

# 一种压缩稀疏用户评分矩阵的协同过滤算法

侯翠琴, 焦李成, 张文革

(西安电子科技大学 智能信息处理研究所;

智能感知与图像理解教育部重点实验室, 陕西 西安 710071)

**摘要:** 提出了一种通过降低用户评分矩阵维数来解决数据稀疏问题的协同过滤算法(基于项目多类属概率潜在语义的协同过滤算法). 首先将概率潜在语义分析法中的隐变量集固定为项目的多类属集, 明确隐变量的意义, 限制隐变量的变化范围; 而后迭代学习隐变量的分布, 即用户的兴趣模型, 压缩用户评分矩阵; 最后用学到的兴趣模型度量用户的相似度, 对目标用户做出推荐. 仿真实验结果表明: 该算法有效解决了数据稀疏问题, 平均绝对误差低于基于记忆的协同过滤算法 4%; 与通过概率潜在语义分析法降低用户评分矩阵维数来解决数据稀疏问题的协同过滤算法相比, 该算法明确了隐变量的意义, 提高了对系统的理解, 并取得了富有竞争力的推荐性能.

**关键词:** 项目多类属; 概率潜在语义分析; 迭代方法; 协同过滤; 算法

**中图分类号:** TP181 **文献标识码:** A **文章编号:** 1001-2400(2009)04-0614-05

## Collaborative filtering algorithm via compressing the sparse user-rating-data matrix

HOU Cui-qin, JIAO Li-cheng, ZHANG Wen-ge

(Ministry of Education Key Lab. of Intelligent Perception and Image Understanding

Research Inst. of Intelligent Information Processing, Xidian Univ., Xi'an 710071, China)

**Abstract** The paper proposes a novel memory-based collaborative filtering algorithm—Multi-label Probabilistic Latent Semantic Analysis based Collaborative Filtering, which improves the quality of recommendations by reducing the dimension of the user-rating-data matrix by multi-label probabilistic latent semantic analysis when the matrix is extremely sparse. Firstly, it confines the set of latent variables of probability latent semantic analysis to the set of multi-label of items to make latent variables have meanings of corresponding labels. Then it learns the probabilistic distribution of latent variables, i. e., the model of use's interest, to compress the user-rating-data matrix. Finally, it computes the similarity between different users based on the above learned model and makes recommendations. Compared to memory-based collaborative filtering algorithms, the proposed algorithm decreases the mean absolute error 4 percents averagely on test dataset by reducing the dimension of the user-rating-data matrix. The proposed algorithm makes the recommendation system understandable and obtains competitive recommendations compared to the filtering algorithm which reduces the dimension of the user-rating-data matrix by probabilistic latent semantic analysis.

**Key Words:** multi-label of items; probabilistic latent semantic analysis; iterative method; collaborative filtering; algorithms

随着信息技术的飞速发展, 网络上的资源呈指数膨胀, 推荐技术日益受到用户的青睐, 也受到越来越多的学者和电子商务网站的关注. 学者们提出众多推荐算法, 其中协同过滤算法<sup>[1]</sup>是被广泛关注的一类算法.

收稿日期: 2008-06-10

基金项目: 国家自然科学基金资助(60703107, 60703108, 60703109, 60702062); 国家 863 项目资助(2006AA01Z107, 2007AA12Z136, 2007AA12Z223); 973 项目资助(2006CB705700); 教育部长江学者和创新团队支持计划资助(IRT0645)

作者简介: 侯翠琴(1983-), 女, 西安电子科技大学博士研究生, E-mail: houcuiqin0304@163.com.

协同过滤算法认为对一些产品具有相似评价的用户,对别的产品也有相似的喜好程度.这类算法主要利用用户喜好的相似性向用户推荐产品.

