第 34 卷第 12 期 系统工程理论与实践 Vol.34, No.12
2014 年 12 月 Systems Engineering — Theory & Practice Dec., 2014

文章编号: 1000-6788(2014)12-3238-12 中图分类号: TP18 文献标志码: A

协同过滤推荐算法研究: 考虑在线评论情感倾向

王 伟, 王洪伟, 孟 园

(同济大学 经济与管理学院, 上海 200092)

摘 要 协同过滤推荐算法通常是基于兴趣相似的用户行为来实现个性化推荐, 其核心是定义用户之间的兴趣相似度. 本文在传统的协同过滤推荐算法基础上, 考虑在线评论对用户相似度识别的影响. 在混合商品推荐中, 粗粒度识别评论情感极性; 而在同类商品推荐中, 细粒度识别每个商品特征的情感极性. 如果用户对产品的某个特征评价次数大于平均次数, 表明用户对该特征较关注; 如果对产品的某个特征评价低于平均评价, 表明用户对该特征较挑剔. 进而根据用户评论来建立用户偏好模型, 用户在评论中反映出来的相似度越高, 表明用户之间的偏好越一致. 实验表明, 同传统的协同过滤算法相比, 基于在线评论情感分析的用户协同过滤算法在准确率和召回率指标上有显著提升.

关键词 推荐系统; 推荐算法; 协同过滤; 在线评论; 情感分析

The collaborative filtering recommendation based on sentiment analysis of online reviews

WANG Wei, WANG Hong-wei, MENG Yuan

(School of Economics and Management, Tongji University, Shanghai 200092, China)

Abstract Collaborative filtering recommendation algorithm bases on user behavior with similar interests to produce personalized recommendation. The core of the algorithm is to define the distance between the user's interest similarities. The paper considers the online review sentiment impact on user similarity recognition. In mixed products recommendation, coarse-grained sentimental polarity is identified; while in same category products recommendation, fine-grained sentimental analysis is employed for each feature. If the user's evaluation frequency is greater than the average on a special feature, it indicates that the user pays close attention to the feature; while if the user's rating is smaller than the average rating on a special feature, it means the user has a strict requirement on this feature. And then the user's preference model is created according to reviews, the higher the similarity between users in the reviews, the more consistent preferences between users. Experiment results show that the proposed collaborative filtering algorithm based on sentiment analysis of online reviews improves the traditional algorithm significantly on accuracy and recall.

Keywords recommendation system; recommendation algorithm; collaborative filtering; online review; sentiment analysis

1 引言

互联网环境下, 信息的过度丰富对电子商务带来挑战: 用户为了获取真正需要的商品信息, 花费的成本越来越高, 已经超出可容忍的范围 [1]. 个性化推荐系统的出现, 为用户提供了解决信息过载的工具. 这类系统通常是基于兴趣相似的用户行为, 通过定义用户之间的距离 (N- 范数, 内积等), 实现个性化的推荐. 在线

收稿日期: 2013-09-02
资助项目: 国家自然科学基金 (70971099, 71371144); 上海市哲学社会科学规划课题一般项目 (2013BGL004); 中央高校基本科研业务费专项资金 (1200219198)
作者简介: 王伟 (1982–), 男, 汉, 重庆人, 博士研究生, 研究方向: 商务智能与情感计算, E-mail: wayswang@gmail.com; 通讯作者: 王洪伟 (1973–), 男, 汉, 辽宁人, 副教授, 博士生导师, 博士, 研究方向: 商务智能与情感计算, E-mail: hwwang@tongji.edu.cn; 孟园 (1982–), 女, 汉, 湖北人, 博士研究生, 研究方向: 电子商务与商务智能, E-mail: nancymeng5544@163.com.

评论 (online reviews) 是指用户通过互联网以文字和数字形式对商品进行正向或负向的评价, 是洞察用户偏好的重要信息来源. 作为网络口碑的重要形式, 在线评论可以通过互联网传播, 促进消费者之间的交流.

实践表明, 协同过滤推荐算法在一定范围内比较有效. 然而, 由于忽略用户对商品的评论, 因此影响了推荐效果. 换句话说, 如果用户在评论中对某商品强烈不满, 那么推荐算法就不应该推荐该产品给相似用户; 反之亦然. 更进一步, 推荐系统通过分析历史评论, 获取不同用户对产品特征的关注度, 进而改进推荐效果. 假设从用户 A 和 B 的历史评论中, 得知 A 和 B 对商品的颜色都非常在意. A 购买手机后对该款手机总体评价较好, 但对颜色相当不满, 这时由于 B 也对颜色非常重视, 所以应该慎重推荐该手机给 B. 同时, 如果 B 对手机颜色的评价低于手机颜色的平均评价得分, 这表明 B 对手机颜色较挑剔, 给 B 推荐产品时需要结合考虑商品的颜色特征. 可见, 融合了细粒度在线评论情感分析的推荐算法可能会提高协同推荐系统的效率.

2 文献综述

2.1 在线评论情感分析的研究

情感分析 (sentiment analysis) 是利用文本挖掘技术, 对在线评论进行语义分析, 旨在识别用户的情感趋向是 "高兴" 还是 "伤悲", 或判断用户的观点是 "赞同" 还是 "反对". 情感分析涉及多个研究问题. 例如文本的主客观检测 [2−3]; 不同粒度的情感极性及强度分析 [4−6]; 产品特征观点对提取以及产品特征评论与评论的情感合成关系分析等 [7−9].

关于情感分析, 一般分为粗粒度的对整个文本的分析以及细粒度的针对词的分析 [10]. 粗粒度的文本分析包含无监督的机器学习方法 [5,11−12], 半监督学习算法 [13] 以及监督学习算法 [14−15]. 细粒度的情感分析是词语级的分析, 首先计算词典中词语的原子极性, 并根据该原子极性计算词语之间的相关性得出每个评价词的情感, 再综合词语的情感得出句子的情感极性和强度, 最后根据句子情感计算文本情感 [16]. 最近的研究提出了一些综合多种方法的情感分析算法, 例如: MSA-COSRs[17], LET[18], PREF[19], SO-LSA[20] 等. 这些情感分析算法采用了不同模型, 适用于不同应用领域. 在国内研究中, 有研究者把情感分为 4 种 (愤怒, 高兴, 悲伤和中立), 并采用 SVM 模型进行情感识别 [21].

产品特征提取是进行细粒度评论分析的基础, 现有研究已经提出了一些特征挖掘算法, 例如基于领域本体的方法 [22], JMTS 特征提取算法 [23]. 产品特征提取往往和情感分析联系在一起, 同时识别某一具体特征的情感倾向 [24], 处理后的结果是特征观点对形式. 不同类别的产品由于其特征不一致, 进行特征提取的算法也不尽相同, 例如酒店评论情感分析, 由于酒店特征中会有设施, 环境等特征, 这显然与电子产品的特征不一样, 因而需要专门提取不同领域产品的特征项 [25]. 产品特征提取常常会出现高维特征项, 不便于处理, 一些研究提出了各种不同的特征项降维方法, 例如采用最大熵模型降维方法 [26], Aprior 降维方法 [27], 词语相似度降维方法 [28].

