

基于局部模型加权融合的 Top-N 电影推荐算法

汤 颖¹ 孙康高¹ 秦绪佳¹ 周建美²

(浙江工业大学计算机科学与技术学院 杭州 310023)¹

(南通大学计算机科学与技术学院 江苏 南通 226019)²

摘 要 为了解决传统推荐算法使用单一模型无法准确捕获用户偏好的问题,将稀疏线性模型作为基本推荐模型,提出了基于用户聚类的局部模型加权融合算法来实现电影的 Top-N 个性化推荐。同时,为了实现用户聚类,文中利用 LDA 主题模型和电影的文本内容信息,提出了语义层次用户特征向量的计算方法,并基于此来实现用户聚类。在豆瓣网电影数据集上的实验验证结果表明,所提局部加权融合推荐算法提升了原始基模型的推荐效果,同时又优于一些传统的经典推荐算法,从而证明了该推荐算法的有效性。

关键词 推荐系统,模型融合,稀疏线性模型,主题模型

中图法分类号 TP301 **文献标识码** A

Local Model Weighted Ensemble for Top-N Movie Recommendation

TANG Ying¹ SUN Kang-gao¹ QIN Xu-jia¹ ZHOU Jian-mei²

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)¹

(School of Computer Science and Technology, Nantong University, Nantong, Jiangsu 226019, China)²

Abstract In order to solve the problem that the traditional recommendation algorithms can not accurately capture the user preference with a single model, this paper proposed a Top-N personalized recommendation algorithm based on local model weighted ensemble. This recommendation algorithm adopts user clustering to compute the local models and takes the sparse linear model as the basic recommendation model. Meanwhile, the semantic-level feature vector representation of each user was proposed based on LDA topic model and movie text content information, so as to implement user clustering. The experiments of the film data crawled from Douban show that our local model weighted ensemble recommendation algorithm enhances the recommendation quality of the original base model and outperforms some traditional classical recommendation algorithms, which demonstrates the effectiveness of the proposed algorithm.

Keywords Recommendation system, Model ensemble, Sparse linear model, Topic model

1 引言

随着信息科技和社交网络的快速发展,互联网产生的数据近年来呈指数暴涨,大数据时代来临。随着数据量的增多,人们越来越难以从海量数据中发现自己真正想要的信息。此时,推荐系统可发挥最大应用价值。根据用户资料 and 用户历史行为数据,推荐系统能够准确预测用户的喜好,个性化地为用户推荐他们可能感兴趣的东 西,大大降低了用户发现目标信息的成本。现代化的推荐系统主要有两个任务:1)评分预测;2)在现实商业场景中应用最多的 Top-N 推荐。本文将研究 Top-N 推荐算法。

不同用户群体内部往往会形成各自独特的行为模式。例如,爱好军事战争的用户群体在军事题材电影中的打分行为往往比较多,但该群体在文艺题材电影中的打分行为是比较稀疏的。反之,爱好文艺的用户群体在文艺题材电影中的打分

行为较多,但在军事题材电影中的打分行为则比较稀疏。而对于一些热门电影,无论其是什么题材,每个用户群体中的用户对它们的打分行为通常都很活跃。而基于邻域的协同过滤是通过分析用户对物品产生的行为来计算物品之间的相似度,其认为物品 A 和物品 B 具有很大的相似度是因为喜欢物品 A 的用户大都也喜欢物品 B,这就意味着热门电影和战争电影之间的相似度在以上两个用户群体中是不一样的,在军事爱好者群体中相似,在文艺爱好者群体中不相似。如图 1 所示,如果用用户对电影的隐式反馈行为表示一部电影的特征向量(矩阵中的列向量),显然,电影 i 和电影 j 之间的相似度在用户聚类 A 和聚类 B 中是不同的,在 A 中电影 i 和电影 j 的相似度非常高,在 B 中两部电影的相似度为 0。传统的推荐算法通常对所有用户构建一个单一的全局模型,认为两个相同的物品在任何人群中的相似程度都是一样的,显然这种单一模型无法发现存在于不同用户子群中物品相似性的差

本文受国家自然科学基金(71571160,61672462)资助。

汤 颖(1977—),女,博士,副教授,硕士生导师,CCF 会员,主要研究方向为信息可视化、数据挖掘和计算机图形图像,E-mail:ytang@zjut.edu.cn;**孙康高**(1992—),男,硕士生,主要研究方向为数据挖掘、推荐系统;**秦绪佳**(1968—),男,博士,教授,博士生导师,CCF 会员,主要研究方向为数据可视化、计算机图形学;**周建美**(1977—),女,硕士,副教授,主要研究方向为信息安全、网络数据库。

