

推荐算法综述

杨博^{1,2}, 赵鹏飞^{1,2}

(1. 吉林大学 计算机科学与技术学院, 吉林 长春 130012;

2. 吉林大学 符号计算与知识工程教育部重点实验室, 吉林 长春 130012)

摘要:推荐是解决互联网信息过载的主要途径之一,已被广泛应用于电子商务等多个领域.尽管已存在多种推荐算法,建造出更加智能、更加鲁棒的推荐系统仍面临诸多尚未解决的难题,推荐方法的研究仍是智能信息处理的研究热点.文章首先阐述了推荐方法的研究背景、研究意义,之后分别介绍了协同过滤推荐算法、基于内容的推荐算法、基于图结构的推荐算法和混合推荐算法,分析了各类算法的优点与不足,最后总结了主要的评价方法以及面临的主要问题,提出了改进的方法和未来可能的研究方向.

关键词:信息过载;推荐系统;协同过滤;信息检索;数据挖掘;机器学习

中图分类号:TP393 **文献标识码:**A

0 引言

信息过载是互联网面临的主要挑战.如何从过载的信息中主动为用户定位和推送其感兴趣的内容是推荐系统的主要任务.

随着电子商务的发展,商家提供的商品种类和数量急剧增长.具有明确需求的用户可通过搜索查找想购买的商品.然而,用户需求通常具有不确定性和模糊性.据亚马逊统计,在其网站购物的客户中,有明确购买意向的仅占16%.如果商家能够从海量的商品中把满足用户模糊需求的商品主动推荐给用户,则有望将潜在需求转化为实际需求,不仅能提高电子商务网站的销售量,还有助于提高用户对网站的忠诚度.在此背景下,能够根据用户特征有针对性推荐商品的个性化推荐系统应运而生,并被广泛应用.包括Amazon、eBay、YouTube和Google在内的诸多网站都部署了不同形式的推荐系统,并产生了巨大的商业利润.据统计,2006年推荐系统为Amazon提高了30%的销售额.

推荐系统具有重要的应用价值,不仅成为计算机领域具有挑战性的研究课题之一,还吸引了来自数学、物理、认知、人工智能、管理、市场营销等众多领域的研究者.2006年10月,美国Netflix公司举行了一场奖金高达100万美元的竞赛,征集更为有效的推荐算法.这场历时三年的竞赛吸引了来自186个国家的4万多个团队参加.

推荐系统还面临诸多未被很好解决的理论和应用层面上的难题,仍是信息检索、数据挖掘、机器学习和人工智能等领域的研究热点.在此研究背景下,本文全面介绍了现有主要的推荐方法,分析了各自的优势与不足,总结了该领域目前所面临的主要问题,试图为推荐技术的研究勾勒出一条清晰的主线,为相关研究者提供有益的参考.参考文献[1-2]对推荐系统的分类,第1节分别从协同过滤、基于内容、基于图结构以及混合推荐四个方面介绍现有主要的推荐算法,经常采用的相似度计算方法,评分估计方法和推荐算法评价准则.第2节总结和分析推荐研究所面临的主要问题和现有的解决方法.第3节对全文进行总结.

收稿日期:2011-05-10;修回日期:2011-05-20

基金项目:国家自然科学基金(60873149;60973088);中央高校基本科研业务费专项资金(200903177)

作者简介:杨博(1974-),男,河南新乡人,教授,博士生导师,主要研究领域为Agent系统,数据挖掘,复杂网络分析. E-mail: Ybo@jlu.edu.cn

1 主要推荐算法

1.1 协同过滤推荐算法

协同过滤推荐算法的基本假设是:为用户推荐感兴趣的内容可通过找到与该用户偏好相似的其他用户,将他们感兴趣的内容推荐给该用户.文献[3]中将协同过滤方法分为基于记忆(memory-based)和基于模型(model-based)的方法.

基于记忆方法采用用户-项目(user-item)评分数据,为目标用户估计对某一特定项目的评分或产生一个推荐列表.表1给出了一个示例,其中空白项表示用户没有对此项目评分,“?”表示待估计的评分.系统根据该表估计用户 U_1 对项目 I_3 的评分.评分等级一般为1至5的正整数,数值大小表示用户对项目的偏好程度.

基于模型方法采用统计、机器学习和数据挖掘等方法,根据目标用户的历史数据(评分、购买、浏览、点击等)构造用户模型(user profile),并据此进行推荐.

表1 用户-项目评分矩阵

Table 1 User-item ratings matrix

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	4	3	?			
U_2	3	2	2	4	4	2
U_3			2		3	2
U_4	4	3	3	5		2
U_5	2	4		1		5

1.1.1 基于记忆的推荐算法

基于记忆的推荐算法可以概括为两步:①采用相似度计算方法计算用户/项目之间的相似度,构造相似度矩阵;②采用相应的算法估计评分,并据此为用户进行推荐.该方法可为目标用户估计对某一特定项目的评分,也可产生一个推荐列表.算法中涉及的相似度计算方法将在第1.6节详细讨论.

(1) 基于用户(user-based)的推荐算法

基于用户的推荐算法^[4]为目标用户 $U_i(i=1,2,\dots,n)$ 估计对给定项目 $I_j(j=1,2,\dots,m)$ 的评分 $p_{i,j}$.该方法首先计算用户 U_i 和其他用户间的相似度,选取其他用户中为 I_j 评过分的用户构成集合 U_i^* ,根据所有 $U_k \in U_i^*$ 对项目 I_j 的评分来估计 U_i 对 I_j 的评分.以表1为例说明该方法计算 $p_{1,3}$ 的过程.采用余弦相似度方法计算用户间的相似度可得:

$$\text{sim}(U_1, U_2) = \frac{4 \times 3 + 3 \times 2}{\sqrt{(4^2 + 3^2)(3^2 + 2^2)}} = 0.998$$

同理可得:

$$\text{sim}(U_1, U_3) = 0, \text{sim}(U_1, U_4) = 1, \text{sim}(U_1, U_5) = 0.894.$$

进而计算:

$$p_{1,3} = \frac{\text{sim}(U_1, U_2) p_{2,3} + \text{sim}(U_1, U_3) p_{3,3} + \text{sim}(U_1, U_4) p_{4,3}}{\text{sim}(U_1, U_2) + \text{sim}(U_1, U_3) + \text{sim}(U_1, U_4)} = \frac{0.998 \times 2 + 0 \times 2 + 0.894 \times 3}{0.998 + 0 + 0.894} = 2.5$$

基于用户的推荐方法面对用户数量不断变化的系统,需要经常重新计算用户间的相似度矩阵,因而时间复杂性高,可扩展性较差.