协同过滤算法主要分为基于模型的协同过滤算法和基于记忆的协同过滤算法.基于模型的协同过滤算法是从所有用户对产品的评价数据库中学习一个预测模型<sup>[2]</sup>,然后利用学到的模型向用户推荐.而基于记忆的协同过滤算法<sup>[3]</sup>则是根据用户的历史信息,估算用户之间的相似性,然后找出目标用户的几个最近邻居,最后加权计算最近邻居对产品的评价,对目标用户做出推荐.基于记忆的协同过滤推荐算法以其简单性、高推荐质量,成为比较成功的一类推荐算法.该算法在越多的用户对越多的产品做出评价后,性能会越好,但是在该方法取得一定成功的同时,也存在一些瓶颈问题.例如,当用户评分数据比较稀疏时,推荐系统性能会急剧下降<sup>[3]</sup>.在系统刚开始运行时,每个用户都对极少数产品做出评价,数据极端稀疏,不同用户之间很少有共同评价的产品.基于记忆的协同过滤推荐算法无法给用户做出有效推荐,这种情况称为系统的“冷启动”,是任何一个系统的必经阶段.

针对基于记忆的协同过滤推荐算法在数据稀疏情况下推荐质量差的问题,提出了一种通过压缩用户评分矩阵,来解决此问题的协同过滤算法——基于项目多类属概率潜在语义分析的协同过滤算法.仿真实验结果表明,本算法能有效解决数据稀疏问题,提高基于记忆的协同过滤推荐算法的推荐精度.与直接通过概率潜在语义分析(probabilistic Latent Semantic Analysis,简称为 pLSA)降低用户评分矩阵维数来解决数据稀疏问题的推荐算法相比,本算法明确了隐变量的含义,并取得了较好的推荐效果.

## 1 背景知识

### 1.1 基于记忆的协同过滤推荐算法

基于记忆的协同过滤算法主要有3步骤:计算用户的相似度、选择最近邻、产生推荐.度量用户相似度的方法有很多种,最常用的为相关相似性<sup>[4]</sup>.

**定义1 相关相似性** 用户*i*的评分项目集为 $I_i$ ,用户*j*的评分项目集为 $I_j$ ,而 $I_{ij} = I_i \cap I_j$ 表示用户*i*和*j*的共同评分的项目集.用户*i*和用户*j*的相关相似性为

$$R_{\text{sim}}(i, j) = \frac{\sum_{t \in I_{ij}} (M_{it} - \bar{M}_i)(M_{jt} - \bar{M}_j)}{\sqrt{\left( \sum_{t \in I_{ij}} (M_{it} - \bar{M}_i)^2 \right) \left( \sum_{t \in I_{ij}} (M_{jt} - \bar{M}_j)^2 \right)}} \quad (1)$$

其中 $M_{it}$ 表示用户*i*对项目*t*的评分, $\bar{M}_i = \frac{\sum_{t \in I_{ij}} M_{it}}{|I_{ij}|}$ , $\bar{M}_j = \frac{\sum_{t \in I_{ij}} M_{jt}}{|I_{ij}|}$ ,即用户*i*和用户*j*在共同评分项目集中的平均值.

计算出用户间的相似性后,可以按相似性大小找出目标用户的前*N*个最近邻,而后预测用户*i*对未评分项目*t*的评分:

$$M_{it} = \bar{M}_i + \sum_{j \in N_i} (M_{jt} - \bar{M}_j) \times R_{\text{sim}}(i, j) \Big/ \sum_{j \in N_i} R_{\text{sim}}(i, j) \quad (2)$$

其中 $N_i$ 是用户*i*的最近邻集, $\bar{M}_i$ 和 $\bar{M}_j$ 分别表示用户*i*和用户*j*对项目评分的平均值.最后按预测评分高低对用户推荐项目.

### 1.2 概率潜在语义分析法

概率潜在语义分析(probabilistic Latent Semantic Analysis——pLSA)<sup>[4-5]</sup>和基于SVD分解的潜在语义分析<sup>[6]</sup>一样,都是分析稀疏文本-单词共生矩阵,通过降低矩阵维数解决一义多词和一词多义等问题的一种统计技术,已被广泛应用于信息检索等领域.但与潜在语义分析相比,概率潜在语义分析有更坚实的统计理论基础.

