进行长期累积, 就能形成贴近用户长期偏好的历史数据. 我们可以基于实时反馈信息进行实时推荐, 基于历史数据进行离线推荐, 将二者结合, 就能在考虑用户长期固有偏好的同时也能关注用户短时间内的兴趣焦点变化.

② 系统运行不畅, 往往是由于推荐计算时间过长, 不能及时提供推荐结果导致. 本文提出待推荐池这一概念, 用于存储准备推荐给用户的数据集. 推荐时, 系统首先从待推荐池提取一定数量的数据推荐给用户, 再通过实时推荐与离线推荐产生推荐数据补充到待推荐池内, 这样即使推荐计算运行时间长, 用户依然能够连续多次从待推荐池中提取数据, 保证了系统运行始终流畅. 待推荐池中的数据, 除了在系统运行初期是按类别等比例随机获取的数据外, 后面补充到待推荐池中的数据都是实时推荐与离线推荐的结果. 因此, 长时间运行后, 池内数据会不断贴近用户的偏好, 这样通过分析池内数据, 也可对本系统推荐效果进行评价.

③ 为了提高系统健壮性与可伸缩能力, 在推荐系统中增加控制模块. 一方面, 按照实时推荐模块和离线推荐模块产生的推荐数量, 控制模块能够合理调节两者份额, 补充到待推荐池内, 提高系统的健壮性. 如当离线推荐数据少而实时推荐数据多时, 会适当增大实时推荐比例, 反之亦然. 另一方面, 当系统增加其它推荐算法时, 很容易通过对该模块进行设置, 来选择其它推荐算法的推荐结果, 提高了系统的可伸缩能力.

2.2 推荐系统运行流程

本系统整体运行流程如图 1 所示, 具体步骤包括:

① 初始状态下, 将所要推荐的数据, 如书籍、文本、图片、电影、音乐等数据, 随机等比例地填充到待推荐池内, 保证池内数据全面多样;

② 每次推荐, 系统从待推荐池内获取定额数据推荐给用户, 如每次从池内提取 10 篇文章推荐给用户;

③ 记录模块记录用户对本次推荐的反馈, 如用户对其中某些数据的阅读, 收藏, 点赞, 评论等行为信息;

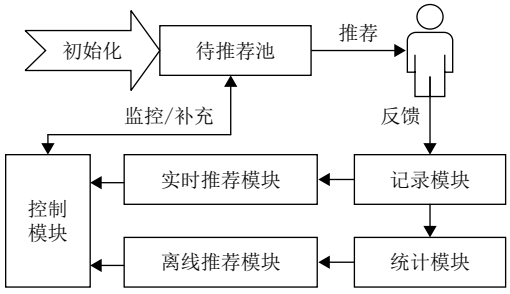

图 1 推荐流程图

④ 用户获取下次推荐数据时, 实时推荐开始运行. 提取记录模块内的所有记录信息, 实时推荐模块根据这些信息计算并产生推荐数据;

⑤ 统计模块获取记录模块信息 (提取后记录模块清空), 与前面获取的历史纪录合并, 离线推荐模块根据这些统计信息计算并产生推荐数据;

⑥ 控制模块监控待推荐池, 对实时推荐数据与离线推荐数据进行去重处理, 包括与用户的历史数据比较去重, 与待推荐池比较去重, 实时与离线推荐数据比较去重; 然后, 根据待推荐池内缺额数, 通过加权或计算 TopN 的方式, 组合两类推荐数据, 补充到待推荐池内.

2.3 待推荐池的运作机制

待推荐池以队列形式存储, 池内最大容积固定, 每次推荐时提取池内排名在前的定额数据给用户. 控制模块根据池内缺额数, 结合实时推荐数据与离线推荐数据补充到待推荐池.