(2) 基于项目(item-based)的推荐算法

通常情况下,网站用户数量不断增长,而被推荐的项目数量则保持相对稳定,计算出的项目相似度矩阵更新频率低,可在较长的一段时间内使用.为此提出了基于项目(item-based)的推荐算法^[5-6].该方法首先计算各项目间的相似度,通过目标用户评过分的的项目,估计对特定项目的评分.仍以表1为例,采用基于项目方法计算 $p_{1,3}$ 过程如下:

$$\text{sim}(I_1, I_3) = 0.998, \text{sim}(I_2, I_3) = 1 \quad p_{1,3} = \frac{0.998 \times 4 + 1 \times 3}{0.998 + 1} = 3.5$$

当推荐系统中的项目数量快速变化时(在线新闻作为被推荐的项目),基于项目的算法同样面临着可扩展性问题。

(3) 基于用户的 top-N 算法

基于用户的 top-N 推荐算法为目标用户产生一个推荐列表,表中项目的先后次序根据用户的偏好度排序。该方法的过程如下:①首先计算目标用户与其他用户的相似度;②选取相似度最大的 K 个用户构成集合 U^* ;③集合 U^* 中所有用户已评过分的项项目构成项目集 I^* ,并删除目标用户已评过分的项项目;④统计 I^* 中各项目被 U^* 中用户评分大于或等于 3 的次数;⑤选取 N 个对应次数最大的项项目作为推荐列表,如果某些项项目的次数相等,则可计算平均分进一步比较。算法中的参考 K 可根据实验结果调整其大小,参数 N 根据应用环境设定。例如,在电影推荐系统中, N 一般选取较小的数,以使用户能在有限时间看完,而对于新闻推荐, N 可以选取比较大的数。

以表 1 为例,采用以上算法为用户 U_1 产生推荐结果如下,这里选取 $K=2$ 和 $N=2$ 。计算相似度可得 U_4 和 U_2 是与 U_1 最相似的 2 个用户,则有 $U^* = \{U_2, U_4\}$ 和 $I^* = \{I_3, I_4, I_5, I_6\}$,其中各项对应的平分超过 3 的次数分别为 1、2、1、0。由于 I_3 和 I_5 次数均为 1,因此需要计算其平均分,分别为 3 和 4。基于以上可得:为用户 U_1 的推荐项项目列表是 $\{I_4, I_5\}$ 。

(4) 基于项项目的 top-N 推荐算法

基于项项目的 top-N 推荐算法比基于用户的 top-N 方法有更好的可扩展性^[7]。该算法为用户 U_i 推荐 N 个项项目的步骤如下^[8]:①首先计算每个项项目间的相似度,为每个项项目构建一个 K 最相似的近邻;②由用户已打过分的项项目集 I_i^* 中每个项项目的 K 个最相似的项项目构造集合 I^* (除去用户已经打过分的项项目);③计算集合 I^* 中每个项项目与集合 I_i^* 的相似度;④选取相似度最大的 N 个作为推荐结果。

还以表 1 为例,采用以上算法为用户 U_1 产生推荐结果,这里选取 $K=3$ 和 $N=2$ 。

$$\text{sim}(I_1, I_2) = 0.919, \text{sim}(I_1, I_3) = 0.998, \text{sim}(I_1, I_4) = 0.974, \text{sim}(I_1, I_5) = 1, \text{sim}(I_1, I_6) = 0.764$$

$$\text{sim}(I_2, I_1) = 0.919, \text{sim}(I_2, I_3) = 1, \text{sim}(I_2, I_4) = 0.774, \text{sim}(I_2, I_5) = 1, \text{sim}(I_2, I_6) = 0.970$$

由此可得与 I_1 最相似的 3 个近邻为 I_5, I_3, I_4 ,与 I_2 最相似的 3 个近邻为 I_3, I_5, I_6 。

$I_1^* = \{I_1, I_2\}$, $I^* = \{I_5, I_3, I_4, I_6\}$ 。 I_5 与 I_1^* 的相似度为 $(\text{sim}(I_1, I_5) + \text{sim}(I_2, I_5))/2 = 1$,同理可得余下各项与 I_1^* 的相似度分别为 0.999, 0.874, 0.867,因此为用户 U_1 产生的推荐项项目列表为 $\{I_5, I_3\}$ 。

基于记忆的推荐算法主要优点是:算法简单,且容易理解和实现。但在实际问题中,用户一项目评分矩阵通常非常稀疏,导致该类算法面临许多问题。例如,冷启动问题(新用户、新项目问题)等。此外,算法采用的相似度计算方法也存在不足。例如,如果两个用户共同评分的项项目都很少,则难以准确的计算出两者的真实相似度。相关问题将在第 2 节详细讨论。

1.1.2 基于模型的推荐算法

推荐可以看做是分类或预测问题。基于模型的推荐方法采用统计学、机器学习、数据挖掘等方法,根据用户历史数据为用户建立模型,并据此产生合理的推荐。这类方法其在一定程度上解决了用户一项目评分矩阵的稀疏性问题。下面给出了一个简单的评分模型^[3]:

$$R(u, i) = \sum_{r=1}^5 p(u, r) \times r \quad (r = 1, 2, \dots, 5)$$

其中, $R(u, i)$ 表示用户 u 对项项目 i 的评分估计(评分等级为 1 到 5 之间的整数), $p(u, r)$ 表示用户 u 打 r 分的概率。基于模型的算法可以借助分类^[9]、线性回归^[5]和聚类^[10]等机器学习实现^[11]。

(1) 基于朴素贝叶斯分类的推荐算法^[9-12]

朴素贝叶斯分类方法假设样本的各个属性相互独立。尽管实际数据大都不能满足该假设,但朴素贝叶斯假设大大简化了问题求解的复杂性,因而在实际问题中经常被采用。假设评分等级分为 5 等,则估计用户 U_i 对项项目 I_j 的评分问题可建模一个分类问题,确定评分所属的 5 个类别之一。假设用变量 y 表示类别, $m=1, 2, 3, 4, 5$ 表示类标识,则有:

$$P(y = m | x_1, x_2, \dots, x_{n-1}) \propto P(x_1, x_2, \dots, x_{n-1} | y = m)P(y = m)$$

进而根据朴素贝叶斯假设有:

$$\propto P(x_1 | y = m)P(x_2 | y = m) \cdots P(x_{n-1} | y = m)P(y = m)$$

其中 x_1, x_2, \dots, x_{n-1} 为其他 $n-1$ 个用户对 I_j 的评分, $P(y=m|x_1, x_2, \dots, x_{n-1})$ 表示在 x_1, x_2, \dots, x_{n-1} 下, 用户对项目 I_j 评分为 m 的概率. 根据用户-项目表计算各个概率 $P(x_i|y=m)$ 和先验概率 $P(y=m)$, 进而计算不同 m 值对应的后验概率 $P(y=m|x_1, x_2, \dots, x_{n-1})$. 选择对应最大后验概率的 m 值作为最终的评分. 通常情况下, 用户对项目的评分是独立的, 因此该算法表现出较好的计算精度. 但如果用户评分之间存在相互依赖关系时, 该算法的准确性将打折扣.

(2) 基于线性回归的推荐方法

基于用户的历史评分, 可采用线性回归方法为用户建立近似的线性预测模型. 令 $u = (x_1, x_2, \dots, x_n)$ 表示用户 u 对 n 个项目的评分, 拟估计用户对第 $n+1$ 个项目的评分 x_{n+1} . 令 $p = (a_1, a_2, \dots, a_n)$ 表示评分系数, 则线性预测模型可表示为:

$$r_{n+1} = p \times u^T + m$$

其中常数 m 表示估计偏差, 进而基于用户-项目矩阵, 估计出模型中的参数. 由于线性回归模型的参数估计也是基于用户-项目矩阵, 受该矩阵的稀疏、噪声等因素影响, 该方法的准确率有可能会降低. 为此, 可采用诸如稀疏因子分解^[13] 和奇异值分解^[11] 等方法进行数据预处理数据, 以降维的方式压缩矩阵, 从而在一定程度上解决数据稀疏性问题. 除了线性回归方法, Slope one 是另一种常用的线性推荐算法, 该算法容易实现, 且能较快地产生推荐结果^[14].

