doi:10.3772/j.issn.2095-915x.2018.05.008

基于改进 LDA 模型的图书推荐方法研究

赵以昕　李铮　汪强兵

南京理工大学经济管理学院　南京　210094

摘 要 当前海量的商品信息困扰着顾客，商用图书推荐系统亟待研究与应用。本文提出一种改进 LDA 模型的图书推荐方法，根据用户微博内容和图书描述分别生成主题模型，并基于主题的词分布将表示用户主题的向量转化成由图书主题构成的向量，从而更精确地计算用户与图书之间的相似度，最后排序得到推荐结果。在实验中，本文对新浪微博用户推荐亚马逊图书，并与传统方法进行了对比。实验结果表明本文的方法效果更佳，可为当前的图书推荐提供参考。

关键词： 图书推荐；主题模型；改进的 LDA 模型

中图分类号： F270；G306；G35

开放科学（资源服务）标识码（OSID）

Book Recommendation Method Based on Improved LDA Model

ZHAO Yixin　LI Zheng　WANG Qiangbing

School of Economics and Management, Nanjing University of Science & Technology, Nanjing 210094, China

Abstract Massive commodity information is puzzling customers, and commercial book recommendation system urgently needs to be studied and applied. This paper proposes an improved LDA model recommendation method. It generates topic model based on user miroblog contents and book descriptions, translates words that represent user topics into words whick represent book topics by means of the word distribution, and calculates the similarity between users and books to get the ranking of recommended results. In the experiment, this paper recommends Amazon books to Sina Microblog users, and compares

基金项目： ISTIC–EBSCO 文献大数据发现服务联合实验室基金项目、南京理工大学本科生科研训练‘百千万’计划项目（201610288039）。

作者简介： 赵以昕（1996–），研究方向：自然语言处理，文本挖掘，信息检索，E–mail：chiaki390625@163.com；李铮（1995–），硕士研究生，研究方向：数据挖掘；汪强兵（1993–），硕士，研究方向：用户建模，自然语言处理。

doi:10.3772/j.issn.2095-915x.2018.05.009

it with the traditional methods. The experimental results show this method presented better performances, which could provide reference for current book recommendation.

Keywords：Book recommendation; topic model; improved LDA model

1 引言

目前大部分图书推荐方法研究都是基于图书馆的，推荐算法大致可分为基于协同过滤的算法、基于中图分类法的算法和基于主题模型的算法[1]。随着互联网的广泛普及和电子商务的迅速发展，顾客从琳琅满目的商品信息中找到自己想要购买的商品更加困难，作为商品推荐的图书推荐也引起了众多学者的关注。现在电商网站基本采用类似于图书馆的协同推荐，

这种协同过滤的方法本身存在冷启动、解释性不强和计算量大等问题。而社交网站则拥有电商网站的所缺少的大量用户内容数据，可以对用户进行详细准确的描述，恰好可以弥补这些问题。新浪微博作为当前流行的一个交互和传播信息平台，拥有海量的用户数据，因而本文选择了新浪微博用户作为研究对象，对其推荐亚马逊图书。

具体而言，本文提出一种基于新浪微博社交平台的图书推荐模型，结合微博内容进行用户画像，通过 LDA 模型生成微博用户的主题向量，同样基于图书描述对亚马逊图书生成图书的主题向量。因图书描述数据与微博内容在语料结构上存在着不可忽略的区别，本文提出隶属度的概念，通过主题词的分布，将用户的主题向量转化为图书的主题向量来描述用户，然后计算用户与图书之间的相似度得到推荐结果。本文的改进 LDA 模型的方法，与传统的基于词向量的方法和基于 LDA 的方法比较推荐效果更佳。

2 相关研究

当前环境下，信息组织的能力提升速度远远赶不上信息爆炸的速度，信息越来越多且呈现碎片化。如何快速准确地为用户推荐喜爱的商品变得尤为重要。经典的推荐方法有基于协同过滤的推荐和基于内容的推荐。

协同过滤的主要思想是借助于其他用户的历史行为来为当前用户推荐，一般认为是由 Goldberg 等在 1992 年首次提出并应用于 Tapestry 系统[2]。冷亚军[3]在对协同推荐的研究中将协同推荐系统分为了基于记忆式的推荐与基于模型的推荐两类。其中前者指基于目标用户的相关用户预测兴趣，而后者指从用户数据中学习一般性的规律并对未知项目预测评分。他在研究中指出，协同推荐主要包括了高泛用性、推荐结果的高新颖度、易于实现等优势，同时也存在抗稀疏性较差、鲁棒性较差、以及扩展性不足等问题。邱均平[4]研究了在图书馆环境下的协同推荐应用，从用户长期偏好和短期偏好两个角度入手，最终发现高校学生的阅读偏好存在由浅入深式的增长规律，同时兴趣的衰减周期也偏短，并由此认为应在对用户偏好分

