软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(6):2974−2998 [doi: 10.13328/j.cnki.jos.006915]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

融合自适应周期与兴趣量因子的轻量级 GCN 推荐*

钱忠胜, 叶祖铼, 姚昌森, 张 丁, 黄 恒, 秦朗悦

(江西财经大学 信息管理学院, 江西 南昌 330013)

通信作者: 钱忠胜, E-mail: changesme@163.com

摘 要: 推荐系统在成熟的数据挖掘技术推动下, 已能高效地利用评分数据、行为轨迹等显隐性信息, 再与复杂而先进的深度学习技术相结合, 取得了很好的效果. 同时, 其应用需求也驱动着对基础数据的深度挖掘与利用, 以及对技术要求的减负成为一个研究热点. 基于此, 提出一种利用 GCN (graph convolutional network) 方法进行深度信息融合的轻量级推荐模型 LG_APIF. 该模型结合行为记忆, 通过艾宾浩斯遗忘曲线模拟用户兴趣变化过程, 采用线性回归等相对轻量的传统方法挖掘项目的自适应周期等深度信息; 分析用户当前的兴趣分布, 计算项目的兴趣量, 以获取用户的潜在兴趣类型; 构建用户-类型-项目三元组的图结构, 并结合减负后的 GCN 技术来生成最终的项目推荐列表. 实验验证所提方法的有效性, 通过与 8 个经典模型在 Last.fm, Douban, Yelp, MovieLens 数据集中的对比, 表明该方法在 *Precision*, *Recall* 及 *NDCG* 指标上都得到良好改善, 其中, *Precision* 平均提升 2.11%, *Recall* 平均提升 1.01%, *NDCG* 平均提升 1.48%.

关键词: 行为记忆; 自适应周期; 兴趣量因子; 图卷积网络; 推荐系统

中图法分类号: TP311

中文引用格式: 钱忠胜, 叶祖铼, 姚昌森, 张丁, 黄恒, 秦朗悦. 融合自适应周期与兴趣量因子的轻量级GCN推荐. 软件学报, 2024, 35(6): 2974–2998. http://www.jos.org.cn/1000-9825/6915.htm

英文引用格式: Qian ZS, Ye ZL, Yao CS, Zhang D, Huang H, Qin LY. Lightweight GCN Recommendation Combining Adaptive Period and Interest Factor. Ruan Jian Xue Bao/Journal of Software, 2024, 35(6): 2974–2998 (in Chinese). http://www.jos.org.cn/1000-9825/6915.htm

Lightweight GCN Recommendation Combining Adaptive Period and Interest Factor

QIAN Zhong-Sheng, YE Zu-Lai, YAO Chang-Sen, ZHANG Ding, HUANG Heng, QIN Lang-Yue

(School of Information Management, Jiangxi University of Finance & Economics, Nanchang 330013, China)

Abstract: Driven by mature data mining technologies, the recommendation system has been able to efficiently utilize explicit and implicit information such as score data and behavior traces and then combine the information with complex and advanced deep learning technologies to achieve sound results. Meanwhile, its application requirements also drive the in-depth mining and utilization of basic data and the load reduction of technical requirements to become research hotspots. On this basis, a lightweight recommendation model, namely LG_APIF is proposed, which uses the graph convolutional network (GCN) method to deeply integrate information. According to behavior memory, the model employs Ebbinghaus forgetting curve to simulate the users' interest change process and adopts linear regression and other relatively lightweight traditional methods to mine adaptive periods and other depth information of items. In addition, it analyzes users' current interest distribution and calculates the interest value of the item to obtain users' potential interest type. It further constructs the graph structure of the user-type-item triplet and uses GCN technology after load reduction to generate the final item recommendation list. The experiments have verified the effectiveness of the proposed method. Through the comparison with eight classical models on the datasets of Last.fm, Douban, Yelp, and MovieLens, it is found that the *Precision*, *Recall*, and *NDCG* of the proposed method are improved, with an average improvement of 2.11% on *Precision*, 1.01% on *Recall*, and 1.48% on *NDCG*, respectively.

Key words: behavior memory; adaptive period; interest factor; graph convolutional network (GCN); recommendation system

* 基金项目: 国家自然科学基金 (62262025); 江西省自然科学基金重点项目 (20224ACB202012)
收稿时间: 2022-06-18; 修改时间: 2022-10-27; 采用时间: 2023-02-07; jos 在线出版时间: 2023-08-09
CNKI 网络首发时间: 2023-08-10

1 引 言

推荐系统主要是在用户需求不明确或复杂时, 能结合历史行为来挖掘用户的特征、偏好等信息, 最终为用户提供服务以匹配相应需求.

协同过滤 (collaborative filtering, CF)[1]作为推荐领域中经典的算法之一, 近几十年来得到了广泛的发展和应用. 它通过结合不同群体 (用户和项目) 之间的相似规律来实现评分填充或项目推荐. 基于协同过滤的推荐模型能较充分地利用用户的全局信息, 获取较好的推荐精度, 但单一的协同过滤推荐模型对于数据的完整性要求极高, 故在实际使用中, 数据稀疏性、训练效率低等问题是这一类模型的通病. 而不可否认的是, 评分、浏览等基础信息确实最能展现用户兴趣也极易从中获取兴趣. 因此, 纵向深挖用户基础信息以获取深度信息 (如本文的自适应周期、兴趣量因子等) 的意义并不亚于横向融入近期热门辅助信息的作用. 换句话说, 基础信息中蕴含的大量深度信息在传统推荐模型中难以挖掘或易过拟合, 这种局限性导致很多推荐模型盲目地朝着融入更多辅助信息的方向发展, 使得模型变得异常复杂.

基于深度学习的推荐模型对挖掘用户-项目间的非线性关系有着独特优势, 且其抽象编码能力可表示更高层次的交互信息, 解决了很多传统推荐模型面临的问题, 其中采用多种神经网络组合成的混合神经网络[2]表现出的优势更为明显. 近些年热门的基于图卷积网络 (graph convolutional network, GCN) 的推荐模型[3−5], 采用 GCN 方法将用户-项目图结构信息输入模型中, 最大程度上保留了潜在信息, 能很好地实现协同过滤的思想. 然而, 这些工作尽管也关注用户的基础信息, 但对用户和项目的高阶交互信息挖掘程度仍较浅. 此外, 很多研究采用繁琐的操作来适应结构更复杂的数据, 导致模型的复杂度成倍增加. 数据的庞杂性加大了对复杂技术的依赖, 使模型构件冗余的现象愈加严重. 但事实上, 结合传统经典技术, 或轻量级神经网络技术的推荐, 不仅能降低复杂度, 性能也并不逊色, 因此, 挖掘合适的深度信息对模型推荐性能的提升至关重要.

针对上面的问题, 本文以深度信息和轻量级推荐为基础 (其中, 深度信息主要指自适应周期和兴趣量因子, 轻量级则主要体现在模型所运用的技术复杂度方面), 提出一种融合自适应周期与兴趣量因子的轻量级 GCN 推荐方法 (lightweight GCN recommendation method combining adaptive period and interest factor, LG_APIF). 主要工作如下.

1) 结合艾宾浩斯 (Ebbinghaus) 遗忘曲线构建用户兴趣模型, 用记忆规律来模拟用户的兴趣变化, 保证用户长期兴趣存活性的同时, 充分发掘潜在短期兴趣.

2) 引入自适应周期和兴趣量因子等深度信息, 采用聚类和线性回归技术对其求解, 这在细化用户兴趣变化过程的同时, 去除传统 GCN 中特征融合等冗余操作, 使模型更有效地学习嵌入表示.

3) 利用 GCN 技术传播用户-类型-项目三元图结构的信息, 使用户的潜在嵌入能融合更多的协作信号, 且让图卷积层能更关注于拟合嵌入关系, 而不易受其他操作带来的噪声影响.

4) 基于 4 个数据集和 8 个经典的相关对比模型, 设计消融、对比、优化等不同类型实验, 由分析结果表明, 模型 LG_APIF 的整体效果最佳, 进一步论证了本文重点挖掘的特征确实可提高推荐效果.

2 相关工作

协同过滤是一种利用用户或项目间相似度来预测的经典算法[1], 基于协同过滤的推荐模型能较充分地利用全局信息, 但单一的协同过滤推荐模型很难适应不同的场景. 比如, 为实现个性化推荐而细化用户的不同兴趣, 就要求 CF 可融入时间等方面的因素并关注到局部信息. 因此, 在推荐系统的发展过程中, 很多基于 CF 的时序预测和用户兴趣预测方法脱颖而出, 主要包括: 时间感知邻域模型 (time-aware neighborhood model)、基于聚类 (clustering) 的方法、基于线性回归 (linear regression) 的方法、基于深度学习 (deep learning) 的方法.

时间感知邻域模型[6]在协同过滤的基础上, 对时序信息进行建模, 即, 在挖掘用户-项目信息的过程中, 将用户交互过的项目按时间轴顺序展开, 使用户对项目的关注度也随时间分布. 这种方法虽扩展了传统模型对时序信息的挖掘与利用, 但未能弥补传统模型固有的缺陷, 比如, 无法区别长短期兴趣.

随着数据挖掘技术的发展, 聚类方法得到广泛研究, 很多学者用其解决推荐问题. 徐晓等人[7]研究一种基

© 中国科学院软件研究所 http://www.jos.org.cn

于密度的聚类改进算法, 并探讨其在推荐领域的应用. Ferdaous 等人[8]利用网络相关性和特征聚类, 推出一种好友推荐算法. Tey 等人[9]为解决推荐中的冷启动问题, 采用一种融合社交影响力的聚类算法. 王俊等人[10]对 CF 算法加以改进, 提出一种基于用户项目聚类的可扩展方法. Xu 等人[11]将用户间的信任关系进行聚类, 并将聚类结果融入 CF 算法中进行推荐. 但这些模型所使用的信任值均由数据集直接给出, 导致数据来源成为这类模型的最大障碍.

线性回归也是推荐中的常用技术, 这得益于它在拟合函数和趋势分析上具有的独特优势. 文献 [12] 通过最小二乘法等方式计算出每种浏览行为的权重, 构建多种浏览行为的多元线性回归函数, 再以每种浏览行为的权重与其量化特征做内积, 获得用户对项目的偏好得分. 刘兴波等人[13]提出一种基于双向线性回归的监督离散型跨模态散列方法, 该方法仅用一个稳定的映射矩阵来描述散列码与相应标签间的线性回归关系, 提升了跨模态散列学习的精度和稳定性. 基于线性回归的推荐大多是根据浏览项目内容大小, 对浏览时间和访问次数进行但用户的多元化属性间大都不存在线性关系, 导致基于线性回归的方法具有较大的局限性.

循环神经网络 (recurrent neural network, RNN) 作为一种优秀的序列问题处理技术, 能非线性化地拟合多元化属性, 被广泛运用到各种推荐[14,15]中. 比如, 动态回归模型 (dynamic regression model) 使用 RNN 来优化自身, 使其在下一个篮子推荐[16]中的效果得到很大的提升. 但 RNN 对数据的要求较高, 需要数据有统一格式, 比如一维向量、二维矩阵、三维张量等, 即欧氏空间的数据, 很难处理类似图结构等复杂混合型的数据. 这些问题的提出, 使得主要解决图论问题的 GNN (graph neural network) 方法开始得到很多推荐领域学者的关注. 但直到 2014 年, Perozzi 等人[17]提出了 DeepWalk, 意味着图表示学习技术在这个深度学习时代才正式被激活, 之后便得到更为深入的应用. Wu 等人[18]提出深度影响传播模型 (DiffNet++) 来建模社交推荐中的递归动态社交扩散过程, 重点考虑了用户之间的社交网络以及用户和项目间的兴趣网络.

之后, 为使图表示学习能更好地应用于协同过滤的相关研究中, 且在嵌入过程中尽可能地保留用户-项目之间的潜在特征, Wang 等人[3]运用 GCN 技术来实现协同过滤, 提出 NGCF 模型. 在此基础上, Wu 等人[19]认为 GCN 具有不必要的复杂性, 通过消除非线性和多个权重矩阵, 提出一个简易的 GCN 模型. 而 He 等人[4]在 NGCF 的基础上, 更大程度地去除特征融合及非线性转化的过程, 提出一种轻量级的 GCN. 这些工作给予了很多学者启示, 更多相关且性能更好的推荐模型[20,21]不断推出.

上述已有研究表明, 大多数协同过滤推荐模型主要考虑如何对基础信息进行填充或引入辅助信息加以利用, 未更大程度地挖掘在用户与项目基础信息中所蕴藏的深层信息; 此外, 在抽象编码用户行为表示时, 也仅实现最显式的关系, 未考虑更高阶的用户-项目交互关系, 且尽管 GCN 技术在推荐中表现优异, 但对其一些关键构件的必要性仍存在争议. 因此, 本文提出 LG_APIF 模型来捕捉用户的整体兴趣, 先利用较为轻量的传统技术来挖掘用户的自适应周期和兴趣量因子等深度信息, 再进一步归结行为规律, 预测用户的兴趣项目类型, 以建立用户-类型-项目三元图结构信息, 最终结合相对轻量的 GCN 技术构建推荐模型.

3 融合深度信息的 GCN 自适应推荐策略

传统的协同过滤推荐模型, 大多数将侧重点放在用户的相似性上, 即用户之间是否浏览了同一类项目, 或项目的相似性上, 即项目之间是否存在功能或特征上的替换性及互补性. 但在用户和项目基础特征中, 还蕴藏着很多对推荐效果起着重要作用的深度信息, 它们十分有助于推荐系统的发展. 因此, 提出一种轻量级推荐模型 LG_APIF, 来挖掘与利用这些深度信息, 并最终实现推荐.

下面先给出本文模型 LG_APIF 的整体框架, 如图 1 所示; 再详细介绍兴趣模型构建、自适应周期预测、兴趣量因子解析以及融合深度信息的轻量级 GCN 推荐. 模型 LG_APIF 的工作原理如下.

1) 构建兴趣模型. 将用户的历史记录转化成以时间间隔为单位的行为记忆序列, 结合艾宾浩斯遗忘曲线构建可计算用户兴趣程度的兴趣模型, 同时引入并应用自适应周期和兴趣量因子这两个重要的深度信息.

2) 预测自适应周期. 结合转化后的行为记忆序列, 将同种项目类型下不同用户间的兴趣周期进行聚类, 再利用

© 中国科学院软件研究所 http://www.jos.org.cn

线性回归方法, 拟合出同种项目类型在不同兴趣形态下的兴趣函数, 最后根据目标用户对该项目类型的当前兴趣形态, 匹配兴趣函数以预测下一个行为的自适应周期.