(3) 基于马尔科夫决策过程 MDP 的推荐算法

借鉴强化学习(reinforcement learning)的思想, 文献[15]中把推荐过程建模为 MDP 最优决策问题, 即如何产生一个能最大用户收益的推荐项目列表. 该工作将 MDP 模型定义为一个 4 元组 (S, A, R, Pr) , 其中 S 表示由项目序列构成的状态集, A 表示动作集, 每个动作是向用户推荐某个项目, R 表示对每种状态/动作的奖励函数, Pr 表示状态转移概率. 推荐过程对应的 MDP 过程可描述如下:

$$V_i(s) = R(s, \pi_i(s)) + r \sum_{s_j \in S} Pr(s, \pi_i(s), s_j) V_i(s_j) \quad (1a)$$

$$\pi_{i+1}(s) = \arg \max_{a \in A} \left[R(s, a) + r \sum_{s_j \in S} Pr(s, a, s_j) V_i(s_j) \right] \quad (1b)$$

其中, s 表示当前状态, $V_i(s)$ 表示在状态 s 下采取某种动作获得的收益值, $\pi_{i+1}(s)$ 表示在状态 s 为获得最大受益而应执行的动作, 即向用户推荐某个项目. 从初始活动 $\pi_0(s) = \arg \max_{a \in A} [R(s, a)]$ 开始, 迭代公式(1a)和(1b)直至收敛. 以此方式产生一个最大化收益的推荐项目列表.

除以上介绍的方法外, 基于模型的协同过滤方法还包括基于聚类的 Gibbs 抽样方法^[16], 概率相关方法^[17] 和极大熵方法^[18] 等. 基于模型的协同过滤算法能在一定程度上解决基于记忆的推荐算法面临的主要困难, 在推荐性能上更优, 但通常算法复杂, 计算开销大.

1.2 基于内容的推荐算法

协同过滤方法只考虑了用户评分数据, 忽略了项目和用户本身的诸多特征, 如电影的导演、演员和发布时间等, 用户的地理位置、性别、年龄等. 如何充分、合理的利用这些特征, 获得更好的推荐效果, 是基于内容推荐策略所要解决的主要问题.

1.2.1 文本推荐方法

基于内容的推荐方法根据历史信息(如评价、分享、收藏过的文档)构造用户偏好文档, 计算推荐项目与用户偏好文档的相似度, 将最相似的项目推荐给用户. 相比于多媒体信息(视频、音频、图片等), 文本类项目(新闻、网页、博客)的特征提取相对容易, 因而基于内容的推荐方法在文本类推荐领域得到了广泛应用.

用户偏好文档和推荐项目文档都采用关键字表示特征, 进而采用 TF-IDF^[19] (term frequency-inverse document frequency) 方法为各个特征确定权重. 如网页推荐系统 Fab 采用 100 个关键字表示一个网页^[20], Syskill 和 Weber 系统采用 128 个关键词表示一个文档^[21].

TF-IDF 方法的基本思想是:一方面,关键字 k 在文档 D 中出现的次数越多,表示 k 对文档 D 越重要,越能用该关键字表示文档 D 的语义;另一方面,关键字 k 在不同文档中出现的次数越多,表示 k 对区别文档的贡献越少.综合考虑以上两个方面,提出 TF-IDF 的特征权重设置方法.设文档集包含的文档数为 N ,文档集中包含关键字 k_i 的文档数为 n_i , f_{ij} 表示关键字 k_i 在文档 d_j 中出现的次数, k_i 在文档 d_j 中的词频 TF_{ij} 定义为:

$$TF_{ij} = \frac{f_{ij}}{\max_z f_{zj}}$$

其中 z 表示在文档 d_j 中出现的关键字. k_i 在文档集中出现的逆频 IDF_i 定义为:

$$IDF_i = \log \frac{N}{n_i}$$

采用 k 维向量 $\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{kj})$ 和 $\vec{d}_c = (w_{1c}, w_{2c}, \dots, w_{kc})$ 分别表示项目文档和用户 c 的配置文档, k 是关键词的个数,向量中的各个分量可计算如下:

$$w_{ij} = TF_{ij} \cdot IDF_i = \frac{f_{ij}}{\max_z f_{zj}} \cdot \log \frac{N}{n_i}$$

可采用多种方法计算项目文档和用户配置文档的相似度,其中夹角余弦方法最为常用,如下所示.

$$sim(c, d_j) = \cos(\vec{d}_c, \vec{d}_j) = \frac{\sum_{i=1}^k w_{ic} w_{ij}}{\sqrt{\sum_{i=1}^k w_{ic}^2} \sqrt{\sum_{i=1}^k w_{ij}^2}}$$

1.2.2 基于潜在语义分析的推荐

关键词的同义和多义现象往往导致文档相似度计算不准确.针对该问题,提出了潜在语义分析方法(Latent Semantic Analysis, LAS)^[22].其基本思想是:采用文档-词矩阵奇异值分解的方法将文档和词语映射到同一个低维的潜在语义空间中,在此空间中可灵活计算文档之间、词之间或文档与词之间的相似度.用户提出的查询请求也被映射到相同的语义空间中,计算各文档与用户查询之间的相似度,据此返回最相关的文档. LSA 的不足是:采用奇异值分解得到的潜在语义空间物理意义不明确,矩阵的奇异值分解计算量大.文献[24]中提出了概率潜在语义分析(probabilistic latent semantic analysis, PLSA)模型可以很好地克服 LSA 模型的不足. PLSA 模型通过计算文档中共现词的分布分析文档的语义.文献[23-24]中实验表明,基于 PLSA 模型的推荐准确度优于基于聚类模型、贝叶斯模型和皮尔森相关性的推荐方法.

1.2.3 自适应推荐

如何构建和更新用户偏好文档是基于内容推荐的关键所在.用户的兴趣会随时间动态变化,因此需要及时更新偏好文档,才能始终为用户推荐准确的内容.

针对该问题,文献[25-26]提出了更新用户偏好文档的自适应过滤方法.该方法将与用户偏好文档相似度高文档项目推荐给用户,同时使用相似度高文档项目更新用户偏好文档各分量的权重,以此方式实现用户偏好文档向新兴趣的动态调整.在此基础上,文献[27]通过阈值法提高了推荐系统的运行效率,只有当文档项目与用户偏好文档相似度高于设定的阈值时,才对用户配置文档进行更新.文献[28]进一步将用户兴趣细分为长期和短期兴趣,并认为短期兴趣更能反映用户当前关注的内容,赋予短期兴趣关键字较大的权重,进一步提高对用户兴趣建模的准确性.除以上方法外,一些系统还采用了 Bayesian 分类^[29-30]、决策树、聚类和人工神经网络^[29]等机器学习方法建立和更新更为复杂的用户偏好模型.

不能向用户推荐新项目是协同过滤推荐方法的主要缺点之一.新项目进入推荐系统后,协同过滤推荐方法只有当该项目被多个相关用户评价之后,才能为其他用户进行较为准确的推荐.基于内容的推荐方法在一定程度上解决了该问题.新项目进入推荐系统后,基于内容的推荐方法为其提取特征,进而建立刻画其内容的特征向量,然后根据用户偏好文档决定是否向用户推荐.

基于内容的推荐方法也面临一些问题,如新用户问题、过拟合问题、多媒体信息特征难提取问题等,本文将在第 2 节详细讨论.

1.3 基于图结构的推荐算法

用户-项目矩阵可建模为一个二部图(bipartite graph),其中节点分别表示用户和项目,边表示用户对项目的评价.基于图结构的推荐算法通过分析二部图结构给出合理的推荐.

