软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2017,28(3):708−720 [doi: 10.13328/j.cnki.jos.005163]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 融合主题模型和协同过滤的多样化移动应用推荐*

黄 璐[1], 林川杰[1], 何 军[1], 刘红岩[2], 杜小勇[1]

[1](数据工程与知识工程教育部重点实验室(中国人民大学 信息学院),北京 100872)

[2](清华大学 经济管理学院,北京 100084)

通讯作者: 何军, E-mail: hejun@ruc.edu.cn

**摘 要**: 随着移动应用的急速增长,手机助手等移动应用获取平台也面临着信息过载的问题.面对大量的移动应用,用户很难找到最适合的;而另一方面,长尾应用淹没在资源池中不易被人所知.已有推荐方法多注重推荐准确率,忽视了多样性,推荐结果中多是下载量高的应用,使得推荐系统的数据积累越来越偏向于热门应用,导致长期的推荐效果越来越差.针对这一问题,首先改进了两种推荐方法,提出了将用户的主题模型和应用的主题模型与 MF 相结合的 LDA_MF 模型,以及将应用的标签信息和用户行为数据同时加以考虑的 LDA_CF 算法.为了结合不同算法的优点,在保证推荐准确率的条件下提升推荐结果的多样性,提出了融合 LDA_MF,LDA_CF 以及经典的基于物品的协同过滤模型的混合推荐算法.使用真实的大数据评测所提推荐算法,结果显示,所提推荐方法能够得到推荐多样性更好且准确率更高的结果.

**关键词**: 主题模型;矩阵分解;推荐系统;推荐多样性;协同过滤

**中图法分类号**: TP311

中文引用格式: 黄璐,林川杰,何军,刘红岩,杜小勇.融合主题模型和协同过滤的多样化移动应用推荐.软件学报,2017,28(3):708−720. http://www.jos.org.cn/1000-9825/5163.htm

英文引用格式: Huang L, Lin CJ, He J, Liu HY, Du XY. Diversified mobile app recommendation combining topic model and collaborative filtering. Ruan Jian Xue Bao/Journal of Software, 2017,28(3):708−720 (in Chinese). http://www.jos.org.cn/1000-9825/5163.htm

## Diversified Mobile App Recommendation Combining Topic Model and Collaborative Filtering

HUANG Lu[1], LIN Chuan-Jie[1], HE Jun[1], LIU Hong-Yan[2], DU Xiao-Yong[1]

[1](Key Laboratory of Data and Knowledge Engineering (School of Information, Renmin University of China), Beijing 100872, China)

[2](School of Economics and Management, Tsinghua University, Beijing 100084, China)

**Abstract**: With rapid growth of mobile applications, users of mobile app platforms are facing problem of information overload. Large number of apps make it difficult for users to find appropriate ones, while many long tail apps are submerged in the resource pool and are unknown to most users. Meanwhile, existing recommendation methods usually pay more attention to accuracy than diversity, making popular apps as most recommended items. As a result, the overall exposure rate of mobile apps is low as behavior data accumulated by the system is gradually biased towards popular apps, which leads to a poor recommendation performance in the long run. To solve this problem, this article first proposes two recommendation methods, named LDA_MF and LDA_CF, to improve existing methods. LDA_MF combines user topic model and app topic model with matrix factorization model MF, and LDA_CF takes both tag information of apps and user behavior data into consideration. In order to take advantages of different algorithms and increase the diversity of recommendation

* 基金项目: 国家自然科学基金(71272029, 71490724, 61472426); 国家高技术研究发展计划(863)(2014AA015204); 北京市自然科学基金(4152026)

Foundation item: National Natural Science Foundation of China (71272029, 71490724, 61472426); National High-Tech Study and Develement Plans (863) (2014AA015204); Beijing Municipal Natural Science Foundation under (4152026)

收稿时间: 2016-07-31; 修改时间: 2016-09-14; 采用时间: 2016-11-01; jos 在线出版时间: 2016-11-29

CNKI 网络优先出版: 2016-11-29 13:35:12, http://www.cnki.net/kcms/detail/11.2560.TP.20161129.1335.015.html

results without sacrificing accuracy, a hybrid recommendation algorithm is also provided to combine LDA_MF, LDA_CF and item-based collaborative filtering models. A large real data set is used to evaluate the proposed methods, and the results show that the presented approach achieves better diversity and good recommendation accuracy.

**Key words**: topic model; matrix factorization; recommended system; recommendation diversity; collaborative filtering

目前,移动应用(mobile applicaton,简称 app)的爆炸式增长,给人们带来了信息过载的问题.当前,大部分移动资源获取平台(例如 360 手机助手、豌豆荚等)都采用推荐系统为用户提供个性化推荐服务.针对应用的推荐,一方面,不同于学术领域常常研究的评分预测问题,用户没有评分信息,只有浏览、下载、安装等隐式行为数据和应用描述等内容信息;另一方面,移动应用存在长尾效应,大部分用户安装、下载了小部分应用,如何让更多的应用被展示,发现用户感兴趣的长尾应用,增加推荐系统的多样性,也成为推荐系统关心的问题.现阶段,推荐算法的研究大都以推荐准确率为标准,很少考虑推荐结果的多样性.缺乏多样性的推荐结果,一方面,展示给用户的信息单一,应用的整体曝光率低;另一方面,会使得系统的数据积累越来越偏向于热门应用,导致系统推荐效果越来越差.所以,本文主要研究应用推荐中如何提升推荐结果多样性的问题,在提升多样性的同时不降低推荐的准确度.

提高应用推荐结果的多样性是具有挑战性的研究.

- 首先,用户对于应用的下载行为很稀疏,行为数据大都集中在热门应用上.本文采集的 100 万用户对于 100 万应用的下载数据集中,在一天行为中,只有 17 万用户有下载行为,而用户在这天下载的不同应用数目约为 4.5 万个,占所有应用的 4.5%左右;
- 其次,在内容信息方面,应用有标签信息,但是并不全面,而用户没有任何内容信息;
- 再次,增加推荐结果的多样性势必会影响准确性,如何在保证准确性的前提下增加推荐结果的多样性,也是一个难题.

为了解决以上问题,本文主要研究同时利用用户行为数据和应用基本信息为用户进行个性化的多样化的应用推荐方法.提出了融合主题模型和协同过滤算法的 LDA_MF 算法,结合内容信息和行为信息的 LDA_CF 算法以及融合多种推荐算法的混合算法 Hybrid_Rec.

本文工作的创新性主要包括如下几点:

- 为了将多种信息融合,改善长尾应用行为信息不足,准确地表示用户兴趣,本文提出将用户的主题模型 LDA 和应用的主题模型 LDA 与矩阵分解(MF)相结合的 LDA_MF 模型.同时,为了解决用户和应用缺乏描述信息的问题,本文将 Linked_LDA[11]模型用于推荐算法,将应用的标签信息和用户行为数据同时加以考虑,据此设计推荐算法 LDA_CF;
- 观察到各种不同算法在多样性和准确度方面各具特点,以此为出发点,提出了融合 LDA_MF,LDA_CF 以及基于物品的协同过滤算法(item_CF)的方法,提出了旨在保证准确度的情况下,提升多样性的混合算法;
- 在真实的大数据集上做实验,与多种算法在准确度和多样性两方面进行比较,得到了较好的结果.

