基于主题效能的学术文献推荐算法

杜永萍[1]，杜晓燕[1]，姚长青[2]
(1. 北京工业大学 计算机学院，北京 100124；2. 中国科学技术信息研究所，北京 100038)

摘 要： 针对文献推荐问题，提出了一种基于主题效能的学术文献推荐算法，该算法使用潜在狄利克雷分布(latent Dirichlet allocation，LDA)对候选文献和用户发表的文献进行建模，挖掘出具有高效能的主题集合，并根据候选文献中高效能主题的分布情况来计算它与用户兴趣之间的相似度，最后向用户推荐有价值的文献. 实验结果表明：提出的算法比基于频繁项挖掘的算法具有更高的推荐准确率和推荐召回率，可同时满足用户对个性化和文献质量两方面的需求.

关键词： 推荐系统；效能；主题模型
中图分类号： TP 391 **文献标志码：** A **文章编号：** 0254－0037(2015)02－0215－08
doi： 10.11936/bjutxb2014070009

Recommendation Algorithm Based on Topic Utility for Academic Papers

DU Yong-ping[1], DU Xiao-yan[1], YAO Chang-qing[2]
(1. College of Computer Science, Beijing University of Technology, Beijing 100124, China;
2. Institute of Scientific and Technical Information of China, Beijing 100038, China)

Abstract: To solve paper recommendation problem, an academic paper recommendation algorithm based on topic utility is proposed. This approach uses latent Dirichlet allocation (LDA) model to build the model of candidate papers and users´ published papers, and then the topic sets with high utility are mined. The similarity between the user interest and the candidate papers is calculated according to the distribution of the high utility topics. Finally, the valuable papers are recommended to the users. Experimental results show that this method is effective, and it can get higher precision and recall than the algorithm based on apriori. Meanwhile, this method can meet the user demand for both the quality and the personalization.

Key words: recommender system; utility; topic model

随着信息科技时代的到来，数据信息爆炸式地增长，怎样才能在海量的数据中获取重要的信息，是现如今亟待解决的问题. 在这样的形势下，推荐系统[1]应运而生，并逐渐成为信息时代不可或缺的组成部分.

推荐算法主要包括2种：基于协同过滤(collaborative filtering，CF)的推荐和基于内容(content-based，CB)的推荐[2]. 基于内容的推荐方法需要定义推荐对象和用户模型，计算推荐对象的内容特征和用户模型中的用户兴趣之间的相似度，从而选择相似的推荐对象作为推荐结果. 推荐对象和用户模型可采用关键字表示特征，从而使用词频

收稿日期： 2014-07-08
基金项目： 国家科技支撑计划资助项目(2013BAH21B02-01)；北京市自然科学基金资助项目(4123091)；ISTIC－ELSEVIER期刊评价研究中心开放基金资助项目
作者简介： 杜永萍(1977—)，女，副教授，主要从事自然语言处理、信息检索、机器学习、数据挖掘方面的研究，E-mail：ypdu@bjut.edu.cn

反文档频率(term frequency-inverse document frequency, TF-IDF)方法计算各个特征的权重;然而,一义多词和一词多义的现象往往导致关键字并不能准确表示文档特征,因此有学者提出了基于聚类模型[3]、基于语义贝叶斯[4]以及基于概率潜在语义分析[5]的推荐方法. 针对用户兴趣随时间变化的问题,Somlo 等[6]和 Zhang 等[7]提出了更新用户模型的自适应过滤方法,该方法使用相似度高的推荐对象来更新用户模型; Chang 等[8]将用户兴趣进一步分为长期兴趣和短期兴趣,赋予短期兴趣较大的权重来提高用户兴趣建模的准确性;除此之外,还有学者提出使用决策树[9]、人工神经网络[10-11]等机器学习方法来建立和更新更为复杂的用户模型. 协同过滤推荐[12]使用统计技术等搜索目标用户的若干最近邻居,然后根据该最近邻居对项的评分预测目标用户对项的评分,再根据评分的高低选择前 N 个最高评分(Top-N)作为推荐列表;协同过滤推荐技术是目前电子商务系统中应用最为成功的推荐技术之一.