初始状态下, 推荐程序由于无用户历史数据和即时反馈数据, 无法进行实时推荐与离线推荐, 系统自动将各类数据等比例随机选取填充到待推荐池内, 一方面保证池内数据全面多样, 使用户在池内能够获得所有类型数据; 另一方面, 系统后续可从用户对各类数据的反馈中获取用户的兴趣焦点与偏好, 从而进行实时推荐与离线推荐.

推荐初期, 用户历史数据没达到一定规模, 实时推荐正常而离线推荐无法有效进行. 这时, 提取实时推荐数据中至多需补充数额量的数据, 将这些数据的 80% 插入到待推荐池队首, 其余 20% 数据插入到队尾, 保证下次推荐给用户的数据既能围绕用户当前的兴趣焦点, 又能让用户有机会接触到池内其它种类的数据, 此外也保证了待推荐池内的数据量.

随着用户历史数据不断积累, 离线推荐开始正常运行. 补充待推荐池时, 组合两类推荐数据至需补充数额量, 将其中实时推荐部分插入到队首, 使下次推荐的数据中包含大量实时推荐数据与原池内少量数据, 保证推荐给用户的数据大部分围绕用户的兴趣焦点上, 还有小部分是池内存储的用户之前可能感兴趣的数据; 将离线推荐数据部分插入到队尾, 逐渐改变待推荐池内数据类型, 使其逐渐贴合用户的历史偏好.

为保证系统流畅性, 用户每次获取数据时, 直接从待推荐池内提取, 然后控制模块选择调节两种推荐方法的结果数据进行数据补充. 当多次推荐用户无反馈,

即实时推荐模块不能正常产生推荐数据, 而同时用户的历史数据稀疏, 离线推荐模块也没有推荐数据时, 会出现待推荐池内数据不断减少的现象, 这时就需要设置预警阀值, 以避免待推荐池内数据量为空. 当池内数据小于预警阀值时, 系统需要及时补充数据, 所补充的数据可依据统计模块中已有的用户偏好信息产生, 若没有, 可按照初始数据填充方式, 将各类数据等比例随机选取, 产生一定量的数据进行补充.

3 推荐实验及结果分析

3.1 实验方案

将上述推荐系统应用到微信文章推荐上. 微信文章是一种数量大、种类多的文章类型. 文章数据主要来源于微信精选网站 (https://fzn.cc/),使用 Python 爬虫 Scrapy 进行数据抓取, 提取包括美文哲理、幽默笑话、趣味测试、经验总结、美食美景及保健养生 6 大类文章类型. 同时补充搜狗微信网站 (https://weixin.sogou.com/) 中搞笑、养生堂、旅游、美食类文章数据, 微信群网站 (https://weixinqun.com/article) 中养生之道、搞笑段子、人气美食、旅游美景、人生哲理、经验总结类文章数据.

(1) 数据存储形式

采用 MySQL 数据库与 Redis 数据库存储文章数据. MySQL 是一个开源的关系型数据库管理系统, 文章数据爬取后存在该数据库中; 而 Redis 是一个基于内存的, 存储形式为 Key-Value 的非关系型数据库 (NoSql)[10], 推荐系统中的待推荐池及其他数据池均使用 Redis 数据库存储.

为实现微信文章推荐, 每个用户拥有六个数据池 (如下表 1 所示), 各数据池的命名格式为“用户账号_数据池特性_存储类型”.

(2) 推荐方式选择

本推荐过程中, 实时推荐采用基于内容数据的推荐方式, 根据用户近期阅读的文章, 将含有这些文章标签的文章推荐给用户, 从而聚焦在用户兴趣焦点上. 离线推荐采用基于用户行为数据的推荐方式, 将相似用户阅读过而本用户没有阅读过的文章推荐给用户, 拓展推荐数据的兴趣广度.