析中加入时间因素。冯勇 [5] 等提出了协同推荐在社交网络场景下的改进方案，通过用户关系数据精确计算用户相关性，从而改进了推荐效果。卢丹等 [6] 通过用户所购买的商品得到复杂网络社团发现，能够在用户只有较少历史数据时仍可保证推荐的新颖度和覆盖度。Xu 等 [7] 构建了一种基于主题建模的统计模型用于适应不同类型的 UGC，进一步优化了传统方法。Bu 等 [8] 观察到用户之间存在具有差异性的局部偏好，并在传统协同过滤的基础上进行扩展，制定了一种多类集群模型细分偏好关系，并成功提高了推荐性能。但基于协同过滤的方法普遍存在稀疏性问题 [9] 和冷启动问题。推荐系统依赖于用户的偏好记录，在数据缺失的情况下很难保障推荐结果的可靠性。

基于内容的推荐算法起源于信息检索和信息过滤，是根据用户喜好、习惯及行为等记录构造用户偏好文档，基于推荐项目与用户的偏好文档进行相应的推荐。徐清 [10] 等提出了一种适用于电商网站的推荐模型，基于 web 日志抽取访问类特征，与内容特征融合后计算相似度。徐雅斌 [11] 等提取了用户不同维度的文本特征，将分别计算的相似度作为特征向量，并通过逻辑回归预测推荐结果。国外的研究偏向于将基于内容的推荐与协同推荐相结合。Ana 等 [12] 设计了一套结合协同与内容推荐的电视节目推荐系统。其中内容推荐方面使用了常规的相似度排序，另外将相似度作为权值加入到协同推荐中，共同生成推荐结果。Dieleman 等 [13] 设计了一种潜在因素模型用于音乐推荐，结合了用户行为特征与音频采样特征，正则化后搭建 eep 卷积神经网络用于评估预测，并取得了不错的效果。Narducci 等 [14] 考虑了跨语言场景下的内容推荐问题，使用了维基百科与 BabelNet 的知识构建了概念模型，成功处理了多语言场景的推荐。虽然基于内容的推荐方法也存在一些问题，如过拟合问题、结果同质化等等，但能够有效解决缺乏历史数据情况下的冷启动问题，同时也不存在数据稀疏问题，对于社交平台推荐有一定的优越性，因而本文选择基于内容的推荐方法。

此外，用户作为推荐的受众，对其的特征刻画也极为重要。邵秀丽等 [15] 使用用户浏览内容，结合浏览时间和操作时间描述用户兴趣及程度。Tang 等 [16] 提出了树形结构的条件随机域提取文本信息，通过概率主题模型将其整合构建用户兴趣模型。Ma 等 [17] 认为用户信息分布在不同信息中，使用多种数据包括 Twitter、Facebook 和 LinkedIn、个人主页等来源，研究用户显性兴趣和隐性兴趣之间的语义关系以拓宽针对用户兴趣的文本层次分析。He 等 [18] 通过新浪微博用户的原始和转发数据揭示用户兴趣，模型所发现的兴趣词涵盖了用户的本身标签并在此基础上有所扩展。Tu[19] 利用微博用户的个人信息构建二分类器进行职业预测，包括自我描述、用户标签、验证信息、微博内容和提及的用户、网页链接、命名实体和主题标签，并通过社区结构完善职业预测。可以发现研究者会使用尽可能多的数据来进行用户画像，以保证用户兴趣能够更完整地表达，而在很多情况下，很难获取同一个用户的大量数据，行为数据甚至是跨平台数据。

结合上述分析，本文提出基于内容的图书推荐方法，仅使用用户微博内容描述用户兴趣，

doi:10.3772/j.issn.2095-915x.2018.05.008

通过改进 LDA 的方法使推荐变得更简单、易操作，同时保证一定的准确率。

3 图书推荐模型

由于对图书的新颖性与拓宽性的推荐准确性难以评估，本文仅考虑用户感兴趣领域内的图书，最大化推荐结果的准确率，不考虑类似新颖度、宽泛性一类推荐系统的其他评价指标。基于这种假设，本研究将推荐任务视为单纯的图书描述和微博内容的相似度计算，提出一种基于改进 LDA 的图书推荐模型，将由不同文本生成的主题模型通过词语分布关联到一起，达到能够互相转换的效果。整个推荐流程包括数据预处理、样本筛选、文档表示与推荐方法四部分，具体如图 1 所示，所涉及关键技术如下介绍：