借鉴动态网络资源分配过程,文献[31]提出了基于二部图结构分析的推荐算法,该方法的推荐过程如下:

① 建立推荐二部图.推荐系统包含 n 个用户和 m 个项目,若用户 i 偏好项目 j ,则令 $a_{ij} = 1$,否则令 $a_{ij} = 0$,以此方式可建立一个包含 $n+m$ 个节点的二部图.

② 计算资源分配矩阵 W .项目 j 到项目 i 的资源分配权重 w_{ij} 计算如下:

$$w_{ij} = \frac{1}{D_j} \sum_{k=1}^n \frac{a_{ik} a_{jk}}{D_k} \quad \begin{array}{l} i,j \text{ 表示项目} \\ k \text{ 表示用户} \end{array} \quad (2)$$

其中 D_j 表示节点 j 的度.

③ 针对指定用户计算各项目的资源分配.令 $f_i = (a_{i1}, a_{i2}, \dots, a_{im})$ 表示用户 i 的对 m 个项目的初始资源分配, f'_i 表示用户 i 的对 m 个项目的最终资源分配,则有 $f'_i = Wf_i$.

④ 根据 f'_i 产生推荐列表.按 f'_i 中从大到小的顺序排列生成一个推荐项目列表(用户已经偏好的项目除外).

图 1 表示一个由 5 个项目和 3 个用户组成的推荐系统,在其对应的二部图中, x 表示项目, y 表示用户, y_1 为目标用户.

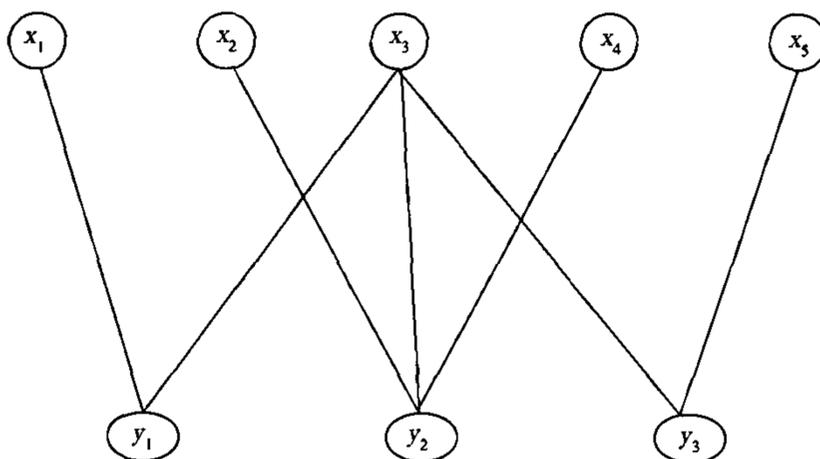


图 1 推荐系统对应的二部图

Fig. 1 Bipartite graph of recommender system

由图可知, y_1 的初始资源分配为: $f_1 = (1, 0, 1, 0, 0)^T$, 由公式(2)可计算出项目间的资源分配矩阵:

$$W = \begin{pmatrix} 1/2 & 0 & 1/6 & 0 & 0 \\ 0 & 1/3 & 1/9 & 1/6 & 0 \\ 1/2 & 1/3 & 7/18 & 1/3 & 1/3 \\ 0 & 1/3 & 2/9 & 1/3 & 1/3 \\ 0 & 0 & 1/9 & 1/6 & 1/3 \end{pmatrix}$$

目标用户 y_1 的最终资源分配如下:

$$f'_1 = Wf_1 = \begin{pmatrix} 1/2 & 0 & 1/6 & 0 & 0 \\ 0 & 1/3 & 1/9 & 1/6 & 0 \\ 1/2 & 1/3 & 7/18 & 1/3 & 1/3 \\ 0 & 1/3 & 2/9 & 1/3 & 1/3 \\ 0 & 0 & 1/9 & 1/6 & 1/3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 1/9 \\ 8/9 \\ 2/9 \\ 1/9 \end{pmatrix}$$

因此为目标用户 y_1 推荐的列表为 (x_4, x_2, x_5) (最终资源分配相等的项目随机排列).

文献[32]提出另外一种计算矩阵 W 的方法:

$$w_{ij} = \frac{1}{D_i} \sum_{k=1}^n \frac{a_{ik} a_{jk}}{D_k} \quad (3)$$

实验结果表明,采用(2)式产生的推荐列表倾向于向用户推荐热门项目,注重推荐的准确性,而采用(3)

式倾向于向用户推荐与众不同的项目,注重多样性. 两者的性能都优于基于 Pearson 相似度的推荐方法. 文献[32]融合公式(2)和公式(3),得到兼顾准确性和多样性的推荐算法. 文献[33]根据用户之间的朋友关系、用户参加的组关系、用户收听历史,将用户、组、音乐家用图表示,采用基于图的节点相似度方法,为用户推荐音乐家,实验室结果表明,准确度比基于用户的协同过滤算法提高了 8%. 文献[34]提出一种基于两层混合模型的推荐算法,采用小世界网络描述用户层中用户-用户节点间关系,采用贝叶斯网络描述项目层中项目-项目节点间关系及用户-项目节点间的偏好关系. 通过对用户进行聚类,采用贝叶斯网络推理产生推荐.

基于图结构计算推荐结果前需要预处理数据集. 例如,MovieLens 是用户对电影的评分数据集,评分等级分为 5 等,需要通过预处理将评分数据转换为二部图结构. 一种方法是,如果用户 i 对项目 j 的评分超过一个事先设定的阈值(如 3 分),则令 $a_{ij} = 1$. 如何选择合适的阈值以得到合理的推荐结果是该预处理方法面临的一个主要问题. 此外,大多数基于图分析的推荐算法都不考虑边的权重. 文献[35]将用户对不同项目的评分作为边的权重,采用 Markov 随机游走的方法,计算推荐结果. 实验表明,考虑权重的推荐方法在推荐准确性方面优于不考虑边权重的推荐方法.

1.4 混合推荐算法

混合推荐是为了解决协同过滤、基于内容和基于图结构推荐算法各自问题而提出的,达到“相互取长补短”的推荐效果. 例如,基于内容方法可以解决协同过滤中“新项目”问题,而协同过滤可降低基于内容算法面临的“过拟合”问题.

混合推荐可以独立运用协同过滤、基于内容和基于图结构的推荐算法,将两者或多者产生的推荐结果进行融合,再将融合后的结果推荐给用户. 文献[1]中提出协同过滤和基于内容推荐的 4 种混合策略:(1) 两种方法单独进行,将结果进行融合;(2) 基于内容融合到协同过滤方法中;(3) 协同过滤融合到基于内容方法中;(4) 将两者混合到一个框架下产生一种新推荐方法.

以第二种混合策略为例具体分析不同的策略如何“相互取长补短”. 协同过滤一般只考虑打分信息,因此如果一个项目没有被用户打分就不会出现在推荐结果中,即所谓的“新项目”问题. 而基于内容的推荐算法可利用用户或项目的特征信息,为用户推荐与用户偏好相似度高的项目,较好的解决这一问题. 另一方面,基于内容的推荐系统,只有当项目文档与用户偏好文档相似时才进行推荐,为用户推荐的结果与用户之前偏好的项目过于类似,会出现过拟合现象. 而协同过滤可在一定程度上解决该问题,为用户推荐新颖的项目.