pLSA的核心是一个被广泛接受的文本产生式模型示象模型(aspect model),它定义了文本-单词对 $(d_i, w_j)$ 的产生过程.用 $D = \{d_1, d_2, \dots, d_m\}$ 表示文本集, $W = [w_1, w_2, \dots, w_n]$ 表示单词集,用 $m \times n$ 维矩阵 $C$ 表示共生矩阵,其元素 $C_{ij} = w_{ij}$ 表示文本 $d_i$ 中单词 $w_j$ 发生的次数.则示象模型定义共生矩阵元素文本-单词对 $w_{ij} = (d_i, w_j)$ 的产生过程为:首先以概率 $P(d_i)$ 选择文本 $d_i$ ,而后以概率 $P(z_k | d_i)$ 选择隐变量 $z_k$ ,最后以概率 $P(w_j | z_k)$ 产生单词 $w_j$ .

基于此产生过程, 可得  $P(d_i, w_j) = P(d_i)P(w_j | d_i)$ ,  $P(w_j | d_i) = \sum_{z \in Z} P(w_j | z)P(z | d_i)$ . 因此文本集  $D$  的 log 似然函数  $L = \sum_{i=1}^m \sum_{j=1}^n n(d_i, w_j) \log P(d_i, w_j)$ , 其中  $n(d_i, w_j)$  表示文本  $d_i$  中单词  $w_j$  发生的次数,  $P(d_i, w_j) = \sum_{z \in Z} P(d_i)P(z | d_i)P(w_j | z)$ . 由于隐变量  $z$  无法从文本集中观测到, 因此概率潜在语义分析法采用 EM 算法<sup>[7-8]</sup> 最大化 log 似然函数  $L$  求得模型中参数  $P(d_i)$ ,  $P(z | d_i)$  和  $P(w_j | z)$  的估计值. 而后可用  $P(d_i, z)$  代替  $w_{ij} = (d_i, w_j)$  来近似表示文本, 达到降低文本特征维数的目的.

2 基于项目多类属 pLSA 的协同过滤算法

在传统的分类任务中, 类别是互斥不相容的, 一个个体最多只能属于其中一类. 而对于多类属分类问题<sup>[9-10]</sup>, 类别是不互斥的, 即一个个体可以同时属于几个不同的类别. 例如, 一幅图片包括沙滩、日落以及人群等, 那么它就可以同时属于景色类和人物类. 现实中, 很多个体从不同的角度看, 它将属于不同的类别. 而在推荐系统中, 如图片, 音乐, 视频等项目的类属一般是多类属的, 又是已知的. 因此笔者尝试将 pLSA 中的隐变量集固定为项目的已知多类属集, 限定隐变量的意义及变化范围.

2.1 基于项目多类属的 pLSA

为了简单地表示和直观地理解, 以下将系统中的项目具体化为电影, 但所提算法适合任何项目. 把协同过滤推荐系统中的  $m$  个用户组成的用户集  $U = [u_1, u_2, \dots, u_m]$  看作是信息检索领域中的文本集合, 同时把  $n$  部电影组成的电影集  $M = [m_1, m_2, \dots, m_n]$  看作是单词集合, 则用户评分矩阵  $V(m, n)$  对应文本-单词共生矩阵. 应用 pLSA 法, 可得:

$$P(u_i, m_j) = P(u_i)P(m_j | u_i) \quad , \tag{3}$$

$$P(m_j | u_i) = \sum_{z \in Z} P(m_j | z)P(z | u_i) \quad . \tag{4}$$

由于提前不知道最优隐变量的数目, 因此在用 EM 估计模型参数时, 同时需要以隐变量  $z$  的数目为参数, 比较在不同个隐变量下得到的 log 似然函数值. 另外, 在推荐系统中给用户推荐产品时, 需要解释推荐理由, 而 pLSA 中隐变量的意义不明确. 基于此, 笔者考虑将 pLSA 中的隐变量集固定为项目的多类属集, 克服上述 pLSA 的两个缺点. 以推荐系统中的电影为例, 则 pLSA 中的隐变量集被固定为电影所属类别的集合  $S$ . 若电影  $m_j$  是教育类的儿童剧, 又有科幻性, 则用户  $u_i$  喜欢电影  $m_j$  的程度——公式(4)可理解为: 电影  $m_j$  以一定的概率属于儿童剧类、教育类以及科幻类, 而用户  $u_i$  又以一定的概率喜欢这几类电影, 最后综合这部电影所属的这几个类别可得用户  $u_i$  喜欢电影  $m_j$  的程度. 用户  $u_i$  喜欢电影  $m_j$  的概率可表示为