本文探讨融合主题模型和协同过滤模型的多样化应用个性化推荐方法.本文第 1 节介绍相关工作.第 2 节介绍应用候选集的生成,提出 LDA_MF 和 LDA_CF 算法.第 3 节介绍不同算法生成推荐候选集的融合,使用逻辑回归学习融合权重.第 4 节给出实验结果,并对实验结果加以分析.最后,在第 5 节对本文加以总结,并指出进一步的工作方向.

## 1 相关工作

本节将从两个方面——个性化推荐和多样化推荐,进行已有工作的总结和分析.

### 1.1 个性化推荐方法现状

近年来,随着隐语义模型越来越受到关注,矩阵分解被广泛应用于推荐系统中.传统的矩阵分解算法有

SVD、非负矩阵分解(NMF)、概率矩阵分解(PMF)等方法,矩阵分解适合于基于大量的评分数据进行推荐的情况.文献[1–4]都是关于矩阵分解应用与推荐系统的研究.文献[1]使用的是 SVD++方法,该方法是在 SVD 的基础上加入用户的偏置,也就是独立于用户和物品的因素部分,例如用户的打分高低喜好等.PMF(probabilistic matrix factorization)[2]是从概率生成的角度来解释用户和物品(item)的隐含特征,PMF 是在 SVD 的基础上假设用户和物品的隐式特征向量服从高斯先验分布,通过最大化后验概率来求解用户和物品的隐式特征矩阵.BPR (Bayesian personalized ranking)算法[3]针对的是用户的隐式反馈行为数据,该算法将用户对物品行为(正反馈为1,无反馈为 0)处理为一个 pair 对的集合$\langle i,j\rangle$,其中,$i$ 评分为 1 也就是有行为数据的物品,$j$ 评分为 0 也就是没有行为数据的物品.该方法基于 pair-wise 的偏序优化,可以避免 point-wise 模型在对无反馈行为涉及项目进行预测时失效(因为无反馈行为涉及项目在训练时全被标记为 0)的问题.文献[4]提出不直接计算物品和物品之间的相似性,将这种相似性转化为两个因子矩阵相乘,这样避免使用物品共现来学习物品相似矩阵.因为没有被用户同时下载的物品也可能是相似的.文献[1–4]的工作都改进了基本的矩阵分解方法,但是以上方法都只考虑了用户的行为数据,没有考虑内容信息,对于在某段时间内用户行为没有涉及的物品没有纳入算法中,这就造成了冷门物品不会出现在推荐结果中,这可能造成马太效应,热门物品越来越热门,冷门物品被系统淘汰.但是有的冷门物品比较小众,也可能是用户感兴趣的,只是用户没有接触到而已.

文献[5]提出 CTR(collaborative topic regression)模型将矩阵分解和主题模型相结合,先计算物品的主题分布,把主题模型得到的物品主题分布作为物品的初始隐含特征(latent factor)矢量,在主题分布的基础上进行矩阵分解.该方法将物品的内容和行为数据结合到一个模型中,可以解决冷启动问题,新的物品只要有描述信息也能够被推荐.该模型并没有考虑用户的隐含特征矢量的初始化问题,而且对于隐式行为数据的处理,将用户没有行为的物品全部看成负例,会对于用户可能感兴趣但是因为没有被推荐的物品产生偏见.

### 1.2 多样化推荐方法现状

推荐系统的一个目标是向用户推荐满足个性化要求的物品,而如果只以当前的推荐准确度为目标进行推荐,将影响推荐系统的长期性能,导致长尾应用无法得到推荐.因此,多样化推荐引起了学术界和业界的关注.现阶段,多样性的研究主要关注两方面:个体多样性(individual diversity)和总体多样性(aggregate diversity).个体多样性可以根据每个用户的推荐列表来计算,衡量的是一个推荐列表中物品的多样性,重点在于避免向同一个用户推荐过于相似的物品.个体多样性的衡量一般是计算给每个用户的推荐列表中的物品之间的相异度(dissimilarity)来衡量.总体多样性则衡量为整个用户群推荐的所有物品的多样性.达到较高的个体多样性并不代表总体多样性高.例如,给所有用户推荐 5 个畅销物品(物品之间彼此各不相似),这样的推荐结果有很高的个体多样性,但是总体多样性却很低.总体多样性能够为用户提供更加广泛的选择,而并非那些用户通常自己就能发现的畅销物品.而且有助于长尾物品的推荐,让推荐系统能够获得越来越多有意义的数据累积.总体多样性的衡量也有若干方法,包括用已推荐的不同物品占所有可推荐物品的百分比,常被称为覆盖率(coverage);还有使用推荐给所有用户的不同物品数作为总体多样性的评价指标等.

提升多样性的研究大多是利用对推荐候选集重排的方式增加推荐结果的多样性,推荐候选集的重排方法一般分为启发式方法和根据评价指标选择排序,启发式方法又分为按物品流行度的降序排序和随机排名方法.文献[6]提出根据物品的流行度对推荐列表进行重排,主要分 3 个步骤进行:第一,根据预测评分对推荐列表排序;第二是设定评分阈值,对大于评分阈值的物品根据物品流行度重排;第三是再过滤,提高评分阈值,将低于阈值的物品剔除,提升准确率.文献[7]提出 UC-BCF(usage context-based collaborative filtering)模型,通过物品的共现性将用户物品之间的关系转化为物品与物品之间的关系,发现新颖物品.文献[6,7]对于增加推荐结果的整体多样性有一定帮助,但是单纯使用用户的行为数据作为候选集选择,推荐候选集中还是会偏向热门物品.

## 2 应用推荐候选集生成

### 2.1 结合主题模型和矩阵分解方法的算法LDA_MF

为了将多种信息融合,增加用户兴趣表示,本文提出了结合主题模型LDA和矩阵分解MF的LDA_MF算法.LDA[8]是一种文档主题生成模型,能够识别语料中潜藏的主题信息.而矩阵分解模型(MF)[9]属于隐语义模型的一种,它的核心思想是,通过隐含特征(latent factor)联系用户兴趣和物品.矩阵分解能够在一定程度上解决数据稀疏的问题,在众多推荐算法中推荐准确率较高.但是矩阵分解的结果会倾向于用户行为较多的应用,也即推荐结果偏向于热门应用.为了改善矩阵分解算法的推荐结果,增加非热门应用的信息,我们受到文献[1]中 CTR (collaborative topic regression)模型的启发,提出将用户的主题模型和应用的主题模型同时与矩阵分解模型相结合.CTR 模型只考虑了物品的主题模型,而在改进的 LDA_MF 模型中,将用户的主题模型也纳入考虑,使用 LDA 训练用户和应用的主题分布,然后再将该主题分布和矩阵分解相结合.LDA_MF 将用户的兴趣(利用用户有行为 app 的 tag 信息进行刻画)也纳入模型中,在推荐过程中,不仅仅是根据用户的下载行为学习用户和 app 的隐含特征矢量,还将用户和 app 的语义层面也纳入考虑范围,对于行为过少的 app 信息起到补充作用.