3) 解析兴趣量因子. 重组行为记忆序列, 对时间进行切块, 构建用户-项目兴趣浮点模型, 生成项目兴趣指数, 将该指数输入到变换后的 *Sigmoid* 函数, 计算各个兴趣类型的项目兴趣量.

4) 生成推荐列表. 通过兴趣模型计算用户的项目类型兴趣占比, 结合用户兴趣项目类型建立用户-类型-项目三元图结构, 利用 GCN 技术传播每个嵌入层, 最终以用户和项目的最后一层嵌入向量内积作为用户对项目的得分, 经排序后, 输出项目个体推荐列表.

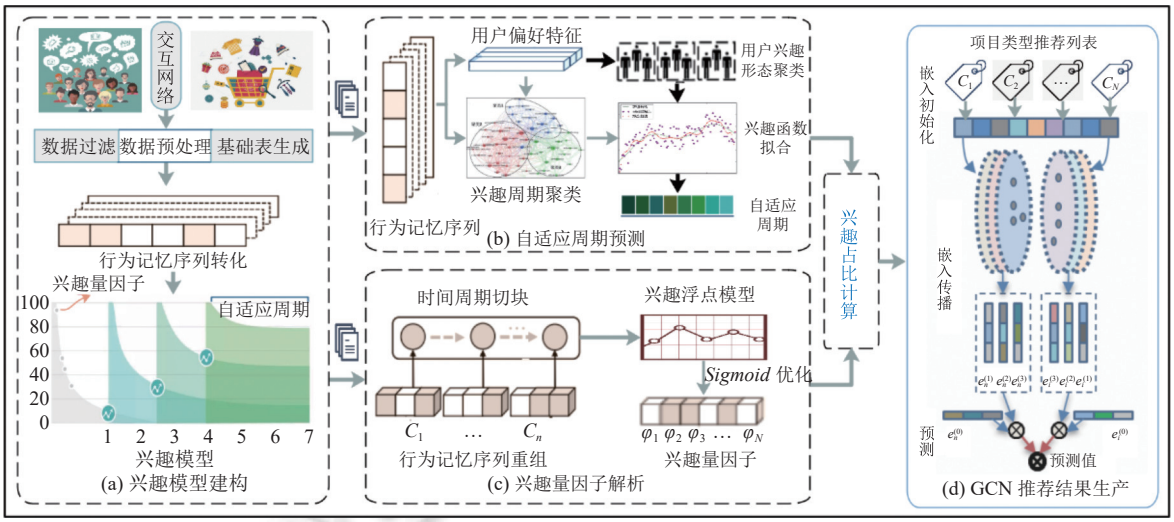


图 1 融合深度信息的轻量级 GCN 推荐模型整体框架

3.1 兴趣模型构建

本节将介绍如何把历史序列转换成行为记忆的形式, 以及如何在兴趣模型的构建中, 计算用户兴趣程度.

3.1.1 行为记忆

为表达各用户的兴趣规律, 先定义模型中每个用户的数据表示形式. 用户与项目的交互时序信息对于用户未来的行为倾向具有很大影响. 因此, 如果用 T_i ($i\in[1, n)$, n 为交互的总次数) 表示相邻行为时刻的间隔, 即目标项目对同一项目类型第 i 次和第 i+1 次浏览的时间间隔, 那么用户 u 在一个周期内对项目类型 i 产生的行为集可表示为 $\{T_1,T_2,T_3,\ldots,T_{n-1}\}$. 进一步地, 将用户一个周期的行为记录拓展为行为记忆的表示形式:

$$B_u^i=\{T_1,T_2,T_3,\ldots,T_{n-1},\tau,T_p\},$$

其中, 深度信息 T_p (p 标记该变量为预测变量) 表示预测用户下一次行为与最近一次行为的时间间隔, 最主要的特点是其不仅受自身历史行为的影响, 还受其他用户对该项目的历史行为影响, 导致其会随着整体行为规律的变化而改变, 因此, 将其定义为自适应周期 (详见第 3.2 节); τ 表示最近一次对该项目行为的时间戳.

因此, 用户在一个周期内的行为记忆可理解为用户浏览或者购买项目的行为规律. 据此, 通过用户的一个行为记忆来预测下一次浏览该项目的时间点, 正是推荐模型需要关注的要点.

3.1.2 兴趣程度

由行为记忆的定义可知, 行为记忆的丰富与否并不能直观地体现出用户的兴趣关联, 但在构建兴趣模型, 以及量化用户兴趣程度时, 往往可体现其独特的优势. 在推荐中, 用户兴趣一般可分为长期兴趣和短期兴趣, 但这两种兴趣在一个较短的浏览周期内的表现形式是相似的, 导致长短期兴趣难以区分. 因此, 在用户兴趣变化过程中, 能使长期兴趣保留下来, 同时继续挖掘用户的潜在兴趣, 是保证推荐效果的关键. 显然, 这种思想十分符合艾宾浩斯遗忘曲线的规律.

© 中国科学院软件研究所 http://www.jos.org.cn

下面引入艾宾浩斯遗忘曲线来模拟用户的兴趣变化过程, 包括解析行为记忆以构建兴趣模型, 以及求解兴趣程度的方式. 艾宾浩斯遗忘曲线如图 2 所示.

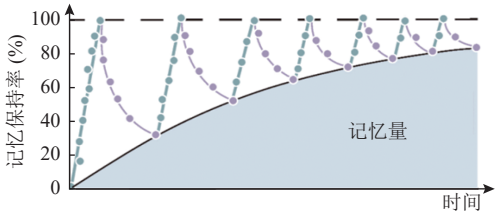


图 2 艾宾浩斯遗忘曲线

艾宾浩斯遗忘曲线最显著的特点就是在每一个时间点上, 均有其对应的记忆保持率, 其中有一种记忆保持率函数 j 表示被相关学者[22]充分运用, 如公式 (1) 所示:

$$j=\frac{20\mathrm{e}^{b}}{(\lambda t+t_0)^{c}} \tag{1}$$

其中, t 是两次相邻记忆时刻的时间差, t_0=0.00255, e 为自然数底, b 和 c 均为常数, 为使艾宾浩斯遗忘曲线更好地模拟兴趣变化过程, 将时间轴放大 λ 倍.

例 1: 以项目 A 为例, 用 φ_{A} 表示对 A 的兴趣量. 结合用户的记忆行为, 取第 1 次浏览结束的时间点作为时间原点, 计算第 1 次与第 2 次浏览间的时间差 t. 将 t 带入记忆保持率函数, 得到第 2 次开始浏览的记忆保持率 j_{A}. 然后将 j_{A} 与之前得到的 φ_{A} (浏览信息 A 最近 1 次表现出的兴趣量) 相乘, 得到第 2 次浏览时推荐项目 A 的兴趣程度 f_{A}. 之后的第 2 次推荐, 可取第 2 次浏览结束后的时刻作为新的时间点. 若用户在下一次浏览中继续浏览同一类信息, 则说明他的兴趣未改变. 为体现用户未改变兴趣, 将新浏览得到的兴趣量 φ_{A} 以新一次浏览结束时刻作为新时间点来计算记忆保持率, 记为 j_{A}, 以进一步确定对项目 A 的兴趣程度.

通过例 1, 就可预测出用户下次浏览该类信息的兴趣程度 f, 从而挖掘出用户的潜在兴趣. 如公式 (2) 所示:

$$f=j\times\varphi=\frac{20\mathrm{e}^{b}}{(\lambda t+t_0)^{c}}\times\varphi \tag{2}$$

由公式 (2) 可知, 用户对信息相邻两次浏览间的时间差 t 及兴趣量 φ 是影响兴趣程度的两个关键因素. 在进行推荐时, 时间差 t 应能随着当前浏览周期的兴趣分布特性自动生成, 从而预测用户下一次可能会再次关注该兴趣的时刻; 兴趣量则可看成是当前浏览周期结束后对该兴趣的兴趣剩余量. 其中, 该时间差即为行为记忆序列中的自适应周期 T_p; 一段浏览周期结束后, 用户对目标兴趣的兴趣剩余量称为兴趣量因子 φ.

显然, 上述这两个参数均是用户与项目基础信息中无法轻易得到的, 而需经过一系列对基础信息的归纳和转化, 因此, 我们将这类信息称为深度信息. 接下来, 第 3.2 节主要求解自适应周期; 第 3.3 节主要求解兴趣量因子. 结合这两个特征, 便可清晰地表示出用户的兴趣分布特性, 从而作出更为准确的推荐.

3.2 自适应周期生成

通过研究发现, 用户的浏览频数 α 和浏览周期 β 可很好地分别描述行为记忆的深度和广度, 但简单地将用户历史记录中总的浏览频数和浏览周期解析出的兴趣表现, 只是用户对项目的整体兴趣程度. 事实上, 用户历史记录中, 哪怕是对同一个项目, 在不同时间段都可能会表现多种不同兴趣. 因此, 将整体兴趣表现来标识近期的兴趣程度, 将会丢失大量重要信息.

为更形象地描述用户对项目在不同周期内的兴趣表现, 这里定义兴趣形态, 以区别于兴趣程度. 其中, 兴趣程度仅发生在某一时间跨度, 某一时刻兴趣的唯一独立的兴趣表现; 兴趣形态则不局限于某一时间跨度, 可在同一时刻表示多种兴趣表现的共现情况, 即兴趣形态可指任一时间跨度, 任一维度下的一种兴趣表现.

例 2: 在图 3 中, 第 i 个浏览周期内的第 j 个浏览时刻点用 t_{ij} 表示, 其中 t_{1j}, t_{2j}, t_{3j} 与 t_{4j} $(0\leqslant j\leqslant 3)$ 均表示同一用户对目标项目的不同历史浏览周期或不同用户对目标项目的同一浏览周期, 则 t_{i0}, t_{i1}, t_{i2} 与 t_{i3} $(1\leqslant i\leqslant 4)$ 表示对

© 中国科学院软件研究所 http://www.jos.org.cn

应浏览周期内的某一个浏览时刻. 其中, 兴趣程度只能是周期跨度中的某一时间轴$\{t_{1j}, t_{2j}, t_{3j}, t_{4j}\}$直线上任一时刻中用户对目标项目的兴趣表现, 而本文将同一用户对目标项目的不同历史浏览周期或不同用户对目标项目的同一浏览周期置于同一时间平面内进行研究, 因而, 兴趣形态不受单一跨度或维度的约束, 可为 t_{ij} 中的任一时刻中用户对目标项目的兴趣表现.

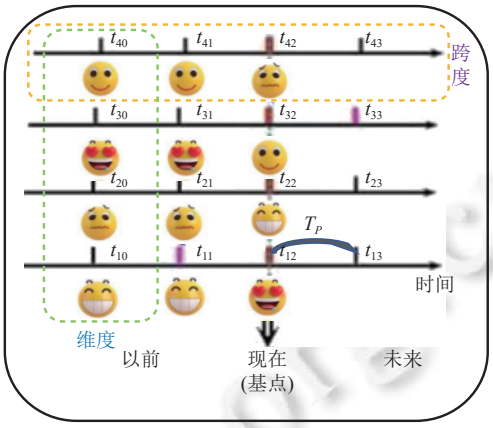


图 3 兴趣表现示例

兴趣形态更适用于分析项目主体下的深度信息. 进一步考虑到用户对项目在历史记录中可能存在多种兴趣形态, 而当前的兴趣形态主要由最近一段时间的数据起主导作用. 我们在不舍弃历史数据的情况下, 指定一个距离当前时刻跨度合适的时间周期, 综合该周期内的行为记忆来计算浏览次数 T_α 和浏览周长 T_β .

浏览次数 T_α 是一个标量, 必须为整数, 如公式 (3) 所示:

$$T_\alpha = a\%b \quad (b = 1,2,3,\ldots) \tag{3}$$

浏览周长 T_β 则为倒数第 T_α 次行为的时间戳开始, 每两个相邻时间戳差值的总和, 如公式 (4) 所示:

$$T_\beta = \sum\nolimits_{i=\alpha-T_\alpha}^{\alpha} (t_{i+1} - t_i) \tag{4}$$

其中, t_i 代表第 i 次浏览行为的时间戳.

可见, 所设计的浏览次数 T_α 和浏览周长 T_β 对项目的近期兴趣形态有着关键作用, 根据这些因素可准确地对兴趣进行 K 均值聚类, 进而利用线性回归生成自适应周期.

3.2.1 基于 K 均值的兴趣形态聚类

项目评分或评价在推荐中一直被认为是最能直观反映用户兴趣程度的信息, 但其稀疏性高、代表性不强等问题, 导致推荐者不得不从更广泛角度去研究用户兴趣. 下面介绍本文确定用户兴趣程度的方法.

模型在学习用户的浏览行为规律过程中, 不仅要区别用户的多种兴趣表现 (比如, 长期兴趣和短期兴趣), 还需关注同种兴趣的兴趣形态分布 (比如, 长期兴趣的变化情况). 因此, 与传统模型根据用户基本信息与项目评价等信息计算相似度的方式不同, 本文主要利用 T_α 和 T_β 提取用户特征, 进而转换成向量表示用以聚类. 其优势是, 这种考虑用深度信息融合用户数据的方式, 不易受缺失值或噪声的干扰, 且 T_α 和 T_β 都是数值标量, 更直观地、客观地反映用户的行为信息, 不易受用户及环境的影响.

K 均值算法是典型的基于距离的聚类方法, 在数值特征的聚类中表现优秀, 其技术原理相对轻量, 十分适用于本文模型. 算法采用距离作为评判相似度的标准, 距离越大, 两者间的相似度越小. 就用户对项目的第 j 个特征的任意两个兴趣, 分别用向量 $\boldsymbol{u}_{j,m}$, $\boldsymbol{u}_{j,n}$ 表示, 两者间的距离采用 Euclidean 距离 (欧氏距离) 度量, 如公式 (5) 所示:

$$d(m,n) = \sqrt{\sum\nolimits_{j=1}^{k} (\boldsymbol{u}_{j,m} - \boldsymbol{u}_{j,n})^2} \tag{5}$$

其中, k 表示模型中特征的数量.

将每个用户对同一类项目的 T_α 和 T_β 作为在该项目类型中用户的标识, 对用户进行分类: ① 随机选取 k 个标

识, 作为质心, 每个被标识的用户都选择距离自己最近的质心, 进行归类; ② 对 k 个类分别求平均值, 作为新的质心, 继续进行归类, 直到质心位置不再变化或者小于一个阈值. 此时把每个用户与其类中质心的距离再进一步转化成兴趣形态相似度, 如公式 (6) 所示:

$$sim(m,n) = 1/(1+d(m,n)) \tag{6}$$

兴趣形态相似度高的用户归到一类, 可得到用户对每种项目在不同周期的兴趣形态, 这在很大程度上反映出用户的兴趣规律, 特别是最近一个周期内的兴趣形态, 对于生成自适应周期起到关键作用.