异,从而无法准确捕获用户的喜好,推荐效果一般。用户和电影的隐式反馈矩阵如图 1 所示。

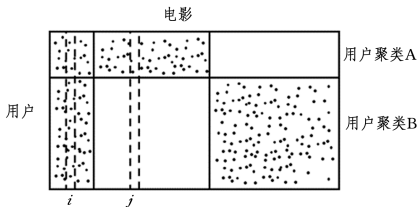


图 1 用户和电影的隐式反馈矩阵

本文提出的推荐算法致力于通过构建局部模型来解决以上问题。为了解决单个模型无法准确捕获用户偏好的问题,本文提出基于用户聚类的局部模型加权融合推荐算法,通过用户子群划分训练得到局部模型,最后与全局模型加权融合来提升推荐的质量。因为局部模型在训练过程中只用到了局部训练数据,会丢失掉同样重要的全局信息,所以本文提出的融合算法是局部模型和全局模型之间的线性加权融合。局部模型对全局模型的预测起到了辅助和修正作用,通过调节权重参数来控制不同模型对于融合模型的重要度。另外,为了实现用户子群的划分,本文利用电影的文本内容信息,提出采用 LDA 主题模型计算用户特征向量,并基于该特征向量使用谱聚类算法实现用户聚类。本文提出的推荐算法是基于内容的推荐、基于邻域的协同过滤和基于模型的协同过滤这三者的有效结合,因此同时具备了 3 种算法推荐结果可解释、推荐速度快、推荐质量高的优点。利用爬取自豆瓣网的电影数据集,通过多个实验证明了所提推荐算法的有效性。

本文第 2 节介绍相关工作;第 3 节介绍推荐系统结构和参数记号表示;第 4 节详细介绍推荐算法的原理;第 5 节给出实验及模型评估;最后总结全文。

2 相关工作

2.1 Top-N 推荐算法

Top-N 推荐算法通过为用户推荐一个经过排名且大小为 N 的物品列表的方式让用户选择自己感兴趣的東西。Top-N 推荐模型主要分为两种类型,其中一种类型是基于邻域的协同过滤方法,该类型又可细分为基于用户的协同过滤 (User-KNN) 和基于物品的协同过滤 (ItemKNN)^[1]。基于用户的协同过滤是推荐系统中最古老的算法,它通过寻找目标用户的一组相似用户,为目标用户推荐相似用户喜欢的物品,推荐结果着重反映与目标用户相似的小群体的热点;而基于物品的协同过滤是目前业界应用得最多的推荐算法,它主要通过分析用户的行为记录来计算物品之间的相似度,接着给目标用户推荐与其喜好物品相似的物品。该算法的推荐结果着重于维系用户的历史兴趣,其优点是模型简单且推荐效率高,缺点是没有学习到物品更深层次的特征表示,因此无论是推荐质量还是准确性都比较低。为了解决这一问题,Karypis 等提出了稀疏线性模型 SLIM^[2],通过机器学习的方式从用户物品反馈矩阵中学习得到物品的相似度矩阵,从而极大地提升了推荐的效果。

另一种 Top-N 推荐算法的类型是基于模型的协同过滤,尤其是以矩阵分解为代表的隐因子模型,该类模型无论是在推荐算法的比赛中还是在实际的应用中都发挥了很重要的作

用。Funk 提出 FunkSVD^[3],通过对原始的用户物品反馈矩阵进行带 L2 正则项的矩阵分解,得到在同一维度空间下用户和物品的低维隐式因子矩阵,通过计算特定用户和物品隐因子向量的点积来得到当前用户对某物品的预测偏好值,最后按预测值从高到低排序,向用户推荐一个 Top-N 的物品列表。Koren 等在 Netflix 比赛中也提出了多种基于 FunkSVD 改进的推荐算法,包括 BiasSVD^[4], SVD++^[5], time-SVD++^[6] 等,这些推荐算法也适用于评分预测。另外,通过结合用户的隐式反馈信息,Hu 等提出 WRMF^[7],把推荐问题转化成一个加权正则化的最小二乘问题。Kabbur 等提出 FISM^[8],直接从评分矩阵中学习得到物品的特征表示,而不考虑用户维度,通过两个低维的隐式因子矩阵,学习物品之间的相关性,并以此来预测、推荐物品。另外,由于训练推荐模型的评分矩阵通常是很稀疏的,存在大量缺失值,因此基于矩阵填充的推荐算法也得到了很大的发展。Cremonesi 等提出用 0 填充评分矩阵中的缺失值,再用奇异值分解 PureSVD^[9] 的方式直接分解矩阵进行 Top-N 推荐,取得了不错的效果。Kang 等提出了基于低秩假设同时又保持数据原始信息来实现矩阵填充的推荐算法^[10],其推荐效果超越了 PureSVD。此外,Top-N 推荐问题本质上是一个排名问题,而之前所有的推荐算法都是通过最小化损失函数来优化模型的。Rendle 等利用贝叶斯概率模型,通过最大化最佳排名的后验概率,提出了一个基于物品对优化的通用标准——贝叶斯个性化排名标准 BPR^[11],该优化标准可以与之前多种推荐算法结合使用 (BPRkNN,BPRMF),且均能得到不错的效果。