图 1 所示为 LDA_MF 模型,其中的变量含义见表 1.

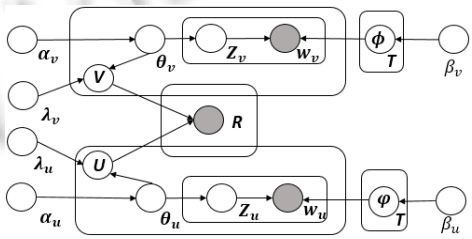


Fig.1 LDA_MF model

图 1 LDA_MF 模型

**Table 1** Symbols in LDA_MF

**表 1** LDA_MF 中的符号含义

| 变量名 | 变量含义 |
|---|---|
| $\alpha_v$ | 应用-主题(app-topic)的先验分布 |
| $\alpha_u$ | 用户-主题(user-topic)的先验分布 |
| $\beta_v$ | 应用标签集合主题-词(topic-word)的先验分布 |
| $\beta_u$ | 用户标签集合主题-词(topic-word)的先验分布 |
| $\theta_v$ | 应用-主题(app-topic)的后验分布 |
| $\theta_u$ | 用户-主题(user-topic)的后验分布 |
| $\varphi$ | 应用标签集合中主题-词的后验分布 |
| $\phi$ | 用户标签集合中主题-词的后验分布 |
| $W_v$ | 应用标签集合中所包含的词 |
| $W_u$ | 用户下载安装应用的标签集合所含词 |
| $Z_v$,$Z_u$ | 应用和用户标签集合中的主题 |
| $U$ | 用户隐含特征矩阵 |
| $V$ | 应用隐含特征矩阵 |
| $T$ | 主题模型中主题数目 |
| $R$ | 隐式反馈矩阵 |
| $\lambda_v$,$\lambda_u$ | 分别为应用和用户的正则项 |

图 1 中:上部分$\alpha_v,\theta_v,Z_v,W_v,\phi,\beta_v$是对应用进行主题建模,表示以 app 标签信息为输入的 LDA 模型,用以模拟标签的生成过程,发现 app 的主题分布;而下部分的$\alpha_u,\theta_u,Z_u,W_u,\varphi,\beta_u$是对用户进行建模,将用户下载、安装的应用标签来表达用户,通过主题模型模拟这些标签的生成过程,从而发现用户的主题分布,即,用户的兴趣分布.其中,$\alpha_v,\alpha_u$分别表示 app 和用户主题分布$\theta_v,\theta_u$的先验分布,$Z_v,Z_u$表示 app 和用户标签集合中的主题,$W_v,W_u$分别表

示 app 标签集合中所包含的词以及用户下载安装应用的标签集合所含词,$\phi,\varphi$分别表示 app 和用户标签集合的主题-词分布,$\beta_v,\beta_u$则表示$\phi,\varphi$的先验分布.使用 LDA 模型能够得到 app 的主题分布$\theta_v$以及用户的兴趣分布$\theta_u$.在矩阵分解部分,基于主题模型的结果初始化用户隐含特征矩阵 $U$ 和应用的隐含特征矩阵 $V$.其中,$\lambda_v,\lambda_u$ 分别为 app 和用户的正则项,$R$ 是隐式反馈矩阵,其中每个元素表示用户是否下载某一应用:下载则为 1,否则为 0.在新模型中,我们不采用随机的方式去初始化用户隐含特征矩阵 $U$ 和物品的隐含特征矩阵 $V$,而是基于主题模型得到的用户的主题分布和 app 的主题分布$\theta_u$和$\theta_v$作为矩阵分解 $U$ 和 $V$ 的输入.在求解过程中使用交替最小二乘法 ALS(alternating least squares)[10]来求解.

LDA_MF 模型使用 LDA 学习应用的主题分布$\theta_v$以及用户的兴趣分布$\theta_u$.LDA 使用的是内容信息获取用户和移动应用的主题分布,即隐含特征.在 LDA 和矩阵分解融合中,我们加入用户的下载行为数据,调整 app 和用户的特征分布.我们认为,app 和用户的特征分布是接近于 LDA 学习出的主题分布的,但也存在偏差值$\varepsilon$.所以每个用户 $i$ 的隐含特征由用户的主题分布特征 $\theta_{u_i}$ 和偏差值 $\varepsilon_{u_i}$ 构成,每个应用 $j$ 的隐含特征由主题分布 $\theta_{v_j}$ 和偏差值 $\varepsilon_{v_j}$ 构成.我们希望通过模型学习得到用户的隐含特征 $U$ 以及 app 的隐含特征 $V$.

$$u_i = \varepsilon_{u_i} + \theta_{u_i} \tag{1}$$

$$v_j = \varepsilon_{v_j} + \theta_{v_j} \tag{2}$$

$$f(U,V) = \sum_{(i,j)\in I} (r_{ij} - u_i^T v_j)^2 + \lambda_u \sum_i \| (u_i - \theta_{u_i}) \|^2 \tag{3}$$

公式(3)是需要最小化的目标函数,对于每一个用户 $i$ 和 app $j$ 的对$(i,j)$,$r_{ij}$ 表示用户 $i$ 对于 app $j$ 的行为,有下载行为则为 1,否则为 0.$\lambda_u$和$\lambda_v$为正则项系数.使用交替最小二乘法求解,最终通过迭代更新每个用户 $i$ 隐含特征 $u_i$ 以及每个应用 $j$ 的隐含特征 $v_j$.更新的公式如公式(4)和公式(5)所示,由于篇幅有限,求解推导的过程没有列出.

$$u_i = (VV^T + \lambda_u E)^{-1}(VR_i + \lambda_u \theta_{u_i}) \tag{4}$$

$$v_j = (UU^T + \lambda_v E)^{-1}(UR_j + \lambda_v \theta_{v_j}) \tag{5}$$

公式(4)中,$E$ 是单位矩阵,$R_i$ 是 $R$ 矩阵中第 $i$ 行向量的转置.公式(5)中,$E$ 是单位矩阵,$R_j$ 是 $R$ 矩阵中第 $j$ 列向量.在得到满足最优化的解,得到用户的隐含特征 $U^*$以及 app 的隐含特征 $V^*$后,我们计算用户 $i$ 对于 app $j$ 的喜好 $r_{ij}^*$,选取 Top-50 应用作为混合推荐的候选,在选取的过程中,将用户已安装的应用过滤掉. $r_{ij}^*$ 的计算公式如下.

$$r_{ij}^* = (u_i^*)^T v_j^* \tag{6}$$

### 2.2 结合内容和行为信息的LDA_CF算法

考虑到大部分用户只对小部分的移动应用有过下载、浏览、安装行为,对于没有行为数据的移动应用,可以使用应用的标签信息来补充表示.使用内容信息来表示应用特征,进而做出推荐.这样能够提升长尾应用加入推荐候选集中的可能性,为增加推荐结果多样性创造可能.

