rec} 两个操作空间选择的输出结果。若结果为 a_{ask} , 则通过属性预测模块选取属性候选集中得分最高的属性进行下一步询问操作; 若结果为 a_{rec} , 则将项目预测模块中得到的 top-k 项目向用户进行推荐。

3.3 更新阶段-候选集更新模块

更新模块将在用户确认请求的属性之后被触发。如章节 2.1 中所描述的, 将确认的属性添加到用户偏好属性集 P_u 中, 初始项目候选集被更新为 $P_{cand} = AA_t(P_u \cup P_{rej})$ 。需要注意的是, 假设一个属性被用户拒绝, 只需要将它从属性候选集中删除, 而不需要进行其它操作。在更新阶段之后, 对话推荐系统进入到下一个对话回合。

4 实验

4.1 数据集

因为本文的研究侧重于对话策略的研究而并非语义的理解, 所以选用的数据集为推荐系统通用的交互数据集。为了更好地对比, 本文选用与 EAR 模型^[12]、CPR 模型^[8]一致的 LastFM (<https://grouplens.org/datasets/hetrec-2011/>) 和 Yelp (<https://www.yelp.com/dataset/>) 数据集进行实验。LastFM 是 2011 年 HetRec 会议发布的音乐推荐数据集, 来源于英国的音乐社 Last fm, 包含用户的听歌记录、用户属性等, 其中存在 1801 个用户、27 675 个项目和

76 693 个交互; Yelp 数据集是由美国点评网站 Yelp 公开的内部数据集, 现已成为通用数据集, 包含用户点评数据、商家属性等, 同时用户可以在该平台上交流购物体验, 其中存在 27 675 个用户、70 311 个项目和 1 368 606 个交互。基本的数据统计信息见表 2。对于两个数据集, 删除少于拥有 10 个相关项目的属性。

表 2 基本数据统计信息

数据集		LastFM	Yelp
类别	用户	1801	27 675
	项目	7432	70 311
	交互	76 693	1 368 606
	属性	8438	590
图	实体	17 671	98 576
	关系	4	3
	三元组	228 217	2 533 827

其中关系的详细数据见表 3。

表 3 关系的详细数据

关系	描述	LastFM	Yelp
交互	用户-项目	76 696	1 368 606
朋友	用户-用户	23 958	688 209
喜欢	用户-属性	33 120	0
属于	项目-属性	94 446	477 012

4.2 评价指标

对话推荐系统的评价指标分为两个级别^[14]。第一个级别是对推荐结果的评价, 其中值得评价的是推荐结果的质量或性能, 主要采用推荐算法中常用的召回或排序的指标, 如 MSE、RMSE、Precision、Recall、MRR、NDCG、HR 等。第二个是对话级别的评价, 其中一类值得评价的是自然语言生成的质量, 指标包括 BLEU 与 ROUGE 等; 另一类评价针对对话推荐系统的效率, 主要关心的是对话的平均回合数 (average turns, AT), 在对话推荐任务中, 越早推中用户喜欢的项目越好, 故对话的回合数越小越好。另外一个指标是对话在固定最大回合数的情况下推荐成功的概率 (Success Rate at t -th turn, SR@ t)。

因为本文主要研究的对话推荐策略的改进, 所以不考虑自然语言的生成, 针对对话推荐的结果本文采用 MRR 和 HR 评价指标。MRR 为平均倒数排名, 其强调在成功推荐的对话回合, 是否将用户的真实访问项目置于推荐项目列表中更靠前的位置, 定义为

$$MRR = \frac{1}{M} \sum_{i=1}^M \frac{1}{p_i} \quad (12)$$

其中, M 表示用户的个数, p_i 是对于第 i 个用户, 用户的

真实访问项目在推荐列表的位置。HR 为命中率, 反映在推荐项目列表中是否存在用户的真实访问项目, 定义为

$$HR = \frac{1}{N} \sum_{i=1}^N hit(i) \quad (13)$$

其中, N 表示推荐的次数, 即用户的对话回合数。 $hit(i)$ 表示是否命中用户的真实访问项目, 即用户选择的项目是否在推荐列表中, 若存在, $hit(i)$ 值为 1, 否则为 0。针对对话系统效率的评价指标为 AT 和 SR@ t 。AT 定义为