$$P(m_j | u_i) = \sum_{s \in S_j} P(m_j | s)P(s | u_i) \quad , \tag{5}$$

其中,  $S_j$  为电影  $m_j$  的类属集.

把隐变量集固定为项目的类属集, 将隐变量的意义明确化, 同时也限定了隐变量的变化范围. 如上例中, 只有  $s \in S_j$  时,  $P(m_j | s)$  才大于 0, 否则等于 0, 其中  $S_j$  为 {儿童剧类, 教育类, 科幻类}. 即若  $s \notin S_j$ , 则  $P(m_j | s) = 0$ . 因此在用 EM 算法最大化 log 似然函数时,  $P(m_j | s)$  为 0 的项应始终保持为 0. 基于项目多类属的 pLSA 的执行步骤如下:

Step 1 对  $P(s | u_i, m_j)$  赋 0 到 1 之间的随机数, 并归一化使  $\sum_{s \in S} P(s | u_i, m_j) = 1$ ;

Step 2 计算 
$$P(s | u_i) = \frac{\sum_{m_j \in M} P(s | u_i, m_j)}{\sum_{m_j \in M, s \in S_j} P(s | u_i, m_j)}, \quad P(m_j | s) = \begin{cases} \frac{\sum_{u_i \in U} P(s | u_i, m_j)}{\sum_{u_i \in U, m_j \in M} P(s | u_i, m_j)}, & \text{若 } s \in S_j, \\ 0, & \text{其他;} \end{cases}$$

$$\text{Step 3 计算 } P(s|u_i, m_j) = \begin{cases} \frac{P(s|u_i)P(m_j|s)}{\sum_{s \in S_j} P(s'|u_i)P(m_j|s')} & , \quad \text{若 } s \in S_j \\ 0 & , \quad \text{其他;} \end{cases}$$

Step 4 判断算法结束条件是否满足, 如果满足, 则结束, 否则转 Step 2.

基于项目多类属的 pLSA 中, Step 2 是 EM 算法在最大化似然函数时的  $M$  步, 而 Step 3 是  $E$  步. 基于项目多类属的 pLSA 设置足够大的迭代次数来保证算法收敛. 在算法结束后, 可以得到用户兴趣模型的估计  $P(s|u_i)$ , 并可用此来近似表示用户评分矩阵, 达到压缩稀疏用户评分矩阵的目的.

## 2.2 基于项目多类属 pLSA 的协同过滤算法

基于项目多类属 pLSA 的协同过滤算法与基于记忆的协同过滤推荐算法基本步骤类似, 但在计算用户的相似度时, 用基于项目多类属的 pLSA 算法所得的隐变量分布, 即用户兴趣模型估计来计算用户的相似度. 另外, 由于每个目标用户所受相邻用户的影响不同, 笔者考虑用阈值法来确定目标用户的最近邻. 基于项目多类属 pLSA 的协同过滤算法概述如下:

Step 1 调用基于项目多类属的 pLSA, 获得用户兴趣模型估计  $P(s|u_i)$ ;

Step 2 计算用户间的相似度  $R_{\text{sim}}(i, j) = \sum_{t=1}^T \left[ P(z_t|u_i) \log \frac{P(z_t|u_i)}{P(z_t|u_j)} + P(z_t|u_j) \log \frac{P(z_t|u_j)}{P(z_t|u_i)} \right]$ ;

Step 3 确定目标用户  $u_i$  的最近邻. 若  $R_{\text{sim}}(i, j) > \alpha$ , 则用户  $u_j$  为  $u_i$  的最近邻;

Step 4 按公式  $M_{it} = \overline{M}_i + \frac{\sum_{j \in N_i} (M_{jt} - \overline{M}_j) \times R_{\text{sim}}(i, j)}{\sum_{j \in N_i} R_{\text{sim}}(i, j)}$  预测未评分项目的评分, 对目标用户做出推荐.