学术文献推荐[13-14]是推荐系统的一个应用方向,可帮助用户在海量文档中找出有价值的文献. 传统的文献推荐技术存在以下 2 点不足: 1) 这些技术利用 TF-IDF 方法将文献表示为以词频为维度的向量,通过计算文献间的相似度,进行基于内容的推荐. 然而,词汇的相似并不足以表示文献的近似关系,TF-IDF 方法仅统计文献中单词的词频信息,无法捕捉到文献的语义特征. 2) 文献数据库中的每篇文献都将作为推荐的候选对象,无论该篇文献自身质量的高低. 在这种情况下,用户得到的推荐结果和用户兴趣相似,但质量并不能保证. 针对上述提出的问题,有学者提出了基于效能的学术资源推荐算法[15]. 该算法挖掘出频繁被引的文献集合,并允许用户使用权威度、出版日期、流行度等主观值来表达自己的个性化需求,从而挖掘出高效能文献集合推荐给用户. 在学术文献推荐领域,该算法提出的"效能"的概念为学者提供了一个新的研究思路,但它并没有利用文献的内容进行推荐.

本文提出了基于主题效能的学术文献推荐算法. 首先,使用潜在狄利克雷分布(latent Dirichlet allocation, LDA)主题模型将每篇文献表示为以主题为维度的向量,并挖掘出所有频繁出现的主题集合;其次,计算每个频繁主题集合对于用户的效能. 频繁出现的主题通常代表学术研究的热点,因此计算频繁出现的主题对于用户的效能值可保证推荐的文献都是包含热点主题的文献. 若将文献的质量定义为其研究主题的热度,那么可认为这些文献都是高质量的. 算法的第 2 步则保证了推荐的文献是符合用户兴趣的,即同一主题集合对于不同用户的效能有可能不相同,而推荐给用户的一定是对于该用户具有较高效能的主题集合的文献.

1 相关工作介绍

1.1 LDA 主题模型

LDA 是一种非监督机器学习技术,可用来识别语料库中潜藏的主题信息. 语料库中的每篇文档与 T 个主题的一个多项分布相对应,将该多项分布记为 θ. 每个主题又与词汇表中的 V 个单词的一个多项分布相对应,将这个多项分布记为 ϕ. θ 和 ϕ 分别有一个带有超参数 α 和 β 的 Dirichlet 先验分布. 对于语料库中的每篇文档,LDA 定义了如下生成过程,这个生成过程如图 1 所示[16].

1) 对于一篇文档 D 中的每个单词,从该文档所对应的多项分布 θ 中抽取一个主题 z;

2) 从主题 z 所对应的多项分布 ϕ 中抽取一个单词 w;

3) 将这个过程重复 N_d 次,就产生了文档 D,这里的 N_d 是文档 D 的单词总数.

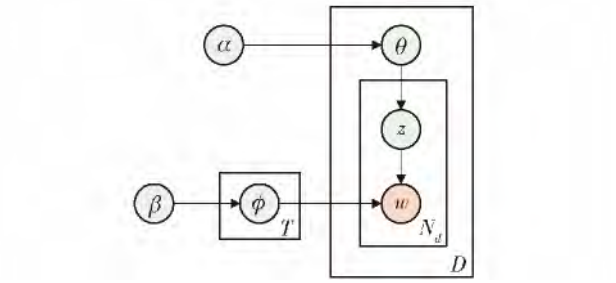

图 1 LDA 主题模型

Fig. 1 LDA topic model

该模型有 2 个参数需要推断: 一个是"文档-主题"分布 θ,另一个是"主题-单词"分布 ϕ. 通过学习这 2 个参数,可获取文档作者感兴趣的主题及每篇文档所涵盖的主题比例等信息. 推断方法主要有变分 EM 算法和常用的 Gibbs 抽样法.

1.2 频繁项集挖掘

频繁项集是指那些经常共现的物品集合,其中"频繁"是由设定的阈值(即支持度,support)来衡量的,一个项集的支持度被定义为数据集中包含该项集的记录所占的比例. 频繁项挖掘是指挖掘出满足最小支持度的项集.

目前频繁项集挖掘已经有很多比较成熟的算法，最经典的算法是 Apriori 算法，它的原理是：如果某个项集是频繁的，那么它的所有子集也是频繁的. 也就是说，如果一个项集是非频繁集，那么它的所有超集也是非频繁的. 该算法首先会生成所有单个物品的项集列表，接着扫描交易记录来查看哪些项集满足最小支持度要求，那些不满足最小支持度的集合会被去掉. 然后，对剩下来的集合进行组合以生成包含 2 个元素的项集. 接下来，再重新扫描交易记录，去掉不满足最小支持度的项集. 该过程重复进行，直到所有不满足最小支持度的项集都被去掉.

1.3　高效能挖掘

“效能”(utility) 用来度量每个项目的一个或多个属性的综合质量，例如利润、受欢迎度、重要度、影响力等；高效能项目集挖掘是指挖掘具有较高效能的项目，它允许用户使用重要性值来表达自己对项目集中对象的兴趣偏好，例如一件商品带来的利润；项目集中的对象已有的信息称为对象的客观值，例如交易中售出的物品数量；项目集中对象的客观值和重要性值共同决定了该项目集的效能. 直观地说，效能是对项目集的“有用程度”的一个量化的反映[15].