3.2 文章标签规范化处理

文献[2]指出基于内容数据的推荐方式需要进行预处理获得其项目特性. 在来源网站上, 微信文章被设置多个标签来代表其特性, 标签数多为 3 个, 但存在无标签、少标签、多标签、标签不规范等问题, 为此, 需要对获取的文章进行预处理, 规范文章标签. 使用 Python 结巴分词结合 TF-IDF 统计方法实现.

文章标签设置标准如图 2 所示. 选取方法如下:

表 1 数据池设计表

数据池名称	数据池命名格式	所在模块	作用
待推荐池	(account)_recommend_list	-	存放待推荐给用户的文章数据
分析池	(account)_analysis_set	记录模块	存放用户本次推荐中阅读的文章数据
已阅读池	(account)_readed_set	统计模块	存放用户已经阅读过的文章数据
类型统计池	(account)_types_zset	统计模块	统计用户已经阅读的所有文章类型
标签统计池	(account)_labels_zset	统计模块	统计用户已经阅读的所有文章标签
被分享池	(account)_shared_set	离线推荐模块	存放相似用户分享给该用户的文章数据

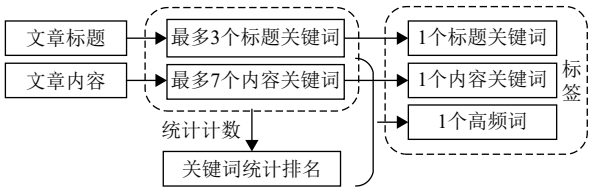

图 2 标签提取规则

① 使用 Python 结巴分词, 提取 3 个标题关键词, 7 个除标题关键词外的内容关键词, 两类关键词提取词数不足时, 以能取到的词数为准;

② 将所有文章的所有关键词进行统计计数形成一个关键词统计排名;

③ 针对每篇文章, 从标题关键词中提取 TF-IDF 最大的一个标题关键词, 同理从内容关键词中提取一词. 然后针对剩余的最多 8 个关键词, 对照关键词统计排名获得排名最高的一个高频词, 以这三个关键词作为文章标签.

分词过程中, 引用用户字典, 确保如“微信”“朋友圈”等网络词语能够被识别出来; 引用停用词词库, 避免分词结果出现常用而无意义的词语; 引用同义词词库 (如菜鸟和新手同义), 使词语标准化; 引用包含词词库 (如颜色包含白色、灰色等), 使标签更具代表性.

确定标签提取规则后, 我们对 527 篇具备标签的微信文章进行测试, 以提取的最多 10 个关键词是否包含标签词作为分词合格标准, 以最终提取的 3 个关键

词是否包含标签词作为提取合格标准进行测试, 测试结果如图 3 所示. 分词合格率为 91%, 提取合格率为 87%, 基本满足需求.

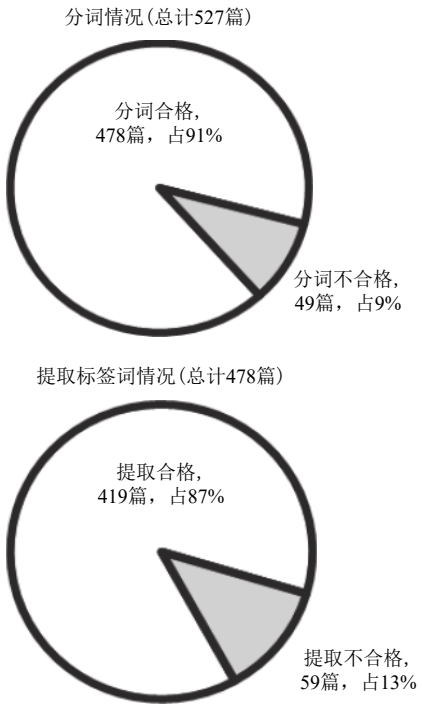

图 3 标签词提取测试情况

然后对所有文章进行标签处理. 以 3 个优先 (优先保留合格标签, 优先添加标题关键词, 优先保留标签与关键词的重叠词) 为标准结合关键词统计排名进行设置, 如图 4 所示.