3.2.2 基于线性回归的自适应周期生成

由用户近期的行为规律, 来预测下次的行为表现, 是多数推荐模型的主要思想. 尽管这些模型也考虑了用户的行为表现, 但只是期望在下一次推荐就得到用户的反馈. 而自适应周期 T_p 反映的是用户未来可能会产生相关行为的时刻. 特别对于长期兴趣, 其存活期长, 但可能并不活跃, 自适应周期 T_p 便可灵活地随着不同的兴趣分布做出变化. 因此, 自适应周期 T_p 是影响推荐效果的关键.

前面通过聚类操作, 已明确用户在不同时期的兴趣形态分布, 接下来将同种项目类型下相同兴趣形态的项目行为数据分别进行拟合. 线性回归方法在拟合过程中已取得不错的表现, 而为体现个性化特征, 本文采用多元线性回归方式, 融入多个影响因子, 以拟合出可计算自适应周期的曲线函数, 如公式 (7) 所示:

$$\begin{cases} \text{if}\begin{cases} x=F_1: T'_p = b_1+\omega_1\times T_\alpha+\omega_2\times(T_\alpha)^2; \\ x=F_2: T'_p = b_2+\omega_3\times T_\alpha+\omega_4\times(T_\alpha)^2; \\ \vdots \\ x=F_n: T'_p = b_n+\omega_{2n-1}\times T_\alpha+\omega_{2n}\times(T_\alpha)^2, \end{cases} \\ T_p = T'_p+\omega_{2n+1}\times T_\beta+\omega_{2n+2}\times(T_\alpha)^2 \end{cases} \tag{7}$$

接下来引入 δ 条件表达式函数, 若条件表达式为 True, 则 δ=1; 若为 False, 则 δ=0. 因此, 可将公式 (7) 转换为一个线性模型, 如公式 (8) 所示:

$$\begin{cases} T_p = b_1\times\delta(x=F_1)+\omega_1\times\delta(x=F_1)T_\alpha+\omega_2\times\delta(x=F_1)(T_\alpha)^2+b_2\times\delta(x=F_2)+\omega_3\times\delta(x=F_2)T_\alpha+\omega_4\times\delta(x=F_2)(T_\alpha)^2+\ldots \\ \quad +b_n\times\delta(x=F_n)+\omega_{2n-1}\times\delta(x=F_n)T_\alpha+\omega_{2n}\times\delta(x=F_n)(T_\alpha)^2+\omega_{2n+1}\times T_\beta+\omega_{2n+2}\times(T_\beta)^2 \\ \text{且}\delta(x)=\begin{cases} 1, & \text{if } x=F_i \quad (1\leqslant i\leqslant n) \\ 0, & \text{otherwise} \end{cases} \end{cases} \tag{8}$$

据此, 便可得到用户兴趣形态的自适应周期模型. 为训练出更好的模型, 这里引入正则项, 采用损失函数来评估, 如公式 (9) 所示:

$$L=\sum_j^n\left(T_p^j-\left(b+\sum_i^m\omega_i T_i\right)\right)^2+\lambda\sum_i^m(\omega_i)^2 \tag{9}$$

其中, n 为迭代次数, m 为参数个数.

自适应周期生成的过程, 即提取出基于显隐式信息下的用户某个周期内的浏览次数 T_α 和浏览周长 T_β, 先将这两个深度信息转换成向量形式进行聚类操作以确定该项目的兴趣形态, 再将相同兴趣形态下, 同种项目的行为数据分别进行拟合, 最后根据用户自身的兴趣形态, 训练自适应周期模型, 便可准确地预测该项目的自适应周期 T_p. 具体过程见算法 1.

算法 1. 自适应周期生成.

输入: 用户所有的行为记忆 B_u^i, 兴趣特征数量超参 k;

输出: 用户-项目类型的自适应周期 T_p.

Begin

1. $B_u^i=\{T_1,T_2,T_3,\ldots,T_{n-1},\tau,T_p\}$ /* 行为记忆初始化, 见第 3.1.1 节*/

2. $I=\varnothing$ /* 初始化用户-项目兴趣形态集*/

© 中国科学院软件研究所 http://www.jos.org.cn

3. $a = T_\alpha(z_1,\ldots)$ /* 指定浏览次数, 见公式 (3)*/
4. $b = T_\beta(a, k)$ /* 计算浏览周长, 见公式 (4)*/
5. **For** B in B_u^i:
6. **For** j in range(k):
7. $c = dis(B_m^j, B_u^j)$ /* 计算用户的项目兴趣距离, 见公式 (5)*/
8. $d = sim(c)$ /* 将距离转化为兴趣形态相似度, 见公式 (6)*/
9. **End For**
10. **Output** $I(u, i)$ /* 输出用户对各种项目的兴趣形态*/
11. **End For**
12. **For** B in B_u^i:
13. **For** i in I:
14. $e = F(i, T_\alpha, T_\beta)$ /* 构建自适应周期模型, 见公式 (7)*/
15. **End For**
16. **End For**
17. **For** B in B_u^i:
18. **For** i in I_i: /* 遍历目标用户的项目兴趣形态集*/
19. $T_p = Li(B, i)$ /* 训练模型并生成自适应周期, 见公式 (8)*/
20. **End For**
21. **End For**
22. **Output** T_p /* 输出自适应周期*/

End

3.3 兴趣量因子解析

已有研究将用户的兴趣量定义为用户对项目的总浏览时长或总行为次数, 显然, 这未注意到兴趣会随时间变化及当前兴趣应起主导作用的问题, 且这种方式通用性不强, 缺乏充足的说服力. 因此, 引入时间影响因素及提高通用性对于衡量用户的兴趣量具有重要意义.

3.3.1 行为记忆重组

不同情景的推荐, 需要参考的用户历史行为跨度是不一致的. 比如, 商品推荐中, 用户一个月前的数据依然具有很好的参考价值, 但在音乐推荐中, 用户的爱好随心情变化明显, 一周前数据的价值就已大打折扣. 为使近期行为能在兴趣量中起主导作用, 却不丢失长久行为的作用, 就应让用户历史行为时间跨度能随着兴趣分布特性而变化. 为此, 先定义一个周期间隔数 (time_period_num, TPN), 将其作为单位周期间隔长度, 进而用户的历史行为总周期被切分为 n 块. 同时, 用兴趣指数描述用户的兴趣量变化情况, 该指数会随着切分周期的改变呈起伏变化, 表现为一个浮动的小点. 为此, 给出兴趣浮点模型, 通过该模型演算出用户对项目的兴趣量, 如图 4 所示.

例 3: 在图 4 中, 用户历史记录按照 TPN 被划分为 n 个单位周期, 在一个单位周期中, 并未考虑浏览次数、浏览周长, 而仅考虑行为是否发生, 因为用户的历史行为时间跨度是不断变长的, 历史行为周期内行为发生的过程和程度不是关键, 甚至很容易给当前兴趣带来噪声, 因而只需关注该行为发生与否. 假设用户对每种项目的起始点 a 的初始兴趣量均为 0, 根据时间周期的推进可演算用户的兴趣量. 在规定一个切分周期内, 若用户对某一项目类型产生行为, 则兴趣指数加一个单位, 否则减一个单位, 直到最近的一个时间节点, 该兴趣指数即对应用户对该项目类型的兴趣量.

但采用例 3 的这种简单方式演算出的数值作为兴趣量, 存在很大弊端, 如图 4 中的 b–c 段及 e–f 段, 由于在历史的某个时期内, 用户对该种项目的未浏览 (或连续浏览) 周期过长, 均会快速减小 (或增大) 该兴趣指数, 使其局

© 中国科学院软件研究所 http://www.jos.org.cn

部过小 (或过大), 哪怕近段周期内存在 (或不存在) 对该种项目产生新的行为, 也会演算出较小 (或较大) 的兴趣量. 因此, 下面对该兴趣量的演算方式进行优化.

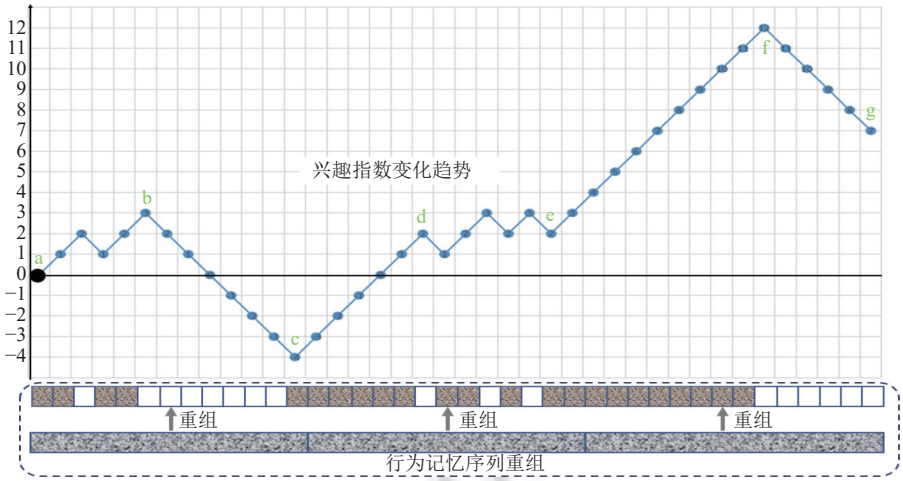


图 4 兴趣浮点模型

3.3.2 兴趣量演算优化

我们引入双曲正切 *Sigmoid* 函数对兴趣量进行优化. 该函数在数据工程里面, 通常被用作归一化处理, 以减小某些离散值对整体数据分布的影响. 为更好地模拟用户的行为变化规律, 并充分利用 *Sigmoid* 函数的分布特点, 将第 3.3.1 节提出的兴趣指数变化转变为 *Sigmoid* 函数中横坐标值 (x 值) 的变化, 而兴趣量变化则转变成纵坐标值 (y 值) 的变化. 先要设置合理且有效的 TPN, 对兴趣指数进行累计演算, 然后代入 *Sigmoid* 函数, 求解出的值作为当前周期中的目标项目兴趣量因子, 这样就有效地规避了局部数值过大的风险.

参考 *Sigmoid* 函数曲线可知, 函数的曲率变化及曲线两端的近似值, 对解析兴趣量因子有很大影响. 因此, 为使兴趣量因子能呈现出更好且更合理的变化趋势, 并限定在一个可控的范围内, 我们将 *Sigmoid* 函数进行一定的优化 (将函数分别进行横向、纵向对称拉伸), 如公式 (10) 所示:

$$Sigmoid(x) = (z - z\mathrm{e}^{-2\mu x})/(1 + \mathrm{e}^{-2\mu x}) \tag{10}$$

通过这种优化, 可将解控制在 $(-z, z)$ 区间 (z 是兴趣量饱和值), 通过控制变量 μ 的值即可调节曲线斜率, 这样更能体现用户对项目的兴趣变化规律. 为更好地模拟兴趣量因子变化, 设定几点兴趣量因子解析规则.

1) 有效切分指数从第一次出现用户对该项目产生行为开始计数, 逐个切分周期递进, 有行为则加一个单位, 否则减一个单位.

2) 若兴趣量达到饱和时, 则其不再发生变化 (或变化可忽略不计), 故需预先设置有效的饱和值.

3) 若用户对新项目产生行为, 则兴趣指数从 0 开始计数; 若用户对旧项目再一次产生行为, 则兴趣指数在原来的数值上继续计数.

综上兴趣量因子解析过程, 可准确且显式地表示出用户对每个项目类型在其历史行为中的兴趣量变化情况, 并且最终计算出用户对每个项目类型的兴趣量. 具体过程见算法 2.

算法 2. 兴趣量因子解析.

输入: 用户所有的行为记忆 B_u^i, 有效饱和值 z (正数);
输出: 用户对应项目类型的兴趣量集 ϕ.

Begin
1. $\phi = \varnothing$ /* 初始化兴趣量集*/

2. T = (time_max(B_u^i) – time_min(B_u^i)) / n /* 获取单位切分周期长度*/
3. **For** B **in** B_u^i :
4. $index = 0$
5. **For** i **in** TPN: /* 按切分周期长度为单位, 遍历用户的整个周期*/
6. **If** action(i) ≠ NULL: /* 判断该切分周期内行为是否发生*/
7. **If** $index < -z$ or $index > z$: /* 判断是否在有效饱和值内*/
8. Continue
9. **Elif**:
10. $index$ += 1 /* 有值则加 1, 否则减 1*/
11. **Else**:
12. $index$ –= 1
13. **End If**
14. $\varphi_{An} = \sigma'(index)$ /* 将得出的兴趣指数经 *Sigmoid* 优化, 见公式 (10)*/
15. **End If**
16. **End For**
17. $\phi = f(\varphi_{An})$ /* 更新用户的兴趣量集*/
18. **End For**
19. **Output** ϕ

End

3.4 融合深度信息的 GCN 推荐

用户短期内兴趣量大的兴趣易被模型学习, 相反地, 将会被遗忘的兴趣却常被忽略, 但往往更可能是潜在兴趣, 甚至其推荐价值要超过当前新挖掘的兴趣. 这就要求我们的推荐不仅要考虑当前的热门兴趣, 还需兼顾到下一次推荐或用户下一次行为发生前将会被遗忘的兴趣 (注意, 这种兴趣可能来自前几次推荐). 显然, 前面第 3.2–3.3 节特征挖掘的工作能够保证这一点.

但现实中, 用户一般不会主动再去浏览将会被遗忘的兴趣个体, 也就是说用户的此次兴趣再次作用到上一个周期兴趣项目的概率并不大. 但却常体现在同一类项目中, 即再次浏览与上一次兴趣项目同类的项目个体. 一般认为, 用户潜在兴趣是先表现在某类项目上, 再指向该类的某个具体项目上的想法是合理的.

例 4: 假设喜剧类电影《喜剧之王》是用户上个历史周期的兴趣项目, 但这个周期对动作类电影浏览量更高. 因此, 在推荐中考虑动作类电影的同时, 也不能放弃电影《喜剧之王》带来的历史兴趣信息. 然而, 再推荐一次电影《喜剧之王》意义不大, 但可推荐同为喜剧类的其他电影, 比如《功夫》.