表 1 给出了 6 条商品交易记录，每个列对象 G_i 表示 1 件商品，表中的元素代表销售数量，这张表给出了待挖掘对象(即商品)的客观值. 各个商品的利润，即对象的重要性值，商品 $G_1 \sim G_6$ 的利润分别为 2、7、4、1、3、6. 以项集 $\{G_2, G_5\}$ 为例，其效能值 $u\{G_2, G_5\} = (4\times7+3\times3)+(3\times7+5\times3)+(1\times7+2\times3)=86$，若设效能阈值为 80，则该项集为高效能项集，若设效能阈值为 90，则该项集为低效能项集.

表 1　商品交易记录

Table 1　Commodity trading records

交易	商品					
	G_1	G_2	G_3	G_4	G_5	G_6
T_1	0	4	1	0	3	0
T_2	1	0	3	1	8	0
T_3	2	2	0	5	0	1
T_4	0	3	0	1	5	6
T_5	2	0	2	0	0	1
T_6	1	1	0	0	2	0

高效能项集挖掘的目标是发现所有效能值超过设定阈值的项目集. 相关定义如下：

1) $G=\{G_1, G_2, \cdots, G_n\}$ 表示对象集合，例如在表 1 中，对象即为商品，$G=\{G_1, G_2, G_3, G_4, G_5, G_6\}$.

2) $T=\{T_1, T_2, \cdots, T_m\}$ 表示交易记录，其中 $T_i=\{o_{i,1}, o_{i,2}, \cdots, o_{i,n}\}$，$o_{i,j}$ 表示交易 T_i 中对象 G_j 的客观值，例如在表 1 中，交易记录即为销售记录，$T=\{T_1, T_2, T_3, T_4, T_5, T_6\}$，$T_4=\{o_{4,1}, o_{4,2}, o_{4,3}, o_{4,4}, o_{4,5}, o_{4,6}\}=\{0, 3, 0, 1, 5, 6\}$.

3) $s(G_p)$ 是主观值，它表示对象 G_p 的重要性，用户可通过设定此值的大小来表达自己对 G_p 的偏好程度，例如在商品销售中，主观值即为商品的利润，$s(G_2)=7$，$s(G_6)=6$.

4) $u(G_p, T_q)=o_{q,p}\, s(G_p)$ 是效能函数，表示对象 G_p 在交易 T_q 中产生的效能，例如在表 1 中，有 $u(G_2, T_4)=o_{4,2}\cdot s(G_2)=3\times7=21$，表示商品 G_2 在销售记录 T_4 中给商家带来的利润为 21.

5) $u(C, T_q)=\sum_{G_p\in C} u(G_p, T_q)$ 且 $(\forall G_p\in C, o_{q,k}\neq0)$，或 $u(C, T_q)=0$ 且 $(\exists G_p\in C, o_{q,k}=0)$ 是项集 C 在交易 T_q 中的效能，其中 $C\subseteq G$，例如在表 1 中，$u(\{G_2, G_5\}, T_1)=u(G_2, T_1)+u(G_5, T_1)=4\times7+3\times3=37$，$u(\{G_2, G_6\}, T_1)=0$.

6) $u(C)=\sum_{T_q\in T} u(C, T_q)$ 是项集 C 的效能，例如在表 1 中，$u(\{G_2, G_5\})=u(\{G_2, G_5\}, T_1)+u(\{G_2, G_5\}, T_4)+u(\{G_2, G_5\}, T_6)=(4\times7+3\times3)+(3\times7+5\times3)+(1\times7+2\times3)=86$.

设定效能阈值 threshold，当 $u(C)\geqslant$ threshold 时，称 C 是高效能项集. 高效能项集挖掘算法的目的是找到所有具有高效能的项集 $C_{\mathrm{H}}=\{C \mid C\subseteq G, u(C)\geqslant\text{threshold}\}$.

2　基于主题效能的学术文献推荐算法

基于主题效能的学术文献推荐算法首先将候选文献表示成主题维度上的向量，之后挖掘出频繁出现的主题集合，并计算这些主题集合对于每名用户的效能值. 每个主题集合对于用户的效能用该集合中的主题和用户兴趣主题之间的相似度来度量.