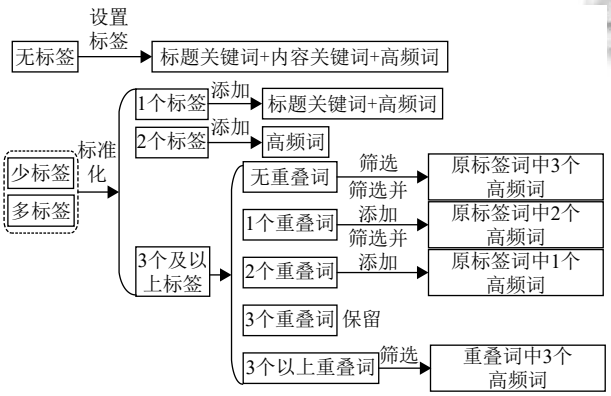

图 4 标签设置规则

3.3 推荐方法

3.3.1 实时推荐

文章标签规范化处理后, 每篇文章固定存在 3 个标签代表文章特性. 实时推荐模块采用基于内容数据的推荐方式进行推荐, 具体流程如图 5 所示.

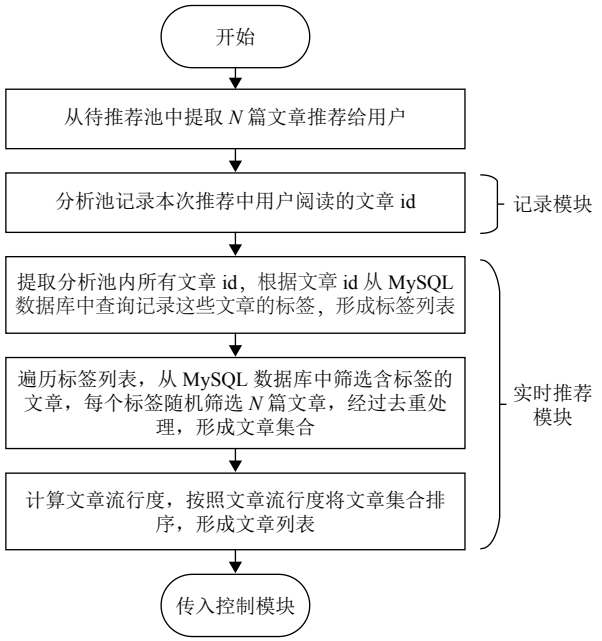

图 5 实时推荐流程

图 5 中文章流行度涉及到标签流行度[11]的概念, 标签流行度是指用户查看的所有文章中某标签出现的频率高低, 计算方法如式 (1) 所示

$$popularity_{(u,t_i)} = \frac{Freq_{(u,t_i)}}{\sum_{t_j \in T(u)} Freq_{(u,t_j)}} \tag{1}$$

其中, $T(u)$表示用户 u 阅读的文章所有标签集合, $Freq_{(u,t_i)}$ 表示用户 u 阅读所有文章中标签t_i的次数.

文章流行度是指用户对文章的偏好程度, 可以用文章标签流行度累加和表示, 如式 (2) 如下:

$$Pre_{popu}(u,r) = \sum_{t \in T(r)} popularity_{(u,t)} \tag{2}$$

其中, $T(r)$表示文章 r 的所有标签. 对于本文研究的微信文章, 该公式所计算的就是每篇文章中三个标签的流行度之和.

3.3.2 离线推荐

用户统计模块统计用户阅读的所有文章的 id、类型以及标签, 离线推荐模块根据这些历史数据采用基于用户行为数据的推荐方式进行推荐, 具体推荐流程如图 6 所示.