也有一些研究专注于情感的强度研究, 有些词虽然表达相同的情感极性, 但是强度是不一样的. 例如 "好" 和 "优秀", 虽然情感极性都是正面的, 但是后者比前者的强度大得多, 一些研究专门探讨了词语的情感强度识别 [16].

2.2 协同过滤算法研究

基于协同过滤算法 [29] 的推荐系统在电子商务领域得到了广泛应用. 协同过滤算法基于关联挖掘算法中的 "支持度 - 置信度" 思想. 考虑用户的历史购买行为, 计算用户之间的兴趣相似度, 然后推荐相似用户也有购买行为的产品. 电子商务中协同过滤算法分为基于产品的协同过滤算法和基于用户的协同过滤算法. 基于用户的协同过滤算法需要先识别用户偏好 [30], 用户偏好应用于计算用户偏好相似度, 用户偏好相似度再应用于推荐算法. 用户偏好是各种各样的, 可以据此发现热门商品, 热门信息, 甚至用于新闻推荐, 行为监控 [31] 等.

大多数用户一般只对少数商品有购买行为, 因此协同过滤算法存在数据稀疏问题, 数据稀疏问题常常导致推荐系统效率非常低 [32−33]. 解决这一问题的办法是对数据进行聚类, 但是单纯的聚类会降低算法精确度, 因此有研究者提出一种基于情境聚类和用户评级的协同过滤模型, 取得了不错的效果 [34]. 另一个思路是基于主成分分析 (PCA, principle component analysis) 和 SOM (self-organizing map) 聚类的混合协同过滤模型 [35].

2.3 情感分析在推荐系统中的应用研究

情感分析在协同过滤推荐算法中的研究刚刚起步. 研究发现, 关系结构会对推荐系统的效率产生影响 [36].

冷启动问题是推荐系统的一个常见问题,量化社会关系为冷启动问题提供了解决方法[37],这种量化的社会关系涉及年龄,性别,职业等.有研究者提出了一种基于 n 序访问解析逻辑的冷启动消除方法,即首先通过 Web 日志来获取用户访问项序,进而定义 n 序访问解析逻辑将其分解为用户访问子序集来解决冷启动问题[38].基于话题模型的协同过滤推荐算法能够更好地显示用户偏好,因为文字评论可以对产品的各个方面进行详细的评价,而数字评论只有一个综合得分,欠缺足够的参考价值[33].同时考虑用户评论特征的推荐算法,对于用户相似度的识别更加有效[32].这种方法可以用于购买同类产品的用户相似度计算和推荐,但是在电子商务中产品类别繁多,各种类别产品的特征千差万别,例如手机有"通话质量"这一特征,而硬盘有"转速"这一特征,这两个特征是不能直接比较的.用户评论中包含了大量有用的意见和情感信息,利用自然语言处理技术和模糊计算构建基于消费者在线评论的产品模糊推荐系统,可以引导用户做出更加正确的购物决策[39].产品涉及多个特征维度,有研究者从信息粒度和信息来源的角度研究了商品的属性知识和用户对商品属性的偏好信息在推荐系统中的应用,该方法在一定程度上提升了矩阵中数据元素的密度,进而解决数据稀疏和冷启动问题[40].

热门商品会影响推荐系统的多样性和准确性[41],推荐系统算法应当把商品的热门程度考虑进去,因为产品越热门,购买的用户越多,用户之间的相似性越不可靠.降低热门商品的权重有助于提高推荐系统的效率.用户可以通过评论对商品打分,但是该打分可能出现偏差,即噪音或者偏见[42].有些电子商务网站提供对评论的有用性评价,"赞"或者"踩",这种互荐的机制可以纠正用户偏好识别误差,提高正确率.有研究者根据这类用户与系统之间的交互,结合情境因素对顾客消费的影响,构建推荐规则[43].

综上所述,在线评论的特征提取,情感极性及强度分析已有较多的研究,方法也较为成熟.协同过滤算法通常根据用户的购买行为识别用户偏好和相似度,目前也有诸多改进算法.但是,基于在线评论情感分析的协同过滤算法研究较少.特别是,细粒度的在线评论情感分析更有助于识别用户的满意度,偏好和相似度.此外,有关推荐系统的研究,还没有区分同类产品和混合产品推荐.由于二者的特征不同,采用不同的算法显然更加科学.为此,本文把推荐系统分为同类产品推荐和混合产品推荐.对于同类产品推荐,采用细粒度的在线评论情感分析技术来识别和量化用户对商品特征的兴趣度;而对于混合商品推荐,则采用粗粒度的在线评论情感分析技术.

3 算法描述

3.1 在线评论抓取

使用自己编写的爬虫程序抓取在线评论.一些网站的在线评论采用 Ajax 方式,需要采用特殊的爬虫策略.爬虫的技巧包括修改 http 头信息,字符编码转化,json 格式解析等.抓取的在线评论保存在数据库中.常见的爬虫策略有深度优先策略和广度优先策略.为了提高爬虫效率,采取有限深度的深度优先策略,即设置爬行深度为 2.抓取的评论结构如下:

产品 ID: 812683
评论者 ID: 94567192
评论 ID: 17894435
商品类别: 笔记本电脑
商品名称: ThinkPad E430(3254-C18)14 英寸笔记本电脑
评论日期: 2013-4-3 11:21:40
总体评分: 5
评价内容: 觉得还是蛮值得的,性能稳定,散热好! 外观漂亮,就是声音不好……. ……

抓取到的评论记为 $D=\{D_1,D_2,\cdots,D_n\}$,其中 D_i 表示第 i 条评论的内容.

3.2 在线评论的预处理

根据标点符号对评论语料进行分句.我们观察到,评论中标点符号的使用极不规范,甚至有些用户根本不使用标点符号,仅仅使用空格来分隔分句.因此,采用以下 8 个标点符号作为分割句子的标准:" ","。","."," ",",","! ",";","?".分割后的分句称为子句.

任何一条评论文本可能由客观子句和主观子句构成,客观子句对于评论的情感表达没有影响,本文只关

注主观子句. 基于文献 [44] 的方法, 识别主观子句和客观子句, 并在语料中剔除客观子句, 仅保留主观子句.

剔除不一致的评论. 有些评论的文字内容是正面的, 但是星级评分是 1 星; 而有些评论的文字内容是负面的, 但星级评分是 5 星. 需要剔除这类自相矛盾的评论, 因为这类评论的存在会极大影响算法的有效性.

对这类噪音评论的处理, 实际上属于粗粒度的情感分析. 文本的情感计算采用文献 [12] 的方法. 判断文本的情感极性与用户的打分比较, 一星和二星的评论视为负面评论, 四星和五星的评论视为正面评论. 然后对比该计算结果和用户打分, 剔除不一致的评论.