根据例 4, 提出的模型 LG_APIF 遵从用户的兴趣变化规律, 先从更为宏观的角度去挖掘用户兴趣. 比如, 先推荐出用户的兴趣项目类型, 再从兴趣项目类型推荐列表中预测用户的兴趣项目个体.

3.4.1 兴趣项目类型推荐

协同过滤、协同深度学习等模型[23,24]的嵌入函数缺乏对关键关联信息的精确编码, 往往这种编码揭示了隐藏在用户-项目交互中的用户 (或物品) 间的相似性. 更具体地说, 大多数现有的方法为降低模型复杂度或使高阶传播更好地拟合嵌入, 仅使用描述性特征 (例如 ID) 来构建嵌入函数, 这很难考虑用户-项目的一般交互或更高阶交互关系. 因此, 当嵌入不足以捕获协作信息时, 我们采用具有交互关系的操作来弥补, 且为避免过度增加高阶传播的负担, 先将用户-项目的一般交互关系转变成具有高阶连接性的形式, 如图 5 所示.

受最近发展的 GNN[3,4,25]的启发, 模型 LG_APIF 在传统用户-项目图结构的基础上, 加入项目类型信息, 以构建用户-类型-项目三元图结构, 使传播网络输入的图结构数据开始就拥有较为显式的协作信号. 此外, 本文采用 GCN 技术来传播用户-项目嵌入信息, 对用户与项目的高阶交互信息进行挖掘. 由图 5 可知, 将每种潜在项目类型

© 中国科学院软件研究所 http://www.jos.org.cn

内的所有项目作为一个嵌入层, 该嵌入层聚合了用户-项目的交互信息来进一步细化用户 (或项目) 的嵌入. 每增加一个嵌入传播层, 就能包含更多用户 (或项目) 的交互信息, 且用户 (或项目) 的嵌入表示均由上一个传播层细化得到. 因此, 在传播过程中, 更容易挖掘到更高阶用户-项目交互信息, 或者增强用户-项目间的协作信号.

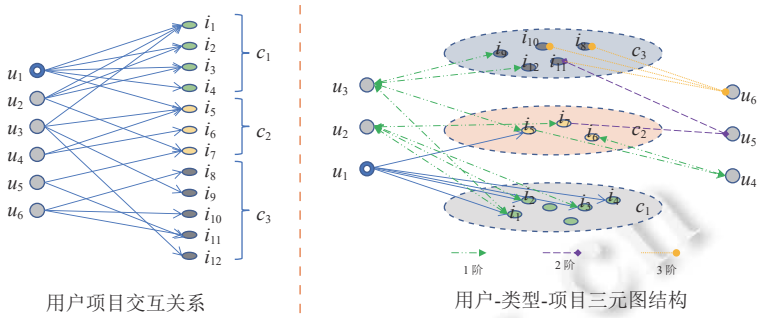


图 5 用户项目交互关系及用户类型-项目三元图结构

例 5: 叠加 1 层 (初始嵌入层) 捕获 $u_1 \leftarrow i_1 \leftarrow u_2$, $u_1 \leftarrow i_1 \leftarrow u_3$ 的 1 阶行为相似性; 叠加 2 层捕获 $u_1 \leftarrow i_5 \leftarrow u_4$ 的 1 阶行为相似性, $u_1 \leftarrow i_1 \leftarrow u_2 \leftarrow i_7 \leftarrow u_5$ 的 2 阶行为相似性, 及 $u_1 \leftarrow i_1 \leftarrow u_2 \leftarrow i_7$, $u_1 \leftarrow i_5 \leftarrow u_4 \leftarrow i_6$ 的 1 阶潜在推荐; 叠加 3 层捕获 $u_1 \leftarrow i_1 \leftarrow u_3 \leftarrow i_9$, $u_1 \leftarrow i_1 \leftarrow u_3 \leftarrow i_{12}$ 的 1 阶潜在推荐, $u_1 \leftarrow i_1 \leftarrow u_2 \leftarrow i_7 \leftarrow u_5 \leftarrow i_{11}$ 的 2 阶潜在推荐, $u_1 \leftarrow i_1 \leftarrow u_2 \leftarrow i_7 \leftarrow u_5 \leftarrow i_{11} \leftarrow u_6$ 的 3 阶行为相似性, 及 $u_1 \leftarrow i_1 \leftarrow u_2 \leftarrow i_7 \leftarrow u_5 \leftarrow i_{11} \leftarrow u_6 \leftarrow i_8$, $u_2 \leftarrow i_7 \leftarrow u_5 \leftarrow i_{11} \leftarrow u_6 \leftarrow i_{10}$ 的 3 阶潜在推荐.

信息流的强度 (由层间可训练权重估计) 决定了项目的推荐优先级. 模型 LG_APIF 将嵌入层以用户兴趣项目类型 c (即用户的兴趣类型) 为界, 需清楚嵌入层间的联系与权重 (即兴趣项目类型及相应的权重).

自适应周期和兴趣量因子的特征求解完成后, 用公式 (2) 求得用户对所有项目类型的兴趣度, 再将用户对每类项目的兴趣度与所有项目类型的兴趣度相比, 即用 f_{A} (用户对项目类型 A 的兴趣度) 除以 f_{All} (用户对所有项目类型的兴趣度), 便可得到推荐比例, 记为 w. 项目类型 A 在此次兴趣中的占比 w_{A} 的计算方式, 如公式 (11) 所示:

$$w_{\mathrm{A}} = f_{\mathrm{A}} / f_{\mathrm{All}} = \left(\frac{20\mathrm{e}^{b}}{(\lambda T_{\mathrm{A}} + t_0)^{c}} \times \varphi_{\mathrm{A}} \right) / \sum_{i=1}^{N} \left(\frac{20\mathrm{e}^{b}}{(\lambda T_{\mathrm{A}} + t_0)^{c}} \times \varphi_{i} \right) \tag{11}$$

在实验中可发现, 用户的潜在项目兴趣类型集中所包含的潜在项目的数量依然很大, 且不同类型的项目间极可能存在相近的兴趣度. 为对用户近期行为做出及时反馈, 将最近一个周期内浏览过的项目类型权重加大, 这也符合用户的浏览习惯, 同时不易丢失一些潜在兴趣. 得到对应项目类型 A 的权重计算方式, 如公式 (12) 所示:

$$w'_{\mathrm{A}} = \mu \left(1 + c_{\mathrm{A}} / \sum_{I}^{M} c_I \right) \times w_{\mathrm{A}} \tag{12}$$

其中, μ 为用户意图系数, M 为一个周期内用户浏览项目的类型数, c_I 为对应项目类型 I 中项目的浏览次数.

由公式 (12) 可知, 用户意图系数 μ 表示最近周期内的浏览行为对未来行为的影响权重. 比如, 目的性强的用户一般很明确自己的意图, 浏览的项目类型稀疏性相对较小; 目的性弱的用户则不明确自己的意图, 浏览的项目类型稀疏性相对较大. 为此, 用户意图系数 μ 值定义为最近一个或多个周期内的各类浏览项目数与对应项目类型中项目总数之比并求和, 如公式 (13) 所示:

$$\mu = \sum_{J}^{N} \sum_{I}^{M_J} (c_I / M_I) \tag{13}$$

其中, M_J 表示选择距离当前时刻最近 N 个周期内的第 J 个周期的浏览项目类型总数, c_I 表示对应周期内用户对项目类型 I 的浏览次数, M_I 表示项目类型 I 中的项目总数.

根据用户的兴趣占比 (见公式 (11)) 及其意图系数 μ (见公式 (13)), 计算每个项目类型的得分, 排序得到多个用户潜在项目类型, 再据此构建多个嵌入传播层. 接下来具体阐述如何捕获协作信号以进行项目个体推荐.

3.4.2 兴趣项目个体推荐

由文献 [3] 和文献 [4] 的设计思想及其实验可知, 使用 GCN 技术来模拟协同过滤, 并不能简单地、直接地套用 GCN 框架, 因为其中很多构件对协同过滤任务并不一定有用. 此外, 本文构建的用户-类型-项目三元图结构中

© 中国科学院软件研究所 http://www.jos.org.cn

节点已有了较为明确的分类, 且其初始嵌入表示含义也较为明显, 无需再对特征进行提取, 而仅需使用 GCN 框架中的邻域聚合功能. 因此, 为降低模型的复杂度, 我们借鉴了文献 [4] 提出的轻量级 GCN 来设计推荐框架.

本文模型的 GCN 项目个体推荐模块的总体结构如图 6 所示, 初始输入为用户-类型-项目三元图结构组成的嵌入表示, 输出为预测值 (即, 用户对项目的偏好得分). 该 GCN 项目个体推荐模块由嵌入层 (生成并初始化用户与项目的嵌入)、图传播层 (在多层传播中优化嵌入) 及预测层 (精化嵌入并输出预测值) 组成.

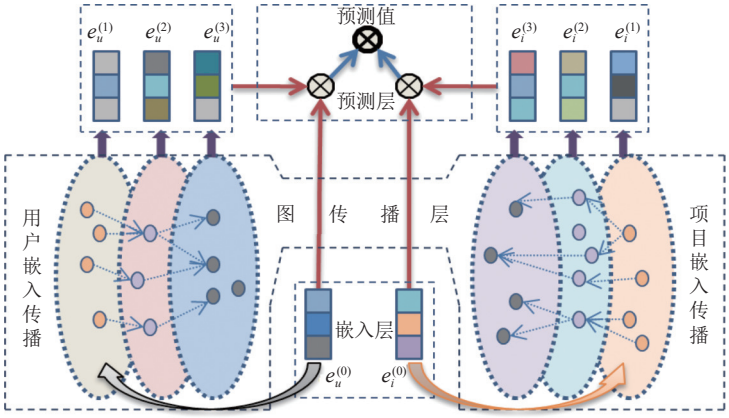


图 6 项目个体推荐模块

● 嵌入层. 与一些基于嵌入式的推荐模型[3,26]相似, 模型 LG_APIF 的 GCN 模块同样设定初始的自由嵌入, 给定用户和项目的索引, 从用户自由嵌入矩阵 $\boldsymbol{P}$ 和项目自由嵌入矩阵 $\boldsymbol{Q}$ 中输出自由用户潜向量 $\boldsymbol{p}_u$ 和自由项目潜向量 $\boldsymbol{q}_i$. 直观地说, 浏览的项目被视为该用户的特征, 并用来衡量两个用户的相似度; 同理, 消费一个物品的用户可被视为该物品的特征, 并用来衡量两个物品的相似度. 因此, 通过这种关系来建立用户 u 的相关特征向量 $\boldsymbol{x}_u$, 以及项目 i 的特征向量 $\boldsymbol{y}_i$.

值得注意的是, 与传统推荐模型 (如 MF 和神经协作过滤) 不同的是: 它们将这些用户-项目 ID 嵌入直接馈送到交互层以获得预测分数. 而在模型 LG_APIF 中, 通过在用户-类型-项目交互图上传播嵌入来对其优化. 由于嵌入优化步骤将协作信号显式地注入到嵌入中, 因此可为推荐提供更有效的嵌入. 因而, 对于用户 u, 以 $\boldsymbol{p}_u$ 及其相关特征向量 $\boldsymbol{x}_u$ 作为输入, 输出一个用户初始嵌入 e_u^0 (表示用户的初始兴趣), 如公式 (14) 所示:

$$\boldsymbol{e}_u^0 = g(\boldsymbol{W} * [\boldsymbol{x}_u, \boldsymbol{p}_u]) \tag{14}$$

其中, $\boldsymbol{W}$ 是一个变换矩阵, $g(\cdot)$ 是个非线性函数. 为了便于标记, 省略了全连接神经网络中的偏置项.

同理, 对于项目 i, 以 $\boldsymbol{q}_i$ 及其相关特征向量 $\boldsymbol{y}_i$ 作为输入, 输出一个项目初始嵌入 e_i^0 (表示项目的初始特征), 如公式 (15) 所示:

$$e_i^0 = \sigma(\boldsymbol{F} * [\boldsymbol{q}_i, \boldsymbol{y}_i]) \tag{15}$$

其中, $\boldsymbol{F}$ 是一个变换矩阵, $\sigma(\cdot)$ 是个非线性函数.

● 图传播层. 该层对用户的潜在偏好在关系网络中扩散的动态变化进行建模, 其为多层结构, 每层输入前一层的用户嵌入, 在关系扩散过程结束后输出用户的更新嵌入.

对于用户 u, e_u^k 表示其第 k 层的潜在嵌入. 通过将第 k 层的输出输入到 k+1 层, k+1 层的影响扩散操作将用户 u 的潜在嵌入更新为 $e_u^{(k+1)}$, 如公式 (16) 所示:

$$e_u^{(k+1)} = \sum\nolimits_{i \in N_u^i} \frac{1}{\sqrt{|\boldsymbol{N}_u^i||\boldsymbol{N}_i^u|}} e_i^k \tag{16}$$

其中, $\boldsymbol{N}_u^i$ 表示用户 u 的交互项目集, $\boldsymbol{N}_i^u$ 表示项目 i 的交互用户集.

同样, 对于项目 i, e_i^k 表示其第 k 层的潜在嵌入. 通过将第 k 层的输出输入到 k+1 层, k+1 层的影响扩散操作将项目 i 的潜在嵌入更新为 $e_i^{(k+1)}$, 如公式 (17) 所示:

© 中国科学院软件研究所 http://www.jos.org.cn

$$e_i^{(k+1)} = \sum_{u \in N_i^u} \frac{1}{\sqrt{|N_i^u||N_u^i|}} e_u^k \tag{17}$$

● 预测层. 当给定并初始化所有用户的嵌入 e_u^0 和所有项目的嵌入 e_i^0 时, 便可分别通过公式 (16) 和公式 (17) 中设置的传播方式来层层优化嵌入. 直到嵌入达到最后一层 (第 K 层) 时, 再将每一层用户和项目的嵌入 (包括初始嵌入) 组合起来, 得到用户 (或项目) 的最终表示, 如公式 (18) 所示:

$$e_u = \sum_{k=0}^{K} \omega_k' e_u^k, \quad e_i = \sum_{k=0}^{K} \omega_k' e_i^k \tag{18}$$

其中, 第 k 层嵌入的权重为 ω_k' (即, 用户对第 k 种兴趣项目类型的选择概率).