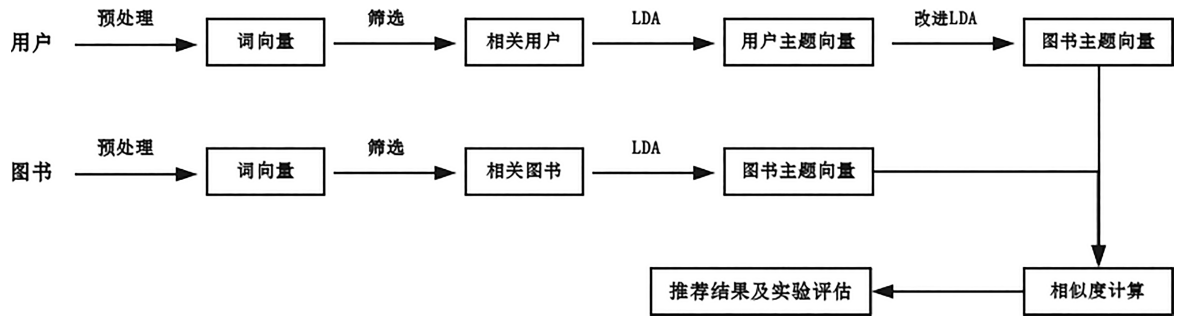

图 1 推荐模型流程图

3.1 数据预处理

鉴于 LDA 的输入是词向量，因此需对每篇文档进行预处理。首先对每篇微博分词后，去除其中的停用词与罕见词，其中罕见词指在全部文档中出现频率低于 0.1% 的词。同时，微博中存在一些常见词，其在全部文档中出现频率高于 50%，而并不具备实际区分度，例如“转发”、“分享”等词，这些常见词会影响到主题模型的质量与最终结果的解释性，在此过程中也一并去除。最后将每位用户发过的全部微博合并，形成能表示用户的词向量。图书的预处理工作与微博内容的预处理基本相同。

3.2 文本特征提取

根据抽样结果统计显示，每位微博用户的特征词数量处于平均 500~1000 个左右的水准，其中仅包含了 5%~10% 左右用户感兴趣领域的相关词，然而每个图书领域的总特征词普遍在几万个左右。由于领域词的比重过低，导致了词向量对用户的描述存在偏差，从而导致在推荐结果中混杂了不相关的图书。由此需要借助更高级的模型来表达用户特征。

主题模型基于贝叶斯概率模型，将由词向量构成的文档转化成语义层次更高的主题向量，以牺牲一定的精度为代价，为高维度稀疏的词向量补充语义信息。潜在狄利克雷分布（Latent Dirichlet Allocation，LDA）[20] 是一种常见的主题模型。该模型认为一篇文档中的每个词的生成过程是：作者先按某种概率决定了某个主题，然后从这个主题中以一定的概率选择了某个词，即这篇文章中的每一个词都是属于唯一一个主

题的。因此可以将词的生成过程用数学公式（1）表示：

$$P(w|d)=P(w|t)\times P(t|d) \quad \text{公式（1）}$$

其中，P(w|d) 表示文档 d 生成词语 w 的概率，P(w|t) 表示词语 w 属于话题 t 的概率，P(t|d) 表示生成话题 t 的概率。

最初生成主题向量，首先假设每个 topic 的 P(t|d) 相等，选择使词语生成概率最大的主题作为该词所属的主题，即 argmax(P(w|d))。得到文档中每个词所属的主题，累加后即可得到文档新的主题向量，即 P(t|d)。根据新的主题向量可以更新 P(w|d)，反复迭代后，即可得到文档稳定的主题向量。

3.3 样本筛选

主题模型对于文档的特征表示能力依赖于输入文档集合稳定的语义结构。由于微博用户本身涉及的领域复杂多样，其中各种领域的用词习惯，主题结构等均存在一定程度上的偏差。这种误差会严重影响到主题模型的表征能力与解释性能。为了排除这种干扰因素的影响，本研究设计了一种非监督迭代算法，筛去了与图书领域不相关的用户以及少数关联性较低的图书，从而便于构建出精确的主题模型。

本算法中使用了文本处理中常用的加权技术 TFIDF。此外，我们在构建主题模型的过程中发现，微博与图书语料中干扰样本的词语分布与语料整体存在差异。通过排除与整体差异过大的样本可以提升主题模型的性能。KL 距离是一种衡量不同分布之间差异的指标，适用于此处文本相似性的刻画。因此我们将 KL 距离引入到样本筛选算法之中。具体筛选步骤如表 1 所示：

表 1 用户与图书的样本筛选算法描述

算法名称：用户与图书的样本筛选算法
输入：图书集 {B} 和用户集 {U}，其中 U 表示用户的词向量，B 表示图书的词向量
输出：筛选过后的用户集 {U} 和图书集 {B}
步骤：

Step1：对 U，B 分别进行 TFIDF 加权。对于 U，B 中第 i 个词的权值，其计算公式如下：

$$weigh\ t_i=\frac{word_i}{\sum_{j\in U/B} word_j}\times\log(\frac{M}{M_i}) \quad \text{公式（2）}$$

$word_i$ 表示第 i 个词的词频，M 表示用户或图书的总数，M_i 表示 {U} 或 {B} 中包含词 i 的用户数或图书数；

Step2：将 {B} 累加平均，得到图书整体词向量 B_{total}：

$$B_{total}=\frac{\sum word_i\cdot weight_i}{M} \quad \text{公式（3）}$$

Step3：设 {T}={U} ∩ {B}。计算 {T} 中全部元素与 B_{total} 的 KL 距离，具体公式如下：

$$D(T)=\sum P(w\mid T)\times\log\frac{P(w\mid T)}{P(w\mid B_{total})} \quad \text{公式（4）}$$