3.3 在线评论的细粒度情感分析

使用 SCWS(simple Chinese word segmentation) 分词系统 [45] 对评论语料进行分词和词性标注, 该分词系统在中文分词中具有良好的准确性. 它是一套开放源代码软件, 且提供 API 接口. 词性标注集采用北京大学汉语词性标记集 (共 39 个词性标注 [46]).

提取产品特征 - 观点对. 例如, 对于某款相机的评论, 特征 - 观点对形式如下: pair("照片", "清晰"), pair("相片", "大气"), pair("像片", "鲜艳") ……. 已有研究发现, 产品特征通常为名词或者名词短语 [27,47]. 使用上一步已经标注的词性, 再结合文献 [44] 的方法, 提取出产品特征.

研究表明, 绝大部分情感词汇都是形容词或者副词 [23]. 因此, 提取距离特征词最近的形容词和副词作为观点候选词汇. 由于本文只需计算情感极性而不用关注情感强度, 对于有 2 个或 2 个以上的观点候选词的情况, 只需判断特征词与观点词距离. 按照公式 (1) 选择观点词.

$$l_i = \begin{cases} \max\ distance\,(f_i, a_i)\,, \text{where} f_i \rightarrow a_i \\ \min\ distance\,(f_i, a_i)\,, \text{where} f_i \leftarrow a_i \\ \max\ distance\,(f_i, a_i)\,, \text{where} f_i \leftrightarrow a_i \end{cases} \quad ,\forall a \in \{candidate\ opinions\} \tag{1}$$

根据词之间的依存关系 [48], 当特征词出现在候选观点词前面时, 选择距离最远的观点词; 而当特征词出现在候选观点词后面时, 选择距离最近的观点词; 当观点词同时出现在特征词前面和后面时, 选择距离最远的观点词. 考虑下面 3 个例子:

1. 非常新潮时尚的款式.
2. 布料非常时尚新潮.
3. 新潮的样式相当时尚.

以上 3 个例子的产品特征分别是 "款式", "布料" 和 "样式". 第 1 个例子的候选观点词是 {"非常", "新潮", "时尚"}, 由于观点词在特征词前面, 故选择距特征词最近的 "时尚" 作为观点词; 而在第 2 个例子中候选观点词仍然是 {"非常", "时尚", "新潮"}, 由于观点词在特征词后面, 故选择距特征词最远的 "新潮" 作为观点词; 在第 3 个例子中候选观点词是 {"新潮", "相当", "时尚"}, 由于特征词前后都出现了观点词, 故选择距特征词最远的 "时尚" 作为观点词. 这在汉语语法中是符合逻辑的, 日常语言中, 最后出现的修饰词往往是最重要的修饰词.

对于否定副词, 只要取其观点词的相反极性即可.

3.4 合并产品特征

在线评论时常使用不同的词汇描述相同的产品特征. 如果不对这类产品特征进行合并, 分析结果将会出现极大的偏差且不便于理解. 例如关于数码相机评论中, "照片", "相片", "像片" 描述的都是同一产品特征, 应该合并为一个特征. 首先计算产品特征的汉语相似度 [27], 并据此来合并产品特征, 见公式 (2).

$$Sim\,(w_1, w_2) = \frac{\alpha}{Dis\,(w_1, w_2) + \alpha} \tag{2}$$

其中, $Dis(w_1, w_2)$ 表示 2 个词 w_1, w_2 的距离, 如果 w_1, w_2 是义原词, 则 $Dis(w_1, w_2)$ 代表义原相似度; α 是可调节参数, α 的含义是当相似度为 0.5 时的词语距离值. 当相似度 $Sim(w_1, w_2)$ 大于阈值时, 合并 2 个产品特征. 以 3.3 节的例子为例 ($\alpha = 1.6$, 默认值):

Sim("款式", "样式")=0.927778, Sim("款式", "布料")=0.042904.

令相似度阈值为 0.85, 则 "样式" 和 "款式" 应该合并为一个特征; 而 "款式" 和 "布料" 不应该合并.

特征合并后, 输出格式为 D_1={("f_1", "$-1|0|1$"), ("f_2", "$-1|0|1$"), ⋯, ("f_n", "$-1|0|1$")}. 其中 f_i 表示第 i 个产品特征, $-1|0|1$ 分别表示负面情感, 中性情感和正面情感. 在 3.1 节的例子, 进行细粒度情感分析的结果是:

D_1={("性能","1"),("散热","1"),("外观","1"),("声音","-1")}.

3.5 混合商品用户兴趣相似度计算

当用户 A 需要个性化推荐时, 通常先找到与 A 兴趣相似的用户, 然后把其他用户喜欢的而 A 没有购买过的产品推荐给 A. 这涉及 2 个问题: 1) 寻找与目标用户兴趣相似的用户集合; 2) 寻找该集合用户喜欢的且目标用户没有购买过的物品.

如果考虑用户的评论, 用户兴趣度的识别需要过滤用户给出负面评价的商品而保留正面评价的商品. 通过余弦相似度, 计算用户的相似度, 如公式 (3). 其中, $N^+(u)$ 表示用户 u 曾经有过正反馈的商品集合, $N^+(v)$ 表示用户 v 曾经有过正反馈的商品集合.

$$W_{uv} = \frac{|N^+(u) \cap N^+(v)|}{\sqrt{|N^+(u)|\,|N^+(v)|}} \tag{3}$$

对于不同类别的商品而言, 粗粒度情感分析是合适的, 因为对不同类别的商品区分细粒度的评价是没有意义的. 例如: 对笔记本电脑和饭店而言, 二者特征完全不一样, 对于这类商品的协同推荐, 只需要识别整体评论是正面的还是负面的就可以了.

3.6 同类商品用户兴趣相似度计算

对于同类商品推荐, 为了有效识别用户相似度, 采用细粒度的评论分析. 令 $\overline{R}_i$ 表示对某类商品的特征 f_i 的总平均评价次数, $\overline{G}_i$ 表示对某类商品的特征 f_i 的总平均好评率; $R_i(u)$ 表示用户 u 对某类商品的特征 f_i 的平均评价次数, $G_i(u)$ 表示用户 u 对某类商品的特征 f_i 的平均好评率. 采用如下两条规则来识别同类商品的用户相似度.

1) 若用户 u 对商品特征 f_i 的平均评价次数高于该特征的总平均评价次数, 即 $R_i(u) > \overline{R}_i$, 则该用户对特征 f_i 的关注度大于大部分用户.

2) 若用户 u 对商品特征 f_i 的平均好评率低于该特征的总平均好评率, 即 $G_i(u) < \overline{G}_i$, 则该用户对于特征 f_i 的要求高于大部分用户, 换句话说, 该用户对特征 f_i 比较挑剔.

公式 (4) 计算了用户 u 对商品特征 f_i 的关注度.

$$Concern\,(u, f_i) = \frac{count\,(u, f_i)}{count\,(u)+1} \times \frac{N}{count\,(f_i)+1} \tag{4}$$

$Concern(u, f_i)$ 为用户 u 对特征 f_i 的关注度; $count(u, f_i)$ 为用户 u 对商品特征 f_i 的评论次数; $count(u)$ 为用户总评论数量; N 是商品获得的总评论数; $count(f_i)$ 代表特征 f_i 获得的评论次数.