LDA 近年来广泛被用来发现文档中的主题,Linked-LDA[11]是对于 LDA[8]的一个改进[12].LDA 能够发现文档中隐含的主题层,每个文档可以表示为主题的分布,而每个主题可以使用文档中的词来描述.Linked-LDA 将每篇文章的引用和引用的上下文文本作为一个整体,发现文章中隐含的主题分布,相较于 LDA 效果更好.本文 LDA_CF 推荐算法的主要方法为:使用 Linked-LDA 模型,将每个应用的标签和下载该应用的用户集合作为输入,学习出每个应用的主题分布,该主题分布可以作为应用的特征表示;然后,对于每一个用户,我们利用该用户一周内下载的应用的主题分布均值表示该用户,该均值分布可以认为用户在主题上的兴趣分布;最后,我们根据 app 的特征和用户兴趣给出推荐结果.

本文使用 Linked-LDA 的思想来构建 app 的特征表示,将每个 app 作为一个文档,该文档由两部分组成:一部分是 app 的标签所包含的词的信息,另一部分则为下载该 app 的用户集合.我们认为:对于应用来说,拥有相似的标签描述而且被相同用户同时下载较多则更为相似.对于一些长尾应用,下载行为比较少,下载的用户集合也会比较少甚至没有,若只有标签信息,则这部分应用则多由标签信息来构建特征表示.

图 2 为 Linked-LDA 模型示意图,表 2 列出了 Linked-LDA 模型中主要变量的含义.

如图 2 所示:我们将每一个移动应用 $v$ 表示为一个文档,该文档由 app 的标签集合中包含的所有词 $W^{(v)}$以及下载该 app 的用户集合 $U^{(v)}$构成.我们给定 app 中隐含的主题数目为 $K$,最终我们需要得到每一个 app 的主题分布 $\theta$,$\theta$为 $K$ 维向量,每一维表示该 app 在相应主题下的概率.通过主题模型,发掘每一个 app 隐含特征,将 app 由标签表示映射到主题向量表示.

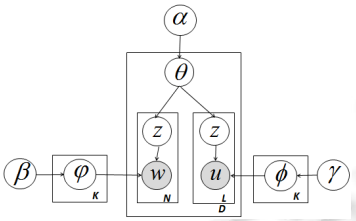


Fig.2 Linked-LDA model

图 2 Linked-LDA 模型

**Table 2** Symbolsin Linked-LDA

**表 2** Linked-LDA 中的符号含义

| 变量名 | 变量含义 |
|---|---|
| $\alpha$ | 移动应用-主题(app-topic)的先验分布 |
| $\beta$ | 主题-词(topic-word)的先验分布 |
| $\gamma$ | 主题-用户(topic-user)的先验分布 |
| $\theta$ | 移动应用-主题(app-topic)的后验分布 |
| $\varphi$ | 在整个 app 集合中,主题-词的后验分布 |
| $\phi$ | 在整个 app 集合中,主题-用户的后验分布 |
| $Z$ | 采样得到的主题 |
| $U$ | 有过下载行为的用户的集合,$u$ 代表其中一个用户 |
| $W$ | 应用标签词的集合,$w$ 代表其中的一个词 |
| $K$ | 主题的个数 |

在 Linked-LDA 模型中,app 的标签描述词汇以及下载该 app 的用户集合生成过程如下.

1) 根据主题-词(topic-word)的先验分布$\beta$,采样得到 app 集合上 $K$ 个主题的 topic-word 后验分布$\varphi$;根据主题-用户(topic-user)的先验分布$\gamma$,采样得到 app 集合上 $K$ 个主题的 topic-user 后验分布$\phi$.其中,

$$\varphi \sim Dirichlet(\beta),\phi \sim Dirichlet(\gamma);$$

2) 对于每一个 app,首先从应用-主题分布(app-topic)的先验分布$\alpha$中采样得到该 app 的 app-topic 后验分布$\theta$,其中,$\theta \sim Dirichlet(\alpha)$;

- 对于 app 的每一个标签词,首先根据 app-topic 后验分布$\theta$采样得到主题 $Z$;然后,根据主题 $Z$ 和 topic-word 后验分布$\varphi_z$,采样得到 app 的标签描述词 $W$;
- 对于每一个下载 app 的用户,首先根据 app-topic 后验分布$\theta$采样得到主题 $Z$,也就是认为用户在下载某一应用的时候,会首先选择一个主题;然后,根据主题 $Z$ 和 topic-user 后验分布$\phi_z$,采样得到下载 app 的用户 $u$.

本文采用吉布斯采样(Gibbs sampling)[13]求解 Linked-LDA,在采样过程中,我们需要不断调整 app 的每个标签词属于每个主题的概率以及下载 app 的每个用户选择每个主题的概率,直到收敛.采样更新规则如下.

$$p(z_i = k \mid \vec{z}_{\neg i}, \vec{w}) \propto \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,\neg i}^{(k)} + \alpha_k)} \cdot \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t \in W} n_{k,\neg i}^{(t)} + \beta_t} \tag{7}$$

$$p(z_j = k \mid \vec{z}_{\neg j}, \vec{u}) \propto \frac{n_{m,\neg j}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,\neg j}^{(k)} + \alpha_k)} \cdot \frac{n_{k,\neg j}^{(u)} + \gamma_u}{\sum_{u \in U} n_{k,\neg j}^{(u)} + \gamma_u} \tag{8}$$

其中,$n_{m,\neg i}^{(k)}$表示在第 $m$ 个 app 文档中(该 app 文档由标签中的词以及下载该 app 的用户构成),除当前需要更新的

第 $i$ 个词汇外,其他被分配到主题 $k$ 的词和用户的数目; $n_{m,\neg j}^{(k)}$ 表示除当前需要更新的第 $j$ 个用户外,其他被分配到主题 $k$ 的词和用户的数目;符号 $\neg i$ 表示除去当前第 $i$ 个词汇;$\neg j$ 表示除去当前第 $j$ 个用户;同理, $n_{k,\neg i}^{(t)}$ 表示除了第 $i$ 个当前词 $t$,被分配到主题 $k$ 的标签词 $t$ 的数目; $n_{k,\neg j}^{(u)}$ 含义类似,表示除了当前抽样的第 $j$ 个用户 $u$,其他被分配到主题 $k$ 的用户 $u$ 的数目.

当吉布斯采样收敛后,我们根据每个 app 的标签以及下载 app 用户分配到主题的情况推导模型的参数估计 $\theta, \theta_k^{(v)}$ 表示每个 app $v$ 在主题 $k$ 下的概率,见公式(9).

$$\theta_k^{(v)} = \frac{n_v^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_v^{(k)} + \alpha_k)} \tag{9}$$

其中, $n_v^{(k)}$ 表示在 app$v$ 的文档中,$v$ 的标签中的词和下载该 app 的用户被分配到主题 $k$ 的数目.于是,每个 app 可以表示为 $\theta^{(v)} = (\theta_1^{(v)},...,\theta_k^{(v)})$.