最后, 我们将模型预测定义为用户和项目最终表示的内积, 如公式 (19) 所示:

$$S_{u,i} = e_u^{\mathrm{T}} e_i \tag{19}$$

综上分析, 引入艾宾浩斯遗忘曲线构建兴趣模型, 分别求得自适应周期 (见算法 1) 和兴趣量因子 (见算法 2) 等关键深度信息, 再根据兴趣模型得到用户潜在推荐项目类型, 进一步构建用户-类型-项目三元图结构, 用来表示用户和项目的嵌入表示, 最后结合 GCN 模拟协同过滤思想, 以生成推荐列表. 具体过程见算法 3.

算法 3. 融合深度信息的 GCN 自适应推荐.

输入: 用户当前内的所有的行为记录 B_u^i;

输出: 用户的兴趣项目个体推荐列表 *List*.

Begin

1. $\Gamma_{T_p} = \varnothing$, $List = \varnothing$ /*参数初始化*/
2. **For** B **in** B_u^i:
3. Compute T_p /* 计算所有用户对应所有项目的自适应周期, 见算法 1*/
4. $\Gamma_{T_p} = f(T_p)$
5. **End For**
6. **For** B **in** B_u^i:
7. Compute φ /*计算所有用户对应所有项目的兴趣量因子, 见算法 2*/
8. $\phi = f(\varphi)$
9. **End For**
10. **For** B **in** B_u^i:
11. **For** each user's $T_p \in \Gamma_{T_p}$ and $\varphi \in \phi$:
12. $J = j(T_p)$
13. $f_A = \varphi * J$ /* 计算用户对项目类型的感兴趣程度, 见公式 (2)*/
14. **End For**
15. $p = \omega(f_A)$ /* 计算用户-项目类型兴趣占比, 见公式 (11)*/
16. $q = \omega'(p)$ /* 将兴趣占比转换成权重值, 见公式 (12)*/
17. Sort and update the set of user interest items /* 排序并更新用户兴趣项目类型集*/
18. **End For**
19. $e_u^0 = g(*)$, $e_i^0 = \sigma(*)$ /*计算用户和项目的初始潜在嵌入, 见公式 (14) 和公式 (15)*/
20. $e_u^{(k+1)} = f(e_u^0)$, $e_i^{(k+1)} = f(e_i^0)$ /*计算用户和项目的 k 层潜在嵌入, 见公式 (16) 和公式 (17)*/
21. $e_u = \sum e_u^k$, $e_i = \sum e_i^k$ /*计算用户和项目的最终潜在嵌入, 见公式 (18)*/
22. $s = S(e_u, e_i)$ /* 计算项目个体的得分, 见公式 (19)*/
23. $List = sort(s)$ /* 排序获得项目个体推荐得分*/

© 中国科学院软件研究所 http://www.jos.org.cn

24. **Output** *List*
End

4 实验设计与对比分析

为验证 LG_APIF 的有效性, 本文选择 4 个数据集进行综合实验对比与分析, 重点回答以下几个问题.

RQ1: 不同超参数对模型效果有何影响?

针对此问题, 在 4.4.1 节设置调参实验, 对超参数在 4 种数据集下的调优过程进行了阐述. 优化模型 LG_APIF 的参数, 并选择了周期间隔数、潜在项目类型数两种关键性参数进行研究, 发现它们与选取的数据集有关, 在不同的数据集下具有不同的最优值, 且它们所影响的特征与推荐效果具有强相关性.

RQ2: 模型的主要构件有哪些, 以及它们存在的必要性?

针对此问题, 在第 4.4.2 节设置消融实验, 就自适应周期、兴趣量因子及用户-类型-项目三元图结构 3 个主要构件, 设计了 3 个变体模型分别分析这些构件对模型的作用. 实验结果显示, 这 3 种变体模型的推荐效果均在不同程度上劣于提出的模型 LG_APIF, 说明这些构件对模型具有重要影响.

RQ3: 与经典的、最新的模型相比, 本文模型的推荐效果如何?

针对此问题, 在第 4.4.3 节设置对比实验, 分别将本文模型与 8 种典型模型就性能方面在 4 种数据集下做对比, 阐述模型 LG_APIF 的优势. 在其中 3 种数据集上, 本文模型无论是 *Precision*, *Recall*, 还是 *NDCG* 指标都具有明显优势, 且整体平均提升达 1.53%, 其主要原因是我们的模型比较关注项目的分类方式. 尽管另一个数据集存在类标签与评分信息不具代表性等问题, 但模型 LG_APIF 还是优于大部分对比模型.

RQ4: 与端到端的模型、推荐框架结构相似的模型相比, 本文模型的效率如何?

针对此问题, 在第 4.4.4 节展开了复杂度分析, 先对本文模型 LG_APIF 的时间复杂度 (即, 各模块时间复杂度之和) 进行分析, 再与一些端到端的模型, 或推荐框架结构相似的模型在时间复杂度方面做对比, 以评估本文模型的效率. 经分析得出, 模型 LG_APIF 在时间复杂度上确实有一定的牺牲, 但结合第 4.4.3 节中该模型的较好推荐效果, 并展开进一步分析, 可知模型 LG_APIF 的设计思想是合理的、有意义的.

RQ5: 模型有哪些地方还可进一步优化?

针对此问题, 在第 4.4.5 节设置优化实验, 展开 LG_APIF 模型优化前后的对比, 以阐明优化效果. 我们对兴趣量因子解析过程中的切分指数进行重新定义, 使其由定长改为可随距离当前时刻远近而变化的变长形式, 从而实现兴趣量因子的自适应性. 通过对比, 优化后的性能得到一定提升, 整体平均提升达 5.04%.

以下实验均在处理器为 AMD Ryzen 7 3700X 8-Core, 内存为 32 GB, 操作系统为 Win 10 64 位, 编程工具为 PyCharm 2021, 编程语言为 Python 3.6 的环境下进行.

4.1 数据集

本文选取的 4 个公开数据集均源于真实业务场景, 它们均为近几年被大量学者使用和研究的经典数据集, 分别是 Last.fm, Douban, Yelp 以及 MovieLens. 为更好地比较本文模型在各数据集上的表现, 从多角度分析数据集的特点, 统计结果如表 1 所示.

表 1 数据统计表

统计数据	Last.fm	Douban	Yelp	MovieLens
用户数	1 892	129 490	963 923	71 567
项目数	17 632	18 541	2 353 207	10 681
类别数	916	658	72	427
时间区间	1个月	1 000时间标记	5个月	7个月
大小	2 KB	20 MB	100 MB	10 MB

© 中国科学院软件研究所 http://www.jos.org.cn

1) Last.fm (源自 https://grouplens.org/datasets/hetrec-2011/): 音乐收听数据. 数据标签包括歌曲播放次数、歌曲 ID、歌曲类型等.

2) Douban (源自 https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban): 电影点评数据. 数据标签包括用户对电影的评分记录 (每个用户最少评分电影数为 20)、用户的人口统计信息 (年龄、性别)、评分时间、评论评分, 以及电影的统计信息 (名称、类别、描述).

3) Yelp (源自 https://www.yelp.com/dataset): 餐厅点评数据. 数据标签包括用户对商家的评论及评分等.

4) MovieLens (源自 https://grouplens.org/datasets/movielens/): 电影点评数据. 数据标签包括用户与电影的基本特征、评分、ID 和电影类别等.

4.2 评价指标

性能评价主要体现在精确度上, 为验证提出的算法是否能提高推荐性能, 选取常用的准确率 (*Precision*)、召回率 (*Recall*) 和归一化折损累计增益 (normalize discounted cumulative gain, *NDCG*) 作为评价指标.

1) *Precision* 和 *Recall*: 用来直接评价推荐算法的精确度, 如公式 (20) 所示:

$$Precision=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|},\quad Recall=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{20}$$

公式 (20) 中, $R(u)$ 和 $T(u)$ 分别为用户 u 推荐的项目列表、用户实际选择的项目列表, $|R(u)|$, $|T(u)|$ 为其对应数量; $R(u)\cap T(u)$ 为最终推荐列表中被用户实际选择项目的集合, $|R(u)\cap T(u)|$ 为该集合中的项目数量.

2) *NDCG* 是对位置敏感的评价指标, 推荐项在列表中位置越靠前, 其贡献越大, 反之越小, 如公式 (21) 所示:

$$NDCG=\sum_{u\in U_{\text{test}}}\frac{1}{Y_u}\sum_{i=1}^{k}\frac{2^{t_i}-1}{\log_2(i+1)} \tag{21}$$

其中, U_{test} 为测试用户集合; Y_u 为用户 u 的最大 *NDCG* 值; k 为已推荐项目的数量; t_i=1 表示击中, t_i=0 表示未击中.

4.3 对比模型

下面将本文提出的 LG_APIF 模型与相关模型 BPR[27], DiffNet++[17], DHCF[28], DeepFM[29], LightGCN[3], S^2-MHCN[20], SEPT[21]及 FAWMF[30]做对比实验, 验证 LG_APIF 模型的有效性.

1) BPR: 一种流行且经典的基于贝叶斯个性化排名的推荐模型, 通过学习用户每对已评分与未评分项目的关系来模拟候选项目的顺序.

2) DiffNet++: 一种最新的基于 GCN 的社交推荐方法, 模拟了用户和项目空间中的递归动态社交扩散.

3) DHCF: 一种基于超图卷积网络的方法, 用于建模用户和项目之间的高阶相关性来实现推荐.

4) DeepFM: 一个神经网络架构, 结合用于推荐的分解机器和用于特征学习的深度学习技术, 推出一个端到端的学习用户复杂行为交互的模型, 并强调低阶和高阶特征交互的重要性.

5) LightGCN: 一种基于 GCN 的通用推荐模型, 利用用户与项目的邻近性来学习节点表示并生成推荐, 被认为是目前比较先进的方法之一.

6) S^2-MHCN: 一种基于超图卷积网路的社交推荐方法, 将用户与用户之间的深层关联性采用超边表示.

7) SEPT: 一种集成了三训练的通用社会感知 SSL (secure sockets layer) 框架的推荐模型. 该框架在增强视图上构建了 3 个图形编码器 (其中 1 个为推荐), 利用 2 个编码器生成的其他用户的自我监督信号对每个编码器进行迭代改进, 以增强推荐性能.

8) FAWMF: 一种基于变分自编码的快速自适应加权矩阵分解方法. 采用参数化的神经网络自适应分配个性化的数据置信度权重, 并从数据中推导出网络, 对所有反馈数据进行训练, 用于推荐.

选取的这几种作为对比的深度模型, 分别考虑了不同的信息源. BPR 无论是对于基础或深度信息的利用率都不高; DiffNet++, DHCF 及 LightGCN 则相继对基础信息进行了更充分地利用, 并引入一些深度方法; DeepFM, S^2-MHCN, SEPT 及 FAWMF 在引入深度方法的同时, 还加强了在基础信息下对深度信息的挖掘与利用. 这有利于

© 中国科学院软件研究所 http://www.jos.org.cn

从各方面将提出的模型 LG_APIF 与上述模型展开对比分析, 并据此来验证模型 LG_APIF 的有效性. 为更直观地阐明本文模型与对比模型的联系与区别, 对各模型进行整体比较, 如表 2 所示.

表 2 各模型特征对比

模型	基础信息	深度信息	方法复杂度
BPR	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆
DiffNet++	★☆☆☆☆	☆☆☆☆☆	★☆☆☆☆
DHCF	★☆☆☆☆	☆☆☆☆☆	★☆☆☆☆
DeepFM	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆
LightGCN	★★☆☆☆	☆☆☆☆☆	★★☆☆☆
S^2-MHCN	★★★☆☆	★☆☆☆☆	★★★☆☆
SEPT	★★★★☆	★★☆☆☆	★★★☆☆
FAWMF	★★★★☆	★★☆☆☆	★★★☆☆
LG_APIF	★★★★★	★★★★★	★☆☆☆☆

注: 采用五星打分制表示信息利用程度或方法复杂程度的深浅, “★”数量越多表示程度越深

4.4 实验结果分析

下面结合各节的实验或分析验证本文模型的有效性. 其中, 在第 4.4.1 节对模型的 2 种关键参数进行调优操作, 以回答 RQ1; 在第 4.4.2 节针对本文模型的 3 大构件设计了 3 种变体模型, 将这些模型与本文模型进行消融实验对比分析, 以回答 RQ2; 在第 4.4.3 节将本文模型与 7 种经典的、最新的相关模型做对比实验, 通过对比结果, 详细分析并阐明本文模型的有效性及其优势, 以回答 RQ3; 在第 4.4.4 节对本文模型与相关对比模型做复杂度对比与分析, 阐明了本文模型的效率及依存的优势, 以回答 RQ4; 在第 4.4.5 节对本模型在设计上的不足进行补充, 通过模型优化前后的对比结果, 来阐述模型优化的效果及可行性, 以回答 RQ5.

4.4.1 模型参数调优

在 4 种不同主题的典型数据集 (Last.fm, Douban, Yelp 及 MovieLens) 上进行实验, 删除了用户与项目类型的交互次数均小于 50 的用户, 以及不足 10 个项目的项目类型数据. 将各数据集的 70% 作为训练集, 30% 作为测试集, 且进一步将训练集和测试集分别按照时间节点以 7:3 的比例划分为模型训练数据和真实标签数据. 超参数分别包括周期间隔数 (Time_Period_Num, TPN)、潜在项目类型数 (CAT_Top_N, CTN).

1) 周期间隔数 TPN

TPN 是用户兴趣变化周期间隔大小的因素, 代表用户的浏览历史时间间隔的切分块数. TPN 过大会导致单位时间间隔过小, 造成数据稀疏性加大, 兴趣变化幅度过于敏感; 而 TPN 过小会导致单位时间间隔过大, 则会丢失某些重要时刻的兴趣变化. 因此, TPN 在很大程度上决定用户整体的兴趣变化趋势. 实验中, TPN 的值设置为{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25}, 其各指标随 TPN 的变化情况如图 7 所示.

由图 7 可知, 随着 TPN 增大, 模型 LG_APIF 在不同数据集下的表现有较明显差异: 在 Last.fm 中, TPN 取 3 时, 各项指标都处于最大值; 在 Douban 中, 比较 3 个评价指标的变化趋势, TPN 的最优值分布在区间 [15, 19], 其中 TPN 取 15 时, *Precision* 和 *Recall* 值均达到最大, *NDCG* 值还有一些上升, 但不明显; 在 Yelp 中, TPN 取 7 时, *Recall* 达到最大值, 且 *Precision* 和 *NDCG* 值尽管不是最大, 但和最大值相差甚小; 在 MovieLens 中, TPN 取 11 时, *NDCG* 达到明显的峰值, *Precision* 和 *Recall* 值在区间 [9, 13] 波动不大.