2.2 局部模型推荐算法

通过训练多个局部推荐模型,再融合局部模型来提升总体推荐效果的推荐算法近年来受到人们越来越多的关注。这个思路最早由 Connor 等^[12]在其工作中提出,他们使用聚类算法对评分矩阵中的物品进行划分,每个划分都会做出独立的推荐,最后将推荐结果融合,该算法在一定程度上提升了推荐精度。Xu 等^[13]提出把用户和物品聚类成多个用户-物品簇,每个用户可能属于多个簇,每个簇通过协同过滤算法都会产生局部推荐结果,最终把当前用户所属各簇中权重最高的簇对应的推荐模型产生的预测分作为当前用户关于该物品的预测分。Lee 等^[14]认为评分矩阵在特定的由用户物品对定义的度量空间邻域内是低秩的,他们提出采用局部最小二乘矩阵近似法 LLORMA 来最小化排名误差,获得局部用户和物品的隐式因子矩阵,最后融合局部模型进行推荐^[15]。Christakopoulou 等^[16]提出了 GLSLIM,先训练一个全局模型,再对用户进行子群划分,分别为每个子群训练一个局部模型,每个用户的推荐结果由该用户所在子群的局部模型和全局模型加权融合得到;另外,在模型训练的每次迭代过程中,用户会被重新分派到不同子群,模型训练和用户子群划分同步进行。

由于稀疏线性模型具有推荐速度快、推荐精度高、适用于 Top-N 推荐等优点,因此本文提出采用稀疏线性模型作为局部基本推荐模型,通过局部模型训练得到局部电影相似度矩阵,最后加权融合全局和局部模型来进行 Top-N 推荐。另外,通过 LDA 主题模型把电影的文本信息结合到推荐算法中,使得推荐结果可解释。因此,本文提出的推荐算法在某种程度上是多种类型推荐算法的融合,汲取了每种推荐算法的优点,又弥补了相互之间的不足,提升了推荐效果。

3 推荐模型结构和参数记号表示

本文的推荐场景为豆瓣电影的个性化 Top-N 推荐,推荐模型的总体流程和结构如图 2 所示。除评分数据外,本文还在豆瓣网抓取了电影的标签信息。每部电影都附属有 8 个标签,由观看过这部电影的用户为这部电影打的所有标签中出现频次最多的 8 个标签组成,该数据能在一定程度上表示这部电影的语义。通过这些标签,利用 LDA 主题模型训练得到用户特征向量并完成用户聚类;再利用评分数据,给每个用户聚类训练局部模型并为所有用户训练一个全局模型;最后通过全局和局部模型线性加权融合生成每个用户的 Top-N 电影推荐列表。

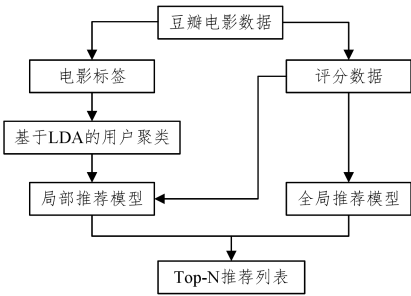


图 2 推荐模型的结构

本文使用到的参数记号及其含义如表 1 所列。

表 1 参数记号表示

定义	描述
w_n	一篇文档的第 n 个单词
θ	一篇文档的主题分布
z_n	一篇文档中第 n 词的主题
α	每篇文档下主题的多项分布的 Dirichlet 先验参数
β	每个主题下词的多项分布的 Dirichlet 先验参数
u_i	用户 $i, u_i \in U, U = n$
v_j	物品 $j, v_j \in V, V = m$
R_u	用户 u 的评分集合
g	线性加权因子,全局模型权重
α	模型惩罚项的系数
ρ	L1 正则化参数的比例
$\tilde{r}_{ui}$	物品 i 对用户 u 的预测推荐分
A	用户-物品评分矩阵
W	物品-物品相似度矩阵
W^{P_u}	用户 u 所在的用户子群 P_u 的物品-物品相似度矩阵
a_i^T	矩阵 A 的第 i 行,表示用户 i 的购买/评分历史信息
a_j	矩阵 A 的第 j 列,表示物品 j 被购买/被评分历史信息

4 基于局部模型加权融合的推荐算法

4.1 基于 LDA 主题模型的用户聚类

为了构建局部推荐模型,首先需要对用户进行聚类,在语义层次对用户进行划分是一种很好的尝试。本文采用 LDA 主题模型来训练得到用户的特征向量,然后采用谱聚类算法对用户进行聚类。