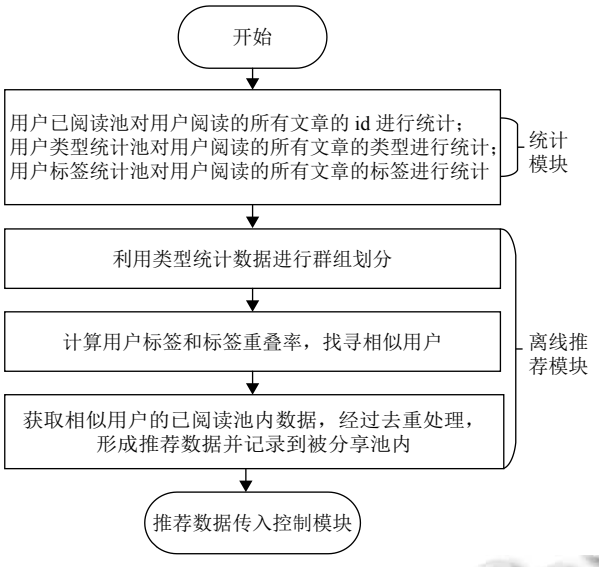

图 6 离线推荐流程

(1) 群组划分

用户统计模块中类型统计池对用户阅读的所有文章的类型进行统计, 即记录了用户阅读美文哲理、幽默笑话、趣味测试、经验总结、美食美景及保健养生 6 类文章的文章数. 将该数据记为一个代表用户阅读类型偏好的六维向量, 计算两用户间余弦相似度, 余弦值越接近 1 表示越相似, 越接近 0 表示越不相似. 计算方法如式 (3) 所示, 其中x_i,y_i分别表示用户 x 与用户 y 在统计模块中类型统计池内 i 类型文章的统计数.

$$\cos\theta = \frac{\sum_{i=1}^{n}(x_i * y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} * \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{3}$$

在用户群体中随机抽取一名用户, 计算该用户与剩余所有用户的余弦相似度, 取阀值为 0.8, 将余弦相似度大于等于 0.8 的用户以及该用户划分为一个群组从用户群体中剥离. 在余下的用户群体中再随机抽取一名用户, 如上述方法计算, 直至将用户群体完全划分.

(2) 寻找相似用户

用户标签是指用户经常看的文章标签集合, 以该数据代表用户阅读主题偏好. 统计模块中标签统计池对用户阅读的所有文章的标签进行了统计, 利用这些数据可以计算用户标签, 具体方法为: 去除标签统计池内用户只阅读过一次的标签, 然后将剩余标签根据标签流行度 (式 (1)) 排序, 排名前 80% 的标签集合作为用户标签. 通过计算用户间标签重叠率, 重叠率越接近 1 越相似, 计算方法如式 (4) 所示, 其中$labels_i$表示用户 i 的标签, $overlap_{(i,j)}$表示用户 i 与用户 j 的标签重叠率:

$$overlap_{(i,j)} = \frac{num(labels_i \cap labels_j)}{num(labels_i)} \tag{4}$$

群组划分后, 对于每个群组, 遍历群组成员, 计算该成员与其他成员的标签重叠率, 取阀值 0.5, 重叠率大约等于 0.5 的成员即为该成员的最终相似用户.

(3) 被分享池

将这些相似用户的已阅读池内的数据, 经过去重处理, 记录到本用户的被分享池内. 被分享池内的文章数据即为离线推荐数据.

3.3.3 整体推荐策略

本推荐系统中实时推荐在每次用户获取数据时运行一次, 运行次数取决于用户的获取次数, 而离线推荐每天定时执行一次, 所产生的推荐数据存储在用户的被分享池内.

控制模块一方面对传递过来的实时/离线推荐数据进行去重处理, 包括与用户的历史数据比较去重, 与待推荐池内数据比较去重, 实时与离线推荐数据比较去重. 另一方面, 监控待推荐池内数据量变化, 以加权及具体场景具体分析的方式组合两类推荐数据补充待推荐池, 具体组合方式如表 2 所示. 当需要增添其他推荐方式时, 通过对该模块进行设置, 结合这些推荐算法的推荐结果, 从而提高了系统的可伸缩能力, 如图 7 所示.