2.1　数据表示

基于主题效能的学术文献推荐算法首先使用 LDA 主题模型将每篇文献表示为以主题为维度的向量，得到文献-主题矩阵，如图 2(a) 所示；之后，算法筛选出文献中概率值较高的主题，将这些主题在矩阵中的取值置为 1，将其余主题在矩阵中的取值置为 0，从而生成高效能项集挖掘算法的数据集，

如图 2(b) 所示.

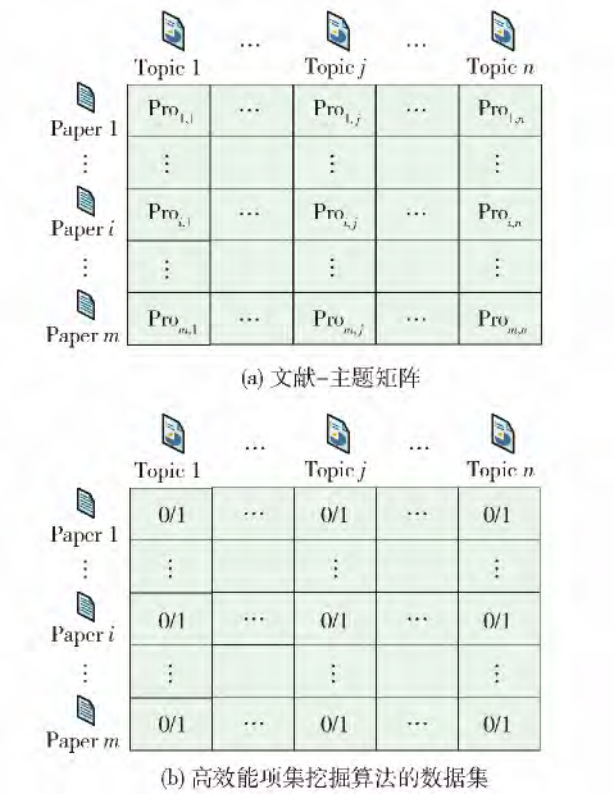

图 2 数据表示

Fig. 2 Data representation

2.2 挖掘高效能项集

挖掘高效能项集的过程分为 2 步,分别为挖掘频繁出现的主题集合以及挖掘高效能主题集合.

2.2.1 挖掘频繁出现的主题集合

使用 Apriori 算法可从上述的数据集中挖掘出频繁出现的主题集合,每个集合中包含一个或多个主题. 设定一个最小支持度,挖掘出支持度大于等于该最小支持度的主题集合. 例如,用编号 1 ~ N 表示 N 个主题,获取到的频繁主题集合示例如表 2 所示.

表 2 频繁主题集合示例

Table 2 Sample of frequent topic set

序号	主题集合	支持度
1	[5 ,28 ,67]	0.13
2	[13 ,78]	0.21
3	[11 ,47 ,98]	0.09
⋮	⋮	⋮

2.2.2 挖掘高效能主题集合

用户对于每个主题的兴趣度可通过对其发表的文献进行 LDA 主题建模,得到文献在主题维度上的向量表示:{ V_1, V_2,…, V_n} ,而向量中的元素值 V_i 就是该文献的作者对主题 i 的兴趣度. 本文使用兴趣度作为主观值,使用候选文献在每个主题上的概率分布作为主题的客观值,从而计算每个主题集合对于用户的效能值,最终保留效能值高的集合,过滤掉效能值低的集合,挖掘高效能主题集合算法描述如下所示.

输入: 文献-主题概率矩阵 M; 用户 u 发表的文献的主题向量 V; 频繁主题集构成的集合 C_F; 效能阈值 threshold.

输出: 高效能主题集构成的集合 C_H.

$C_H = \{\}$
for $\forall c \in C_F$ /* 遍历每一个频繁主题集 */
  total = 0
  for $\forall \text{paper}_i \in M$ /* 遍历每一篇候选文献 */
    /* 遍历频繁主题集中的每一个主题 */
    for $\forall \text{topic} \in C_0$
      /* 获取主题的编号 */
      the number of topic→j
      total = total + $M_{ij}V_j$
    end for
    end for
    if total ⩾ threshold
      add C_0 to C_H
end for
return C_H

从上面的算法描述中可看出,尽管在频繁项挖掘阶段需要将文献-主题矩阵转换为由 0 和 1 构成的矩阵,但在计算主题集合的效能时还需要使用文献-主题矩阵中候选文献在不同主题上的概率分布值,而不是简单的 0 或 1.

2.3 产生推荐

在得到每名用户的高效能主题集合后,根据候选文献中主题的分布情况来为用户推荐学术文献.