如图 3 所示,LDA^[17] 主题模型是一个文档-主题-单词的 3 层贝叶斯网络,它的联合概率如式(1)所示:

$$p(\theta,z,w|\alpha,\beta)=p(\theta|\alpha)\prod_{n=1}^Np(z_n|\theta)p(w_n|z_n,\beta) \tag{1}$$

该模型主要用来进行文档的信息提取和语义分析。给定一个语料库,该模型可以分析该语料库中每篇文档的主题分布,以及每个主题的词分布。通过设定固定的主题数,LDA 模型可以将文档集中每篇文档以主题向量的形式表示出来,

向量中的每个值代表了当前文档关于对应主题的一个隶属度。通过利用这些文档-主题向量,便可以对文档进行主题聚类或文本分类。

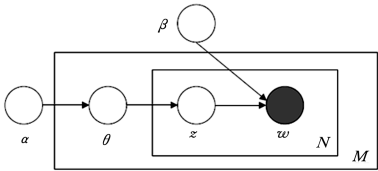


图 3 LDA 主题模型

在豆瓣电影数据集中,每部电影都有多个用户给它赋予的标签。把一个电影标签映射成一个单词 w ,把一个用户观看过的所有电影的标签组成的集合映射成一篇文档 d ,把用户所偏好的一类特定的电影类型映射成一个主题 z 。若数据集中共有 n 个用户,则可生成一个含有 n 篇文档的语料库以及一个字典,语料库中的每篇文档用字典长度的向量表示,向量中的每个值是对应字典中的标签在该用户文档及语料库中的 TF-IDF 值。为了能区分出更加独特的用户群体,我们希望不同主题之间的差异性越大越好。因此,为了确定最佳主题个数,通过设置多个主题数训练多个模型,计算每个模型训练得到的主题向量之间的平均相似度,取主题向量平均相似度最小的模型对应的主题数作为模型的最佳主题个数。本文将在 5.3 节通过实验来确定最佳主题数。

为了确定每个主题的语义,并以此来评价主题模型训练结果的好坏,取每个主题向量中概率值排名前 10 的标签来观察,实验中的某次训练结果如表 2 所列。

表 2 主题分布

主题	标签
0	南海十三郎 谢君豪 高志森 杜国威 Change 2015 日剧 2013 2012 2011
1	人狗奇缘 给我感动的电影 感人至深 樱井孝宏 纯情罗曼史 艾玛·罗伯茨 EmmaRoberts 郭子健 泰迪罗宾 陈观泰
2	日本动画 日剧 2011 二次元 2010 电视剧 2012 日本 动画 2013
3	周星驰 香港电影 童年回忆 国产动画 动画片 大陆 刘德华 杜琪峰 香港 张国荣
4	黒の契約者 J.C.STAFF 钉宫理惠 龙虎斗 日本动画 二次元 2015 2013 2011 2014
5	醉打金枝 nancy punk sid 2015 2014 像素大战 中国大陆 柯南 名侦探柯南
6	港剧 TVB 韩剧 欧阳震华 电视剧 看过的电视剧 关咏荷 黄宗泽 英剧 2013
7	初雪 Cashmere Mafia 成功女人 女人帮 Heima 冰岛 SigurRós Sigur_Rós 冰岛电影
8	2015 2014 2013 中国大陆 2012 Marvel 2011 超级英雄 2016 冒险
9	情色 黑帮 法国电影 AlPacino 意大利 法国 黑色 英国电影 暴力 军事

由表 2 可知,实验中训练得到的 10 个主题向量大致分出了几种用户喜爱的特定电影类型。例如,主题 2 表示存在喜欢观看日本动漫的用户,主题 3 表示存在喜欢观看香港电影和国产动画的用户,主题 6 表示存在喜欢观看港剧的用户,主题 9 表示存在喜欢观看欧美电影的用户。因此,主题模型确实能够发现不同用户群体独特的观影喜好,人群独特的观影喜好形成了独特的行为模式。

通过 LDA 模型训练,得到每一篇文档的主题分布 $\vec{\theta}$,用它来表示每一个用户的特征向量。我们使用该用户特征向量,通过谱聚类算法对用户进行聚类,共聚成 k 个类。谱聚类是一种基于图论的聚类方法,其基本思想是对利用样本数据

的相似矩阵进行特征分解后得到的特征向量进行聚类,谱聚类能够识别任意形状的样本空间且收敛于全局最优解。对于每个用户聚类,把原始训练矩阵 A 中不属于该聚类的用户行向量都置为 0,共得到 k 个聚类的局部训练矩 A^{P_u} 。 P_u 表示聚类编号,且 $P_u \in \{1, \dots, k\}$ 。本文将在 5.3 节通过主题强度来确定最佳聚类数 k 。