由上可知, 模型 LG_APIF 在不同数据集下性能最优时, TPN 的取值均不一样, 我们认为这是合理的, 因为每个数据集的时间周期跨度均不一致, 只有当切分周期的周长设置合理时, 模型的性能才可达到最优. 因此, 我们在 Last.fm, Douban, Yelp 及 MovieLens 中, 对 TPN 的最优取值分别为 3, 15, 7 和 11.

2) 潜在项目类型数 CTN

CTN 是兴趣类型分布范围的关键因素, 它代表需要确定用户的潜在兴趣类型数量, CTN 过大会导致推荐的关

© 中国科学院软件研究所 http://www.jos.org.cn

注点过于分散, 使得模型很难侧重于某些具体的项目, 会很大程度上降低推荐的准确率; CTN 过小会导致推荐的关注点太过于集中, 模型很容易因为某次推荐的偏差过大, 而导致整个推荐结果表现不佳, 模型的稳定性会大幅降低. 因此, CTN 决定了模型的稳定性. 实验中的 CTN 值设置为{1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60}, 各指标值随 CTN 的变化趋势如图 8 所示.

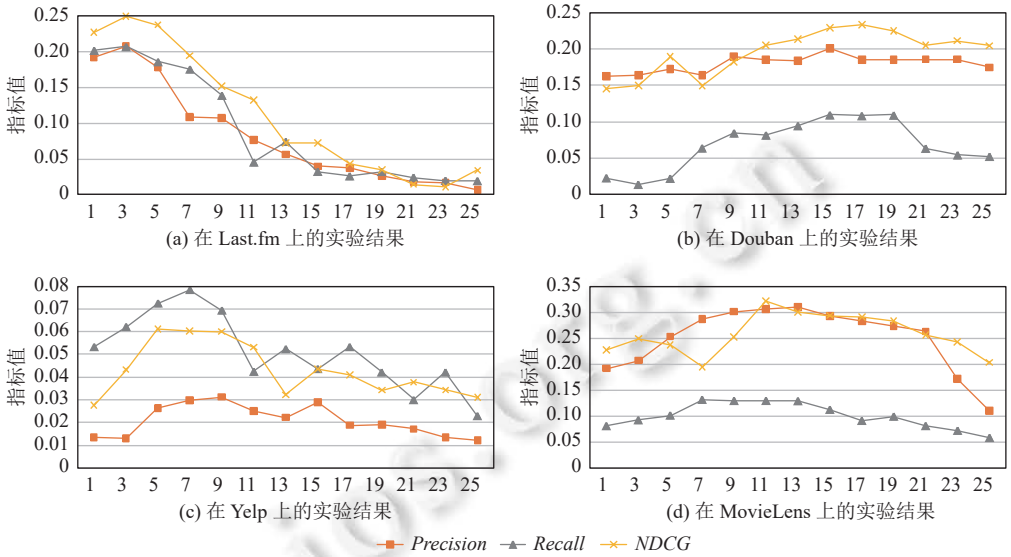


图 7 周期间隔数的影响对比

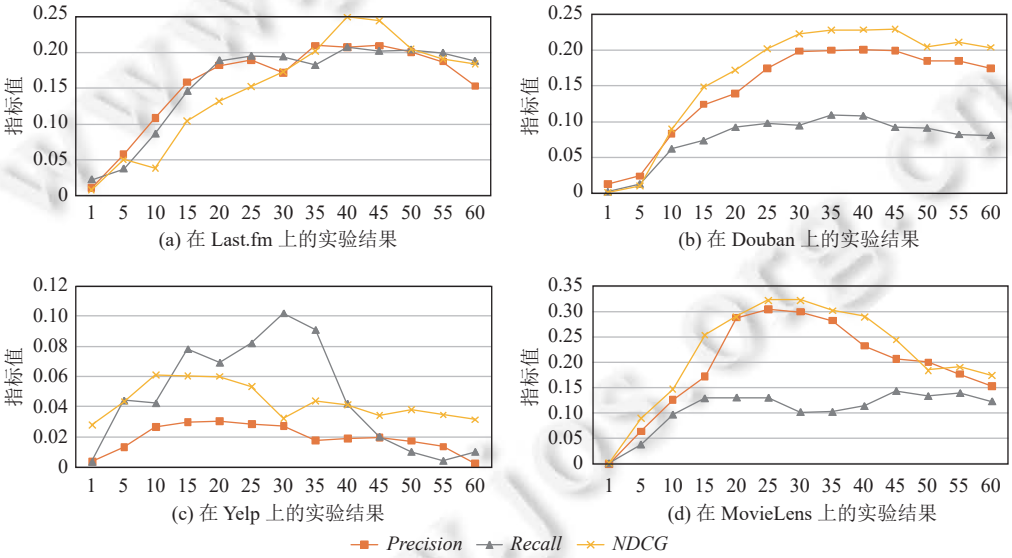


图 8 潜在项目类型数的影响对比

由图 8 可知, 随着 CTN 增大, 模型 LG_APIF 在不同数据集下的表现存在较明显差异: 在 Last.fm 中, CTN 取 40 时, *Recall* 和 *NDCG* 值均达到最大值, *Precision* 值在区间 [35, 45] 变化不大; 在 Douban 中, *Precision* 和 *NDCG* 值在区间 [35, 45] 波动不大, 且 *Recall* 值在 TPN 取 35 时, 达到最大值; 在 Yelp 中, *Precision* 和 *NDCG* 值在区间 [10, 20] 相差甚微, *Recall* 值虽在 CTN 取 30 时, 取得最大值, 但其他两项指标值均大打折扣, 明显低于最优值; 在 MovieLens 中, TPN 取 25 时, *Precision* 值和 *NDCG* 值均取得最大值, *Recall* 值在区间 [15, 25] 变化不大.

© 中国科学院软件研究所 http://www.jos.org.cn

由上可知, 模型 LG_APIF 在不同数据集下性能最优时, CTN 的取值差异很大, 因为每个数据集的项目分类方式, 以及项目类型数和个体数的占比均不一致, 且各用户在不同领域中表现出的偏好方式也有所区别. 因此, 我们在 Last.fm, Douban, Yelp 及 MovieLens 中, 对 CTN 的最优取值分别为 40, 35, 15 和 25.

综合以上两个重要参数的调优实验, 旨在选取能使模型 LG_APIF 达到最优推荐效果的参数. 此外, 不难看出 TPN 和 CTN 均直接影响着兴趣量因子等特征的求解和用户-类型-项目三元图结构的构建, 这些均足以表明本文重点挖掘的深度信息确实深刻地影响着推荐效果.

4.4.2 不同模型构件消融实验分析

由前面可知, 模型 LG_APIF 的关键是求解自适应周期和兴趣量因子, 即分别为 3.1.2 节遗忘曲线规律的公式 (2) 中两个重要参数 t 和 φ, 利用公式 (2) 求出用户感兴趣的项目类型, 来进一步构建用户-类型-项目三元图结构, 最终通过轻量级 GCN 的高阶传播, 求出用户或项目的最终嵌入表示, 即用户对项目的最终得分.

显然, 自适应周期、兴趣量因子及用户-类型-项目三元图结构是模型 LG_APIF 的 3 个主要构件, 为验证它们的必要性, 构建 3 种变体模型 LG_XIF、LG_APX 及 X_APIF 与本文的 LG_APIF 进行消融实验对比. 表 3 是几种变体模型的影响因素对比情况 (其中, “×”表示该构件被替换; “○”表示该构件未被替换).

表 3 各模型构件影响对比

变体模型	自适应周期	兴趣量因子	三元图结构
LG_XIF	×	○	○
LG_APX	○	×	○
X_APIF	○	○	×
LG_APIF	○	○	○

但与传统消融实验不同的是, 这些变体模型并不是去掉原模型的某些构件, 而是进行构件替换. 比如, 本文的自适应周期是历史兴趣在此次推荐时刻与其在未来将会被遗忘的时刻 (该时刻与推荐时刻无关) 之差, 而在大多数模型中直接被认为就是两次推荐时刻之差, 所以变体模型 LG_XIF 中的自适应周期求解部分采用两次推荐的时刻差进行替换; 变体模型 LG_APX 的兴趣量因子解析部分采用大多数模型一样的方式进行替换, 即通过用户的评分或浏览次数直接求解兴趣量; 变体模型 X_APIF 则是将传统的用户-项目交互图替换成用户-类型交互图, 求得目标兴趣项目类型后, 再通过协同过滤的方式确定类型中的目标项目.

在 Last.fm, Douban, Yelp 及 MovieLens 这 4 个公开数据集上实验, 考虑到变体模型的目的是验证模型 LG_APIF 中各关键构件对推荐性能的影响, 这里仅选择准确率 (*Precision*) 指标进行评估, 如图 9 所示.

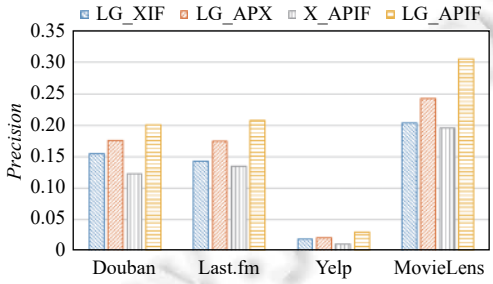


图 9 各变体模型与 LG_APIF 的 *Precision* 对比

从图 9 可看出, 用户-类型-项目三元图构件的影响最为明显, 变体模型 X_APIF 基于用户-类型交互图, 先预测潜在项目类型, 再进行个体预测, 显然其效果远低于模型 LG_APIF. 另外, 替换自适应周期为两次推荐时刻差的变体模型 LG_XIF 的推荐准确率也远低于模型 LG_APIF. 此外, 采用评分或浏览次数来计算兴趣量的变体模型 LG_APX 的推荐性能也逊色于模型 LG_APIF.

1) 用户-类型-项目三元图构件对推荐性能的影响最大. 替换三元图构件的变体模型 X_APIF 虽然也遵从本文

© 中国科学院软件研究所 http://www.jos.org.cn

所提到的用户兴趣规律, 即先求得项目类型再推荐项目个体. 但高阶传播中以项目类型为节点, 最终可能确实可找到用户的兴趣项目类型, 但对于项目个体的预测, 依然很困难. 相比之下, 本文模型 LG_APIF 将项目类型用作连接用户与项目个体间的关系, 来构建用户-类型-项目三元图结构, 个体之间不仅蕴含着类型的潜在信息, 而且蕴含更多的个体协作信息, 这十分有利于个体信息在高阶传播中的学习. 图 9 的实验结果也显示, LG_APIF 的推荐性能远优于 X_APIF.

2) 自适应周期在很大程度上影响着推荐性能. 自适应周期的取值能随着数据集的变化和在多次推荐之后伴随着数据的变化而变化. 因此, 自适应周期的值在每一次推荐时是动态变化的, 这和变体模型 LG_XIF 所设置的固定值具有很大区别. 图 9 中, 本文模型 LG_APIF 的效果远优于变体模型 LG_XIF, 可见本文模型具有更好的适应性, 进而表明自适应周期对推荐效果的提升具有积极作用.

3) 运用本文的兴趣量因子解析方法可在一定程度上改善模型推荐效果. 变体模型 LG_APX 和提出的模型 LG_APIF 在处理兴趣量因子的方式上有相似之处. 比如, 这两个模型都利用了用户浏览次数等基础信息, 只不过与 LG_APX 直接将该基础信息转换成兴趣量因子的方法不一样的是, LG_APIF 进行了更深入的处理, 包括兴趣浮点模型的利用、兴趣量演算的优化等. 通过图 9 中的对比, LG_APIF 在很大程度上优于变体模型 LG_APX, 这不仅说明 LG_APIF 中兴趣量因子解析模块的有效性, 还反映了在改善推荐效果的过程中, 利用重组、优化等方法挖掘到的深度信息要比显式的基础信息更有效.

综上所述, 自适应周期、兴趣量因子等深度信息对提升推荐效果具有很大作用, 而用户-类型-项目三元图结构则是在对深度信息求解后的应用, 因此其不仅自身影响着推荐性能, 还融合自适应周期和兴趣量因子对模型的影响. 结合图 9 可看出, 变体模型 X_APIF 远比其他变体模型的推荐效果差, 说明三元图结构在本文模型中起到至关重要的作用, 同时也表明对基础信息进行深度挖掘利用, 是提升推荐效果的关键.

4.4.3 各模型性能对比

为评估提出的模型 LG_APIF 效果如何, 表 4 给出了各模型在不同数据集下的 *Precision* 值、*Recall* 值和 *NDCG* 值 (在表中分别简写为 P, R, N) 的对比情况.

表 4 各模型在不同数据集上的指标对比

数据集	评价指标	模型名称									性能改善情况 (%)
		BPR	DiffNet++	DHCF	DeepFM	LightGCN	S^2-MHCN	SEPT	FAWMF	**LG_APIF**	
Last.fm	P	0.1561	0.1849	0.1688	0.1991	0.1921	0.2005	**0.2019**	—	**0.2076**	↑0.84
	R	0.1582	0.1874	0.1713	0.1842	0.1948	0.2038	**0.2049**	—	**0.2073**	↑1.17
	N	0.1895	0.2231	0.2074	0.1783	0.2340	0.2440	**0.2451**	—	**0.2499**	↑1.96
Douban	P	0.1567	0.1753	0.1687	0.1839	0.1779	0.1851	0.1883	**0.1902**	**0.2001**	↑5.20
	R	0.0516	0.0621	0.0576	0.0434	0.0625	0.0668	0.1081	**0.0880**	**0.1092**	↑1.02
	N	0.1748	0.1970	0.1866	0.1632	0.1989	0.2104	**0.2264**	0.2215	**0.2288**	↑1.06
Yelp	P	0.0200	0.0248	0.0300	0.0293	0.0259	**0.0300**	0.0290	—	**0.0297**	↓1.00
	R	0.0517	0.0635	0.0599	0.0601	0.0653	**0.0789**	0.0746	—	**0.0783**	↓0.76
	N	0.0384	0.0431	0.0470	0.0431	0.0500	**0.0606**	0.0578	—	**0.0602**	↓0.66
MovieLens	P	0.2891	—	—	—	—	—	—	**0.3046**	**0.3055**	↑0.30
	R	0.1032	—	—	—	—	—	—	**0.1290**	**0.1301**	↑0.85
	N	0.2980	—	—	—	—	—	—	**0.3183**	**0.3228**	↑1.41

其中, 本文模型和目前效果最佳模型的指标值已加粗显示, 最后一列给出提出的模型 LG_APIF 相对于某一最佳模型 (粗体数据表示) 的性能改善情况 (比如, 在 Last.fm 上, SEPT 在对比模型中的 *Precision* 表现最优, 相对于它, 本文模型 LG_APIF 提升 0.84%). 对于 MovieLens, 由于它与 Douban 参考的均是电影主题数据, 只在信息侧重

© 中国科学院软件研究所 http://www.jos.org.cn

点上表现不一致, 因此仅考虑模型 BPR 和 FAWMF 在 MovieLens 上的表现. 另外, 由于 Last.fm 和 Yelp 缺少模型 FAWMF 需要的相关信息, 因此仅考虑该模型在 Douban 和 MovieLens 上的表现.