关于第二条规则, 令 $Nitpick(u, f_i)$ 为用户 u 对特征 f_i 的挑剔程度, 公式 (5) 为 $Nitpick(u, f_i)$ 的计算方法.

$$Nitpick\,(u, f_i) = \frac{\bar{G}_i - G_i\,(u)}{G_i\,(u)+1} \times \frac{N}{\bar{G}_i+1} \tag{5}$$

若 $\overline{G}_i > G_i(u)$, 特征 f_i 的总体评论好评率大于用户 u 所给的好评率, 此时 $Nitpick(u, f_i) > 0$; 当 $\overline{G}_i < G_i(u)$ 时, 特征 f_i 的总体评论好评率小于用户 u 所给的好评率, 此时 $Nitpick(u, f_i) < 0$; 而当 $\overline{G}_i = G_i(u)$ 时, 特征 f_i 的总体评论好评率等于用户 u 所给的好评率, 此时 $Nitpick(u, f_i) = 0$. 换句话说, 用户对某方面特征 f_i 越挑剔, $Nitpick(u, f_i)$ 的值越大, 反之亦然. 结合公式 (4) 和 (5), 计算出用户 u 对某方面特征的偏好度, 见公式 (6). $Preference(u, f_i)$ 代表用户 u 对产品特征 f_i 的偏好程度.

$$Preference\,(u, f_i) = Concern\,(u, f_i) \times Nitpick\,(u, f_i) \tag{6}$$

$Preference(u, f_i) = 0$ 时, 用户 u 对产品特征 f_i 是中性的, 即与大多数用户评价标准是一致的; $Preference(u, f_i) > 0$ 时, 表示用户 u 对产品特征 f_i 显得挑剔, 且 $Preference(u, f_i)$ 值越大, 用户 u 的要求越苛刻; $Preference(u, f_i) < 0$ 时, 表示用户 u 对产品特征 f_i 有容忍度, 且 $Preference(u, f_i)$ 绝对值越大, 用户 u 的容忍度也越大.

将用户 u 对商品 p 的偏好记为向量 $p_u = (p_{u1}, p_{u2}, p_{u3}, \cdots, p_{un})$, 其中 p_{ui} 表示用户 u 对第 i 个特征的偏好程度. 公式 (7) 计算用户特征相似度集合, W_{uv} 代表用户 u 和 v 的相似性.

$$W_{uv} = \frac{\sum_{i=1}^{n} |p_{ui} \times p_{vi}|}{\sum_{i=1}^{n} |p_{ui} - p_{vi}| + 1} \tag{7}$$

3.7 热门商品的处理

社会关系学的马太效应理论认为强者更强, 弱者更弱 [49]. 信息科学的相关研究也广泛存在马太效应, 例如电子商务热门商品 [50], 信息检索 [51], 推荐系统 [52] 等. 推荐系统如果会增大热门商品和非热门商品的流行

度差距, 那么可以认为该推荐系统存在马太效应. 热门商品具有高购买频度, 往往与其他商品一起购买. 一个产品的热度越大, 就越容易频繁出现在其他用户的推荐列表中, 因此该商品就会越来越流行; 反之, 不能达到一定热度的商品往往不能进入用户的推荐列表中, 流行度会越来越差, 导致该产品越来越冷门. 这种现象在协同过滤算法中普遍存在, 不过由于推荐算法可以控制列表, 因此通过适当地提高长尾商品的推荐频率, 可以提高推荐系统的覆盖率, 因此可以更好的发掘信息长尾 [53].

实践中, 人们发现在亚马逊网站的推荐系统中, 似乎所有的商品都与《哈利波特》相关 [54]. 也就是说, 购买任何商品的用户都会购买《哈利波特》, 这不是因为《哈利波特》与其他商品都有关, 而是因为《哈利波特》在热门商品排行榜中占据显著位置, 很多用户都会在购物时顺带买一本. 推荐系统存在大量类似案例, 如果 2 个用户都曾购买《大学英语》, 并不能认为他们兴趣相似, 因为绝大多数中国学生都要使用该书. 但是, 如果 2 个用户同时购买《统计自然语言处理》, 就可认为他们的兴趣相似, 因为通常只有研究自然语言处理的用户才会购买该书. 换句话说, 用户对越冷门的商品采取行为越能说明他们的兴趣相似度越高 [55]. 因此应降低用户 u 和 v 共同兴趣列表中热门物品对他们相似度权重的影响. 对推荐算法的实践证明, 基于用户的方法更擅长于热门推荐而基于项目的方法更擅长于长尾推荐 [56].

借助 TF-IDF 的思想, 对热门商品给予适度惩罚. 公式 (3) 可以改写为公式 (8).

$$W_{uv}=\frac{\sum\limits_{i\in N^+(u)\cap N^+(v)}\frac{1}{\log(1+|P^+(i)|)}}{\sqrt{|N^+(u)|\,|N^+(v)|}} \tag{8}$$

其中, $P^+(i)$ 是商品热门度, 表示用户 u 和 v 共同拥有的正反馈商品的集合.

同理, 对同类商品推荐而言, 在考虑商品热门度后, 公式 (7) 对热门商品降权后可以改写为公式 (9).

$$W_{uv}=\frac{\frac{\sum_{i=1}^{n}|p_{ui}+p_{vi}|}{\log\left(1+\sum_{i=1}^{n}|p_{ui}\times p_{vi}|\right)}}{\sum_{i=1}^{n}|p_{ui}-p_{vi}|+1} \tag{9}$$

其中, $\frac{1}{\log\left(1+\sum_{i=1}^{n}|p_{ui}\times p_{vi}|\right)}$ 是对热门商品的惩罚度.

3.8 算法优化

对于不同类别的产品, 采用商品用户倒查表解决计算上的时间复杂度问题 [55], 其时间复杂度由 $O(|\mathrm{U}|\times|\mathrm{U}|)$ 降低到 $O(|\mathrm{U}|)$.

对于同类商品, 可以借鉴商品用户倒查表的思路创建商品特征用户倒查表. 使用如下方法构建商品特征用户倒查表, 假设某个用户的评论分析输出为 D_1={(“f_1”,“−1|0|1”), (“f_2”, “−1|0|1”), $\cdots$, (“f_n”, “−1|0|1”)}, 为每个特征建立一个二维数组, 格式为 Array[feature][user]=[|positive|, |negative|, |neutral|], 其中 |positive|, |negative|, |neutral| 分别表示对该特征正面评论的次数, 负面评论的次数以及中立评论次数. 如果用户对某方面特征没有评价, 则认为用户对该特征是中性的.