其中，$P(w|T)$ 表示词向量 T 中词的频率 w，$P(w|B_{total})$ 表示词向量 B_{total} 中词 w 的频率；

Step4：对全部 D(T) 排序，并从 {T} 中去除排序结果最后 10% 的用户或图书；

Step5：基于新的 {T}，计算相应的 B_{total}；

Step5：重复 step2-step5，直至 min（D（T））高于指定阈值 m。

doi:10.3772/j.issn.2095-915x.2018.05.008

3.4 基于LDA的推荐方法

为了便于计算相似度，需要将用户与图书的主题向量整合到相同维度上（同样的主题）。首先尝试了传统思路，将用户与图书视为同类文档，共同生成主题模型。通过实验发现，用户与图书之间的主题重合度很低，并且主题模型的质量也没有达到预期，这种方式生成的主题模型对图书与用户的表达效果都不够精准。这一点在实验部分会具体介绍。同时观察到仅由用户或是仅由图书生成的主题模型中领域特征较为明显，对于用户与图书的表述更为完善，因此考虑对全部用户与全部图书分别生成主题模型，将用户的主题向量转化为基于图书词的主题向量，然后根据所有词所属的主题数确定主题分布。

由于用户与图书语料在用词上存在差异性，所以直接根据用户词分布和图书的主题 - 词分布生成图书主题的效果不理想。例如，一个专注于计算机某一领域的用户，在微博中也会使用大量其他领域的特征词。而图书的领域词则较为集中，因此由用户词分布直接得到的图书主题会带来很大程度的偏差。由此，我们考虑从主题本身上进行重新编码。

在 LDA 模型中，主题对应的词语分布一定程度上反映了主题的内容。基于 LDA 的生成原理，本文提出一种假设：如果主题 A 与主题 B 在内容上的相似程度近乎一致，那么作者在生成文档时可以使用主题 B 近似代替主题 A。基于这种假设，本文继而提出主题的隶属度这一概念，这里记为 $R(t_i,t_j)$，表示 t_j 隶属于 t_i 的比重。$t_i \cap t_j$ 在 t_j 中的比重越高，则表明 t_j 隶属于 t_i 的程度越高。在本研究中，使用 KL 距离计算这一指标。此外，对于本主题模型之外的主题，这种隶属度的累积也可以推测出潜在的主题概率。例如，某位用户类似“黑客”，“网络攻防”一类的主题居多，则“信息安全”的隶属度之和较高，其主题概率也会偏高。

本文将隶属度的概念引入到图书推荐模型中。我们近似的将隶属度看成主题间的条件概率 $R(t_i,t_j)$。假设已经生成了用户的主题模型 $\{t_i\}$ 与图书的主题模型 $\{t_j\}$，通过计算 t_i 与 t_j 之间的隶属度，进而通过贝叶斯概率模型计算出用户文档中图书主题的概率 $P(t_j|d)$。同时我们对同一 t_j 做了归一化处理加以约束。对于任意图书主题 t_i，具体计算方法如公式（5），（6）所示：

$$P(t_j \mid d)=\frac{\sum_i P(t_i \mid \mathrm{d}) \cdot \mathrm{R}(t_i,t_j)}{\left\|\overrightarrow{P(t_i \mid \mathrm{d}) \cdot \mathrm{R}(t_i,t_j)}\right\|} \quad \text{公式（5）}$$

其中，

$$\mathrm{R}\left(t_i,t_j\right)=\sum P(\mathrm{w} \mid t_i) \times \log \frac{P(w \mid t_i)}{P(w \mid t_j)} \quad \text{公式（6）}$$

通过计算 $\{t_i\}$ 与 $\{t_j\}$ 两两之间的相似度，我们可以将用户主题构成的主题向量重新编码为图书主题构成的主题向量：

$$\overrightarrow{P(t_j|\mathrm{U})}=\sum_j \frac{\sum_i P(t_i \mid \mathrm{U}) \cdot \mathrm{R}(t_i,t_j)}{\left\|\overrightarrow{\mathrm{P}(t_i \mid \mathrm{U}) \cdot \mathrm{R}(t_i,t_j)}\right\|} \quad \text{公式（7）}$$

在得到用户的图书主题向量后，我们已经将用户与图书整合到相同的空间中，进而可以得到二者的相似度。在此我们通过余弦值这一指标计算：

$$\mathrm{sim}(\mathrm{U},\mathrm{B})=\frac{\sum_{i=0}^{\{t_B\}} P(t_i \mid user) \cdot P(t_i \mid B)}{\| \overrightarrow{T_B \mid U} \| \cdot \| \overrightarrow{T_B \mid B} \|} \quad \text{公式（8）}$$