目前大量网站都提供采用标签(tags)对内容进行标注的功能,将这些信息融合到基于图结构的推荐算法中^[36-37],也有重要的研究价值.

此外,在协同过滤中,基于模型方法与基于记忆方法的混合也行之有效,实验结果表明该策略比单纯基于记忆和基于模型有更好的性能^[38-39].

1.5 其他推荐算法

除了以上介绍的推荐算法,在商业推荐系统中还存在其他的推荐方法.

基于关联规则的推荐根据大量的用户购买历史数据,为用户推荐有相似购买行为的人所购买过的商品. 如美国连锁超市沃尔玛的“啤酒+尿布”推荐,看上去没有关联的两种商品摆放在一起销售,却能够获得意想不到的收益. Agrawal 等提出的 Apriori 算法是经典的关联规则挖掘算法^[40-41],但在实际推荐系统中,待处理的数据量通常巨大,算法效率低下,为此提出了众多的改进方法. 其中,Han 等提出的 FP-Growth 算法能显著提高 Apriori 算法的运行效率^[42].

基于知识的推荐^[43]不是建立在用户需求 and 偏好基础上,而是采用领域知识或规则进行推理. 例如,效用知识(functional knowledge)是关于一个项目如何满足某一特定用户的知识,不仅能产生推荐,还能较好的解释产生推荐的原因所在.

1.6 相似度计算方法

1.6.1 基于共同评分项目相似度计算方法

常用的相似度计算方法包括:皮尔森相关性(pearson correlation)^[4,44]、余弦相似度(cosin similarity)^[3]、调整余弦相似度(adjust cosin similarity)^[5],它们都是基于用户对共同项目的评分来计算.

$$sim(x, y) = \frac{\sum_{i \in I_{x,y}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{x,y}} (r_{x,i} - \bar{r}_x)^2 \sum_{i \in I_{x,y}} (r_{y,i} - \bar{r}_y)^2}} \quad (4a)$$

$$sim(x, y) = \frac{\sum_{i \in I_{x,y}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_{x,y}} r_{x,i}^2 \sum_{i \in I_{x,y}} r_{y,i}^2}} \quad (4b)$$

其中, $I_{x,y}$ 为用户 x 和 y 共同评过分的项集合, $r_{x,i}$ 表示用户 x 对项目 i 的评分, \bar{r}_x 表示用户 x 的平均评分. 公式(4a)计算皮尔森相关性, 公式(4b)计算余弦相似度. (4b)式没有考虑两者平均评分不同的情况, (4a)式考虑了该问题. 调整余弦相似度与公式(4a)的计算形式相似. 在设计基于共同评分的相似度计算方法时, 还需考虑和解决如下问题.

(1) 评分尺度不同

首先规范化用户-项目矩阵中的原始评分, 再计算用户间的相似度. 文献[45]中提出两种用户评分规范化方法.

①高斯规范化方法

$$\overline{r_{u,i}} = \frac{r_{u,i} - \bar{r}_u}{\sqrt{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^2}}$$

式中 I_u 表示用户 u 已评过分的项集合.

②分解规范化方法

文献[46]中把用户-项目评分矩阵分为两个模型: 评分模型(rating model)和偏好模型(preference model). 由用户历史评分值构造用户评分模型, 根据如下公式计算出用户偏好模型, 进而分别计算用户评分相似度和偏好相似度.

$$P(r \text{ is preferred}) = P(\text{rating} \leq r) - P(\text{rating} = r) / 2$$

(2) 公共评分的项目过少

当公共评分的项目很少时, 有可能出现两用户的相似度过高的现象. 存在多种方法解决该问题. 可通过默认投票预处理用户-项目评分矩阵, 使用用户平均评分代替空白项, 增加公共评分项. 文献[47]中根据项目间的关系, 将用户-项目矩阵转换成用户-项目类矩阵, 由各用户对各类进行评分, 依此方式降低评分矩阵的稀疏度. 文献[48]提出一种基于云模型的相似度计算方法, 在用户评分整体层面上粗粒度计算用户之间的相似度, 根据每个用户评分频度向量, 利用逆向云算法得出用 3 个云参数表示用户评分偏好的向量, 采用相似度方法, 如余弦相似度计算用户之间的相似度. 此外, 还有倒排用户频率、实例放大、主流加权、时间加权等方法^[49-52].

(3) 隐含评分

用户-项目评分矩阵给出的是显示评分. 然而, 互联网环境中很多评分是隐式的, 如浏览历史、购买历史、点击行为、下载行为等. 文献[53]针对手机图片下载推荐任务考虑了以上问题. 根据用户下载和发布图片的时间对用户-项目矩阵赋值. 文献[54]采用能表示用户与项目偏好关系的有向无环图代替显示的评分矩阵, 进而给出相应的相似度计算方法.

(4) 相似关系不满足传递性

以上方法采用两个用户的共同评分项目或两个项目的共同评分用户计算两者之间的相似度, 容易产生有偏的结果. 例如, 已知三个用户 A, B 和 C 的评分数据, 采用以上方法分别计算他们之间的相似度, 有可能 A 与 B 相似度高, B 与 C 相似度也高, 但由于 A 和 C 之间没有共同评分的项目而导致两者不相似. 而根据相似度的传递性原则, A 与 C 也应具有高的相似度. 下面介绍的基于图结构的相似度计算方法可有效处理该问题.

1.6.2 基于图结构的相似度计算方法

文献[55]提出一种基于二部图的相似度计算方法. 如 1.3 节所述, 根据用户-项目矩阵可构造相应的二

部图,其中边上的权重表示用户对项目偏好的强弱. A, L 和 P 分别表示二部图的邻接矩阵,拉普拉斯矩阵和马尔科夫随机游走概率转移矩阵. $m(k|i)$ 表示随机游走从节点 i 首次到达节点 k 的期望步长,可采用以下迭代公式计算:

$$\begin{cases} m(k|k) = 0 \\ m(k|i) = \sum_{j=1}^n p_{ij} [1 + m(k|j)] = 1 + \sum_{j=1}^n p_{ij} m(k|j) \quad i \neq j \end{cases}$$

$m(k|i)$ 可作为节点 i 和 k 之间的相似度. 由于 P 为非对称矩阵,因此 $m(j|i) \neq m(i|j)$ 对于 $i \neq j$. 为了使相似度具有对称性,可采用 $n(k,i) = m(k|i) + m(i|k)$ 度量两节点之间的相似度. $m(k|i)$ 还可通过矩阵 L 的穆尔-彭罗斯广义逆矩阵 L^+ 求出:

$$m(k|i) = \sum_{j=1}^n (l_{ij}^+ - l_{ik}^+ - l_{kj}^+ + l_{kk}^+) d_{jj}$$

$$n(k,i) = (l_{ii}^+ + l_{kk}^+ - 2l_{ik}^+) \sum_{j=1}^n d_{jj}$$

其中 $l_{ij}^+ = [L^+]_{ij}$. 上式可用矩阵形式表示,设节点 i 用 n 维列向量 e_i (其中第 i 维为 1,其他维为 0) 表示,则有 $n(k,i) = (e_i - e_j)^T L^+ (e_i - e_j)$. 向量 e_i 可通过空间投影变换为 x_i' , 则有 $(x_i')^T x_i' = l_{ii}^+$, 即为两个节点的内积,可用于表示节点之间的相似度.

实验表明,采用映射后节点间的内积作为相似度,推荐性能要优于其他方法. 该方法不仅可计算用户/用户,项目/项目的相似度,还可用于计算用户和项目间的相似度,作为推荐的直接依据. 而其他相似度计算方法只能计算用户之间或项目之间的相似度.