当得到 app 的主题分布后,利用该用户一周内下载应用的主题分布均值作为用户的特征表示,然后根据 app 和用户的特征计算 app 和用户的相似性,将相似性大的应用推荐给用户.

具体地,对于用户 $u$,设 $V^{(u)} = (V_1^{(u)},...,V_{N_u}^{(u)})$ 为用户 $u$ 所下载的应用集合,其中,$N_u$ 为 $u$ 所下载的应用数,则每个用户 $u$ 的兴趣特征可以使用用户下载过的应用的特征来表示: $f^{(u)} = (f_1^{(u)},...,f_k^{(u)})$ ,其中, $f_i^{(u)}$ 表示用户 $u$ 对第 $i$ 个主题感兴趣程度,见公式(10).

$$f_i^{(u)} = \frac{1}{N_u}\sum_{v_t^{(u)} \in V^{(u)}} \theta_i^{(v_t^{(u)})} \tag{10}$$

其中, $v_t^{(u)}$ 为用户下载集合中的应用.

计算 $f^{(u)}$后,我们将 $f^{(u)} = (f_1^{(u)},...,f_k^{(u)})$ 进行归一化处理,得到 $f'^{(u)} = (f_1'^{(u)},...,f_k'^{(u)})$ ,然后计算用户和物品的相似性.

$$sim(u,v) = \frac{f'^{(u)}(\theta^{(v)})^T}{\| f'^{(u)} \|\| \theta^{(v)} \|} \tag{11}$$

在计算用户和移动应用之间的相似性后,为每个用户选取 Top-50 应用作为混合推荐的候选,在选取的过程中,将用户已经安装的应用过滤掉.

## 3 融合多种算法的混合算法 Hybrid_Rec

在混合推荐算法中,我们使用本文提出的 LDA_MF,LDA_CF 以及传统协同过滤方法中的 Item-based 协同过滤(item_CF)作为推荐候选集生成算法.选择以上 3 种算法生成推荐候选集是基于如下考虑:LDA_MF 算法是基于矩阵分解的改进,能够提供较高的准确率,但推荐结果的多样性欠佳;而 LDA_CF 和 Item_CF 能够分别在语义层和行为层面给出推荐结果,推荐列表多样性丰富,而准确率欠佳.将多种算法融合,能够取长补短,获得准确率高而多样性较好的推荐结果.

本文针对每个用户在每种算法上选取了 Top-50 的结果加入推荐候选集合.使用逻辑回归来学习不同算法融合权重,并最终为用户给出 Top-10 的推荐结果.我们设 Item_CF 获得的候选集为 $R_{item}$,LDA_CF 推荐候选集为 $R_{LDACF}$,LDA_MF 获得的推荐候选集合为 $R_{LDAMF}$,对于某一个应用 $v$,在 Item_CF 推荐得分为 $s_v^{item}$ ,在 LDA_CF 推荐方法的推荐得分为 $s_v^{LDACF}$, 在 LDA_MF 推荐算法的推荐得分为 $s_v^{LDAMF}$, 则该应用 $v$ 混合推荐后的最终总得分为 $s_v^{hybrid}$. 即:在逻辑回归中,我们将一个 app 在 3 种不同推荐候选集生成算法中的推荐得分作为 3 个特征,这 3 个特征即为自变量.将 3 种推荐算法得分加权后便得到最终推荐列表的得分,我们将推荐 app 在推荐列表的得分作为 app 被推荐的概率.

$$s_v^{hybrid} = w_0 + w_1 s_v^{item} + w_2 s_v^{LDACF} + w_3 s_v^{LDAMF} \tag{12}$$

其中,$w=\{w_1,w_2,w_3\}$为 3 种算法融合的权重,使用逻辑回归得到权重值.本文将该问题看做一个分类问题,类别为用户是否下载了某一个应用.我们在用户下载记录中抽取部分数据作为正例,将用户浏览了而没有下载的应用找到,剔除用户安装了的应用,抽取部分应用数据作为负例.为了增加推荐结果的多样性,我们在正例选择时剔除热门应用.$Y$用来表示用户是否会下载该应用,$Y$=1 表示用户下载了某 app,也即用户对某一个 app 感兴趣; $Y$=0 表示用户浏览了某 app 却没有下载,也就是表示用户对该 app 没有兴趣.$h_w$表示预测函数,则对于 $m$ 个样本,最终的损失函数 $J(w)$见公式(14).

$$h_w(v_i)=\frac{1}{1+\mathrm{e}^{-s_{v_i}^{hybrid}}} \tag{13}$$

$$J(w)=-\frac{1}{m}\sum_{i=1}^{m}[y^i\log h_w(v_i)+(1-y^i)\log(1-h_w(v_i))] \tag{14}$$

为了使损失函数最小,采用梯度下降法求解即可得到算法融合的权重.在梯度下降求解后,得到 $w=\{w_1,w_2,w_3\}$以及 $w_0$,可以计算每个用户的推荐候选集合中不同 app 的最终得分,即 $s_v^{hybrid}$.根据该值选取 Top-10 的应用作为用户最终的推荐列表.对于模型的评判,我们使用 AUC 值来度量分类模型的好坏,进而判断融合权重的好坏.

综合推荐候选集的生成和个性化推荐列表生成过程,将混合推荐算法 Hybrid_Rec 的主要步骤表示如下.

**算法**. Hybrid_Rec.

输入:$D_1$:用户下载行为数据集;

$D_2$:逻辑回归训练数据集;

输出:每个用户 top 10 推荐列表.

主要步骤:

1. $result_1=Item\_CF(D_1)$;
2. $result_2=LDA\_MF(D_1)$;
3. $result_3=LDA\_CF(D_1)$;
4. Model $H$ = *regression* ($D_2$)
5. For each user
6. $R_{item}$=Top 50 apps selected based on $result_1$;
7. $R_{LDAMF}$=Top 50 apps selected based on $result_2$;
8. $R_{LDACF}$=Top 50 apps selected based on $result_3$;
9. For each app $v\in R_{item}\cup R_{LDAMF}\cup R_{LDACF}$
10. Calculate $s_v^{hybrid}$ based on model $H$
11. Select Top-10 apps with high $s_v^{hybrid}$

## 4 实验评估与分析

本文的数据来源于某互联网公司手机助手平台中应用推荐的实际数据,该数据包括用户下载、安装应用的日志信息以及应用的标签信息.手机助手的应用有 100 万左右,用户数据有十几亿.我们选取 100 万活跃用户作为实验对象,为用户做个性化推荐,给出 Top-10 的推荐列表,实验均在分布式平台 Hadoop 上运行.