4.2 基于用户聚类的局部模型加权融合

为了实现对电影的个性化 Top-N 推荐,采用稀疏线性模型 SLIM^[2]作为基本推荐模型。稀疏线性模型兼具 ItemkNN 速度快以及矩阵分解模型知识学习推荐精度高的优点,能够利用大小为 $n \times m$ (n 为用户数, m 为电影数)的原始评分矩阵 A ,学习获得一个 $m \times m$ 的电影相似度稀疏矩阵 W ,最后使用 W 矩阵和 A 矩阵完成基于物品邻域的 Top-N 推荐。SLIM 模型的损失函数如式(2)所示:

$$\begin{aligned} &\underset{W}{\text{minimize}} \quad \frac{1}{2} \|A - AW\|_F^2 + \frac{\alpha(1-\rho)}{2} \|W\|_F^2 + \alpha\rho \|W\|_1 \\ &\text{s. t.} \quad W \geq 0 \\ &\quad \text{diag}(W) = 0 \end{aligned}$$

(2)

该模型为弹性网络(ElasticNet),L1 范数控制 W 矩阵的稀疏程度,L2 范数控制模型的复杂度,防止模型过拟合。可以通过随机梯度下降法,并行训练 W 矩阵的每一列 w_j 来得到最终的 W 矩阵,如式(3)所示:

$$\begin{aligned} &\underset{w_j}{\text{minimize}} \quad \frac{1}{2} \|a_j - Aw_j\|_2^2 + \frac{\alpha(1-\rho)}{2} \|w_j\|_2^2 + \alpha\rho \|w_j\|_1 \\ &\text{s. t.} \quad w_j \geq 0 \\ &\quad w_{j,j} = 0 \end{aligned}$$

(3)

用户 i 关于电影 j 的最终预测推荐度计算公式如式(4)所示:

$$\tilde{a}_{ij} = a_i^T w_j$$

(4)

为了构建全局推荐模型和局部推荐模型,使用 SLIM 模型作为基本推荐模型,利用全局训练矩阵 A 训练得到全局电影相似度矩阵 W ,利用局部训练矩阵 A^{P_u} 训练得到每个聚类对应的局部电影相似度矩阵 $W^1 \dots W^{P_u} \dots W^k$ 。在实验中,我们把评分数据都转换成了 0-1 表示的隐式反馈信息,故最终电影 j 关于用户 u 的线性加权融合推荐度计算公式如式(5)所示:

$$\tilde{r}_{uj} = \sum_{i \in R_u} (g w_{ij} + (1-g) w_{ij}^{P_u})$$

(5)

其中, R_u 为与用户 u 发生过交互的所有电影的集合,参数 g 为全局模型的权重参数。可以通过调节参数 g 来控制全局模型和局部模型在融合模型中的权重比例,通过确定最优权重参数 g 获得融合模型的最佳推荐效果。我们将在 5.4 节中通过实验来确定最佳的全局模型权重参数。在确定了模型中的所有参数之后,计算所有电影关于当前用户 u 的加权融合推荐度,并按从大到小的顺序排序,同时删除已经与当前用户发生过交互的电影,取排名靠前的 N 部电影推荐给当前用户。

5 实验及结果分析

文本采用 Python 作为开发语言,运行环境为:Inter(R) Core(TM) i5-2450M CPU @ 2.50GHz 处理器,8.00GB 运行内存,Windows 10 64 位操作系统,PyCharm 开发平台。

5.1 实验数据集介绍

本文采用在豆瓣网上爬取的数据集来验证提出的推荐模型。初始数据集包括 3327 个用户,28603 部电影,389173 个评分。为了更好地评估推荐模型的性能,对该数据集进行清洗,保证一部电影至少被 20 个用户观看过,同时保证一个用户至少观看了 15 电影。最终,实验用的数据集包括了 3155 个用户,3524 部电影,302662 个评分,4221 个电影标签。数据集样本分布细节如图 4 所示。

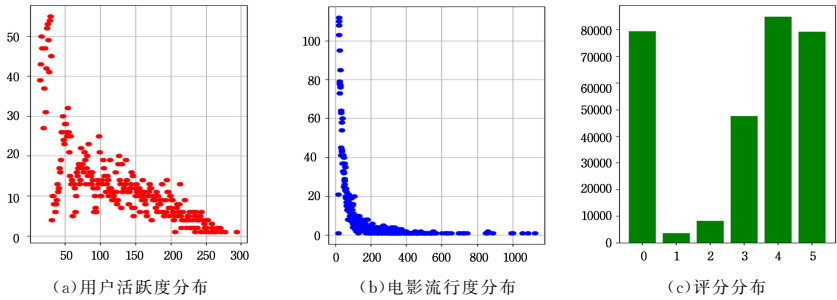


图 4 实验数据集可视化展示

图 4(a)是用户活跃度分布图,横轴表示用户的评分数量,纵轴表示对应评分数量的用户数;图 4(b)是电影流行分布图,横轴表示电影的流行度(被评分的次数),纵轴表示对应流行度的电影数,显然,电影的流行度分布服从长尾分布;图 4(c)是评分分布图,图中横轴表示评分,纵轴表示对应评分的电影数量,豆瓣的用户评分模式是 1~5 星,对应 1~5 分,有很多用户观看过某部电影但没有打分,此类情况记为 0。把这些显式的评分信息转化为隐式的 0-1 反馈信息,只要用观用户看过该电影,对应项就置为 1,否则置为 0,从而生成原始训练矩阵 A 。