由此，我们对用户与图书集中每本书的相似度进行排序，选择排名靠前的书作为推进结果。完整的图书推荐模型步骤见表 2：

表 2 基于改进 LDA 的图书推荐算法描述

算法名称：基于改进 LDA 的图书推荐算法 输入：图书集合 {B}，用户集合 {U} 输出：推荐图书集合 {R} 步骤： Step1：使用表 1 中的算法对 {U} 与 {B} 进行筛选； Step2：根据筛选后的用户生成用户主题模型，并得到用户主题集合 $\{t_U\}$，t_U 对应的词分布 P(w|t_U)，用户 U 对应的主题分布 P(t_u|U)；同样得到图书主题集合 $\{t_B\}$，对应的词分布 P(w|t_B)，以及每本图书的主题分布 P(t_B|B) Step3：对于任意 $t_i \in \{t_U\}$，$t_j \in \{t_B\}$，根据 P(w|t_i) 与 P(w|t_j)，公式（6），计算 P(t_i,t_j)； Step4：根据 P(t_u|U)，$\{R(t_i,t_j)\}(t_i \in \{t_U\}, t_j \in \{t_B\})$，公式（7）计算用户的图书主题向量 P($t_B$|U)； Step5：根据 P($t_B$|U)，P($t_B$|B)，公式（8）计算 B 与 U 的相似度 sin(U,B)，并将相似度排序得到推荐集合 {R}。

4 实验与结果分析

4.1 实验数据概述

本研究共爬取了新浪微博 43596 用户的微博文本数据，以及来自亚马逊网站 47013 本计算机类图书的内容概述，用于构建推荐模型与评估。两类数据样例可见表 3、表 4。

表 3 新浪微博用户数据样例

用户名	微博内容
	犹记《东京消失》，不知为啥特喜欢看航空题材灾难和惊险电影，只是一人乘机通常是起飞前睡着，送饮料时小醒，然后再一睡到落地。
黑客老鹰	裹挟和碾压有利于巨头的超越成长，在无法筛查的不透明数据大海中，台风的力量是强大的，尤其是臃肿拖沓的治理者也需要刺激与数字的时候。
	……

表 4 亚马逊图书数据样例

图书名	图书类别	图书描述数据
华为防火墙技术漫谈	安全与加密	1. 本书为是市场上第一本华为官方出版的防火墙学习用书。2. 凡是以协议、教材面目呈现的书，都有一个致命问题：理论多而实战内容少。本书写作前期充分分析了华为 400 个问题、防火墙网上案例，所以内容能直接命中用户实际场景，命中技术难点和常见问题，在理论结合实战方面做得非常到位。

对全体语料使用 3.3 节的样本筛选后剩余 3900 高质量用户与 45000 本图书。我们使用 java 实现了本文提出的推荐模型，并且使用了开源 LDA 工具 jgibbslda 用于生成 LDA 模型。同时我们在附录中放入了程序的主要代码，以供模型实现参考。由于微博用户普遍内容质量与噪声的问题，在有限的推荐结果中无法直观对比推荐效果，因此本研究在此考虑领域中最具代表性的用户用于测试推荐性能。在此选择了“安全与加密”，“云计算与大数据”两个大类中，与其领域相关度最高的总共五位用户样本用于实验。五位用户的基本情况如下：

doi:10.3772/j.issn.2095-915x.2018.05.008

表 5 实验用户信息表

用户 id	微博昵称	备注	所属分类
1851708063	石晓虹	奇虎 360 副总裁	安全与加密
2108809592	陈怡桦 EvaChen	云计算安全及服务器安全的全球领导品牌——趋势科技 (Trend Micro) 合伙创始人及全球执行总裁	安全与加密
1906343315	vivchar	中科院计算所副研究员，大规模数据计算专家查礼	云计算与大数据
1802786827	谢恩伟	微软大中华区高级副总裁、Xbox 事业部中国区总经理、百家合董事	云计算与大数据
2856831960	孙博凯 – Prakash	微软亚太研发集团首席技术官	云计算与大数据

4.2 测评方法

为了更好地对比算法质量，本次实验采用召回率和准确率评估推荐结果。我们为每位用户推荐固定数量的图书，并通过人工标注的方式，将与用户领域一致、并且内容相关的书目视为正确推荐结果，其余视为错误结果，由此计算出推荐结果的准确率。人工标注通过两人在互不干扰的情况下进行判断推荐效果，经一致性检验得到 kappa 值为 0.72，最后取两人标注“正确推荐”的交集作为最终推荐正确的结果，其余均为错误结果。虽然这种标注模式注重于表示推荐图书与用户的相关度，虽然在描述用户真实兴趣偏好上存在一定程度的偏差，而在各推荐方法的纵向对比上依然具备相对可靠的区分能力。同时我们又结合关键词匹配与人工标注的方式，预先标记出若干高相关度的图书视为用户感兴趣的样本，用于计算召回率。两种指标计算方法如下：