为了得到用户对每个主题的兴趣度,对用户发表的文献进行 LDA 主题建模,将文献在主题维度上的向量作为用户兴趣模型,之后计算候选文献和用户兴趣模型之间的相似度,将相似度高的候选文献推荐给用户. 在计算候选文献和用户兴趣之间的相似度时,需要适当调整每个主题的权重,赋予高效能主题集合中的主题较大的权重 weight_1,赋予其他的主题较小的权重 weight_2,候选

文献和用户兴趣模型之间的相似度计算算法描述如下所示.

输入: 文献-主题概率矩阵 M; 用户 u 发表的文献的主题向量 V; 用户 u 的高效能主题集构成的集合 C_H; 主题权重 weight_1 和 weight_2.

输出: 用户 u 的兴趣模型和每篇候选文献之间的相似度(集合 C).

```
/*  计算所有主题的权重之和,供归一化时使用 */
total = 0
for ∀ topic ∈ M      /*  遍历每一个主题 */
   weight = weight_2
   for ∀ C_0 ⊆ C_H /*  遍历每一个高效能主题集 */
      /*  赋予高效能主题集中的主题较大的权重 */
      if topic ∈ C_0
         weight = weight_1
         break
      end for
      total = total + weight
end for

C = { }
for ∀ paper_i ∈ M      /*  遍历每一篇候选文献 */
   sim = 0
   for ∀ topic_j ∈ M      /*  遍历每一个主题 */
      weight = weight_2
      /*  遍历每一个高效能主题集 */
      for ∀ C_0 ⊆ C_H
         /*  赋予高效能主题集中的主题较大的权重 */
            if topic_j ∈ C_0
               weight = weight_1
               break
      end for
   end for
   /*  相似度计算, weight/ total 表示归一化权重 */
   sim = sim + M_ij V_j weight/total
   add{ i , sim} to C
end for
return C
```

预先设定 2 个权重，一个权重是高效能主题集合中主题的权重 weight_1，另一个权重是不在高效能主题集合中的主题的权重 weight_2. 首先，算法统计所有主题在用户 u 的高效能主题集合中的分布情况，将每个主题所应具有的权重进行求和，用于之后权重的归一化. 其次，算法扫描全部候选文献，计算每篇候选文献和用户兴趣模型在主题维度上的相似度，并进行记录，最后输出全部候选文献的编号以及各自对应的相似度值.

3　实验结果与分析

3.1　数据集

采用 ACL Anthology Network (AAN, http://clair.eecs.umich.edu/aan/index.php) 数据集来对算法的性能进行评测. 在删除缺失标题或摘要的文献后，得到 1965—2012 年的文献数据，共计 15 166 篇. 在进行实验之前，先对每篇文献进行以下几步预处理操作: 1) 抽取每篇文献的题目和摘要，并进行分词; 2) 去除停用词; 3) 提取单词词干.

为了建立用户兴趣模型，将 1965—2011 年发表的 13 908 篇文献作为训练集，将 2012 年发表的 1 258篇文献作为测试集. 对训练集以及测试集中的文献标题和摘要进行 LDA 主题建模，训练集中的文献作为推荐候选文献，测试集中文献的主题模型即作为文献作者的兴趣模型，使用本文提出的算法推荐与用户(即文献作者)兴趣最为相似的前 N 篇候选文献.

3.2　评测标准

采用准确率(precision)和召回率(recall)作为算法性能的评价指标. 准确率表示推荐列表中实际被引用的文献在推荐列表中所占比例，计算方法为

$$\text{precision}_k = \frac{|\text{AL}_k \cap \text{PL}_k|}{|\text{PL}_k|} \tag{1}$$

召回率表示推荐列表中实际被引用的文献在测试文献的参考文献列表中所占比例，计算方法为

$$\text{recall}_k = \frac{|\text{AL}_k \cap \text{PL}_k|}{|\text{AL}_k|} \tag{2}$$

式中: AL_k 为文献 k 实际引用的文献列表; PL_k 为算法针对文献 k 给出的推荐列表. 计算 1 258 篇文献的平均准确率和平均召回率来评测算法性能.

3.3　参数对算法性能的影响

基于主题效能的学术文献推荐算法有 2 个参数，分别是挖掘频繁出现的主题集阶段的最小支持度以及挖掘高效能主题集阶段的效能阈值，下面讨论这 2 个参数对算法性能的影响.

3.3.1 最小支持度对算法性能的影响

设定 LDA 模型的主题个数为 200，最小支持度分别取 0.05、0.10、0.15、0.20、0.25，可找出满足这些最小支持度频繁出现的主题集合. 之后，效能阈值取总效能(每名用户的全部频繁主题集的效能之和)的 0.4%，找出每名用户的高效能主题集合，赋予高效能主题集合中的主题较大的权重 1.0，赋予其他的主题较小的权重 0.48，推荐数量取 5～300，每次增加 5 篇文献，图 3(a)(b)分别给出了不同最小支持度下推荐的平均准确率和平均召回率.