$$AT = \overline{dialogue\ length} \quad (14)$$

成功率指标 SR@ t 定义为在 t 个交互回合内推荐成功的概率, 计算公式如下所示, 其中 $\#$ 表示事件发生的次数

$$SR = \frac{\#successful\ dialogues}{\#dialogues} \times 100\% \quad (15)$$

4.3 对比模型

为了评估 LSCPR 模型的性能, 将 LSCPR 与以下代表性基线方法进行对比实验, 各方法的具体区别见表 4。

表 4 对比模型比较

模型	对话策略 模块类型	技术手段	ASK 动作空间	交互轮次	
				单轮	多轮
Max Entropy	基于人工规则	最大熵	属性		✓
Abs Greedy	基于人工规则	EE 算法	项目		✓
CRM	基于强化学习	LSTM、 强化学习	属性	✓	
EAR	基于强化学习	强化学习	属性	✓	
CPR	基于强化学习	GNN、 强化学习	属性	✓	
SASRec		自注意力机制			
AttRec		自注意力机制			

(1) Max Entropy。这个方法遵循在章节 2.2.2 中介绍的用来构建对话历史记录规则, 每个对话回合都会对候选属性项中熵最大的属性进行询问。

(2) Abs Greedy^[15]。这个方法在每个回合都会推荐项目, 而不会询问任何问题。一旦推荐被拒绝, 它会通过将拒绝的项目作为负面样本来更新模型。根据文献^[15]实验结果, 该方法与常用的线性 bandits 算法 (Thompson 采样^[16]和 UCB-based 算法^[17]) 相比具有同等或者更好的性能。

(3) CRM^[18]。这个模型最初是为单轮次对话推荐而设计的, 即进行推荐行为之后不论是否推荐成功都会结束对话。它将表示对话当前状态和用户偏好的潜在向量空间输入深度策略网络, 来决定下一步是进行推荐还是询问, 同时根据策略梯度算法调整对话策略和更新潜在向量空间。

(4) EAR^[12]。这是首个着重体现推荐模块与对话模块深度交互的模型, 提出使用交叉熵信息、用户反馈信息、

对话回合信息构造状态向量, 作为策略网络的输入, 使策略模型能够更好地适应动态环境, 学习到更加灵活的对话策略。

(5) CPR^[8]。该模型将图推理与对话推荐结合起来, 将用户和项目信息构建成无向异构图, 将用户实时交互的过程建模成图上的路径游走的过程, 通过在图上游走和推理的方式选择具体的属性节点和项目节点, 而策略网络只需要输出询问与推荐两个动作, 使得策略网络的训练变得更加容易。

(6) SASRec^[19]。通过自注意力机制对用户的历史交互信息进行抽象, 将抽象得到的用户信息与所有项目的嵌入做内积交互, 得到用户与每个项目的相关性概率。

(7) AttRec^[20]。采用自注意力机制对用户的历史交互项目之间的关系进行捕获, 得到用户的短期偏好表示; 采用协同度量学习建模用户的长期偏好, 结合后加权得到用户对候选项目的交互概率。

4.4 实验细节

本文使用 pytorch 对所提模型进行实现, 并使用 GPU (GTX 1080Ti) 对训练进行加速。设置 top- k 的值为 10, 最大的对话回合数 T 为 15。在项目和属性预测训练模块根据经验设置如下: FM 的嵌入维度大小设置为 64, 项目和属性预测任务的学习率分别设置为 0.01 和 0.001, 使用了 L2 正则化系数为 0.001 的 SGD 优化器; 在基于 RL 的策略模块中, 训练奖励为: $r_{rec_suc} = 1$, $r_{rec_fail} = -0.1$, $r_{ask_suc} = 0.01$, $r_{ask_fail} = -0.1$, $r_{quit} = -0.3$ 。DQN 的参数设置为: 经验回放内存大小为 50 000, 折扣因子 γ 被设置为 0.999。

4.5 性能对比

4.5.1 对话推荐质量

由于该部分主要对所提出对话推荐方法的推荐结果的质量进行讨论, 所以不采用 Abs Greedy 和 Max Entropy 作为对比方法。本文提出的 LSCPR 方法与其它对比模型在推荐质量方面的性能比较见表 5。