使用以下例子解释: 假设用户 A 对产品 i 的评论输出为 D_{ai}={(“性能”,“1”), (“外观”,“1”)}, 用户 A 对产品 j 的评论输出为 D_{aj}={(“性能”,“0”), (“外观”,“1”), (“声音”,“−1”)}; 用户 B 对产品 i 评论输出为 D_{bi}={(“性能”,“1”), (“外观”,“0”), (“声音”,“−1”)}, 用户 B 对产品 j 评论输出为 D_{bj}={(“ 性能”,“1”), (“声音”,“0”)}; 用户 C 对产品 i 评论输出为 D_{ci}={(“外观”,“1”), (“声音”,“1”)} , 用户 C 对产品 j 评论输出为 D_{cj}={(“外观”,“1”), (“声音”,“1”)}. 则分别创建 “性能”, “外观” 和 “声音” 的二维数组, 然后依次扫描评论, 当评论情感特征值是 1 时, |positive| 加 1; 当评论情感特征值是 −1 时, |negative| 加 1; 当情感特征值是 0 时或者无该项特征情感时, |neutral| 保持不变. 优化处理过程如图 1 所示.

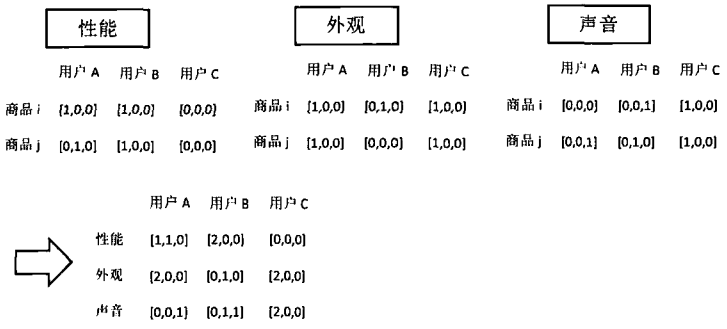

图 1　特征用户倒查表

使用优化算法前的时间复杂度为 $O(|\mathrm{U}|\times|\mathrm{U}|\times|f_n|\times|\ f_n|)=O\ (|\mathrm{U}|^2\times|f_n|^2)$, 使用上面的优化算法后, 时间复杂度降低为 $O(|\mathrm{U}|\times|f_n|)$. 以图 1 为例, 可以得到用户 A 与 B 特征相似度 W_{AB} 如下.

$$W_{AB}=\frac{|0\times(-\frac{4}{5})|+|(-\frac{2}{5})\times\frac{1}{5}|+|\frac{1}{2}\times 1|}{|0-(-\frac{4}{5})|+|(-\frac{2}{5})-\frac{1}{5}|+|\frac{1}{2}-1|+1}=41/175.$$

同理, 用户 A 与 C 的特征相似度 W_{AC}=221/1040, 用户 B 与 C 的特征相似度为 W_{BC}=221/1145.

3.9 用户对商品的兴趣度计算

得到用户之间的兴趣相似度之后, 推荐与该用户兴趣最相似的 K 个用户喜欢的商品. 公式 (10) 度量了协同过滤算法中用户 u 对商品 i 的兴趣度.

$$Interest\,(u,i)=\sum_{v\in S(u,K)\cap N(i)} W_{uv}\times r_{vi} \tag{10}$$

其中, $S(u, K)$ 是指和用户 u 兴趣最接近的 K 个用户集合, $N(i)$ 是对商品 i 有过购买行为的用户集合, W_{uv} 是指用户 u 和用户 v 的兴趣相似度, r_{vi} 代表用户 v 对商品 i 的兴趣权重, 由于本文只使用在线评论作为用户对商品评价的反馈数据, 故 r_{vi}=1. $Interest(u, i)$ 只有一个参数 K, 表示为每个用户选出 K 个和他兴趣最相似的用户, 然后推荐这 K 个用户感兴趣的商品. K 值并非越大越好, K 值过大可能会出现过分拟合问题, 应该在实践中调整 K 值.

4 实验设计

4.1 实验数据

本文把商品推荐算法分为同类商品推荐和混合推荐, 同类商品推荐算法需要识别和区分商品类别, 而混合推荐不区分商品类别. 之所以根据商品类别来区分, 是因为同类商品具有相同的产品特征, 便于识别和对比, 得到的特征偏好对于用户偏好模型的建立更加准确; 而混合推荐涉及不同的产品类别, 产品特征不一致, 因而不能通过特征比较建立用户偏好模型. 这两种算法在电子商务中都有广泛应用, 但是现有研究很少将二者区分开来. 图 2 和图 3 分别展示了混合推荐和同类产品推荐的例子.

图 2 混合推荐的例子

图 3 同类产品推荐的例子

实验采用开源服务器 Linux, Apache, MySQL 和 PHP 实现. 采用我们编写的爬虫程序抓取京东商城 (www.jd.com) 上的产品评论. 然后按照 3.2 节的方法对每条评论进行验证, 剔除不符合要求的评论. 经过上面处理后, 实验的数据是 8,485,656 条评论, 分别来自 1,289,935 个用户对 266,406 个产品的评价, 时间跨度上最早的评论是在 2007-12-14 16:17, 最晚的评论是在 2013-05-29 23:29. 平均每个用户发表 6.58 条评论, 平均每个产品获得 31.85 条评论.

采用真实数据离线实验的方法. 首先将数据按照均匀分布随机分成 M 份 (取 $M=8$), 挑选一份作为测试集, 将剩下的 $M-1$ 份作为训练集. 然后在训练集上建立用户的兴趣模型, 并在测试集上对用户行为进行预测, 统计出相应的评测指标. 为了保证评测指标的稳定性, 需要进行 M 次实验, 每次实验都采用不同的测试集. 最后将 M 次实验得到的评测指标的平均值作为最终的评测指标.

本文采用随机推荐算法作为比较. 该算法是指从用户没有历史购买行为的商品中随机取出 N 个商品推荐给用户, 不考虑产品类别, 也不考虑权重.

图 4 显示了本文实验的流程.

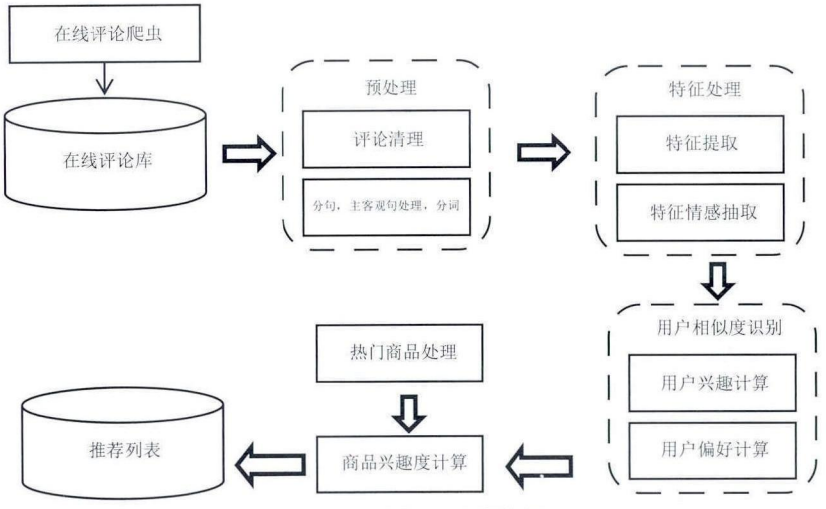

图 4　实验流程

算法 1 显示了程序处理的具体步骤的伪码.