在 100 万活跃用户中,我们将用户在 2015 年 5 月的下载行为作为一个月数据集,2015 年 5 月 25 日~5 月 31 日的下载行为作为一周数据以及 2015 年 6 月 1 日的下载行为作为一天数据.使用 2015 年 6 月 1 日~6 月 3 日这 3 天的下载数据作为评测数据,在评测时,我们使用相同的结果每天评测一次,将 3 天的评测结果的均值作为最终评测结果.见表 2,100 万活跃用户一个月平均下载为 20 个应用,一周平均下载 7 个应用,一天平均下载 3 个应用.

分析中发现,有的用户一天下载应用数超过 1 000 个应用.我们将这种用户判定为刷量用户,将刷量用户剔除,不做考虑.

**Table 3** User download data

**表 3** 用户下载数据概况

| 数据集 | 用户数 | 平均下载量 | 下载的不同应用数 |
|---|---|---|---|
| 一月数据 | 749 191 | 20 | 199 597 |
| 一周数据 | 509 080 | 7 | 115 356 |
| 一天数据 | 173 873 | 3 | 44 752 |

### 4.1 实验评测指标

#### 4.1.1 推荐准确性评测

在推荐准确性评测方面,我们采用两方面的指标.

- 面向用户的命中准确率,用 $P_{hit\_user}$ 表示.将推荐结果与用户一天中实际下载结果进行对比,推荐的应用中只要有用户下载的应用,则认为推荐命中了该用户, $N^u_{hit}$ 为推荐结果命中的用户数量, $P_{hit\_user}$ 则为命中的用户占这一天下载用户的比例.公式(15)为 $P_{hit\_user}$ 的计算方法.

$$p_{hit\_user}=\frac{N^u_{hit}}{N^u_{download}} \tag{15}$$

其中, $N^u_{download}$ 表示一天内所有下载了应用的用户数量.

- 基于应用来考虑的,$P_{hit\_app}$ 为推荐列表中被用户下载的应用数目占所有用户实际下载应用的比例,公式(16)为 $P_{hit\_app}$ 的计算方法.

$$p_{hit\_app}=\frac{N^v_{hit}}{N^v_{download}} \tag{16}$$

其中, $N^v_{hit}$ 表示推荐列表中被用户下载的 app 数目,即推荐准确的 app 数目; $N^v_{download}$ 表示一天中用户下载的所有 app 数目.因为用户下载应用的随机性加上有些用户在评测当天没有下载行为,所以在评测数据选择方面,我们以 3 天的数据(2015 年 6 月 1 日~6 月 3 日)分别评测各种算法的推荐结果,使用 3 日评测结果的均值作为最后的准确率结果.

#### 4.1.2 推荐多样性评测

高准确率一直是推荐系统追求的标准,但是推荐准确率高而多样性低的推荐结果并不利于推荐系统的长期发展.推荐多样性的评测包括多个方面:一是推荐系统发掘长尾应用的能力,即对于整个推荐结果,评测能够展示的不同应用数目以及在推荐结果中推荐准确的不同应用数目;二是推荐列表的多样性,该多样性描述了推荐列表中物品两两之间的不相似程度.

对于推荐系统发掘长尾应用的能力,我们使用信息熵来评测.我们认为:如果所有的应用都能出现在推荐列表中,而且出现的次数差不多,那么推荐系统就有很强的发掘长尾应用能力.信息熵能够很好地表示推荐系统发掘长尾应用的两个方面:一是在推荐次数差不多的情况下,推荐列表中出现的不同应用数越多信息熵越大;二是在推荐应用数一定的情况下,推荐次数的分布越均匀信息熵越大.我们沿用文献[13]中物品流行度的概念,将推荐列表中不同应用出现次数称为该应用在推荐结果中的流行度,则信息熵计算见公式(17).

$$H=-\sum_{i=1}^{n}p(i)\log p(i) \tag{17}$$

其中,$p(i)$由应用$i$的流行度除以所有应用流行度之和计算.应用的流行度分布越平均,则信息熵越大,说明推荐系统覆盖率越高.所以信息熵越大,表示推荐系统发掘长尾能力越好.另外,为了将推荐的准确性也一定程度地在多样性度量中加以体现,除了计算推荐结果中能够展示的所有不同应用的熵,记为 $H(all\ app)$之外,我们还计算了推荐列表中推荐准确的不同应用的熵,记为 $H(hit\ app)$.

针对用户推荐列表的多样性,我们采用公式(18)计算每个用户 $u$ 推荐列表 $R(u)$中两两 app 之间的不相似程度,公式(19)求得所有用户推荐列表不相似程度的均值.

$$Diversity(R(u)) = 1 - \frac{\sum_{i,j \in R(u), i \neq j} sim(i,j)}{\frac{1}{2} | R(u) | (| R(u) | -1)} \tag{18}$$

$$Diversity = \frac{1}{| U |} \sum_{u \in U} Diversity(R(u)) \tag{19}$$

### 4.2 算法LDA_CF和LDA_MF

在LDA_CF中,我们选取$K$=100,即100个主题.对于主题模型中的先验分布$\alpha,\beta,\gamma$的取值,根据已有其他学者的实践经验[15],令$\alpha$=50/$K$,$\beta$=$\gamma$=0.01.我们得到app标签信息中的100个主题,每个主题都由topic-word概率分布来描述.选取概率最大的、最能表示该主题的前5个词来表示该主题.由于空间限制,表4展示了得到的部分主题.其中,#1表示主题1,“音乐(0.314)”表示词“音乐”以0.314的概率描述该主题.

**Table 4** Topics and representative words by Linked-LDA

**表4** Linked-LDA得到的app主题及代表词

| 编号 | 主题代表词 |
|---|---|
| #1 | 音乐(0.314) 铃声(0.124) 下载(0.054) mp3(0.038) 乐器(0.013) |
| #2 | 有声读物(0.235) 现代言情(0.036) 恐怖悬疑(0.031) 教育学习(0.023) 儿童读物(0.021) |
| #3 | 消除(0.202) 休闲益智(0.16) 三消(0.068) 泡泡龙(0.021) 对对碰(0.017) |
| #4 | 办公理财(0.192) 效率办公(0.102) 记事(0.092) 理财(0.053) 笔记(0.033) |
| #5 | 理财(0.12) 股票投资(0.112) 投资(0.053) 记账(0.048) 股票(0.039) |

在得到app的主题分布后,我们可以用该主题分布作为特征来表示app.即每个app由100维向量表示,每一维表示该app在主题上的概率.对于用户,我们使用用户下载的app主题分布的平均值归一化处理后的结果作为用户的特征分布.我们认为:用户一周下载行为表示用户的短期兴趣,用户一个月的下载行为是用户的长期兴趣.得到用户兴趣表示后,采用cosin相似度计算用户兴趣和app主题分布相似性,将相似度高的Top-10的app作为推荐结果.