表 5 不同模型对话推荐性能比较 (质量)

Model	LastFM		Yelp	
	MRR	HR@10	MRR	HR@10
CRM	0.0552	0.2946	0.0924	0.4809
EAR	0.0498	0.2790	0.0838	0.4677
CPR	0.0563	0.2983	0.0954	0.5021
SASRec	0.0783	0.3401	0.0794	0.3958
AttRec	0.0787	0.3466	0.0825	0.4191
LSCPR	0.0799	0.3672	0.1001	0.5237

分析实验结果可以得出: 本文提出的 LSCPR 模型在两个数据集中的每个评价指标都优于其它对比模型。观察其

它模型发现,在数据量较大的 Yelp 数据集上对话推荐方法 (CRM、EAR、CPR) 的整体性能会比传统序列推荐方法 (SASRec、AttRec) 更优;在数据量较小的 LastFM 数据集上结果则相反。本文认为当中的原因在于在数据量较大的候选集上,单单利用用户的历史交互信息进行用户兴趣偏好的表示过于片面,并且可能会与用户的当前偏好信息产生偏差,所以通过对话交互的方式获取的用户实时偏好信息能够产生更好的效果。

4.5.2 对话系统效率

该部分主要对对话推荐方法的推荐效果和交互效率进行讨论,所以对比模型为涉及对话推荐思想的方法。本文提出的 LSCPR 和其它对比模型的评价指标实验结果见表 6,其中为更直观地看出在数据量较大的 Yelp 数据集上各模型之间的性能对比。

表 6 不同模型对话推荐性能比较 (效率)

Method	LastFM		Yelp	
	SR@15	AT	SR@15	AT
Abs Greedy	0.482	11.95	0.178	14.94
Max Entropy	0.558	12.88	0.187	14.24
CRM	0.572	10.76	0.183	14.77
EAR	0.633	10.19	0.226	13.81
CPR	0.692	9.69	0.411	12.23
LSCPR	0.710	8.94	0.437	11.17

首先,本文模型 LSCPR 在两个数据集上的评价指标 (推荐成功率和平均对话回合数) 都优于其它对比模型。具体来说,与最优基线模型相比, LSCPR 在 LastFM 数据集上分别提高了 SR@15 和 AT 的 1.8% 和 0.75 回合。同时,在 Yelp 数据集上的性能提升比 LastFM 数据集显著,分别提高了 SR@15 和 AT 的 2.6% 和 1.06 回合。这说明在数据量较大时,历史交互数据的充分利用反而能够显示出更大的优势。图 4 为各模型与最优对比模型 CPR 在每个对话回合中成功率差值的情况。可以看出 LSCPR 模型处于平稳且优于最优模型的趋势。

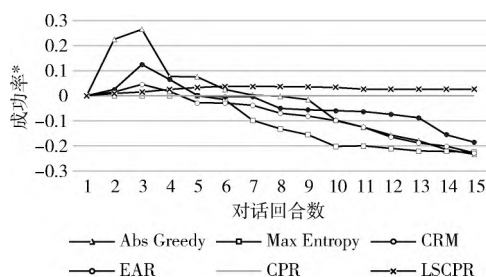


图 4 Yelp 上不同回合数的 SR* 对比

其次,对话推荐模型 (CRM、EAR、CPR、LSCPR)

的整体性能大体上比传统意义上对话推荐方法 (Abs Greedy、Max Entropy) 更优。本文认为原因在于对话推荐模型采用基于强化学习的策略模块对对话环境充分建模,更新用户的实时交互反馈信息,能够及时调整相应的项目和属性候选集。其中 Abs Greedy 虽然性能较差,但是从图 4 可以看出它在前几个回合能够达到最佳效果,之后性能下降,推断原因在于它是一种尝试推荐项目的方法,而其它方法可以得到用户对项目和属性的明确反馈,从而减少候选项目空间,以此实现长期回报。另外,CRM 模型性能略差的原因考虑在于其是单轮次推荐,不管推荐成功与否,自动结束对话。