5.2 评价方法及评价指标

本文使用留一法交叉验证来证明模型的有效性。从每个用户的电影评分集合中随机抽取一部电影放入测试集中,其他电影用来作为模型的训练集;然后用训练好的模型为每个

用户推荐一个 Top-N 的电影列表,观察测试集里该用户对应的那一部电影是否出现在推荐列表中以及其出现在列表中的具体位置 p_i ,本文实验采用 Top-10 推荐;最后用命中率(HR)和平均排名命中率(ARHR)两个指标来衡量模型的推荐质量。其定义分别如式(6)、式(7)所示:

$$HR = \frac{\# hits}{\# users}$$

(6)

$$ARHR = \frac{1}{\# users} \sum_{i=1}^{\# hits} \frac{1}{p_i}$$

(7)

5.3 主题个数和用户聚类数的确定

主题模型训练首先需要确定具体的主题个数。因此,最初先随机设定一组主题数{5,6,7,8,9,10,11,15}训练得到多个主题模型,然后分别计算每个主题模型产生的主题向量之间的平均相似度(这里采用余弦相似度),取平均相似度最小

的模型作为最终模型。经过实验验证,当前数据集的主题数设置为 10 时,平均相似度最低,为 0.645,因此主题数取 10。为了提升模型的训练速度,采用 Hoffman 等提出的 Online LDA 主题模型^[18],该模型把传统 LDA 小时级的训练时间降低到了秒级。最后训练得到 3155 个 10 维的用户特征向量和 10 个 4221 维的主题向量。每个主题的语义见表 2。

谱聚类算法实现用户聚类时首先需要确定聚类个数。因为训练得到的每个用户向量的每一维度表示该用户属于对应主题的隶属度,故为了确定每个主题在当前用户群体中的重要性,把所有用户的特征向量按维度进行累加再取平均,得到一个 10 维的主题强度向量,可视化如图 5 所示(纵轴表示主题强度,横轴为主题)。可以看到,主题 2,9,3,8,6(对应于表 2)在当前数据集中的强度最大,说明喜欢观看这些类型电影的人最多。因此,本文使用谱聚类算法将用户聚成 5 类。

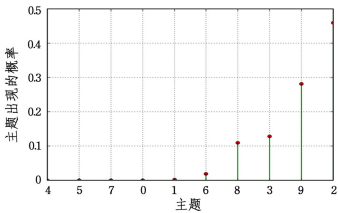


图 5 主题强度分布图

5.4 基于用户聚类的局部模型加权融合的实验

对于每个用户聚类,我们都训练一个局部稀疏线性推荐模型,同时训练一个全局稀疏线性推荐模型,通过交叉验证确定最优参数,使每个模型的实验指标达到最优。训练结果如表 3 所列。

表 3 局部模型和全局模型的训练结果

	用户数	α	ρ	HR	ARHR
局部模型 0	372	0.004410	0.085770	0.204301	0.087799
局部模型 1	656	0.002100	0.400000	0.195122	0.087737
局部模型 2	979	0.004410	0.180117	0.184883	0.086792
局部模型 3	436	0.019449	0.040842	0.206422	0.091025
局部模型 4	712	0.004410	0.009261	0.175562	0.071615
全局模型	3155	0.010000	0.070000	0.194036	0.088414

根据式(5),将局部模型和全局模型融合起来给每个用户推荐 Top-N 的电影列表。为了确定最优的全局权重参数 g ,把 0-1 区间等分成 101 份,分别进行融合实验,得到的实验结果如图 6 所示。其中,当 $g=1.0$ 时,融合模型就是单一的全局模型;当 $g=0$ 时,融合模型就是单一的局部模型。由图可知,当 g 取 0.53 时,HR 达到最高,为 0.205388,同时 ARHR 为 0.091972,超越了单一的全局模型推荐算法($HR=0.194036,ARHR=0.088414$),HR 在原基础上有了 5.85% 的提升,ARHR 有了 4% 的提升。

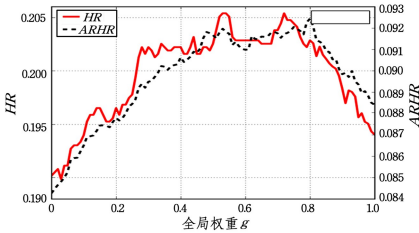


图 6 使用不同全局权重的融合结果

另外,当 g 在 0.5~0.8 范围内取值时,实验效果最好,说明在模型融合过程中,全局模型为主,局部模型为辅,局部模型修正了全局模型在预测上的不足。该实验证明了(LM)构建局部模型的重要性和必要性。