4.2 同类商品推荐评测

选取书籍作为同类商品推荐的类别, 因为书籍属于持续购买的产品, 购买行为频繁.

首先测试在不同 K 值下算法的性能指标, 见表 1. 可以看到, 当 K=120 时, 准确率和召回率最高, 覆盖率和流行度也在可接受的范围内. 因此, 本算法进行同类产品推荐时, 选择 K=120.

对比考虑商品流行度与不考虑商品流行度的算法, 如表 2 所示. 可见, 考虑商品流行度算法时, 以牺牲准确率和召回率为代价, 覆盖率和流行度有一定程度的提升, 证明了本文对于商品流行度处理的有效性.

表 3 显示了几种不同的推荐算法的性能比较. Random 算法的准确率和召回率都非常低, 但是覆盖率相当高. 基准方法是文献 [29] 的算法, 基准算法准确率和召回率远高于 Random 算法, 但覆盖率非常低, 且推荐出的结果非常热门. 本文算法的准确率, 召回率和覆盖率显著提高, 且商品流行度降低.

表 1　同类商品推荐中不同 K 值下的推荐性能

K	准确率	召回率	覆盖率	流行度
10	9.42%	5.78%	26.55%	26.67
20	10.34%	6.14%	24.78%	27.38
50	12.68%	6.87%	22.27%	30.81
80	14.07%	7.32%	20.99%	32.04
100	15.12%	7.91%	19.78%	34.95
120	**15.87%**	**8.26%**	**19.25%**	**35.32**
200	14.76%	8.02%	18.76%	36.82

表 2　同类商品推荐中考虑商品流行度和不考虑商品流行度的性能对比

算法	准确率	召回率	覆盖率	流行度
不考虑流行度	17.21%	9.71%	15.70%	41.32
考虑流行度	15.87%	8.26%	19.25%	35.32

表 3　同类商品推荐中几种推荐算法的性能对比

算法	准确率	召回率	覆盖率	流行度
Random	2.01%	1.04%	100%	37.12
基准算法	13.21%	7.43%	13.97%	50.23
本文算法	15.87%	8.26%	19.25%	35.32

4.3 混合商品推荐评测

混合商品推荐不考虑产品类别, 系统总体性能评价见表 4. 可见在混合推荐中, 当 K=100 时, 可以获得最佳推荐效果, 这个数字比同类产品推荐中的 K 值小 (同类商品推荐中 K=120, 4.2 节), 这是因为混合商品推荐可以在较少的训练样本下获得较多的用户偏好信息, 而在同类产品中由于其购买相对稀疏, 因而需要较多的学习样本.

表 4　混合推荐中不同 K 值下的推荐性能

K	准确率	召回率	覆盖率	流行度
10	8.16%	5.03%	32.37%	30.21
20	9.94%	5.62%	28.89%	31.95
50	11.47%	6.13%	25.64%	32.63
80	12.32%	6.74%	23.37%	32.95
100	**13.82%**	**7.02%**	**21.79%**	**33.72**
120	12.17%	6.28%	20.14%	34.57

Algorithm 1 程序处理流程

输入:

从数据库中循环读取用户训练集 U_{train}=$\{u_1, u_2, \cdots, u_i\}$, 测试用户集 U_{test}=$\{u_1, u_2, \cdots, u_j\}$ 和评论集 D=$\{D_1, D_2, \cdots, D_n\}$, 文献 [28] 的相似度词汇表, 情感极性词汇表;

输出:

对测试集中用户推荐的商品列表以及推荐度;

for each review d in D **do**
  SplitedSentences = SplitSentence(d);
  for each sentence s in SplitedSentences **do**
    FeaturePhrases=ExtractFeature(s, *noun*);
    OpinionPhrases=ExtractOpinion(s, *adj*||*adv*);
    MergeFeatures (*feature*1, *feature*2); // if the similarity greater than threshold, then merge these two features.
  end for
end for
for each user u in U_{test} **do**
  Compute positive reviews number U^+;
  Define K=n as train sample number;
  Select K users from U_{train} where user in (select user from D where u); // select K training users form train data set for every test user in test data set.
  Compute K users' positive reviews number V^+;
  Compute user similarity W_{uv};
  Compute concern (u_i, f_j);
  Compute nitpick (u_i, f_j);
  Compute preference (u_i, f_j); // Compute preference (u_i, f_j)= concern (u_i, f_j)* nitpick (u_i, f_j).
end for
for each product from P where u in U_{test} **do**
  Compute *recommend* degree recommend(u, p);
  if recommend>threshold **then**
    Add to output *list*;
  end if
end for
Order the output *list* by recommend desc;
return *list*;

混合产品推荐的准确率与同类产品推荐的准确率并无可比性, 因为二者适用于不同环境和应用领域. 但总体来说, 对评论细粒度的分析能够提高推荐系统的性能.

在混合推荐中, 还分别研究了考虑产品流行度和不考虑产品流行度的算法性能. 研究表明商品的流行度是呈长尾分布的, 即只有少数商品销量非常大, 流行度非常高; 而大多数商品销售量并不大. 另一个值得注意的现象是, 用户的活跃度也是呈长尾分布的, 即只有少数用户特别活跃. 图 5 描述了本文数据中商品的流行度分布, 以及用户的活跃度分布. 可以看到二者呈幂函数分布, 但是用户活跃度的长尾度比商品流行度的长尾度小得多, 原因在于一个电子商务网站的商品数量相比用户数量来说小得多. 以本文数据为例, 本文用户数量是商品数量的 10 倍以上, 而且用户活跃度分布比商品流行度分布均匀得多.

表 5 显示了考虑商品流行度的算法与不考虑流行度的算法评测, 与同类商品类似, 在牺牲了一定准确率和召回率基础上, 显著提升了覆盖率和流行度指标.

最后, 选用随机推荐算法和基准协同过滤算法进行比较, 见表 6, 本文提出的算法在 4 个评价指标中均有提升.