最后,在对话推荐模型中,使用图的路径推理 (CPR、LSCPR) 比不使用 (CRM、EAR) 更优。原因在于图节点关系的约束有助于消除许多不相关的属性,尤其是属性数量较大时。同时,候选空间限制在邻接属性或者项目使得对话流畅性提升。所以,本文基于图的路径推理来改进推荐方法,将用户历史交互信息采用注意力机制获取有价值的用户偏好表示,将其对项目候选集做进一步限定,从而得到更好的推荐效果。

4.6 消融实验

实验评估了用户长期偏好与短期偏好相结合给对话推荐系统性能带来的影响,为进一步验证用户长短期偏好结合的重要性以及注意力机制在用户长短期偏好信息融合方面所起到的作用,本文构建了以下 LSCPR 变体。

(1) LSCPR-AM: 忽略项目候选集预测模块中对注意力机制的应用,仅采用平均值的方法来代替 LSCPR 中利用注意力机制捕获用户历史交互数据中融合长短期偏好的表示。

(2) LSCPR-ST: 忽略项目候选集预测模块中对用户短期历史交互数据的应用,即不再单独考虑历史交互序列中最近的交互项目对于实时对话交互路径选取的影响。

(3) LSCPR-LS: 忽略项目候选集预测模块中对用户历史交互数据的应用,即仅采用实时对话交互信息来进行推荐,改变后的模型近似于 CPR 模型。

本文使用 SR@15 和 AT 评价指标来评估相应模块设定在 LastFM 数据集和 Yelp 数据集上的有效性。消融实验结果如图 5 所示。首先, LSCPR 模型的性能优于其它 3 个变体,验证了历史交互数据与用户实时对话交互信息相融合的必要,这两类信息相辅相成,互相提供有效信息,历史交互数据为项目候选集的选取增添用户个性化限制,同时通过实时交互获取的邻接项目节点为项目候选集的生成划定了相应的范围。此外,对历史交互数据的应用也应考虑长短期偏好的划分, LSCPR-ST 变体的实验验证了用户最近时间的历史交互项目对之后项目选取具有重要作用。 LSCPR-AM 变体则验证了注意力机制在长短期兴趣偏好融合表示的积极贡献。

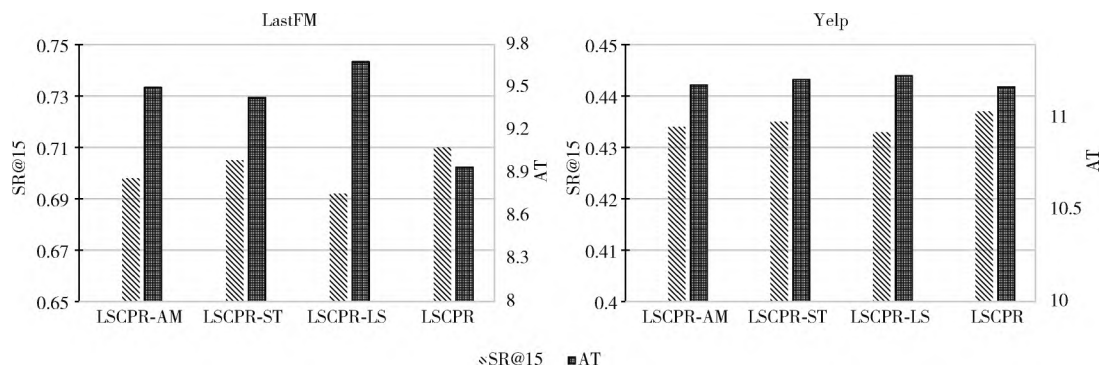


图5 历史交互数据应用对模型的影响

5 结束语

本文提出了一种结合用户长短期偏好信息的对话推荐方法 LSCPR, 该方法首先利用注意力机制将用户历史交互数据中的一般偏好与短期偏好进行加权表示, 再与通过用户、项目、属性三元无向图的路径推理得到的实时更新的未交互项目集进行线性融合, 最终求得项目集中每个项目的预测评分, 得到最终的项目候选集, 为用户提供更具个性化与实时性的推荐结果。最后, 基于两个真实数据集进行相应实验, 验证了对用户历史交互数据进行充分利用的必要性, 同时采用基于路径推理的对话推荐方法更好地实现了推荐结果的可解释性。