由表 4 可知, 相比于其他模型, 本文模型 LG_APIF 在 *Precision*, *Recall* 和 *NDCG* 上均有较大幅度的提升, 整体平均提升达到 1.53%. 从数据集的角度上看: 在 Douban 上的表现明显要优于其他各种模型, 平均提升 2.43%; 在 Last.fm 和 MovieLens 上分别平均提升 1.32% 和 0.85%; 仅在 Yelp 数据集下, 改善效果不是很理想. 从指标值的角度上看: *Precision* 值、*Recall* 值和 *NDCG* 值分别平均提升 2.11%, 1.01% 和 1.48%.

整体来看, 在 Last.fm, Douban 与 MovieLens 上, 模型 LG_APIF 的指标值均高于其他模型; 在 Yelp 上, 模型 LG_APIF 虽较 S^2-MHCN 稍逊色, 但整体指标值仍均明显优于其他对比模型. 主要优势有:

1) 与 8 个对比算法相比, 提出的模型 LG_APIF 在推荐性能方面整体上有较大幅度提升. 特别是在 Douban 数据集下, 优势更加明显, 我们认为这是因为电影项目在分类时, 具有明确的分类标准, 不同的电影表现出的情感和主题有明显区别, 故用户受到分类标签影响更大. 某用户喜欢看青春偶像剧, 那他对该类型电影关注度也可能会更高; 某用户喜欢某位明星, 那他可能会对该明星主演的电影毫不抗拒. 当然, 为验证模型在电影领域应用的适用性, 我们还特别增加了 MovieLens 数据集, 该数据集同样记录有关用户-电影项目的数据信息, 由表 4 可知, 模型 LG_APIF 在该数据集下表现依然优秀, 其原因与上述一致.

2) 在 Last.fm 数据集下, 提出的模型 LG_APIF 表现出来的优势也很明显, 我们认为这是由于音乐在分类方式上界限比较明显, 比如民谣和摇滚音乐, 这两种音乐类型具有很大区别. 另外, 音乐的听取意向更易表达出近段时间内的兴趣倾向. 因此, 这再一次验证了本模型对项目类型具有严格的分类标准. 相对于对比模型, 模型 LG_APIF 在产生推荐结果之前就已经限定了用户的兴趣范围 (即, 项目类型), 所以, 本模型对噪声的反应并不会太敏感, 即具有很好的稳定性, 从而使得本文模型在召回率方面表现突出.

3) 为更准确评估各推荐模型推荐列表中的项目排名质量, 还对各推荐模型的 NDCG 值进行了计算. 由表 4 可知, 模型的 *NDCG* 指数越高, 表明模型推荐的项目与用户最终所选择项目的相关性越高, 模型输出的推荐列表就越符合用户需求. 在其中 3 种数据集实验中, 模型 LG_APIF 的效果均优于其他对比模型, 这也表明推荐系统中的兴趣量因子和兴趣类型的确定对于及时发现并满足用户需求非常重要.

此外, 在 Yelp 中, 尽管模型 LG_APIF 的指标值高于某些模型, 但并不是最高, 尤其对比模型 S^2-MHCN 来说, 本文的模型存在一些不足之处. 本文模型 LG_APIF 在 Yelp 数据集上的表现一般, 其主要原因是由于用户对餐厅的分类以及评分信息敏感性不高, 在餐厅类型标签的设置上一般比较模糊. 比如, 面食店也卖盖浇饭, 西餐厅中也会有少量的中餐. 另外, 用户的评分一般主要针对餐厅某个方面的不足, 用户在选择餐厅时, 评分信息并不是唯一标准, 往往也会被餐厅的位置、装饰或餐厅中某种特定美食所吸引. 可见, Yelp 数据集的数据标签和评分信息不具有代表性, 这也说明本文模型在数据运用上有所欠缺.

在上面模型的对比实验及对结果的分析中 (详见表 4), 就模型 LG_APIF 的优势以及不足之处均进行了详细讨论, 可将它的特点整体概括为以下 3 点.

① 在对应用环境的理解方面. 模型 LG_APIF 在 Douban 数据集上, 无论是 *Precision*, *Recall*, 还是 *NDCG* 指标上都具有明显的优势, 且平均提升 2.43%. 主要原因是由于我们更注重项目分类的方法上, 这一点在同为项目分类方式较为明显的 Last.fm 和 MovieLens 上得到了验证. 因此, 这也表明了模型 LG_APIF 更适用于处理分类方式具有明确规则的环境.

② 在对用户的理解方面. 模型 LG_APIF 在其中的 3 种数据集上的 *NDCG* 指标上都有着比较明显的优势, 且平均提升 1.48%, 这说明我们的模型能比较好地适应用户. 用户的整体兴趣能在很大程度上被长期兴趣影响, 同时由于用户的每一个最近点击也会影响用户的兴趣量, 即模型也能对短期兴趣进行学习, 并试图以此替代长期没有启动的兴趣.

③ 在对项目的理解方面. 随着迭代次数的增加, 模型 LG_APIF 的性能会在一定区间内保持上升状态, 且在多次训练后, 模型的预测效果会越来越好. 主要原因是模型中项目的兴趣定位, 均由每个用户的行为密切影响. 随着模型的成熟 (迭代次数增加), 项目的定位也会变得更为明确, 这也充分说明 LG_APIF 模型设计的有效性.

© 中国科学院软件研究所 http://www.jos.org.cn

4.4.4 模型复杂度分析

所提模型 LG_APIF 先利用相对轻量的传统技术来解析自适应周期、兴趣量因子等深度信息, 再结合轻量级 GCN 技术进行推荐. 与现有模型相比, 推荐步骤相对增多, 尽管模型的效果得到一定程度的提升, 但其复杂度也随之稍有增加. 下面就时间复杂度方面对相关模型的框架展开性能评估.

1) 就本文模型 LG_APIF 的时间复杂度来说, 主要表现在自适应周期生成、兴趣量因子解析以及 GCN 推荐 3 个模块. 对于自适应周期生成部分, 又分为聚类过程和线性回归过程: 聚类过程的时间复杂度为 O(*EnKId*), 其中 *E* 为用户数, *n* 是平均交互项目量, *K* 为聚点数, *I* 是迭代次数, *d* 为数据维度, 通常 *K*, *I*, *d* 均可认为是常量; 线性回归过程采用逆矩阵的求解方式, 即时间复杂度为 O(K^3). 对于兴趣量因子解析部分, 由于行为记忆序列长度和单位周期数量均是预知的, 故其时间复杂度为 O(1). 对于 GCN 推荐部分, 时间复杂度为 O(*MDLd*), 其中 *M* 是交互图节点数, *D* 为扩散深度, *L* 是边数量. 因此, 本文模型的时间复杂度大致为 O(*En*+*MDL*).

2) 就对比模型 BPR (该模型相对最早, 效果最差) 的时间复杂度来说, 时间主要消耗在处理反馈信息的排序问题上. 因此, 时间复杂度大致为 O(*DL*).

3) 就对比模型 LightGCN (该模型相对较新, 与本文模型结构最为相似) 的时间复杂度来说, 时间主要消耗在图传递层的动态扩散过程中. 因此, 时间复杂度大致为 O(*MDL*).

综上分析, 各模型的时间复杂度大小关系为: 本文模型 LG_APIF > LightGCN > BPR. 尽管从复杂度上来看, 模型 LG_APIF 的效率要劣于对比模型, 但结合实验的推荐效果来说, 其设计思想依然是合理的、有意义的, 主要理由如下.

1) 就本文模型 LG_APIF 的时间复杂度与模型 LightGCN 相比: 由于交互图节点数 *M* 一般包括用户数 *E* 和项目数, 且 *n* 为平均交互项目数, 故在大部分情况下, *En* 并不会比 *M* 大很多, 甚至会更小 (比如, 长尾分布严重时). 此外, 尽管 LG_APIF 模型的线性回归部分时间复杂度为 O(K^3), 但 *K* 一般可取随机常量, 具有可控性, 通常数量级也不会很大, 对整体模型的时间复杂度影响有限. 因此, 本文模型的时间复杂度并不会比对比模型 LightGCN 高很多, 甚至在大多情况下, 均处于相同数量级的, 但 LG_APIF 模型的推荐效果却要比 LightGCN 好很多. 这足以说明, 本文模型在时间复杂度上的牺牲是值得的.

2) 就本文模型 LG_APIF 的时间复杂度与模型 BPR 相比: 尽管模型 LG_APIF 的复杂度约为模型 BPR 的 *M* 倍, 但推荐效果上, 却比模型 BPR 提升约 33% (见表 4, 以 Last.fm 数据集及 *Precision* 评价指标为例), 且就目前大多数 GPU/CPU 来说, 本文模型的计算任务并不算大. 因此, 本文模型是可行的、有意义的.

3) 与传统端到端的模型相比, 本文模型 LG_APIF 在组成结构上也具有一定的优势. 由于采用多模块组成形式, 使得模型 LG_APIF 能利用分布式计算的优势, 即只要计算资源充足, 模型 LG_APIF 的整体推荐性能也会得到很大的提升. 此外, 虽然本文的推荐框架增加了特征提取模块, 但采用 GCN 技术是经过简化的, 与传统的基于 GCN 的推荐相比, 在高阶传播时, 所消耗的时间实际上相对较少. 因此, 若与达到相当推荐效果的模型相比, 模型 LG_APIF 是较为轻量的.

4.4.5 兴趣量因子自适应优化

模型 LG_APIF 的主要思想是加大对基础信息的挖掘力度, 以提取出与其他对比模型相区别的、高价值的深度信息. 为进一步验证用户基础信息蕴含的深度信息对推荐性能的影响, 该小节补充了相关优化实验.

由第 3.3 节可知, 周期间隔数 TPN 是模型 LG_APIF 在行为记忆重组 (见第 3.3.1 节) 过程中兴趣指数的关键参数, 影响着兴趣量因子, 进而决定模型的准确性. 但前面的 TPN 均依据数据集的时间区间来手动设置, 这很容易人为地带来误差. 另外, 经实验 (见第 4.4.1 节) 发现, 不同数据集的 TPN 设置差异很大, 难以保证参数设置的公平性, 降低了实验可信度. 为解决上述问题, 对模型 LG_APIF 设计了几组针对性的优化实验, 目标是达到 TPN 自动设置的效果, 实现兴趣量因子的自适应性. 基于此, 对 TPN 的相关定义进行 2 点补充.

① 近期行为比历史行为更重要. 即近期发生的行为往往要比历史中的行为更能影响当前的兴趣. 但前面实验设置的 TPN, 将近段周期和历史周期的划分单位和方式均一样, 无法表达近期行为的重要性. 因此, TPN 的划分方式应有所区别, 且随着用户的历史浏览周期慢慢靠近当前时刻, TPN 还应逐渐变小.

② 兴趣指数应呈“蟒蛇状”分布. 蟒蛇头部的粗大应表现在 *Sigmoid* 函数图像的右端部分, 即浏览的正样例 (有

© 中国科学院软件研究所 http://www.jos.org.cn

行为的单位周期) 部分, 腹部的肿大表现在 *Sigmoid* 函数图像的 y 轴两侧部分, 而纤细的尾部则是 *Sigmoid* 函数图像的左端部分, 即浏览的负样例 (未有行为的单位周期) 部分. 比如, 应认为在一个相对较长的历史周期内, 正样例 (或负样例) 间隔为 m 的两个连续出现的较大次数对最终结果的影响区别不大. 这个差距 m 不能与间隔仍为 m 的两个连续出现的较小次数相对等. 尽管兴趣量演算过程中, 引入双曲正切 *Sigmoid* 函数已解决了局部最大 (小) 的问题, 但当连续的正样例 (或负样例) 过大时, 会导致中间少数出现的负样例 (或正样例) 被丢弃, 很难被记录. 另外, 若能将两端分布的周期进行恰当的合并, 而将重心放在腹部的行为上, 这能大大减少兴趣量因子解析过程的迭代次数, 提升模型效率, 降低噪声影响, 增强稳定性.

因此, 如何合理地设置 TPN 就显得非常重要. 用户的历史周期可看成一段时间轴线段, 设置 TPN 就是如何将一条长度确定的线段进行切分. 假设该线段长度为 L, 第 i 段的切分长度为 L_i (即 TPN). 由第 3.3.1 节可知, 此时的 L_i 为定长, 且分为了 L/i 段, 但根据上述对 TPN 的 2 点补充, 这种设置是有缺陷的. 接下来, 我们将设计 3 种新的切分方式, 来实现上述要求, 以期提升模型效率.

● 方式 1. 模型 LG_APIF-*L*. 针对补充的第①点, 保证长度为 L 的线段从左端点到右端点的切分长度越来越小, 且设置合理的切分段数. 因此, 我们将 $2^n\times(n+1)>L$ (n=1, 2, 3, 4,…) 设为判断式, 递归执行, 直到 n 的取值首次使得判断式成立, 此时, 第 i 段切分长度 L_i 的计算方式如公式 (22) 所示:

$$L_i=\begin{cases}2^{n-1}, & 1\leqslant i\leqslant D\\ 2^{n-2}, & D<i\leqslant(2^2-1)D\\ \vdots & \vdots\\ 2^{n-n}, & (2^{n-1}-1)D<i\leqslant(2^n-1)D\end{cases}\qquad \text{其中, } D=L/(2^{n-1}(n+1)) \tag{22}$$

● 方式 2. 模型 LG_APIF-*Y*. 针对补充的第②点, 为避免正样例 (或负样例) 出现的频数过大, 则需预先设定最大正样例数 E^+ (或最大负样例数 E^-), 且应与浏览总周长相关联. 因此, 我们定义了映射函数, 当兴趣指数超出最大值时, 则通过该函数将其映射回 E^+ (或 E^-) 内, 此时兴趣指数的计算如公式 (23) 所示:

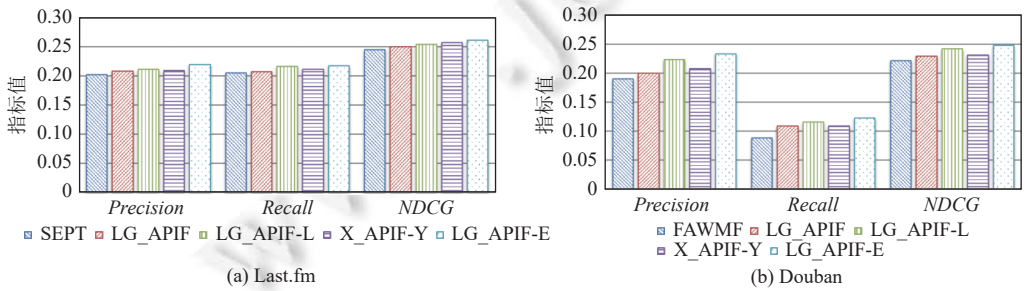
$$Y=E^-+\frac{E^+-E^-}{X_{\max}-X_{\min}}\times(X-X_{\min}) \tag{23}$$

其中, Y 表示当前兴趣指数 X 在超出 E^+ 或 E^- 范围时, 被映射回区间 [E^- , E^+] 的值.