$$\text{Precision} = \frac{N(rs)}{N(s)} \qquad \text{公式（9）}$$

$$\text{Recall} = \frac{N(rs)}{N(p)} \qquad \text{公式（10）}$$

其中 N(rs) 为推荐正确的图书数量，N(s) 为向用户推荐的所有图书数量，N(p) 为预先人工标记的图书总数。

同时，为进一步对比实验效果，我们将本文的模型（改进 LDA）与以下两种传统思路进行对比：

（1）基于词向量的方法

构建全部用户与图书的词向量，并分别进行 TFIDF 加权。对于每位被推荐用户，计算全部图书的词向量与其的余弦值作为相似度。将相似度进行降序排序得到图书推荐结果。

（2）基于 LDA 的方法

将全部用户与图书的词向量视为一类文档，共同构建主题模型（简称混合主题模型），生成每位用户与每本图书的主题向量。对于被推荐用户，计算与全部图书主题向量的相似度，排序后作为推荐结果。

4.3 实验结果分析

4.3.1 过滤算法的有效性分析

首先说明本文采取的过滤策略是有效的。比较过滤前与过滤后两种传统方法的推荐结果具体如表 6 所示：

由此看出，两种传统方法经过过滤准确率得到一定提升，所以使用迭代计算相似度的过滤策略是有效的。因此在后续实验中，对传统方法与本文模型统一使用了这种筛选算法。

表 6 两种传统方法的推荐结果比较

推荐数	词向量过滤前	词向量过滤后	LDA 过滤前	LDA 过滤后
10	0.3	0.4	0.25	0.3
20	0.3	0.4	0.3	0.35
30	0.3	0.33	0.2	0.23
40	0.325	0.375	0.275	0.3
50	0.28	0.42	0.3	0.38

4.3.2 改进 LDA 模型的有效性分析

首先我们通过主题模型的生成效果比较改进算法的有效性。

困惑度（perplexity）最早由 David M. Blei[20] 提出，是目前常用的一种评价主题模型性能的手段，可以表现出主题模型的泛化能力。本文采用困惑度作为主题模型效果评测的方法，其计算公式为：

$$\text{perplexity(D)} = \exp\{\frac{\sum_{d=1}^{M} \log p(w \mid d)}{\sum_{d=1}^{M} Nd}\} \quad \text{公式(11)}$$

其中 M 表示语料库中的文档总数，Nd 表示文档的词语总数，$p(w|d)$ 表示模型生成文档的概率。困惑度越低表示主题模型的性能越好。本实验计算了不同主题数下各主题模型的困惑度变化情况，统一使用 300 次迭代的 gibbs 采样，结果如下：

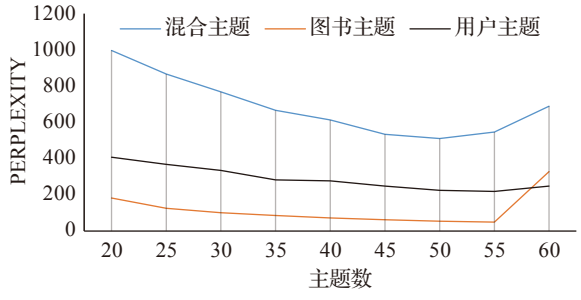

图 2 不同主题数下的 perplexity 水平

由图 2 可见，共同生成的主题模型 perplexity 值远高于图书与用户分别生成的主题模型，泛化能力也不如后者，由此可见分别生成主题模型的必要性。其中图书主题的 perplexity 普遍高于用户主题，因此在模型设计中选择统一使用图书的主题模型。

同时我们在实验中以某数据挖掘领域用户的微博为例，通过可视化对比改进主题模型的性能。观察改进 LDA 主题模型和混合主题模型中此用户主题词的分布情况。我们选择了两者中主题向量中概率最高的主题，情况如下图 3、图 4：