5.5 与其他流行推荐算法的对比实验

本节将本文提出的推荐算法与其他几种经典的推荐算法即基于热度的推荐(TopPop)、基于用户的推荐(UserKNN)、基于物品的推荐(ItemKNN)、加权正则的矩阵分解方法(WRMF)^[7]、稀疏线性模型推荐(SLIM)^[2]进行比较。前 3 种推荐算法是推荐系统发展过程中经典的推荐算法,因为实现简单,所以应用很普遍。WRMF 和 SLIM 是近年来提出的新推荐算法,它们分别矩阵分解和线性拟合训练推荐模型进行推荐,能够较准确地预测用户喜好,取得了非常好的推荐效果。

基于热度的推荐(TopPop)通过筛选热度最高的前 N 部电影推荐给用户,这里的热度是按照电影的评分人数来确定的,评分人数越多的电影越热门。

基于用户的推荐(UserKNN)通过寻找与当前用户最相似的前 k 个用户,推荐其喜欢的电影给当前用户来实现推荐。该方法评价一部影片 i 对一个用户 u 的推荐分 $\hat{r}_{ui}$,如式(8)所示:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_u^k(i)} sim(u, v)(r_{vi} - \mu_v)}{\sum_{v \in N_u^k(i)} sim(u, v)}$$

(8)

其中, μ_u 和 μ_v 表示用户 u 和用户 v 对电影打分的均值。 r_{vi} 表示用户 v 对电影 i 的评分。 $sim(u, v)$ 表示用户 u 和用户 v 之间的余弦相似度,用户 u 和用户 v 用他们在评分矩阵中对应的行向量表示。 $N_u^k(i)$ 表示在对电影 i 打过分的所有用户中与用户 u 最相似的前 k 个用户的集合。基于物品的推荐(ItemKNN)通过将当前用户偏爱物品的相似物品推荐给用户来实现个性化推荐。该方法评价一部影片 i 对一个用户 u 的推荐分 $\hat{r}_{ui}$,如式(9)所示:

$$\hat{r}_{ui} = \mu_i + \frac{\sum_{j \in N_i^k(u)} sim(i, j)(r_{uj} - \mu_j)}{\sum_{j \in N_i^k(u)} sim(i, j)}$$

(9)

其中 μ_i 和 μ_j 表示电影 i 和电影 j 的评分均值。 r_{uj} 表示用户 u 对电影 j 的评分。 $sim(i, j)$ 表示电影 i 和电影 j 之间的余弦相似度,电影 i 和电影 j 用他们在评分矩阵中对应的列向量表示。 $N_i^k(u)$ 表示在用户 u 打过分的所有电影中与电影 i 最相似的前 k 部电影的集合。

WRMF 和 SLIM 的具体实现方法请参见文献[7]和文献[2]。本文使用 HR 和 ARHR 两个评价指标来衡量模型的推荐质量,在豆瓣数据集上的实验结果如图 7、图 8 所示。

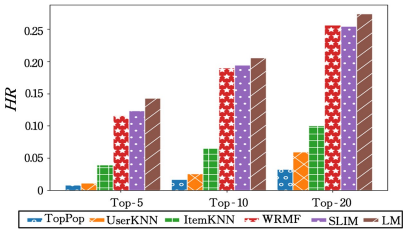


图 7 HR 指标的对比实验结果

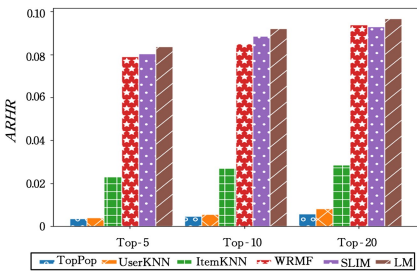


图 8 ARHR 指标的对比实验结果

由图可知,无论是在 HR 还是在 ARHR 上,WRMF,SLIM 和 LM 的推荐效果都远远优于 TopPop,UserKNN 和 ItemKNN。其中,TopPop 的推荐效果是最差的,因为它不是个性化推荐,其给每位用户推荐的都是最热门的电影。而 UserKNN和 ItemKNN 由于没有通过机器学习建模,得到的只是表层的用户相似度或者电影相似度,导致推荐质量也一般。而在 WRMF,SLIM 和 LM 这 3 种基于模型的推荐算法中,WRMF 是通过矩阵分解来实现推荐,SLIM 是 LM 的基本推荐模型,每组实验都表明所提局部加权融合推荐算法的推荐效果优于其他两种推荐算法,从而进一步验证了所提推荐算法的有效性。