● 方式 3. 模型 LG_APIF-*E*. 同时针对补充的第①点和第②点, 从左端到右端切分总长度为 L 的线段时, 不仅切分长度要越来越小, 且最大正样例数 E^+ 和最大负样例数 E^- 也要阶梯式地变小. 因此, 我们在方式 1 和方式 2 的基础上, 将 E^+ 和 E^- 的计算方式定义为如公式 (24) 所示:

$$E^+=L/(2^{n-1}\times(n+1));\quad E^-=-L/(2^{n-1}\times(n+1)) \tag{24}$$

图 10 展示了本文模型在分别考虑 3 种优化方式下的柱状图结果对比情况. 为说明上述优化方式的有效性, 这里选取了优化前本文模型 LG_APIF 表现较好的两种数据集 (Last.fm 和 Douban), 对比模型只选取了其中表现最优的模型 (见表 4), 以及优化后的模型 (LG_APIF-L, LG_APIF-Y, LG_APIF-E), 来比较它们在 3 种评价指标下的实验效果. 其中, 横坐标选取不同评价指标, 纵坐标为评价指标值.



(a) Last.fm (b) Douban

图 10 模型 LG_APIF 在 3 种优化方式下的效果对比

© 中国科学院软件研究所 http://www.jos.org.cn

从图 10 可看出, 模型 LG_APIF 在优化方式 1、方式 2 及方式 3 时, 性能均优于现有对比模型, 且在整体上也要优于优化前的模型自身, 平均提升可达 5.04%. 尽管性能提升不明显, 但也充分说明我们对于模型 LG_APIF 中兴趣量因子自适应的优化思路是可行的、有效的, 也表明加大挖掘用户深度信息力度的必要性, 以及采用更为有效的方法去归结行为规律, 会进一步提高推荐性能, 并带来更强的可解释性.

5 总结与下一步工作

我们提出一种融合自适应周期与兴趣量因子的轻量级 GCN 推荐方法 LG_APIF, 通过运用相对轻量级的技术对用户基础信息的深度挖掘与利用, 不仅改善了模型 LG_APIF 的 *Precision*, *Recall* 和 *NDCG* 指标效果, 还在很大程度上控制了模型的复杂度. 其主要贡献如下.

1) 利用遗忘曲线构建兴趣模型, 更合理地模拟用户的兴趣变化规律. 本文将人类思维规律扩展运用到推荐系统中, 可较显式地表示用户的动态行为变化; 采用遗忘曲线拟合出的函数对用户的兴趣占比进行计算, 更准确地判别出用户当前的兴趣规律.

2) 挖掘自适应周期和兴趣量因子等深度信息, 更为准确地解析用户的整体偏好. 根据相似用户具有相似兴趣的规律对用户行为进行聚类并拟合, 预测用户的自适应周期; 再结合用户-项目的行为规律, 设计了一种更有效、更易理解的兴趣量因子求解方式. 这些深度信息的特征表示与基础信息的表示相似, 却蕴含着更高阶的协作, 有助于捕获用户的整体偏好, 也能为 GCN 推荐框架减负.

3) 设计用户-类型-项目三元图结构, 再结合减负后的 GCN 方法实现协同过滤, 更能降低 GCN 训练的难度并提升模型的推荐效果. 相较于传统的协同过滤方法, 本文模型采用 GCN 技术来完成协同过滤任务, 且为降低传统 GCN 技术的训练难度, 使用较为显式地携带协作信号的三元图结构作为模型的输入, 使模型能更侧重解决邻域聚合任务, 提高预测准确性.

本文主要以用户对项目的兴趣形态表现为中心, 以项目为主体对用户进行聚类分析. 实际上, 也能以用户为主体, 对不同项目进行兴趣周期聚类, 该方法同样适用. 当然, 本文模型也存在一些方面值得改进, 如部分冷启动用户的偏好仍未能得到准确预测. 此外, 本文用到的隐特征并不很全面, 社交关系、标签信息等特征未被使用. 接下来的工作将进一步充分利用其他多种信息, 使推荐模型的性能得到更好的优化.

References:

[1] Al-Shamri MYH. Power coefficient as a similarity measure for memory based collaborative recommender systems. Expert Systems with Applications, 2014, 41(13): 5680–5688. [doi: 10.1016/j.eswa.2014.03.025]

[2] Feng Y, Zhang B, Qiang BH, Zhang YY, Shang JX. MN-HDRM: A novel hybrid dynamic recommendation model based on long-short-term interests multiple neural networks. Chinese Journal of Computers, 2019, 42(1): 16–28 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.00016]

[3] Wang X, He XN, Wang M, Feng F, Chua TS. Neural graph collaborative filtering. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Paris: ACM, 2019. 165–174. [doi: 10.1145/3331184.3331267]

[4] He XN, Deng K, Wang X, Li Y, Zhang YD, Wang M. LightGCN: Simplifying and powering graph convolution network for recommendation. In: Proc. of the 43rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 639–648. [doi: 10.1145/3397271.3401063]

[5] Li QM, Han ZC, Wu XM. Deeper insights into graph convolutional networks for semi-supervised learning. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence and the 30th Innovative Applications of Artificial Intelligence Conf. and the 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI, 2018. 3538–3545.

[6] Garg D, Gupta P, Malhotra P, Vig L, Shroff G. Sequence and time aware neighborhood for session-based recommendations: Stan. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Paris: ACM, 2019. 1069–1072. [doi: 10.1145/3331184.3331322]

[7] Xu X, Ding SF, Ding L. Survey on density peaks clustering algorithm. Ruan Jian Xue Bao/Journal of Software, 2022, 33(5): 1800–1816 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6122.htm [doi: 10.13328/j.cnki.jos.006122]

[8] Ferdaous H, Bouchra F, Brahim O, Imad-Eddine M, Asmaa B. Recommendation using a clustering algorithm based on a hybrid features

© 中国科学院软件研究所 http://www.jos.org.cn

selection method. Journal of Intelligent Information Systems, 2018, 51(1): 183–205. [doi: 10.1007/s10844-017-0493-0]

[9] Tey FJ, Wu TY, Lin CL, Chen JL. Accuracy improvements for cold-start recommendation problem using indirect relations in social networks. Journal of Big Data, 2021, 8(1): 98. [doi: 10.1186/s40537-021-00484-0]

[10] Wang J, Li SJ, Yang S, Jin H, Yu W. A new transfer learning model for cross-domain recommendation. Chinese Journal of Computers, 2017, 40(10): 2367–2380 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.02367]

[11] Xu CH, Ding AS, Zhao KD. A novel POI recommendation method based on trust relationship and spatial-temporal factors. Electronic Commerce Research and Applications, 2021, 48: 101060. [doi: 10.1016/j.elerap.2021.101060]

[12] He XN, Zhang HW, Kan MY, Chua TS. Fast matrix factorization for online recommendation with implicit feedback. In: Proc. of the 39th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Pisa: ACM, 2016. 549–558. [doi: 10.1145/2911451.2911489]

[13] Liu XB, Nie XS, Yin YL. Mutual linear regression based supervised discrete cross-modal hashing. Journal of Computer Research and Development, 2020, 57(8): 1707–1714 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200122]

[14] Zhao N, Long Z, Wang J, Zhao ZD. AGRE: A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder. Knowledge-based Systems, 2023, 259: 110078. [doi: 10.1016/j.knosys.2022.110078]

[15] Zheng Y, Shi XM, Liu JX. Multi-path back-propagation method for neural network verification. Ruan Jian Xue Bao/Journal of Software, 2022, 33(7): 2464–2481 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6585.htm [doi: 10.13328/j.cnki.jos.006585]

[16] Yu F, Liu Q, Wu S, Wang L, Tan TN. A dynamic recurrent model for next basket recommendation. In: Proc. of the 39th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Pisa: ACM, 2016. 729–732. [doi: 10.1145/2911451.2914683]

[17] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 701–710. [doi: 10.1145/2623330.2623732]

[18] Wu L, Li JW, Sun PJ, Hong RC, Ge Y, Wang M. DiffNet++: A neural influence and interest diffusion network for social recommendation. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(10): 4753–4766. [doi: 10.1109/TKDE.2020.3048414]

[19] Wu F, Zhang TY, de Souza Jr AH, Fifty C, Yu T, Weinberger KQ. Simplifying graph convolutional networks. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 6861–6871.

[20] Yu JL, Yin HZ, Li JD, Wang QY, Hung NQV, Zhang XL. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proc. of the 2021 Web Conf. Ljubljana: ACM, 2021. 413–424. [doi: 10.1145/3442381.3449844]

[21] Yu JL, Yin HZ, Gao M, Xia X, Zhang XL, Hung NQV. Socially-aware self-supervised tri-training for recommendation. In: Proc. of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining. Singapore: ACM, 2021. 2084–2092. [doi: 10.1145/3447548.3467340]

[22] Liang YW. Design and implementation of English picture book reading recommendation system based on forgetting curve [MS. Thesis]. Beijing: Beijing Jiaotong University, 2019 (in Chinese with English abstract). [doi: 10.26944/d.cnki.gbfju.2019.000095]

[23] Shi LH, Kou Y, Shen DR, Nie TZ, Li D. Recommendation method based on multi-view embedding fusion for HINs. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3619–3634 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6632.htm [doi: 10.13328/j.cnki.jos.006632]

[24] Liu LL, Tan ZY, Shu J. Node importance estimation method for opportunistic network based on graph neural networks. Journal of Computer Research and Development, 2022, 59(4): 834–851 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.20200673]

[25] Peng L, Hu RY, Kong F, Gan JZ, Mo YJ, Shi XS, Zhu XF. Reverse graph learning for graph neural network. IEEE Trans. on Neural Networks and Learning Systems, 2022. [doi: 10.1109/TNNLS.2022.3161030]

[26] Wu L, Sun PJ, Fu YJ, Hong RC, Wang XT, Wang M. A neural influence diffusion model for social recommendation. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Pairs: ACM, 2019. 235–244. [doi: 10.1145/3331184.3331214]

[27] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In: Proc. of the 25th Int'l Conf. on Uncertainty in Artificial Intelligence. Montreal: AUAI Press, 2009. 452–461.

[28] Ji SY, Feng YF, Ji RR, Zhao XB, Tang WW, Gao Y. Dual channel hypergraph collaborative filtering. In: Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. San Diego: ACM, 2020. 2020–2029. [doi: 10.1145/3394486.3403253]

[29] Guo HF, Tang RM, Ye YM, Li ZG, He XQ. DeepFM: A factorization-machine based neural network for CTR prediction. arXiv:1703.04247, 2017.

[30] Chen JW, Wang C, Zhou S, Shi QH, Chen JB, Feng Y, Chen C. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. Proc. of the AAAI Conf. on Artificial Intelligence, 2020, 34(4): 3470–3477. [doi: 10.1609/aaai.v34i04.5751]

© 中国科学院软件研究所 http://www.jos.org.cn

附中文参考文献:

[2] 冯永, 张备, 强保华, 张逸扬, 尚家兴. MN-HDRM: 长短兴趣多神经网络混合动态推荐模型. 计算机学报, 2019, 42(1): 16–28. [doi: 10.11897/SP.J.1016.2019.00016]

[7] 徐晓, 丁世飞, 丁玲. 密度峰值聚类算法研究进展. 软件学报, 2022, 33(5): 1800–1816. http://www.jos.org.cn/1000-9825/6122.htm [doi: 10.13328/j.cnki.jos.006122]

[10] 王俊, 李石君, 杨莎, 金红, 余伟. 一种新的用于跨领域推荐的迁移学习模型. 计算机学报, 2017, 40(10): 2367–2380. [doi: 10.11897/SP.J.1016.2017.02367]

[13] 刘兴波, 聂秀山, 尹义龙. 基于双向线性回归的监督离散跨模态散列方法. 计算机研究与发展, 2020, 57(8): 1707–1714. [doi: 10.7544/issn1000-1239.2020.20200122]

[15] 郑烨, 施晓牧, 刘嘉祥. 基于多路径回溯的神经网络验证方法. 软件学报, 2022, 33(7): 2464–2481. http://www.jos.org.cn/1000-9825/6585.htm [doi: 10.13328/j.cnki.jos.006585]

[22] 梁艺伟. 基于遗忘曲线的英语绘本阅读推荐系统的设计与实现 [硕士学位论文]. 北京: 北京交通大学, 2019. [doi: 10.26944/d.cnki.gbfju.2019.000095]

[23] 石乐昊, 寇月, 申德荣, 聂铁铮, 李冬. 面向HIN基于多视角嵌入融合的推荐方法. 软件学报, 2022, 33(10): 3619–3634. http://www.jos.org.cn/1000-9825/6632.htm [doi: 10.13328/j.cnki.jos.006632]

[24] 刘琳岚, 谭镇阳, 舒坚. 基于图神经网络的机会网络节点重要度评估方法. 计算机研究与发展, 2022, 59(4): 834–851. [doi: 10.7544/issn1000-1239.20200673]

钱忠胜(1977－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为软件工程, 机器学习, 智能化软件.

张丁(1994－), 女, 硕士生, 主要研究领域为机器学习, 软件工程.

叶祖铼(1997－), 男, 硕士, 主要研究领域为推荐系统, 机器学习.

黄恒(1999－), 男, 硕士生, 主要研究领域为推荐系统, 机器学习.

姚昌森(1998－), 男, 硕士生, 主要研究领域为机器学习, 软件工程.

秦朗悦(1998－), 女, 硕士生, 主要研究领域为机器学习, 软件工程.

© 中国科学院软件研究所 http://www.jos.org.cn