基于矩阵填充和物品可预测性的协同过滤算法

潘涛涛 [1] 文 峰 [2] 刘勤让 [1]

摘 要 针对传统矩阵填充算法忽略了预测评分与真实评分之间的可信度差异和传统 Top-N 方法推荐精度低等问题, 提出了一种改进的协同过滤算法. 该算法首先利用置信系数 C 区分评分值之间的可信度; 然后提出物品可预测性的概念, 综合物品的预测评分与物品的可预测性进行物品推荐并将其转化为 0-1 背包问题, 从而筛选出最优化的推荐列表. 实验结果表明: 该算法能有效缓解稀疏性的影响, 提高推荐性能, 并且算法具有良好的可扩展性.

关键词 协同过滤, 推荐系统, 预测评分, 相似度, 0-1 背包问题

引用格式 潘涛涛, 文锋, 刘勤让. 基于矩阵填充和物品可预测性的协同过滤算法. 自动化学报, 2017, **43**(9): 1597−1606

DOI 10.16383/j.aas.2017.c160644

Collaborative Filtering Recommendation Algorithm Based on Rating Matrix Filling and Item Predictability

PAN Tao-Tao[1] WEN Feng[2] LIU Qin-Rang[1]

Abstract The traditional matrix filling algorithm ignores the difference between true rating and predictive rating, and there is only one standard on the traditional Top-N recommended method. In order to solve these two problems, an improved collaborative filtering algorithm is proposed. Firstly, the confidence coefficient is used to distinguish the credibility of the ratings. Then, a concept of item predictability is proposed. The program recommends items by comprehensively considering the item′s predictive ratings and the predictability, and transforming the program into the 0-1 knapsack problem so as to select the optimized recommended list. Experimental results show that the algorithm can effectively alleviate the effect of sparsity and improve the performance of the recommendation, and that the optimization algorithm has good expansibility.

Key words Collaborative filtering, recommendation system, predictive ratings, similarity, 0-1 knapsack problem

Citation Pan Tao-Tao, Wen Feng, Liu Qin-Rang. Collaborative filtering recommendation algorithm based on rating matrix filling and item predictability. *Acta Automatica Sinica*, 2017, **43**(9): 1597−1606

随着互联网技术的发展, 数据资源急剧增长, 快速而高效地从海量的数据中获取所需的信息变得日益紧迫, 信息过载问题[1−2] 出现. 在此背景下, 搜索引擎和推荐系统是必不可少的技术. 搜索引擎能够满足用户查找明确目标的需求, 而推荐系统能够满足用户的潜在需求并且根据用户的兴趣提供个性化服务[3]. 当前主流的推荐系统有基于内容的推荐系统和协同过滤推荐系统[4−5]. 基于内容的推荐算法主要采用文本分类技术对物品内容进行分析. 协同过滤算法是基于这样一个假设: 若某些用户过去对一些物品的评分比较相似, 则他们对其他物品的评分也比较相似. 基于此假设, 该算法首先找到目标用户的相似用户作为邻居用户, 然后综合邻居用户对物品的评分来预测目标用户的评分. 由于协同过滤算法不依赖于物品的内容, 能很好解决对图片、音乐、电影等资源的推荐, 所以被广泛应用于推荐系统中[6−7].

尽管协同过滤技术取得了很大的成功, 但同时也面临着严峻的稀疏性问题[8−9], 影响了相似度计算的准确性, 降低了推荐性能. 为了解决上述问题, 国内外学者们提出了许多解决方法, 最简单的方法就是设置固定的缺省值, 利用用户或物品的评分均值等对评分矩阵进行填充[10], 由于这种方法忽略了用户或者物品之间的差异, 可信度不高. 邓爱林等[11] 提出基于物品评分预测的方法对矩阵进行填充. Xu 等[12] 通过加权基于用户和基于物品的预测评分值来填充矩阵. 陈刚等[13] 采用 BP 神经网络进行评分预测并填充阵. Jang 等[14] 提出基于信任传播的矩阵填充方法. Eldar[15] 提出基于云模型的矩阵填充方法. 上述矩阵填充方法虽然在一定程度上缓解了稀疏性的