1.7 评分估计方法

设 $U = \{u_1, \dots, u_n\}$ 表示用户集合, $I = \{i_1, \dots, i_m\}$ 表示项目集合, $r_{u,i}$ 表示用户 u 对项目 i 的评分,则估计 $r_{u,i}$ 的常用方法有:

$$r_{u,i} = \frac{1}{n} \sum_{\bar{u} \in \bar{U}} r_{\bar{u},i} \tag{5a}$$

$$r_{u,i} = \frac{\sum_{\bar{u} \in \bar{U}} w(u, \bar{u}) r_{\bar{u},i}}{\sum_{\bar{u} \in \bar{U}} w(u, \bar{u})} \tag{5b}$$

$$r_{u,i} = \bar{r}_u + \frac{\sum_{\bar{u} \in \bar{U}} w(u, \bar{u}) (r_{\bar{u},i} - \bar{r}_{\bar{u}})}{\sum_{\bar{u} \in \bar{U}} w(u, \bar{u})} \tag{5c}$$

其中 \bar{U} 表示与用户 u 相似度高的用户集合, n 是集合 \bar{U} 中的用户数目, $w(u, \bar{u})$ 表示用户 u 与用户 \bar{u} 之间的权重(可用相似度表示), \bar{r}_u 为用户 u 对项目的平均评分. (5a)式通过求用户 u 近邻对项目 i 评分的平均值来估计 $r_{u,i}$,没有考虑权重问题. (5b)式考虑了权重因素,权重高的近邻对估计评分有大的贡献. (5c)式考虑不同用户评分尺度不同问题.

1.8 推荐系统评价准则

根据推荐的目的,推荐算法或为用户估计特定项目的评分,或为用户产生一个推荐项目列表. 前者常采用平均绝对误差(MAE)、均方根误差(RMSE)评价估计的准确性^[56],后者多采用信息检索领域常用的查全率(recall)和查准率(precise)评价推荐列表的准确性.

类似于机器学习研究方法,一般将数据集分为训练集和测试集两个部分,训练集用于模型中的参数估计,测试集用于评价算法的性能.

(1) 平均绝对误差(mean absolute error, MAE)

MAE 用于度量推荐算法的估计评分与真实值之间的差异. MAE 值越小,估计的准确性越高,定义如下:

$$E_{MAE} = \frac{1}{n} \sum_{i \in U, j \in I} |p_{ij} - r_{ij}|$$

其中 p_{ij} 为用户 i 对项目 j 的估计评分, r_{ij} 为用户 i 对项目 j 的真实评分. 集合 U 和 I 分别为测试集中的用户集合和项目集合. n 为 p_{ij} 的个数. 文献[57]考虑到不同的推荐系统可能采用的评分等级不同, 对 E_{MAE} 进行了规范化处理.

(2) 均方根误差(root mean squared error, RMSE)

RMSE 是 Netflix 竞赛采用的评价准则. RMSE 值越小, 算法的准确度越高, 定义如下:

$$E_{RMSE} = \sqrt{\frac{1}{n} \sum_{i \in U, j \in I} (p_{ij} - r_{ij})^2}$$

(3) 查全率(recall)

用于度量推荐列表中是否包含了用户偏好的全部项目. 令 L_i 表示推荐算法为用户 i 产生的推荐列表, R_i 表示测试集中用户 i 偏好的全部项目, 有:

$$\text{Recall} = \frac{1}{n} \sum_{i=1}^n \frac{|L_i \wedge R_i|}{|R_i|}$$

其中, n 表示训练集中用户数目, $|L_i \wedge R_i|$ 表示推荐列表中实际被用户偏好的项目数量.

(4) 查准率(precision)

用于度量推荐列表中是否都是用户偏好的项目, 定义为:

$$\text{Precision} = \frac{1}{n} \sum_{i=1}^n \frac{|L_i \wedge R_i|}{|L_i|}$$

查全率和查准率没有考虑推荐项目的顺序问题——根据用户对项目的偏好程度排序. 一种改进方法是: 依次考察推荐列表中每一项目, 若此项目在测试集中被用户选择过(说明用户偏好该项目), 则计算到此项目为止推荐列表的查准率, 进而对所有这样的查准率进行平均, 作为该推荐结果的查准率.

评价推荐方法的标准还应与具体应用背景相关, 需要将多种标准折中考虑. 例如, 在线图书推荐对查准率要求高, 但对查全率要求可以适当放宽. 为用户推荐大多数用户都会偏好的项目, 如某部流行的电影, 准确度会很高, 但却忽略了用户的个性化程度. 文献[32]给出了一种评价推荐结果多样性的度量方法.

2 问题及分析

2.1 稀疏性问题

数据的稀疏性是推荐系统面临的主要问题. 协同过滤推荐算法基于用户-项目评分矩阵, 但在实际应用中该矩阵通常非常稀疏, 导致推荐结果不准确. 例如, 基于用户的推荐算法通过相似度计算找出与目标用户相似度高的用户, 相似度计算方法基于两个用户对共同项目的评分, 评分数据的稀疏将导致两者只具有很少的共同评分项目, 因此为目标用户计算出的最近邻用户不准确. 当新用户进入推荐系统时, 由于用户评分数据很少, 很难为用户进行个性化推荐. 同样的, 新进入系统的项目由于被很少人评分而不能被正确推荐. 以上两个问题被称为**冷启动问题**.

已提出多种方法用于减少数据稀疏对推荐结果带来的影响. 采用降维技术^[11,13]压缩矩阵是常采用的一种解决方法. 例如, 采用奇异值分解方法删除不重要的或噪音用户和项目, 降低用户-项目评分矩阵的维数. 采用潜在语义索引技术^[22-24]将两个用户投影到一个低维的空间上, 进而计算两者的相似度. 默认投票方法通过增加用户公共评分项目来减少评分数据的稀疏性^[3]. 文献[47]中依据项目间的关系, 将用户-项目矩阵转换成用户-类别矩阵, 刻画用户对各类别项目进行评分, 降低评分矩阵的稀疏度. 文献[58]改进了基于项目的 top-N 推荐算法, 进行预测时不仅考虑项目的 K 个近邻, 还考虑此项目的影响集, 通过项目的近邻和影响集进行预测. 另外, 通过引进其他的数据源, 如用户社交数据, 文献[33]通过用户之间的朋友关系、用户参加的群组关系, 文献[59]通过采用用户之间的信任关系等, 基于这些社交数据产生推荐, 在一定程度上解决了稀疏性问题.

2.2 可扩展性问题

当商业推荐系统的用户数量或项目数量达到百万甚至千万级别时, 推荐算法将面临严重的可扩展问题. 许多在线推荐网站需要对用户产生及时的推荐结果, 对推荐的实效性要求更高. 而现有的大多数推荐算法都

不具有可扩展性.例如,协同过滤算法的时间复杂度为 $O(n^2m)$,对于百万的用户数 n 和项目数 m 而言,计算开销非常大.采用降维、聚类、分类等策略可在一定程度上解决可扩展性问题.例如,SVD等降维技术可压缩矩阵,并能产生较好的推荐结果,但需要花费大量的时间分解矩阵;基于最近邻的 KNN 算法^[8],每次只考虑与目标用户相似度最大的、或相似度高于某一阈值的近邻,可在一定程度减少推荐过程的时间开销,文献[60]提出了一种新的最近邻度量——相似度支持度,并提出了数中能在保持推荐精度的前提下维持合理规模的 k 近邻策略;基于模型的 CF 算法,如聚类协同过滤算法^[61-62],将偏好相似的用户归类,在进行推荐时只考虑与目标用户同在一类中的其他用户.