结束语 本文针对传统推荐系统单一模型无法准确捕获用户偏好的问题,提出采用稀疏线性模型作为基本推荐模型,并基于用户聚类训练局部推荐模型,最后通过与全局模型加权融合来实现电影的 Top-N 个性化推荐。另外,本文充分使用了电影推荐场景中的数据,包括用户电影的评分数据和电影本身的文本数据,以丰富本文算法模型的输入。此外,本文还提出利用 LDA 主题模型计算用户语义层次特征向量的方法。从某种意义上讲,本文提出的局部模型加权融合推荐算法是内容推荐、基于邻域的协同过滤、基于模型的协同过滤三者的有效结合。最终,基于豆瓣电影数据集的实验验证了本文提出的推荐算法在一定程度上解决了单一算法模型所产生的不足,提升了推荐的效果。

但是,本文的工作也存在许多不足,最重要的就是融合模型在评价指标的提升上没有达到预期,这可能是由多方面因素导致的,可能是模型的调参问题,也有可能是基本推荐模型的选择问题,还有可能是特征工程不够完善的问题。此外,LDA 主题模型中主题数对最终推荐模型效果的影响,以及用户聚类过程中聚类数对推荐模型最终效果的影响,都需要我们今后花更多的时间通过大量实验进行探索和研究。再者,由于本文提出的推荐算法各局部模型之间是相互独立的,因此可采用并行计算方法,独立训练各局部模型,从而加快建模的速度。综上,本文只是提出局部融合推荐算法的一个开端,今后也会着重基于以上几方面来提高和完善模型。

近年来,深度学习越来越火热,在很多研究领域都取得了传统机器学习无法得到的成果。在推荐系统领域,也出现了很多基于深度学习来设计的模型,如基于 AutoEncoder 设计的 AutoRec^[19],以及 Google 提出的基于 Wide & Deep Learning 的推荐系统^[20]。我们之后也会考虑把深度学习技术加入到我们的工作中,以进一步提升推荐质量。

参 考 文 献

[1] DESHPANDE M,KARYPIS G. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS),2004,22(1):143-177.

[2] NING X,KARYPIS G. Slim: Sparse linear methods for top-n recommender systems[C]//2011 IEEE 11th International Conference on Data Mining (ICDM). IEEE,2011:497-506.

[3] FUNK S. Netflix Update; Try This at Home [OL]. <http://sifter.org/~simon/journal/20061211.html>.

[4] KOREN Y,BELL R,VOLINSKY C. Matrix factorization tech-

niques for recommender systems[J]. Computer,2009,42(8):30-37.

[5] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining. ACM,2008:426-434.

[6] KOREN Y. Collaborative filtering with temporal dynamics[J]. Communications of the ACM,2010,53(4):89-97.

[7] HU Y,KOREN Y,VOLINSKY C. Collaborative filtering for implicit feedback datasets[C]//Eighth IEEE International Conference on Data Mining,2008(ICDM'08). IEEE,2008:263-272.

[8] KABBUR S,NING X,KARYPIS G. Fism: factored item similarity models for top-n recommender systems[C]//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2013:659-667.

[9] CREMONESI P,KOREN Y,TURRIN R. Performance of recommender algorithms on top-n recommendation tasks[C]//Proceedings of the Fourth ACM Conference on Recommender Systems. ACM,2010:39-46.

[10] KANG Z,PENG C,CHENG Q. Top-N Recommender System via Matrix Completion[C]//AAAI. 2016:179-185.

[11] RENDLE S,FREUDENTHALER C,GANTNER Z,et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press,2009:452-461.

[12] O'CONNOR M,HERLOCKER J. Clustering items for collaborative filtering[C]//Proceedings of the ACM SIGIR Workshop on Recommender Systems. UC Berkeley,1999:128.

[13] XU B,BU J,CHEN C,et al. An exploration of improving collaborative recommender systems via user-item subgroups[C]//Proceedings of the 21st International Conference on World Wide Web. ACM,2012:21-30.

[14] LEE J,KIM S,LEBANON G,et al. Local low-rank matrix approximation[C]//International Conference on Machine Learning. 2013:82-90.

[15] LEE J,BENGIO S,KIM S,et al. Local collaborative ranking [C]//Proceedings of the 23rd International Conference on World Wide Web. ACM,2014:85-96.

[16] CHRISTAKOPOULOU E,KARYPIS G. Local item-item models for top-n recommendation [C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM,2016:67-74.

[17] BLEI D M,NG A Y,JORDAN M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research,2003,3(1):993-1022.

[18] HOFFMAN M,BACH F R,BLEI D M. Online learning for latent dirichlet allocation[C]//Advances in Neural Information Processing Systems. 2010:856-864.

[19] SEDHAIN S,MENON A K,SANNER S,et al. Autorec: Autoencoders meet collaborative filtering[C]//Proceedings of the 24th International Conference on World Wide Web. ACM,2015:111-112.

[20] CHENG H T,KOC L, HARMSSEN J,et al. Wide & deep learning for recommender systems[C]//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM,2016:7-10.