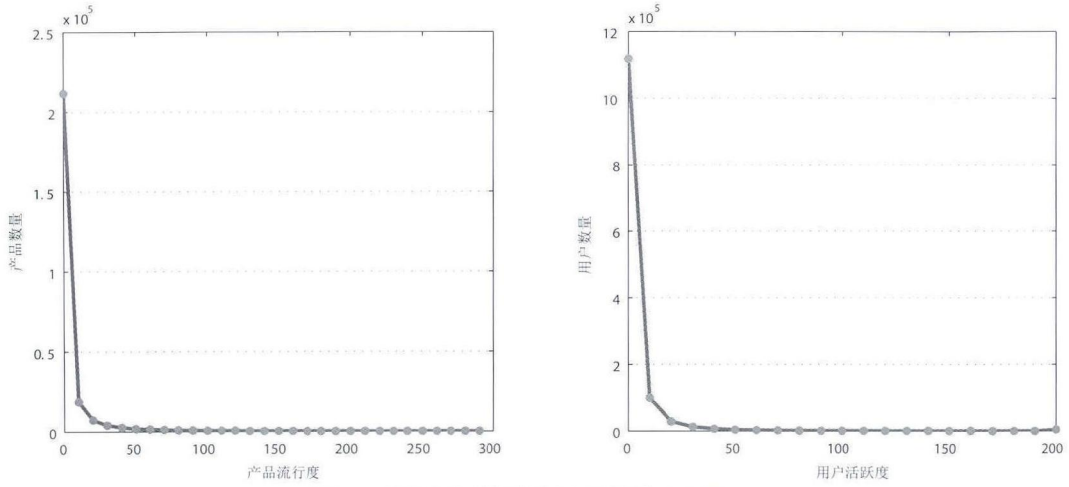

图 5 商品流行度以及用户活跃度分布图

表 5 混合推荐中考虑商品流行度和不考虑商品流行度的性能对比

算法	准确率	召回率	覆盖率	流行度
不考虑流行度	15.26%	8.72%	14.97%	43.79
考虑流行度	13.82%	7.02%	21.79%	33.72

表 6 混合推荐中几种推荐算法的性能对比

算法	准确率	召回率	覆盖率	流行度
Random	0.91%	0.53%	100%	36.31
基准算法	12.71%	6.42%	18.80%	45.38
本文算法	13.82%	7.02%	21.79%	33.72

5 结论和展望

将推荐系统分为同类产品推荐和混合推荐, 根据两种推荐系统的特征, 分别采用粗粒度和细粒度的在线评论情感分析的用户协同过滤算法. 根据用户在评论中表现出来的情感倾向来计算用户的偏好度和相似度, 进而推荐用户满意度高的产品给相似用户. 更进一步, 对于同类产品推荐, 考虑到用户对每个产品特征的评价, 根据用户表现出来的历史反馈, 计算用户对每个产品特征的关注度. 然后使用特征关注度来计算用户之间的相似性, 进而满足用户对不同产品特征的特殊需求. 对于热门商品, 采取了一些方法来降低热门商品的推荐率, 以更好的匹配用户偏好相似度. 实验表明, 本文的方法比传统的协同过滤算法有显著改进.

不可避免有一些不足, 未来的深入研究方向有: 1) 本文把推荐系统分为同类产品推荐和混合推荐, 实际上还有一类产品介于二者之间, 例如手机与平板电脑, 二者共同具有屏幕, CPU 等特征, 还有一些不同的特征, 例如手机有 "通话质量" 特征. 对于这两类产品特征的细粒度识别和计算, 还有待深入研究; 2) 任何一个推荐系统都面临冷启动问题, 传统的解决方法是使用热门商品排行榜替代推荐系统, 抑或是使用人口统计学特征进行推荐. 我们可以考虑加入用户评论来解决冷启动问题; 3) 不少在线评论都提供标签功能, 结合用户标签和在线评论内容能够更加清晰准确地识别用户偏好度和相似度.

参考文献

[1] Rosenberg D. Early modern information overload[J]. Journal of the History of Ideas, 2003, 64(1): 1–9.

[2] Bruce R, Wiebe J. Recognizing subjectivity: A case study of manual tagging[J]. Natural Language Engineering, 1999, 5(2): 187–205.

[3] Wiebe J, Bruce R, Bell M, et al. A corpus study of evaluative and speculative language[C]// Proceedings of the Second SIGdial Workshop on Discourse and Dialogue-Volume 16. Association for Computational Linguistics, 2001: 1–10.

[4] Hu M, Liu B. Mining and summarizing customer reviews[C]// Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2004: 168–177.

[5] Turney P D. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews[C]// Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002: 417–424.

[6] Turney P D, Littman M L. Measuring praise and criticism: Inference of semantic orientation from association[J]. ACM Transactions on Information Systems (TOIS), 2003, 21(4): 315–346.

[7] Jindal N, Liu B. Identifying comparative sentences in text documents[C]// Proceedings of the 29th Annual

International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2006: 244–251.

[8] Liu J, Yao J, Wu G. Sentiment classification using information extraction technique[M]// Advances in Intelligent Data Analysis VI. Springer Berlin Heidelberg, 2005: 216–227.

[9] Popescu A M, Etzioni O. Extracting product features and opinions from reviews[M]// Natural Language Processing and Text Mining. Springer London, 2007: 9–28.

[10] Darena F, Burda K. Grouping of customer opinions written in natural language using unsupervised machine learning[C]// Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on. IEEE, 2012: 265–270.

[11] Chen C C, Chen Z Y, Wu C Y. An unsupervised approach for person name bipolarization using principal component analysis[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(11): 1963–1976.

[12] Paltoglou G, Thelwall M. Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(4): 66.

[13] 黄诗琳, 郑小林, 陈德人. 针对产品命名实体识别的半监督学习方法 [J]. 北京邮电大学学报, 2013, 36(002): 20–23.
Huang Shilin, Zheng Xiaolin, Chen Deren. A semi-supervised learning method for product named entity recognition[J]. Journal of Beijing University of Posts and Telecommunications, 2013, 36(002): 20–23.

[14] Moraes R, Valiati J F, Gaviao Neto W P. Document-level sentiment classification an empirical comparison between SVM and ANN[J]. Expert Systems with Applications, 2013, 40(2): 621–633.

[15] Sayeedunnissa S F, Hussain A R, Hameed M A. Supervised opinion mining of social network data using a bag-of-words approach on the cloud[C]// Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012). Springer India, 2013: 299–309.

[16] Kanayama H, Nasukawa T. Unsupervised lexicon induction for clause-level detection of evaluations[J]. Natural Language Engineering, 2012, 18: 83–107.

[17] Xianghua F, Guo L, Yanyan G, et al. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon[J]. Knowledge-Based Systems, 2013, 37: 186–195.

[18] Kawamae N. Predicting future reviews: Sentiment analysis models for collaborative filtering[C]// Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM, 2011: 605–614.

[19] Leung C W K, Chan S C F, Chung F L, et al. A probabilistic rating inference framework for mining user preferences from reviews[J]. World Wide Web, 2011, 14(2): 187–215.

[20] Danesh S, Liu W, French T, et al. An investigation of recursive auto-associative memory in sentiment detection[M]// Advanced Data Mining and Applications. Springer Berlin Heidelberg, 2011: 162–174.

[21] 秦宇强, 张雪英. 连续汉语普通话中基于 SVM 的说话人情感互相关性算法 [J]. 系统工程理论与实践, 2011, 31(增刊 2): 154-159.
Qin Yuqiang, Zhang Xueying. SVM-based speaker emotional cross-correlation algorithm in continuous Chinese mandarin[J]. Systems Engineering — Theory & Practice, 2011, 31(S2): 154–159.

[22] Yin P, Wang H W, Guo K Q. Feature-opinion pair identification of product reviews in Chinese: A domain ontology modeling method[J]. New Review of Hypermedia and Multimedia, 2013, 19(1): 3–24.