2.3 特征提取问题

对文本类信息的自动特征提取技术相对成熟.但互联网的大量信息以多媒体形式提供,仅为用户推荐文本信息远远不够.由于多媒体信息自动特征提取技术的限制,多媒体信息推荐研究进展缓慢.目前对多媒体信息的推荐多基于用户对多媒体信息的手工标注信息产生推荐.对文本类信息进行推荐时,还面临过拟合的问题.例如,基于内容的推荐算法只推荐与目标用户偏好文档相似度高的文本,因而为用户推荐的项目在内容上十分相似,推荐的多样性较差.可通过混合推荐方法提高基于内容推荐的多样性.

2.4 其他问题

除以上主要问题外,推荐系统还面临其他问题.例如托攻击问题^[63]、隐私问题^[64].许多用户不愿提交自己的评分数据或历史浏览数据.可以通过提升推荐系统的信誉度,并建立和采取相应的制度保护用户的隐私数据解决该问题.此外,在对用户产生推荐时,还需为用户提供“为什么向我推荐这个产品”的合理解释^[65].例如,Amazon 会向用户提供诸如“为您推荐原因是由于您浏览过/买过…”的信息.然而,对现有的多数推荐算法还不能给出令人满意的推荐原因.

3 结论

推荐系统是帮助用户解决信息过载问题的有效工具,已被广泛应用于包括电子商务在内的多个领域.协同过滤、基于内容推荐、基于图结构推荐和混合推荐是目前较为常见的推荐方法.本文综述了相关算法,分析了各自方法的优势与不足,进一步分析了推荐系统中常用的相似性计算方法和评分估计方法,总结了常被采用的推荐算法评价方法和评价准则.

经过 20 多年的发展,推荐技术取得了长足进步,然而现有的推荐算法仍面临诸多困难与挑战,其中数据稀疏、过拟合、可扩展性和多媒体信息特征提取是主要问题^[1,66].现有的技术和方法都不能从根本上解决这些问题.随着应用领域的不断拓展,推荐系统还会面临新的需求与问题.推荐系统的发展与它面临的问题和挑战密不可分,针对以上问题的推荐方法研究仍是信息检索、数据挖掘和机器学习等智能信息处理领域的研究热点.

参考文献:

- [1] Adomavicius G, Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions[J]. *IEEE Transaction on Knowledge and Data Engineering*, 2005, 17(6): 734-749.
- [2] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究与发展[J]. *自然科学进展*, 2009, 19(1): 1-15.
- [3] Breese J S, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[C]//Process of the 14th Conference on Uncertainty in Artificial Intelligence, 1998: 43-52.
- [4] Resnick P, Iakovou N, Sushak M, et al. GroupLens: An open Architecture for Collaborative Filtering of Netnews[C]//Process of the 1994 Computer Supported Cooperative Work Conference, 1994: 175-186.
- [5] Sarwar B, Karypis G, Konstan J. Item-based Collaborative Filtering Recommendation Algorithms[C]//Process of the 10th International Conference on World Wide Web, 2001: 285-295.
- [6] Linder G, Smith B, York J. Amazon. com Recommendations: Item-to-Item Collaborative Filtering[J]. *IEEE Internet Computing*, 2003, 7(1): 76-80.
- [7] Karypis G. Evaluation of Item-based Top-N Recommendation Algorithms[C]//Process of the International Conference on Information and Knowledge Management, 2001: 247-254.

- [8] Deshpande M, Karypis G. Item-based top-N Recommendation Algorithms[J]. *ACM Transaction on Information Systems*, 2004, **22**(1):143-177.
- [9] Chen Y H, George E I. A Bayesian Model for Collaborative Filtering[C]//Process of the 7th International Workshop on Artificial Intelligence and Statistics, 1999.
- [10] Ungar L H, Foster D P. Clustering Methods for Collaborative Filtering[C]//Process of the Workshop on Recommendation System, 1998.
- [11] Billsus D, Pazzani M. Learning Collaborative Information Filters[C]//Process of International Conference on Machine Learning, 1998:46-54.
- [12] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[M]. San Francisco: Morgan Kaufmann, 1988.
- [13] Canny J. Collaborative Filtering with Privacy Via Factor Analysis[C]//Process of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002:238-245.
- [14] Lemire D, Maclachlan A. Slope one Predictors for Online Rating-based Collaborative Filtering[C]//Process of the SIAM Data Mining Conference, 2005.
- [15] Shani G, Heckerman D, Brafman R. An MDP-based Recommender System[J]. *The Journal of Machine Learning Research*, 2005, **6**:1265-1295.
- [16] Ungar L H, Foster D P. Clustering Methods for Collaborative Filtering[C]//Process of the Workshop on Recommender Systems, Menlo Park: AAAI Press, 1998:112-125.
- [17] Getoor L, Sahami M. Using Probabilistic Relational Models for Collaborative Filtering[C]//Process of the Workshop Web Usage Analysis and User Profiling, 1999.
- [18] Pavlov D, Pennock D. A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-dimensional Domains[C]//Process of 16th Annual Conference on Neural Information Processing System, 2002:1441-1448.
- [19] Salton G. Automatic Text Processing[M]. New York: Addison Wesley, 1989.
- [20] Balabanovic M, Shoham Y. Fab: Content-based Collaborative Recommendation[J]. *Communications of the ACM*, 1997, **40**(3):66-72.
- [21] Pazzani M, Billsus D. Learning and Revising user Profiles: The Identification of Interesting Web Sites[J]. *Machine Learning*, 1997, **27**(3):313-331.
- [22] Deerwester S, Dumais S, Furnas G, et al. Indexing by Latent Semantic Analysis[J]. *The Journal of the American Society for Information Science*, 1990, **40**(6):391-407.
- [23] Hofmann T. Latent Semantic Models for Collaborative Filtering[J]. *ACM Transaction Information Systems*, 2004, **22**(1):89-115.
- [24] Hofmann T. Unsupervised Learning by Probabilistic Latent Semantic Analysis[J]. *Machine Learning*, 2001, **42**:177-196.
- [25] Somlo G, Howe A. Adaptive Lightweight Text Filtering[C]//Process of the 4th International Symposium on Intelligent Data Analysis, 2001:319-329.
- [26] Zhang Yi, Callan J, Minka T. Novelty and Redundancy Detection in Adaptive Filtering[C]//Process of 25th Annual International ACM SIGIR Conference, 2002:81-88.
- [27] Roberson S, Walker S. Threshold Setting and Performance Optimization in Adaptive Filtering[J]. *Information Retrieval*, 2002, **5**:239-256.
- [28] 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.
- [29] Park H S, Yoo J O, Cho S B. A Context-aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory[J]. *Fuzzy System and Knowledge Discovery*, 2006, **4223**:970-979.
- [30] Mooney R J, Bennett P N, Roy L. Book Recommending Using text Categorization with Extracted Information[C]//Process of AAAI'98/ICML'98 Workshop on Learning for Text Categorization, 1998:49-54.
- [31] Zhou Tao, Ren J, Medo M. Bipartite Network Projection and Personal Recommendation[J]. *Physical Review E*, 2007, **76**:046115.
- [32] Zhou Tao, Zoltan Kuscik, Liu Jian-guo. Solving the Apparent Diversity-accuracy Dilemma of Recommender Systems[J]. *In Process of the National Academy of Sciences*, 2010, **107**(10):4511-4517.