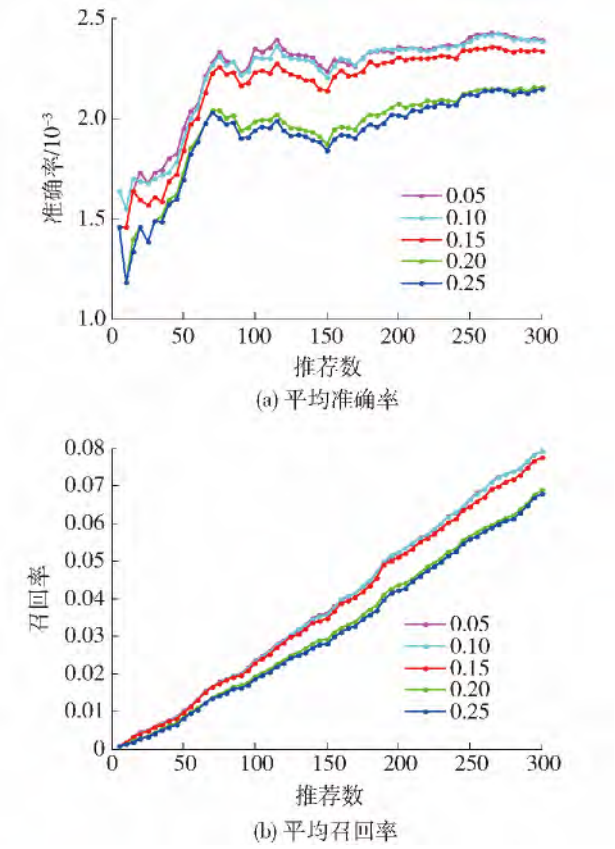

图 3 最小支持度对算法性能的影响

Fig. 3 Performance of algorithm impacted by minimum support

实验结果表明：最小支持度越小，算法的推荐准确率和推荐召回率越大. 在降低最小支持度时，算法挖掘出的频繁出现的主题集合的数量就会增加，此时提供给后续推荐阶段的高效能主题集合的数量便会随之增加，即权重为 $weight_1$ 的主题的数量增加，而权重为 $weight_2$ 的主题的数量减少，此时推荐的平均准确率和平均召回率就会提高.

3.3.2 效能阈值对算法性能的影响

类似地，设定 LDA 模型的主题个数为 200，最小支持度取 0.05，可找出满足该最小支持度的频繁出现的主题集合. 效能阈值分别设为总效能的 0.4%、0.8%、1.2%、1.6%、2.0%，可找出大于等于这些效能阈值的高效能主题集合. 赋予高效能主题集合中的主题较大的权重 1.0，赋予其他的主题较小的权重 0.48，推荐数量取 5～300，每次增加 5 篇文献，图 4(a)(b)分别给出了推荐的平均准确率和平均召回率.

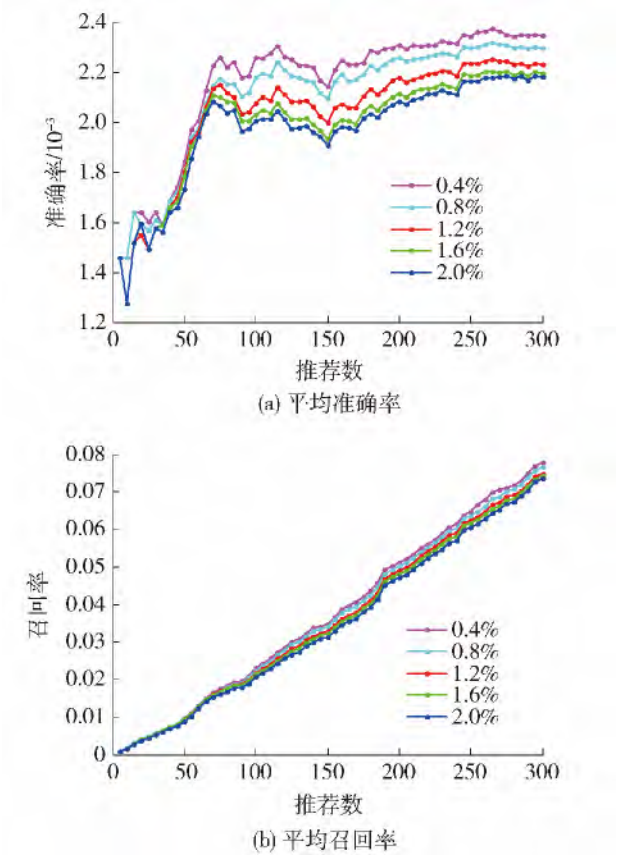

图 4 效能阈值对算法性能的影响

Fig. 4 Performance of algorithm impacted by utility threshold

实验结果表明：效能阈值越小，算法的推荐准确率和推荐召回率越大. 与最小支持度对算法性能的影响类似，在降低效能阈值时，提供给后续推荐阶段的高效能主题集合的数量就会增加，最终提升推荐的平均准确率和平均召回率.