3.4 结果分析

本文设计的推荐系统, 其目的主要在于让推荐的数据能够贴合用户的偏好, 可以通过对待推荐池内数据进行分析, 考查本系统推荐效果.

以 10 788 篇微信文章进行推荐测试, 设置用户待推荐池容积为 100, 每次推荐, 提取池内 10 篇文章数据给用户. 系统模拟 100 名用户, 每名用户进行了 100 次推荐操作, 为贴近实际情况, 保证模拟效果, 所模拟的用户一方面需要具有一定的兴趣偏好, 另一方面还能按照一定概率阅读不在该偏好内的其他文章. 因此, 对模拟用户做如下设置: ① 每名用户随机具有 1~3 个感兴趣的文章类型. ② 每次推荐, 用户会随机阅读 1~3 篇偏好类型的文章. ③ 每次推荐, 用户有 20% 的概率阅读 1 篇不在偏好类型之内的其他类型的文章. 推荐过程中, 前 50 次只进行实时推荐, 第 50 次以后加入离线推荐, 每进行 25 次推荐, 统计所有用户当前待推荐池内, 含有 2 个及以上用户标签的文章所占比例, 计算最大值、最小值、平均值、两平均值总和以及该总和的增长比例. 实验结果如表 3 所示.

表 2 两类推荐数据组合方式

场景	判断方式		两类推荐方法实际推荐数
无实时推荐数据时	$M1=0$		$n1=0$, $n2=\min(M2, N)$
实时推荐数据少于等于需补充数据量的 80%	$0<M1<=0.8*N$		$n1=M1$, $n2=\min(M2, N–M1)$
实时推荐数据大于需补充数据量的 80%，而离线推荐数据小于补充数据量的 20%	$0.8*N<M1$	$M2<0.2*N$	$n1=\min(M1, N–M2)$, $n2=M2$
实时推荐数据大于需补充数据量的 80%，而离线推荐数据大于等于补充数据量的 20%		$M2>=0.2*N$	$n1=0.8*N$, $n2=0.2*N$

注: 设待推荐池需要补充数为 N, 实时推荐可推荐数 $M1$, 实际推荐数 $n1$, 离线推荐可推荐数 (用户被分享池容量)$M2$, 实际推荐数 $n2$

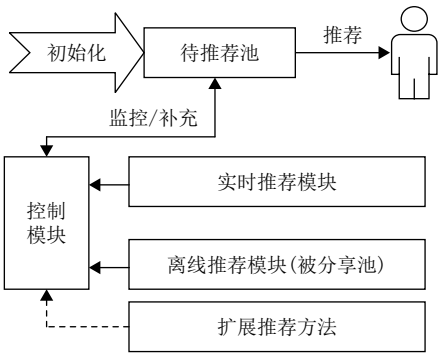

图 7 整体推荐流程

含有 2 个及以上用户标签的文章可以认为是用户感兴趣的文章. 初始状态下, 待推荐池内含 2 个及以上用户标签的文章所占比例为 0. 根据表 3 所示, 经过 25 次推荐, 占比约占 22.97%, 之后每 25 次推荐, 达到 44.39%, 50.75%, 53.48%, 分别增长了 93.25%, 14.33%, 5.38%. 经过 100 次推荐后, 含用户标签的文章总计约占 91.76%, 其中能够代表用户感兴趣的文章 (含有 2 个及以上用户标签的文章) 约占 53.48%, 只含有 1 个用户标签的文章约占 38.28%, 如表 4 所示. 从池内文章占比变化可以看出, 待推荐池内文章能够逐渐贴近用户偏好.