基于项目多类属的 pLSA 每一次迭代的时间复杂度为  $O(mn)$ . 而基于项目多类属 pLSA 的协同过滤算法要对目标用户  $u_i$  推荐项目时, 要计算  $u_i$  与别的用户的相似度, 复杂度为  $O(m)$ , 推荐项目时要预测  $u_i$  对所有未评分项目的评分, 复杂度为  $O(n)$ . 因此基于项目多类属 pLSA 的协同过滤算法要对目标用户  $u_i$  推荐项目时, 总时间复杂度为  $O(N \times m \times n)$ ,  $N$  为基于项目多类属的 pLSA 的迭代次数.

## 3 仿真实验

### 3.1 数据集及评价准则

用 Minnesota 大学进行 GroupLens Research 项目时的 MovieLens 数据 (<http://www.grouplens.org/node/73>) 来评价算法性能. MovieLens 数据包括 3 952 部电影和 6 040 个用户的基本信息, 以及用户对这些电影的 100 多万条评分数据. 其中电影的总类属数为 18, 用户的评分为 1 到 5. 本文中主要研究系统刚启动时的数据稀疏问题, 因此按照一般用户的习惯, 假定每个用户的评分数据为 10~30 条, 平均为 20 条. 在随机确定每个用户的评分数据条数后, 从所有的评分数据中随机选择出评分数据, 生成实验所用的整个数据集, 并分别取前 1 000, 2 000, 3 000 及全部用户的数据生成 4 个包含不同个用户的数据集. 对于每个数据集, 取前 90% 用户的全部评分数据及后 10% 用户对 ID 号小于 3 500 电影的评分数据作为训练数据, 剩余数据作为测试数据.

笔者采用被广泛接受的平均绝对误差 MAE (Mean Absolute Error) 来衡量算法的推荐质量, 其定义为

$$\text{MAE} = \sum_{i=1}^N |p_i - q_i| / N, \text{ 其中, } p_i \text{ 是用户的实际评分, } q_i \text{ 是系统对评分的预测值, } N \text{ 为总的预测评分条数.}$$

显然, MAE 越小, 系统的推荐误差越小.

### 3.2 实验结果

对于每个数据集, 分别运行传统基于记忆的协同过滤推荐算法、基于 pLSA 的协同过滤推荐算法和笔者所提算法. 对于传统基于记忆的协同过滤推荐算法, 用最常用的 Pearson 相关系数来度量用户的相似性. 基于 pLSA 的协同过滤算法中取隐变量数分别为 10, 30, 40 及 50. 笔者所提算法中, 隐变量数与项目的总类属数相同 (为 18). 对于基于 pLSA 的协同过滤推荐算法及笔者所提算法, 设置 EM 算法的迭代代数 50 代,

保证算法的收敛,并取算法运行 30 次的平均结果.另外,对这 3 个算法在找最近邻时,都采用阈值法.不同的用户在同一阈值下,有不同个最近邻.阈值  $\alpha$  的取值为  $\alpha = [(s_{\max} - s_{\min}) / 30] \times n, n = 0, 1, 2, 3, \dots, 29$ ,其中,  $s_{\max}$  和  $s_{\min}$  分别为所有用户间相似度的最大值和最小值,  $n$  控制阈值  $\alpha$  的大小,  $n$  越大,  $\alpha$  越接近用户间最大相似度  $s_{\max}$ .