为了比较利用应用的标签和下载用户信息以及利用应用的描述信息进行主题建模的区别,我们将LDA_CF中的LDA部分替换为对应用的描述文本进行话题建模,相应的算法称为LDA_CF(text).表5为LDA_CF推荐算法以及LDA_CF(text)分别采用一周下载行为和一个月下载行为表示用户兴趣的推荐结果.采用一周数据和一个月数据能够给出推荐的用户数分别为738 764(即98.6%的用户)和495 892(即97.40%的用户).显然,利用一个月的信息可以推荐的用户数多于利用一周的.但是对比表中结果,我们能够看到,LDA_CF算法在一周数据上推荐命中的用户的比例比一个月数据高.因为一周行为数据描述的是用户短期兴趣,在应用推荐中,用户一周兴趣比一个月兴趣稳定.另外,LDA_CF比LDA_CF(text)的性能显然要好,主要原因是应用的描述信息噪音太大,包含很多与app功能不相关的信息.不过,单独使用LDA_CF作为推荐准确率并不高.我们将该算法作为推荐候选集生成算法,使用用户一周行为数据作为LDA_CF的输入,选取Top-50结果加入推荐候选集.

**Table 5** Recommendation results of LDA_CF

**表5** LDA_CF推荐算法效果

| 测试指标 | 一个月数据 | | 一周数据 | |
|---|---|---|---|---|
| | LDA_CF | LDA_CF(text) | LDA_CF | LDA_CF(text) |
| $P_{hit\_user}$ | **0.046 8** | 0.015 1 | **0.056 6** | 0.017 6 |
| $P_{hit\_app}$ | **0.017 4** | 0.006 9 | **0.023 3** | 0.008 7 |

在LDA_MF模型中,根据已有其他学者的实践经验[14],令$\alpha$=50/$K$,$\beta$=0.01,$K$=100.在LDA中使用的主题数目和矩阵分解阶段的隐含特征维度是一样的,即,LDA_MF最后训练出来的用户隐含特征和app的隐含特征均为100维.LDA_MF是对CTR[1]的改进,CTR是将LDA与矩阵分解MF相结合,但是该模型只对item(即app)部分使用LDA,用户部分并没有使用,其参数与LDA_MF设置相同.两个算法推荐准确度的比较见表6.在推荐候选集生成时,我们选择LDA_MF推荐结果的Top-50加入推荐候选集中用于混合推荐.

在准确率方面,LDA_MF 和 CTR 推荐结果 $P_{hit\_user}$,即推荐准确的用户,占当天下载所有用户数的比例差不多.CTR 推荐准确率略高于 LDA_MF,但 LDA_MF 推荐结果推荐准确的不同 app 数目要多于 CTR.LDA_MF 将用户的兴趣纳入模型中,在推荐过程中,不仅仅是根据用户的下载行为学习用户和 app 的隐含特征,还将用户和 app 的语义层面也纳入考虑范围,对于行为过少的 app 信息起到补充作用.在给用户做推荐时,用户和 app 的隐含特征均包含利用 LDA 模型学习的语义信息,所以 LDA_MF 能够将一部分和用户兴趣关联大的非热门物品推荐给用户.与 CTR 相比,LDA_MF 能够在基本保证准确率的基础上,一定程度上提高推荐结果的多样性.

**Table 6** Results comparision of LDA_MF and CTR

**表 6** LDA_MF 和 CTR 推荐算法对比

| 测试指标 | LDA_MF | CTR |
|---|---|---|
| $P_{hit\_user}$ | 0.088 | 0.089 |
| $P_{hit\_app}$ | 0.033 | 0.023 |

### 4.3 混合算法Hybrid_Rec推荐结果

在混合推荐中,为了提升推荐结果的多样性,降低热门应用的推荐.我们在用户下载的应用中剔除了下载量较大的应用,在剩余的用户下载行为中,随机抽样 20 000 个用户下载组合作为正例.而对于负例的选择,我们将用户浏览了而没有下载的应用找到,剔除集合中用户安装了的应用,随机抽样 20 000 个用户-应用组合作为负例.我们将选取的样本集合,随机选取 80%作为训练集,20%作为测试集合.对于所有样本,计算样本在 3 种算法被推荐的概率,分别作为样本的特征.逻辑回归训练完成后,得到的模型结果见公式(20),即 Item_CF 融合权重为 0.162 3,LDA_CF 算法融合权重为 0.114 8,LDA_MF 算法融合权重为 0.722 9.

$$s_v^{hybrid} = 0.0545 + 0.1623 \times s_v^{item} + 0.1148 \times s_v^{LDACF} + 0.7229 \times s_v^{LDAMF} \tag{20}$$

我们在测试样本上计算模型的 AUC 值,结果为 0.71.模型分类效果较好,说明混合算法的权重可以采用.所以计算最终用户推荐候选集合中每个 APP $v$ 的最终的得分 $s_v^{hybrid}$,选择得分最高的 Top-10 作为用户最终的推荐列表.

### 4.4 推荐结果测评

在实验结果评测阶段,我们主要评测本文所提出的 Hybrid_Rec 混合推荐方法和传统方法在推荐准确性以及推荐多样性两方面的表现.我们对比的方法主要有:

- MF:Hu 等人提出的矩阵分解推荐算方法[16],针对于隐式行为的推荐;
- Item_CF:使用对数似然相似度度量的基于物品的协同过滤算法;
- Diverse_Rec:将 Item_CF,LDA_CF 以及 LDA_MF 推荐结果中的 Top-10 的应用作为推荐候选集合,根据 app 下载量逆序排序,将候选集合相对长尾的 Top-10 的应用推荐给用户,属于增加推荐结果多样性的方法.其中,根据下载量逆序排序,重排后推荐给用户的思路类似于文献[6];
- Most Popular:将一个月中下载量最大的 2 000 个应用随机选取 10 个推荐给用户,作为用户推荐列表;
- LDA_MF:本文提出的推荐候选集生成方法之一,将 LDA 与 MF 相结合;
- LDA_CF:本文提出的基于 Linked-LDA 和基于内容的协同过滤算法的推荐方法;
- Hybrid_Rec:本文所提出的混合推荐方法.

我们选择 2015 年 6 月 1 日~6 月 3 日这 3 天数据作为测评,评测用户的推荐列表中是否有用户实际下载过的应用.我们将每天的数据作为评测集合,3 天分别评测,取 3 天评测的平均值作为最后的结果.

在推荐准确率方面,我们主要评测两个指标:$P_{hit\_user}$ 和 $P_{hit\_app}$.在推荐多样性评测方面,我们主要比较局部多样性以及全局多样性.局部多样性使用 $D$(diversity)表示;而在全局多样性方面,主要比较推荐结果能够展示的所有不同应用的熵 $H$(*all app*)以及推荐列表推荐准确的不同应用的熵 $H$(hit app).

表 7 展示了不同算法推荐准确性评测结果.从表中结果可知:MF,LDA_MF 以及 Hybrid_Rec 推荐算法的推荐准确率较高,其中,Hybrid_Rec 为推荐准确率最高的算法;混合推荐算法能够推荐命中更多的用户,而且算法

在所有用户推荐列表中推荐准确的 app 比例也较高,排列第二.