直观上讲，更多的权重为 $weight_1$ 的主题意味着推荐时可参考更多的高效能主题，从而可获得更高的推荐准确率和召回率. 因此，上述 2 个实验证明了本文提出算法的合理性.

3.4 算法有效性验证

3.4.1 用户个性化需求验证

为了验证本文提出的算法在满足用户个性化需求方面的有效性，将本文的算法(Utility)与基于频

繁项挖掘的算法(Apriori)进行比较，基于频繁项挖掘的推荐算法只挖掘出频繁出现的主题集合，并不进行高效能过滤. 同样地，设定 LDA 模型的主题个数为 200，最小支持度取 0.05，效能阈值取总效能的 0.4%，高效能主题集合中的主题的权重设为 1.0，其他主题的权重设为 0.48，推荐数量取 5 ~ 300，每次增加 5 篇文献，图 5(a)(b)分别给出了 2 种算法的平均准确率和平均召回率.

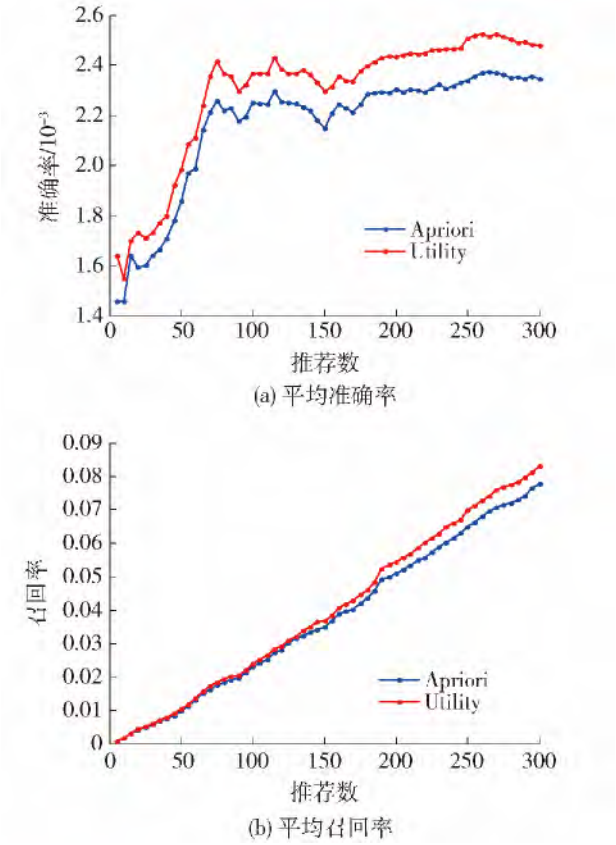

图 5　基于主题效能的学术文献推荐算法(Utility)与基于频繁项挖掘的推荐算法(Apriori)的性能比较

Fig. 5　Performance comparison between Utility and Apriori

实验结果表明：本文提出的算法在推荐准确率和召回率方面都要高于基于频繁项挖掘的推荐算法，由于综合考虑了用户的个性化需求，推荐给用户的文献中的主题既是频繁出现的主题也是符合用户个性化需求的主题.

3.4.2　文献质量验证

为了验证本文提出的算法是否可满足用户对文献质量的要求，类似地，将本文的算法(Utility)与基于频繁项挖掘的算法(Apriori)进行比较，使用平均引用次数来量化文献的质量. 实验设置同 3.4.1 节，推荐数量取 5 ~ 100，每次增加 5 篇文献，表 3 给出了 2 种算法的部分平均引用次数.