参考文献:

- [1] Luo K, Yang H, Wu G, et al. Deep critiquing for VAE-based recommender systems [C] //The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2020; 1269-1278.
- [2] ZHANG Qingbo, WANG Bin, CUI Ningning, et al. Attention-based regularized matrix factorization for recommendation [J]. Journal of Software, 2020, 31 (3): 778-793 (in Chinese). [张青博, 王斌, 崔宁宁, 等. 基于注意力机制的规范化矩阵分解推荐算法 [J]. 软件学报, 2020, 31 (3): 778-793.]
- [3] Zou L, Xia L, Gu Y, et al. Neural interactive collaborative filtering [C] //The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020; 749-758.
- [4] Greco C, Suglia A, Basile P, et al. Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems [C] //Conference of the Italian Association for Artificial Intelligence. Springer, 2017; 372-386.
- [5] Guo Q, Zhuang F, Qin C, et al. A survey on knowledge graph-based recommender systems [J]. Scientia Sinica Informationis, 2020, 50 (7): 937.
- [6] Zhou S, Dai X, Chen H, et al. Interactive recommender system via knowledge graph-enhanced reinforcement learning [C] //The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2020; 179-188.
- [7] Zhou K, Zhao W X, Bian S, et al. Improving conversational recommender systems via knowledge graph based semantic fusion [C] //The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2020; 1006-1014.
- [8] Lei W, Zhang G, He X, et al. Interactive path reasoning on graph for conversational recommendation [C] //Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2020; 2073-2083.
- [9] Liao Lizi, Ma Yunshan, He Xiangnan, et al. Knowledge-aware multimodal dialogue systems [C] //In Proceedings of the 26th ACM International Conference on Multimedia. Association for Computing Machinery, 2018; 801-809.
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need [C] //In NeurIPS, 2017; 5998-6008.
- [11] SUN Jing, SUN Jingyu, LI Can, et al. Recommendation algorithm based on user interest representation and attention mechanism [J]. Computer Engineering and Design, 2021, 42 (3): 814-821 (in Chinese). [孙静, 孙静宇, 李璨, 等. 融合用户兴趣表征与注意力机制的推荐算法 [J]. 计算机工程与设计, 2021, 42 (3): 814-821.]
- [12] Lei W, He X, Miao Y, et al. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems [C] //The Thirteenth ACM International Conference on Web Search and Data Mining, 2020; 304-312.
- [13] WEI Xiaohang, YU Chongchong, TIAN Changli, et al. Internet CTR prediction model on big data platform [J]. Computer Engineering and Design, 2017, 38 (9): 2504-2508 (in Chinese). [魏晓航, 于重重, 田嫦丽, 等. 大数据平台下的互联网广告点击率预估模型 [J]. 计算机工程与设计, 2017, 38 (9): 2504-2508.]
- [14] Gao C, Lei W, He X, et al. Advances and challenges in

- conversational recommender systems: A survey [J]. AI Open, 2021, 2: 100-126.
- [15] Jia Wu, Chen Xiu-Yun, Zhang Hao, et al. Hyperparameter optimization for machine learning models based on bayesian optimization [J]. Journal of Electronic Science and Technology, 2019, 17 (1): 26-40.
- [16] Yuval Lewi, Haim Kaplan, Yishay Mansour. Thompson sampling for adversarial bit prediction [J]. JMLR: Workshop and Conference Proceedings, 2020, 117 (4): 1-37.
- [17] Chen X, Ghahserifard B. Distinguished sets of semi-simple Lie algebras [J]. Journal of Algebraic Combinatorics, 2021, 54 (3): 879-891.
- [18] Sun Y, Zhang Y. Conversational recommender system [J]. AI Open, 2018, 2 (2021): 235-244.
- [19] Kang W C, Mcauley J. Self-attentive sequential recommendation [C] //IEEE International Conference on Data Mining. Singapore: IEEE, 2018: 197-206.
- [20] Shuai Z, Yi T, Yao L, et al. Next item recommendation with self-attention [C] //Recommender Systems and Natural Language Processing, 2019: 315-324.