表 3 待推荐池内含 2 个及以上用户标签的文章占比变化表

推荐次数	第 25 次推荐		第 50 次推荐		第 75 次推荐		第 100 次推荐	
文章占比	文章中含有用户标签数							
	2 个	3 个	2 个	3 个	2 个	3 个	2 个	3 个
最小值	0	0	0.11	0	0.21	0	0.25	0.01
最大值	0.39	0.12	0.59	0.17	0.61	0.24	0.62	0.3
平均值	0.2015	0.0282	0.3807	0.0632	0.4092	0.0983	0.4025	0.1323
总计	0.2297		0.4439		0.5075		0.5348	
增长比例	-		+93.25%		+14.33%		+5.38%	

表 4 待推荐池最终各类文章占比

	不含用户标签	含 1 个用户标签	含 2 个用户标签	含 3 个用户标签
最小值	0	0.17	0.25	0.01
最大值	0.27	0.64	0.62	0.3
平均值	0.0824	0.3828	0.4025	0.1323

4 总结

本文设计了一种结合实时推荐与离线推荐的推荐系统, 能够保证系统运行流畅、具备可伸缩能力以及能够适应用户长期偏好及短期兴趣焦点变化. 基于该系统, 实现了对于微信文章的推荐实验, 采用易理解、可操作、效果可见的推荐方式, 在实验中, 推荐方式具备如下特点: ① 用户历史数据稀疏不影响推荐系统运行; ② 只采集了用户阅读行为, 无评分机制; ③ 系统具备可伸缩能力, 能够根据各推荐模块产生的推荐数据量, 调节两类推荐数据比例, 并且可以增添其他推荐方式; ④ 保证了推荐系统始终运行流畅. 此外, 对于推荐系统评价方面, 本系统通过对待推荐池内数据分析来对本系统推荐效果进行评价, 实验表明待推荐池内数据能够逐步贴近用户兴趣偏好.

参考文献

1 Resnick P, Varian HR. Recommender systems. Communications of the ACM, 1997, 40(3): 56–58. [doi: 10.1145/245108.245121]

2 朱扬勇, 孙婧. 推荐系统研究进展. 计算机科学与探索, 2015, 9(5): 513–525. [doi: 10.3778/j.issn.1673-9418.1412023]

3 王国霞, 刘贺平. 个性化推荐系统综述. 计算机工程与应用, 2012, 48(7): 66–76. [doi: 10.3778/j.issn.1002-8331.2012.

07.018]
4 李媛媛. 存在社会影响的群体推荐用户建模研究[硕士学位论文]. 天津: 天津大学, 2014.
5 赵鹏, 蔡庆生, 王清毅. 一种用于文章推荐系统中的用户模型表示方法. 计算机技术与发展, 2007, 17(1): 4–5, 48. [doi: 10.3969/j.issn.1673-629X.2007.01.002]
6 雷曼, 龚琴, 王纪超, 等. 基于标签权重的协同过滤推荐算法. 计算机应用, 2019, 39(3): 634–638. [doi: 10.11772/j.issn.1001-9081.2018071521]
7 尹祎, 冯丹, 施展. 一种基于效用的个性化文章推荐方法. 计算机学报, 2017, 40(12): 2797–2811. [doi: 10.11897/SP.J.1016.2017.02797]
8 Pazzani MJ. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 1999, 13(5–6): 393–408. [doi: 10.1023/a:1006544522159]
9 Zhang YC, Blattner M, Yu YK. Heat conduction process on community networks as a recommendation model. Physical Review Letters, 2007, 99(15): 154301. [doi: 10.1103/PhysRevLett.99.154301]
10 李挺. 基于 iOS 平台科技新闻推荐系统的设计与实现[硕士学位论文]. 大连: 大连理工大学, 2015.
11 张鹏飞, 王宜贵, 张志军. 融合标签和多元信息的个性化推荐算法研究. 计算机工程与应用, 2019, 55(5): 159–165. [doi: 10.3778/j.issn.1002-8331.1711-0330]