表 3　基于主题效能的学术文献推荐算法(Utility)与基于频繁项挖掘的推荐算法(Apriori)的引用次数比较

Table 3　Reference frequency comparison between Utility and Apriori

推荐数	Utility	Apriori
5	3.226 8	3.165 6
20	6.026 0	6.052 7
40	6.657 8	6.650 7
60	7.416 2	7.481 0
80	7.814 0	7.806 8
100	7.695 3	7.692 7

从实验结果中可看出：2 种算法所推荐的文献的平均引用次数相差不大，经计算，基于主题效能的学术文献推荐算法在所有推荐个数下的平均引用次数的平均值为 7.550 6，而基于频繁项挖掘的算法为 7.550 0，由此可见，基于频繁项挖掘的算法在加入高效能挖掘步骤后并未降低推荐文献的质量，也就是说，本文提出的基于主题效能的学术文献推荐算法可同时满足用户在兴趣个性化和文献质量两方面的需求.

4　结论

1) 提出了基于主题效能的学术文献推荐算法：首先利用 LDA 主题模型将候选文献和用户发表的文献表示为主题维度上的向量，之后使用 Apriori 算法挖掘出高效能主题集合，最后根据高效能主题在候选文献中的分布情况来为用户推荐高质量并且兴趣相似的文献.

2) 实验结果表明，本文提出的算法具备有效性，相比于基于频繁项挖掘的算法，本文提出的算法具有更高的推荐准确率和推荐召回率，并且 2 个算法所推荐文献的平均引用次数几乎相同，从而证明本文提出的算法可同时满足用户对个性化和文献质量两方面的需求.

参考文献：

[1] XU Hai-ling, WU Xiao, LI Xiao-dong. Comparison study of Internet recommendation system [J]. Journal of Software, 2009, 20(2): 350-362. (in Chinese)

[2] WANG Guo-xia, LIU He-ping. Survey of personalized recommendation system [J]. Computer Engineering and Applications, 2012, 48(7): 66-76. (in Chinese)

[8] DILIGENTI M, GORI M, MAGGINI M. Users, queries and documents: a unified representation for Web mining [C] // Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. Washington, DC: IEEE Computer Society, 2009: 238-244.

[4] BASILE P, TINELI E, DEGEMMIS M, et al. Semantic Bayesian profiling services for information recommendation [C] // Proceedings of the 11th International Conference, KES 2007 and XVII Italian Workshop on Neural Networks Conference on Knowledge-based Intelligent Information and Engineering Systems. Vietri Sul Mare: Springer, 2007: 711-719.

[5] WU Hu, WANG Yong-ji, CHENG Xiang. Incremental probabilistic latent semantic analysis for automatic question recommendation [C] // Proceedings of the 2008 ACM Conference on Recommender Systems. New York: ACM, 2008: 99-106.

[6] SOMLO G, HOWE A. Adaptive lightweight text filtering [C] // Process of the 4th International Symposium on Intelligent Data Analysis. Lisbon: Springer, 2001: 319-329.

[7] ZHANG Yi, CALLAN J, MINKA T. Novelty and redundancy detection in adaptive filtering [C] // Process of 25th Annual International ACM SIGIR Conference. Tampere: ACM, 2002: 81-88.

[8] CHANG Ye-in, SHEN Jun-hong, CHEN T L. A data mining-based method for the incremental update for supporting personalized information filtering [J]. Journal of Information Science and Engineering, 2008, 24(1): 129-142.

[9] NIKOVSKI D, KULEV V. Induction of compact decision trees for personalized recommendation [C] // Proceedings of the 2006 ACM Symposium on Applied Computing. New York: ACM, 2006: 575-581.

[10] SHARMA A, DEY S. A document-level sentiment analysis approach using artificial neural network and sentiment lexicons [J]. ACM SIGAPP Applied Computing Review, 2012, 12(4): 67-75.

[11] ZHU Xing, HUANG Shen, YU Yong. Recognizing the relations between Web pages using artificial neural network [C] // Proceedings of the 2003 ACM Symposium on Applied Computing. New York: ACM, 2003: 1217-1221.

[12] DENG Ai-lin, ZHU Yang-yong, SHI Bo-le. A collaborative filtering recommendation algorithm based on item rating prediction [J]. Journal of Software, 2003, 14(9): 1621-1628. (in Chinese)

[13] SUGIYAMA K, KAN M. Exploiting potential citation papers in scholarly paper recommendation [C] // Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries. New York: ACM, 2013: 153-162.

[14] MENG Fan-qi, GAO De-hong, LI Wen-jie, et al. A unified graph model for personalized query-oriented reference paper recommendation [C] // Proceedings of the 13th ACM/IEEE-CS Joint Conference on Conference on Information & Knowledge Management. New York: ACM, 2013: 1509-1512.

[15] LIANG Shen-shen, LIU Ying. A utility-based recommendation approach for academic literatures [C] // Process of the 6th China Management Annual Meeting. Beijing: CSMM, 2011: 91-102. (in Chinese)

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

(责任编辑 吕小红)