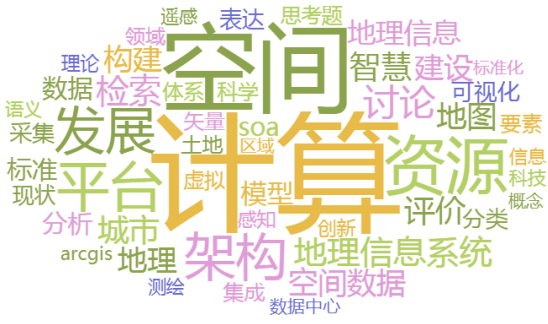

图 3 改进主题模型主题词分布

图 4 混合主题模型主题词分布

doi:10.3772/j.issn.2095-915x.2018.05.008

可以看出改进的主题模型中的特征词较混合模型相比，与此用户研究领域更加相关。不仅对用户的特征表示更准确，同时也更具备解释性。

4.3.3 推荐效果比较分析

改进 LDA 算法与两种传统方法的准确率和召回率如图 5、6 所示：

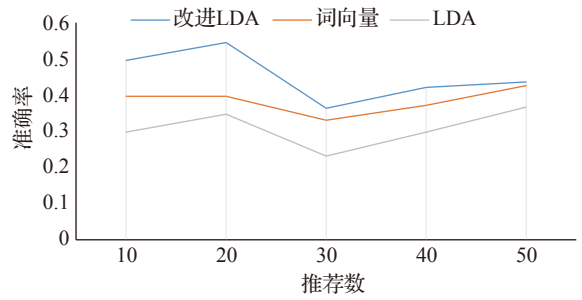

图 5 推荐模型与传统方法的准确率比较

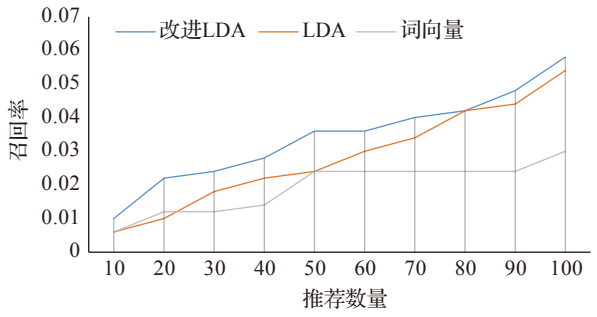

图 6 推荐模型与传统方法的召回率比较

从结果可以看出，基本与预期情况一致。准确率方面传统 LDA 方法全面不如词向量。这也是由于图书和微博语料存在差异，导致主题模型质量较差，从而对用户和图书的特征提取也不够准确。而经过本文中的思路改进后，反而获得了超过词向量的理想效果，由此可以看出本文算法的可靠性。

而召回率方面，词向量虽然对于排名靠前的推荐结果有着不错的准确率保证，然而对于该领域整体相似度计算而言表现并不好，一定程度上反应了词向量较低的容错性与词语表征能力的局限性。同时也由于两类语料本身文本过短的原因，导致词向量方法存在不可忽略的误差，难以得到稳定的效果。而主题模型在词语特征的基础上考虑了文档集合的全局因素，从而确保了更稳定的性能。

5 结语

为了解决社交平台场合下的商品推荐问题，为基于内容的推荐系统提供更准确的解决方案，本文提出了基于改进 LDA 的推荐模型。通过主题间的内容相似性实现了不同主题模型之间的转换，从而将用户主题模型生成的主题向量转化为图书主题模型生成的主题向量。最后在实验中证明了本模型的有效性。

与此同时，本文提出的主题向量计算公式对主题的刻画还不够精确，主题数没有呈现出明显的变化趋势，这可能是由于本文没有考虑到被代替主题自身之间的重复内容导致的，在今后的工作中也会进一步研究。同时也会考虑到多种主题联合的代替可能。此外，本文实验只考虑了图书推荐中的准确性，没有考虑多样性、惊喜度等指标，在使用到实际推荐场合时依然存在改进空间。并且实验对象本身都是领域专家，不仅用户特征较为明显，用词也偏于专业，在映射为图书领域的主题时效果较为稳定。而这并不是微博用户的常态，大多数微博用户的兴趣与领域相对而言更为复杂，对于特征不够明显的用户，本文没有做深入的研究。在实际面向微博用户的图书推荐场合，可能需

要考虑更多的复杂因素。在收集了更多数据之后，也会进一步优化实验方案，提高模型的可解释性。

参考文献

[1] 傅汉霖, 顾小宇. 图书推荐算法综述[J]. 计算机时代, 2016(12):21-23.

[2] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12):61-70.

[3] 冷亚军, 陆青, 梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能, 2014, 27(8):720-734.

[4] 邱均平, 张聪. 高校图书馆馆藏资源协同推荐系统研究[J]. 图书情报工作, 2013, 57(22):132-137.

[5] 冯勇, 李军平, 徐红艳,等. 基于社会网络分析的协同推荐方法改进[J]. 计算机应用, 2013, 33(3): 841-844.

[6] 卢丹, 王君博, 武森. 电子商务中基于复杂网络社团发现的商品推荐研究[J]. 工业技术创新, 2015(1):61-65.

[7] Xu Y, Yin J. Collaborative recommendation with user generated content[J]. Engineering Applications of Artificial Intelligence, 2015(45): 281-294.

[8] Bu J, Shen X, Xu B, et al. Improving collaborative recommendation via user-item subgroups[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2363-2375.

[9] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003, 14(9): 1621-1628.

[10] 徐清. B2C电子商务中商品推荐模型研究[D]. 大连：大连理工大学, 2012.

[11] 徐雅斌, 石伟杰. 微博用户推荐模型的研究[J]. 电子科技大学学报, 2015, 44(2):254-259.

[12] Barragáns-Martínez A B, Costa-Montenegro E, Burguillo J C, et al. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition[J]. Information Sciences, 2010, 180(22):4290-4311.