接着,进一步比较了推荐准确率最高的 Top-3 的算法:MF,LDA_MF 以及 Hybrid_Rec.另外,为了衡量 item_CF 在混合模型中的作用,实现了不包含该算法的推荐算法,称为 Hybrid2.表 8 对比了这 4 种算法推荐准确率以及多样性的各个指标,可以发现:Hybrid_Rec 推荐准确的用户比例最高,推荐准确的 app 比例较 LDA_MF 次之,但也相差不大.在多样性方面,不论全局多样性指标 $H$(hit app)和 $H$(all app)抑或是局部多样性指标 $D$(diversity),Hybrid_Rec 效果都比其他两种算法好.比较 Hybrid2 与 Hybrid_rec 可以发现,不包括 item_CF 的混合算法在全局多样性方面降低得多一些.可见,item_CF 对于提升全局多样性有一定的贡献.

**Table 7** Recommendation accuracy

**表 7** 推荐准确率

| 方法 | $P_{hit\_user}$ | $P_{hit\_app}$ |
|---|---|---|
| MF | **0.088 1** | **0.032 2** |
| LDA_MF | **0.087 8** | **0.033 0** |
| Item_CF | 0.025 7 | 0.009 5 |
| LDA_CF | 0.056 6 | 0.023 3 |
| Most Popular | 0.007 2 | 0.002 1 |
| Diversity_Rec | 0.014 3 | 0.005 1 |
| Hybrid_Rec | **0.101 6** | **0.032 7** |

**Table 8** Performance comparision of top-3 algorithm

**表 8** 推荐性能 top-3 算法测评对比

| 方法 | $P_{hit\_user}$ | $P_{hit\_app}$ | $H$(hit app) | $H$(all app) | $D$(diversity) |
|---|---|---|---|---|---|
| MF | 0.088 | 0.032 | 2.172 | 1.799 | 0.821 |
| LDA_MF | 0.088 | 0.033 | 2.201 | 2.011 | 0.824 |
| Hybrid2 | 0.093 | 0.033 | 2.243 | 2.793 | 0.834 |
| Hybrid_Rec | **0.102** | **0.033** | **2.309** | **2.958** | **0.838** |

综合不同推荐结果,包括准确性和多样性两方面.混合推荐算法结合了多种算法的优点,准确性最高,能够给更多的用户提供准确的推荐列表,在全局上也能够准确推荐更多的 app.在多样性方面,混合推荐方法能够给用户提供推荐多样性最为丰富的列表,也就是对于单个用户,混合推荐算法提供的推荐列表物品不相似性最大,用户有更多的选择范围.而在全局多样性方面,混合推荐算法也能够展示较多的应用.

## 5 结 论

本文研究移动应用的个性化推荐问题.与已有研究大多关注短期推荐准确度不同,本文研究如何提升推荐结果的多样性.我们发现,不同的算法在准确度和多样性方面表现不同.因此,我们主要研究如何将多种算法进行融合.为此:首先,为了进一步改进矩阵分解算法,本文提出将用户的主题模型和应用的主题模型与矩阵分解方法相结合的模型 LDA_MF 算法.为了解决用户和应用缺乏描述信息的问题,我们提出结合主题模型和基于物品的协同过滤算法的推荐算法 LDA_CF.最后,为了在保证准确度的情况下提升多样性,本文提出了将 LDA_MF,LDA_CF 以及 Item_CF 通过逻辑回归进行融合的混合算法 Hybrid_Rec.在真实大数据集上评测证实,Hybrid_Rec 能够得到准确率高而且多样性较为丰富的结果.推荐多样性问题值得深入研究,未来可以进一步研究对用户行为序列进行挖掘,将用户的长期和短期兴趣进行结合的多样化推荐方法.本文提出的混合算法融合的都是常见的推荐算法,在这些算法结果的基础上,通过逻辑回归加以综合,逻辑回归模型无需经常更新,可以设定一定的间隔进行线下更新.如何提高常用的推荐算法如 MF 的效率以便能够在线使用,近年来已有相关研究.未来可以进一步研究如何提升基于话题模型的推荐算法的效率以及如何融合高效率的算法于混合模型等问题,以期同时提升准确度、多样性和推荐效率.

**References**:

[1] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2008. 426–434. [doi: 10.1145/1401890.1401944]

[2] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proc. of the 25th Int'l Conf. on Machine Learning. ACM Press, 2008. 880−887. [doi: 10.1145/1390156.1390267]

[3] Rendle S, Freudenthaler C. Improving pairwise learning for item recommendation from implicit feedback. In: Proc. of the 7th ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2014. 273−282. [doi: 10.1145/2556195.2556248]

[4] Kabbur S, Ning X, Karypis G. Fism: Factored item similarity models for top-*n* recommender systems. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2013. 659−666. [doi: 10.1145/2487575.2487589]

[5] Wang C, Blei DM. Collaborative topic modeling for recommending scientific articles. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2011. 448−456. [doi: 10.1145/2020408.2020480]

[6] Adomavicius G, Kwon YO. Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. on Knowledge and Data Engineering, 2012,24(5):896−911. [doi: 10.1109/TKDE.2011.15]

[7] Niemann K, Wolpers M. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2013. 955−963. [doi: 10.1145/2487575.2487656]

[8] Blei DM, Ng AY, Jordan MI. Latent dirichletallocation. Journal of Machine Learning Research, 2003,3:993−1022.

[9] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009,8:30−37. [doi: 10.1109/MC.2009.263]

[10] Takane Y, Young FW, De Leeuw J. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. Psychometrika, 1977,42(1):7−67. [doi: 10.1007/BF02293745]

[11] Erosheva E, Fienberg S, Lafferty J. Mixed-Membership models of scientific publications. Proc. of the National Academy of Sciences, 2004,101:5220−5227. [doi: 10.1073/pnas.0307760101]

[12] Nallapati RM, Ahmed A, Xing EP, Cohen WW. Joint latent topic models for text and citations. In: Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2008. 542−550. [doi: 10.1145/1401890.1401957]

[13] Casella G, George EI. Explaining the Gibbs sampler. The American Statistician, 1992,46(3):167−174.

[14] Xiang L. Recommender Systme Practice. Beijing: Posts & Telecom Press, 2012 (in Chinese).

[15] Griffiths TL, Steyvers M. Finding scientific topics. Proc. of the National Academy of Sciences, 2004,101(Suppl. 1):5228−5235. [doi: 10.1073/pnas.0307752101]

[16] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In: Proc. of the 8th IEEE Int'l Conf. on Data Mining (ICDM 2008). 2008. 263−272. [doi: 10.1109/ICDM.2008.22]

**附中文参考文献**:

[14] 项亮.推荐系统实践.北京:人民邮电出版社,2012.

**黄璐**(1990－),女,广东饶平人,硕士生,主要研究领域为数据挖掘,主体模型,推荐系统.

**林川杰**(1992－),男,硕士生,主要研究领域为深度学习,数据挖掘.

**何军**(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,社交网络分析,推荐系统.

**刘红岩**(1968－),女,博士,教授,博士生导师,主要研究领域为大数据分析,商务智能,数据挖掘.

**杜小勇**(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库系统,大数据管理,智能信息检索,知识工程.