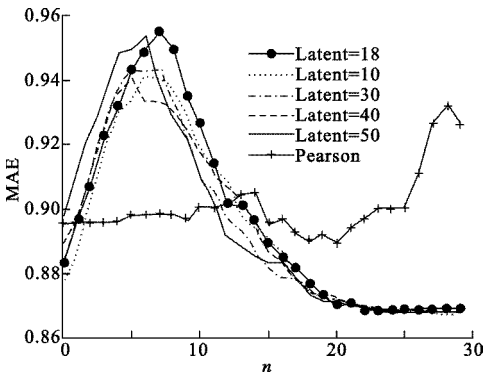


图 1 平均推荐误差比较(用户数为 1000)

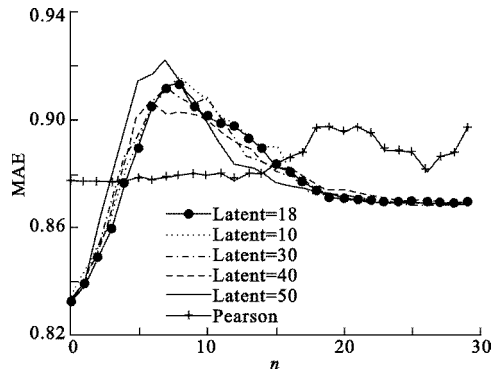


图 2 平均推荐误差比较(用户数为 2000)

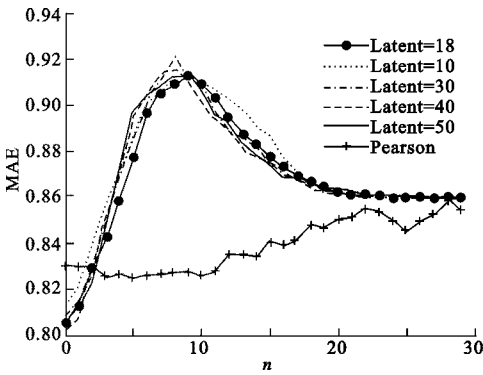


图 3 平均推荐误差比较(用户数为 3000)

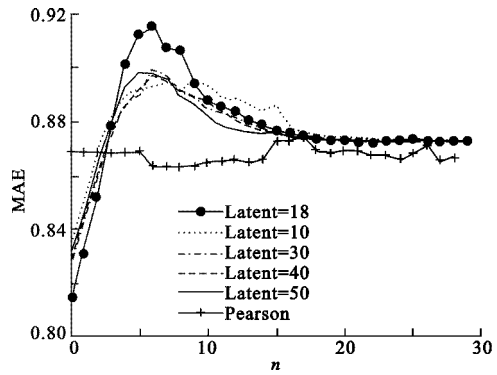


图 4 平均推荐误差比较(用户数为 6040)

3 个算法在 4 个不同数据集上运行的结果如图 1~4 所示.图 1~4 依次表示用户数为 1000, 2000, 3000 和 6040 时各推荐算法在不同阈值  $\alpha$  下的平均推荐误差.比较各图可知,对于不同的数据集,使算法推荐误差最小的阈值各不相同,一般小的阈值可使平均预测评分误差小.本文算法和基于 pLSA 的协同过滤推荐算法的最小推荐误差都远小于基于记忆的协同过滤推荐算法的推荐误差.本文算法的推荐误差比基于记忆的协同过滤推荐算法的推荐误差平均低 4%.这是由于传统的协同过滤算法根据用户对项目的评分直接计算用户的相似度,但在评分数据极端稀疏情况下,用户之间很少有共同的评分项目,不能准确体现用户的相似度.对于用户数为 6040 的数据集,本文算法所得的最小推荐误差则明显小于基于 pLSA 的协同过滤推荐算法在不同个隐变量下所得的最小推荐误差.图 5 更清晰地比较了基于 pLSA 的协同过滤算法在不同隐变量下和笔者所提算法所得的最小推荐误差.

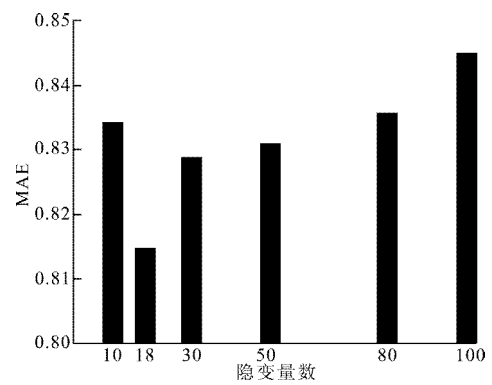


图 5 不同隐变量下平均推荐误差比较

### 4 结 论

笔者提出的基于项目多类属 pLSA 的协同过滤算法是一种通过学习用户模型降低用户评分矩阵维数的

(下转第 638 页)