[23] Alam M H, Lee S K. Semantic aspect discovery for online reviews[C]// Proceedings of the 2012 IEEE 12th International Conference on Data Mining. IEEE Computer Society, 2012: 816–821.

[24] Yu Z Z, Zheng N, Xu M. An automatic product features extracting method in Chinese customer reviews[C]// 7th International Conference on System of Systems Engineering (SoSE), 2012: 455–459.

[25] Kasper W, Vela M. Sentiment analysis for hotel reviews[C]//Computational Linguistics-Applications Conference. 2011: 45–52.

[26] Somprasertsri G, Lalitrojwong P. A maximum entropy model for product feature extraction in online customer reviews[C]// 2008 IEEE Conference on Cybernetics and Intelligent Systems, IEEE, 2008: 575–580.

[27] Hu M, Liu B. Mining opinion features in customer reviews[C]// AAAI, 2004, 4(4): 755–760.

[28] Liu Q, Li S. Word similarity computing based on how-net[J]. Computational Linguistics and Chinese Language Processing, 2002, 7(2): 59–76.

[29] Billsus D, Pazzani M J. Learning collaborative information filters[C]//ICML. 1998, 98: 46–54.

[30] Lin C, Tsai C. Applying social bookmarking to collective information searching (CIS): An analysis of behavioral pattern and peer interaction for co-exploring quality online resources[J]. Computers in Human Behavior, 2011, 27(3): 1249–1257.

[31] Dalcanale F, Fontane D, Csapo J. A general framework for a collaborative water quality knowledge and information network[J]. Environmental Management, 2011, 47(3): 443–455.

[32] Liu H, He J, Wang T, et al. Combining user preferences and user opinions for accurate recommendation[J]. Electronic Commerce Research and Applications, 2013, 12(1): 14–23.

[33] Xu J, Zheng X, Ding W. Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering[C]// 2012 IEEE Ninth International Conference on E-Business Engineering (ICEBE), IEEE, 2012: 9–16.

[34] 邓晓懿, 金淳, 韩庆平, 等. 基于情境聚类和用户评级的协同过滤推荐模型 [J]. 系统工程理论实践, 2013, 33(11): 2945–2953.
Deng Xiaoyi, Jin Chun, Han Qingping, et al. Improved collaborative filtering model based on context clustering and user ranking[J]. Systems Engineering — Theory & Practice, 2013, 33(11): 2945–2953.

[35] 郁雪, 李敏强. 基于 PCA-SOM 的混合协同过滤模型 [J]. 系统工程理论与实践, 2010, 30(10): 1850–1854.
Yu Xue, Li Minqiang. Effective hybrid collaborative filtering model based on PCA-SOM[J]. Systems Engineering — Theory & Practice, 2010, 30(10): 1850–1854.

[36] Hu N, Tian G, Liu L, et al. Do links matter? An investigation of the impact of consumer feedback, recommendation networks, and price bundling on sales[J]. IEEE Transactions on Engineering Management, 2012, 59(2): 189–200.

[37] Tyagi S, Bharadwaj K K. Enhanced new user recommendations based on quantitative association rule mining[J]. Procedia Computer Science, 2012, 10: 102–109.

[38] 李聪, 梁昌勇. 基于 n 序访问解析逻辑的协同过滤冷启动消除方法 [J]. 系统工程理论与实践, 2012, 32(7): 1537–1545.
Li Cong, Liang Changyong. Cold-start eliminating method of collaborative filtering based on n-sequence access analytic logic[J]. Systems Engineering — Theory & Practice, 2012, 32(7): 1537–1545.

[39] 钟佳丰. 基于在线评论的产品模糊推荐系统研究 [D]. 大连: 大连理工大学, 2012.
Zong Jiafeng. Research on product fuzzy recommendation system based on online review[D]. Dalian: Dalian University of Technology, 2012.

[40] 胡新明. 基于商品属性的电子商务推荐系统研究 [D]. 武汉: 华中科技大学, 2012.
Hu Xinming. Research on recommender system based on product attributes[D]. Wuhan: Huazhong University of Science and Technology, 2012.

[41] Gan M, Jiang R. Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation[J]. Expert Systems with Applications, 2013, 40(10): 4044–4053.

[42] Pham H X, Jung J J. Preference-based user rating correction process for interactive recommendation systems[J]. Multimedia Tools and Applications, 2013, 65(1): 119–132.

[43] 金淳, 张一平. 基于 Agent 的顾客行为及个性化推荐仿真模型 [J]. 系统工程理论与实践, 2013, 33(2): 463–472.
Jin Chun, Zhang Yiping. Agent-based simulation model of customer behavior and personalized recommendation[J]. Systems Engineering — Theory & Practice, 2013, 33(2): 463–472.

[44] Song H, Fan Y, Liu X, et al. Extracting product features from online reviews for sentimental analysis[C]// 2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), IEEE, 2011: 745–750.

[45] Xunsearch. SCWS 中文分词 [EB/OL]. [2014-01-19]. http://www.xunsearch.com/scws/

[46] 北京大学. 中文汉语标注集 [EB/OL]. [2014-01-19]. http://icl.pku.edu.cn/icl_groups/corpus/addition.htm.

[47] Nakagawa H, Mori T. A simple but powerful automatic term extraction method[C]// COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology-Volume 14. Association for Computational Linguistics, 2002: 1–7.

[48] Somprasertsri G, Lalitrojwong P. Mining feature-opinion in online customer reviews for opinion summarization[J]. Journal of Universal Computer Science, 2010, 16(6): 938–955.

[49] Merton R K. The Matthew effect in science[J]. Science, 1968, 159(3810): 56–63.

[50] Cho J, Roy S. Impact of search engines on page popularity[C]//Proceedings of the 13th International Conference on World Wide Web. ACM, 2004: 20–29.

[51] Fleder D M, Hosanagar K. Recommender systems and their impact on sales diversity[C]//Proceedings of the 8th ACM Conference on Electronic Commerce. ACM, 2007: 192–199.

[52] Goel S, Broder A, Gabrilovich E, et al. Anatomy of the long tail: Ordinary people with extraordinary tastes[C]// Proceedings of the Third ACM International Conference on Web Search and Data Mining. ACM, 2010: 201–210.

[53] Anderson C. The long tail[M]. Random House Business, 2006.

[54] Linden G. Early amazon: Similarities[EB/OL]. [2014-01-19]. http://glinden.blogspot.com/2006/03/early-amazon-similarities.html.

[55] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012: 23–29, 45–49.
Xiang Liang. Recommendation system in action[M]. Beijing: The People's Posts and Telecommunications Press (Posts & Telecom Press), 2012: 23–29, 45–49.

[56] 刘青文. 基于协同过滤的推荐算法研究 [D]. 合肥: 中国科学技术大学, 2013.
Liu Qingwen. Research on recommender systems based on collaborative filtering[D]. Hefei: University of Science and Technology of China, 2013.