- [33] Yuan Quan, Zhao Shi-wan, Li Chen, *et al.* Augmenting Collaborative Recommender by Fusing Explicit Social Relationships[C]//ACM Conference on Recommender System Workshop on Recommender Systems and the Social Web, 2009.
- [34] 张少中, 陈德人. 面向个性化推荐的两层混合图模型[J]. 软件学报, 2009, 20(zk): 123-130.
- [35] Liu Jie, Shang Ming-sheng, Chen Duan-bing. Personal Recommendation Based on Weighted Bipartite Networks[C]//process of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, 2009.
- [36] Cattuto C, Loreto V, Pietronero L. Semantic Dynamics and Collaborative Tagging[J]. *In Process of the National Academy of Sciences*, 2007, 104(5): 1461-1464.
- [37] Zhang Zike, Lv Lin-yuan, Liu Jian-guo, *et al.* Empirical Analysis on a Keyword-based Semantic System[J]. *The European Physical Journal B*, 2008, 66(4): 557-561.
- [38] Yu K, Schwaighofer A, Tresp V, *et al.* Probabilistic Memory-based Collaborative Filtering[J]. *IEEE Transaction on Knowledge and Data Engineering*, 2004, 16(1): 56-69.
- [39] Pennock D M, Horvitz E, Lawrence S, *et al.* Collaborative Filtering by Personality Diagnosis: a Hybrid memory- and Model-based Approach[C]//Process of the 16th Conference on Uncertainty in Artificial Intelligence, 2000: 473-480.
- [40] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases[C]//Process of the ACM SIGMOD Conference on Management of Data, 1993: 207-216.
- [41] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules[C]//Process of the International Conference on Very Large Databases, 1994: 487-499.
- [42] Han Jia-wei, Pei J, Yin Y, *et al.* Mining Frequent Patterns Without Candidate Generation[J]. *Data Mining and Knowledge Discovery*, 2004, 8: 53-87.
- [43] Burke R. Knowledge-Based Recommender Systems[J]. *Encyclopedia of Library and Information Systems*, 2000, 69(32): 180-200.
- [44] Shardanand U, Maces P. Social Information Filtering: Algorithms for Automating "Word of Mouth"[C]//Process of Conference on Human Factors in Computing Systems, 1995: 210-217.
- [45] Rong Jin, Luo Si. A Study of Methods for Normalizing User Ratings in Collaborative Filtering[C]//Process of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004.
- [46] Rong Jin, Luo Si, Zhai Cheng-xiang. Collaborative Filtering with Decoupled Models for Preferences and Rating[C]//Process of the 12th International Conference on Information and Knowledge Management, 2003: 309-316.
- [47] Zeng Chun, Xing Chun-xiao, Zhou Li-zhu. Similarity Measure and Instance Selection for Collaborative Filtering[C]//Process of the 12th International Conference on World Wide Web, 2003.
- [48] 张光卫, 李德毅, 李鹏, 等. 基于云模型的协同过滤推荐算法[J]. 软件学报, 2007, 18(10): 2403-2411.
- [49] Delgado J, Ishaii N. Memory-based Weighted-majority Prediction for Recommender Systems[C]//Process of the ACM SIGIR'99 Workshop Recommender Systems: Algorithms and E-Valuation, 1999.
- [50] Nakamura A, Abe N. Collaborative Filtering Using Weighted Majority Prediction Algorithms[C]//Process of the 15th International Conference on Machine Learning, 1998: 395-403.
- [51] Yi Ding, Xue Li. Time Weight Collaborative Filtering[C]//Process of the 14th ACM International Conference on Information and Knowledge Management, 2005.
- [52] Yi Ding, Xue Li, Maria E. Recency-based Collaborative Filtering[C]//Process of the 17th Australasian Conference, 2006.
- [53] Lee T Q, Paik Y. A Time-based Approach to Effective Recommender Systems Using Implicit Feedback[J]. *Expert System with Applications*, 2008, 34(4): 3055-3062.
- [54] Chen Yen-liang, Cheng Li-chen. A Novel Collaborative Filtering Approach for Recommending Ranked Items[J]. *Expert Systems with Applications*, 2008, 34(4): 2396-2405.
- [55] Fouss F, Pirotte A, Renders J, *et al.* Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2007, 19(3): 355-369.
- [56] Herlocker J L, Konstan J A, Terveen L G, *et al.* Evaluating Collaborative Filtering Recommender Systems[J]. *ACM Transactions on Information Systems*, 2004, 22: 5-53.
- [57] Goldberg K, Toeder T, Gupta D, *et al.* Eigentaste: a Constant time Collaborative Filtering Algorithm[J]. *Information Retrieval*, 2001, 4: 133-151.

- [58] 陈健,印鉴. 基于影响集的协同过滤推荐算法[J]. 软件学报,2007,18(7):1685-1694.
- [59] Pitsilis G,Knapskog S J. Social Trust as a Solution to Address Sparsity-inherent Problems of Recommender Systems [C]//ACM Recommender System Workshop on Recommender System & The Social Web,2009.
- [60] 罗辛,欧阳元新,熊璋,等. 通过相似度支持度优化基于 K 近邻的系统过滤算法[J]. 计算机学报,2010,33(8):1437-1445.
- [61] Connor M O,Herlocker J L. Clustering Items for Collaborative Filtering[C]//Process of the ACM SIGIR Workshop on Recommender Systems,1999.
- [62] Sarwar B M,Karypis G,Konstan J A,*et al.* Recommender Systems for Large-scale E-commerce:Scalable Neighborhood Formation Using Clustering[C]//Process of the 5th International Conference on Computer and Information Technology,2002.
- [63] 李聪,骆志刚,石金龙. 一种探测推荐系统攻击的无监督算法[J]. 自动化学报,2011,37(2):160-167.
- [64] 张锋,常会友. 基于分布式数据的隐私保持协同过滤推荐研究[J]. 计算机学报,2006,29(8):1487-1495.
- [65] Herlocker J L,Konstan J A,Riedl J. Explaining Collaborative Filtering Recommendations[C]//Process of the ACM Conference on Computer Supported Cooperative Work,2000.
- [66] 徐海玲,吴潇,李晓东,等. 互联网推荐系统比较研究[J]. 软件学报,2009,20(2):350-362.

Review of the Art of Recommendation Algorithms

YANG Bo^{1,2}, ZHAO Peng-fei^{1,2}

(1. College of Computer Science and Technology, Jilin University, Changchun 130012, China;

2. Key Laboratory of Symbolic Computation and Knowledge Engineering
of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract: Recommendation is one of primary ways to addressing the problem of Internet information overload, and has been applied to many areas including E-business. Although there exist different kinds of recommendation algorithms and systems, we are still faced some key challenges which have not been well solved for constructing more intelligent and more robust recommender systems, and the involved research is still the hot point of intelligent information processing. We describe the background and significance of the studies of recommendation techniques, and then introduce main recommendation algorithms including collaborative filtering recommendation algorithms, content based recommendation algorithms, graph structure based recommendation algorithms, and hybrid recommendation algorithms, moreover, discuss the advantages and disadvantages of existing algorithms, and finally conclude main evaluation methods and main issues, improvements and future research directions.

Key words: information overload; recommender system; collaborative filtering; information retrieval; data mining; machine learning