- [7] 王万林. 非均匀环境下的相控阵机载雷达 STAP 研究[D]. 西安: 西安电子科技大学, 2004: 93-100.  
Wang Wanlin. Study on STAP for Phased Array Airborne Radar in Nonhomogeneous Environment[D]. Xi'an: Doctoral Dissertation Xidian University, 2004: 93-100.
- [8] Dipietro R C. Extended Factored Space-time Processing for Airborne Radar Systems[C]//Proceeding of the 26<sup>th</sup> Asilomar Conference on Signal, System, and Computers. Pacific Grove: IEEE, 1992: 425-430.
- [9] 廖桂生, 保铮. 机载雷达时空二维部分联合自适应处理[J]. 电子科学学刊, 1993, 15(6): 575-580.  
Liao Guisheng, Bao Zheng. A Partial Adaptive Spatial-Temporal Processing for Airborne Radars[J]. Journal of Electronics, 1993, 15(6): 575-580.

(编辑: 高西全)

(上接第 618 页)

基于记忆的协同过滤算法, 有效解决了传统基于记忆的协同过滤算法在用户评分数据极端稀疏, 推荐质量急剧下降的问题, 有一定的应用潜力. 与通过 pLSA 降低维数提高推荐精度的算法相比, 笔者所提算法赋予 pLSA 中隐变量的意义为项目的类属, 提高了系统的可理解性, 为系统的后续解释等工作奠定了基础, 并取得了较好的推荐精度.

#### 参考文献:

- [1] Breese J S, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithm for Collaborative Filtering [C]//Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98). San Francisco: Morgan Kaufmann Publisher, 1998: 43-52.
- [2] 王玲, 薄列峰, 刘芳, 等. 稀疏隐空间支持向量机[J]. 西安电子科技大学学报, 2006, 33(6): 896-901.  
Wang Ling, Bo Liefeng, Liu Fang, et al. Sparse Hidden Space Support Vector Machine[J]. Journal of Xidian University, 2006, 33(6): 896-901.
- [3] Wang J, Vries A P, Reinders M J T. Unifying User-based and Item-based Collaborative Filtering Approaches By Similarity Fusion[C]//Proceedings of the 29th Annual International ACM SIGIR Conference (SIGIR'06). New York: ACM Press, 2006: 501-508.
- [4] Hoffmann T. Probabilistic Latent Semantic Indexing [C]//Proceedings of the 22nd Annual International ACM SIGIR Conference (SIGIR'99). New York: ACM Press, 1999: 50-57.
- [5] Ding G, Li T, Peng W. On the Equivalence Between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing [J]. Computational Statistics & Data Analysis, 2008, 52(8): 3913-3927.
- [6] DeerWester S, Dumais S, Furnas G W, et al. Indexing by Latent Semantic Analysis [J]. Journal of the American Society for Information Science, 1990, 41(6): 391-407.
- [7] Duda R O, Hart P E, Stork D G. Pattern Classification[M]. 2nd Edition. New York: Wiley-Interscience, 2000.
- [8] 乔向东, 王宝树, 李涛. 基于 EM 算法的杂波环境下机动目标跟踪研究[J]. 西安电子科技大学学报, 2003, 30(4): 514-519.  
Qiao Xiangdong, Wang Baoshu, Li Tao. Study of Tracking Maneuvering Targets in Clutter Based on the EM Algorithm [J]. Journal of Xidian University, 2003, 30(4): 514-519.
- [9] Zhang M L, Zhou Z H. ML-KNN: a Lazy Learning Approach to Multi-label Learning [J]. Pattern Recognition, 2007, 40(7): 2038-2048.
- [10] Tsoumakas G, Katakis I. Multi-Label Classification: An Overview [J]. International Journal of Data Warehousing and Mining, 2007, 3(3): 1-13.

(编辑: 高西全)