[13] Dieleman S, Schrauwen B. Deep content-based music recommendation[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:2643-2651.

[14] Narducci F, Basile P, Musto C, et al. Concept-based item representations for a cross-lingual content-based recommendation process[J]. Information Sciences, 2016, 374: 15-31.

[15] 邵秀丽, 乜聚科, 侯乐彩,等. 基于综合用户信息的用户兴趣建模研究[J]. 南开大学学报（自然科学版）, 2009, 42(3):8-15.

[16] Tang J, Yao L, Zhang D, et al. A Combination Approach to Web User Profiling [J]. Acm Transactions on Knowledge Discovery from Data, 2010, 5(1):2.

[17] Ma Y, Zeng Y, Ren X, et al. User Interests Modeling Based on Multi-source Personal Information Fusion and Semantic Reasoning[C]. International Conference on Active Media Technology. Springer, Berlin, Heidelberg, 2011:195-205.

[18] Li H E, Jia Y, Han W, et al. Mining User Interest in Microblogs with a User-Topic Model [J]. China Communications, 2014, 11(8):131-144.

[19] Tu C, Liu Z, Sun M. PRISM: Profession Identification in Social Media with Personal Information and Community Structure[M]. Singapore: Springer Social Media Processing, 2015.

[20] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research Archive, 2003(3):993-1022.

doi:10.3772/j.issn.2095-915x.2018.05.008

附录

本研究中使用 java 实现了文中设计的推荐模型，其中关键代码如下：

```
/**
 * 计算用户的图书主题向量
 * @param user_topic_word 用户主题的词分布 P(w|t_U)
 * @param book_topic_word 图书主题的词分布 P(w|t_B)
 * @param user_topic_vec 用户主题向量 P(t_U|U)
 * @return 用户的图书主题向量 P(t_B|U)
 */
    Hashtable<Integer,Double> topicList(
    Hashtable<Integer,Hashtable<String,Double>> user_topic_word,
    Hashtable<Integer,Hashtable<String,Double>>book_topic_word, Hashtable<Integer,Double> user_topic_vec){
    Hashtable<Integer,Double> book_topic_vec = new Hashtable<>();
    double topic_vec_length = 0;
      for(int book_topic_:book_topic_word.keySet()){
        double value = 0;
          for(int user_topic_:user_topic_word.keySet()){
            doublesim=KL(user_topic_word.get(user_topic_),
                                book_topic_word.get(book_topic_));
            value += sim*user_topic_vec.get(user_topic_);
          }
        book_topic_vec.put(book_topic_,value);
        topic_vec_length += value*value;
      }
      topic_vec_length = Math.pow(topic_vec_length,0.5);
    for(int book_topic_:book_topic_vec.keySet())
            book_topic_vec.put(book_topic_,-
    book_topic_vec.get(book_topic_)/topic_vec_length);
    return book_topic_vec;
  }

/**
 * 得到推荐结果
 * @param book_vec 各图书主题向量
 * @param user_vec 用户主题向量
 * @param recommend_count 推荐数量
 * @return 推荐结果列表
 */
ArrayList<String>        recommend(
Hashtable<String,Hashtable<Integer,Double>> book_vec,
Hashtable<Integer,Double> user_vec, int recommend_count ){
      ArrayList<Sim> sim_list = new ArrayList<>();
    for(String book_id:book_vec.keySet()){
      double sum1=0,sum2=0,sum3=0;
      for(int word:user_vec.keySet()){
          double book_value = book_vec.get(-
```

```
book_id).get(word);
            double user_value = user_vec.get(word);
            sum1 += user_value*book_value;
            sum2 += book_value*book_value;
            sum3 += user_value*user_value;
          }
          sim_list.add(new Sim(book_id,sum1/Math.
pow(sum2,0.5)/Math.pow(sum3,0.5)));
      }
      Collections.sort(sim_list,new Sort());
       ArrayList<String> recommend_result = new
ArrayList<>();
      for(int i=0;i<recommend_count;i++)
         recommend_result.add(sim_list.get(i).id);
      return recommend_result;
   }

   /**
    * 计算 KL 散度
    * @param set1 分布 1
    * @param set2 分布 2
    * @return KL 散度
    */
    public static double KL(Hashtable<String,Dou-
ble> set1,Hashtable<String,Double> set2){
      double sum = 0;
      for(String word:set2.keySet()){
         double value1 = set1.get(word)==null?set1.
get(word):0;
         double value2 = set2.get(word);
         sum += value1*Math.log(value1/value2);
      }
      return sum;
   }
   class Sim{
      String id;
      double sim;
      Sim(String id,double sim){
         this.id = id;
         this.sim = sim;
      }
   }
   class Sort implements Comparator {
      public int compare(Object o1, Object o2) {
         Sim s1=(Sim)o1;
         Sim s2=(Sim)o2;
         if(s1.sim<s2.sim)return 1;
         else if(s1.sim>s2.sim)return -1;
         else return 0;
      }
   }
```