收稿日期 2016-09-08 录用日期 2017-01-16
Manuscript received September 8, 2016; accepted January 16, 2017
国家高技术研究发展计划 (863 计划) (2014AA01A), 国家自然科学基金 (61572520) 资助
Supported by National High Technology Research and Development Program (863 Program) (2014AA01A), National Natural Science Foundation of China (61572520)
本文责任编委 周涛
Recommended by Associate Editor ZHOU Tao
1. 国家数字交换系统工程技术研究中心 郑州 450002 2. 江南计算技术研究所 无锡 214000
1. China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002 2. Jiangnan Computing Technology Research Institute, Wuxi 214000

影响, 但仍存在以下两个问题: 1) 矩阵填充后未区分预测评分与真实评分之间的差别, 从而影响了推荐性能; 2) 传统 Top-N 推荐精度低, 仅根据预测评分的高低进行物品筛选, 没有考虑物品的可预测性. 例如: 某物品的预测评分比较高, 但是该物品的可预测性非常低, 即大部分用户对此物品的预测评分与真实评分的偏差都非常大, 则此物品预测评分的可信度就比较低, 所以仅根据物品预测评分的高低生成推荐列表不够准确.

针对上述两个问题, 本文首先在矩阵填充后引入置信系数 $C\,(0 < C < 1)$ 来区分预测评分与真实评分之间的可信度, 进而区分不同物品之间的意义大小; 然后提出物品可预测性的概念, 在推荐环节综合物品的预测评分与物品的可预测性, 并将其转化为 0-1 背包问题, 从而筛选出最优化的推荐列表.

本文结构安排: 第 1 节划分了物品的层次, 定义了物品可预测性的概念并设计了基于矩阵填充和物品可预测性的协同过滤算法 MfP-CF; 第 2 节对算法的性能进行分析; 第 3 节通过实验来验证第 2 节的算法性能; 第 4 节总结并指出下一步研究方向.

1 相关概念及定义

针对传统算法存在的问题, 本文划分了物品层次, 定义了评分值可信度、物品可信度以及物品可预测性等概念, 并引入了 0-1 背包问题, 具体描述如下.

1.1 物品层次划分

假设系统中用户集合为 $U = \{u_1, u_2, \cdots, u_m\}$, 物品集合为 $I = \{i_1, i_2, \cdots, i_n\}$, 用户评分信息可以用矩阵 $R \in \mathbf{R}^{m\times n}$ 表示. 其中 $r_{u,i}$ 表示用户 u 对物品 i 的评分, 用 1∼5 这 5 个评分等级来表示用户对物品的偏好程度, 用户 – 物品评分矩阵如式 (1) 所示:

$$R = \begin{pmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{m,1} & \cdots & r_{m,n} \end{pmatrix} \tag{1}$$

假设用户 u 和用户 v 的已评分物品集合分别为 $I_u = \{i|r_{u,i} \neq 0, u \in U, i \in I\}$ 和 $I_v = \{i|r_{v,i} \neq 0, v \in U, i \in I\}$, 则物品集合 I 可以分成 3 个层次: 共同评分物品集合 $I^1_{u,v}$、单一用户评分物品集合 $I^2_{u,v}$、共同未评分物品集合 $I^3_{u,v}$, 如下式所示:

$$I^1_{u,v} = \{i|i \in (I_u \cap I_v)\} \tag{2}$$

$$I^2_{u,v} = \{i|i \in (I_u \cup I_v), i \notin (I_u \cap I_v)\} \tag{3}$$

$$I^3_{u,v} = \{i|i \notin (I_u \cup I_v), i \in I\} \tag{4}$$

$$I = I^1_{u,v} \cup I^2_{u,v} \cup I^3_{u,v} \tag{5}$$

用户 u 和用户 v 的评分物品并集 $I_{u\cup v}$ 如式 (6) 所示:

$$I_{u\cup v} = I_u \cup I_v = I^1_{u,v} \cup I^2_{u,v} \tag{6}$$

1.2 相关定义

定义 1. 评分值可信度 (Rating reliability): 定义填充矩阵中各评分值的可信度, 设为 $tr_{u,i}$. 矩阵经过评分填充后, 用户的评分值可分为真实评分值与预测评分值, 显然两者的可信度是不同的. 本文引入置信系数 $C\,(0 < C < 1)$ 来区分预测评分值与真实评分值之间的差异, 即真实评分的可信度为 1, 预测评分的可信度为 C.

定义 2. 物品可信度 (Item reliability): 定义矩阵填充后, 不同层次物品之间的可信度. 传统矩阵填充算法的目的是将用户间的评分物品并集 $I_{u\cup v}$ 转换成共同评分物品集合即预测单一用户评分物品 $I^2_{u,v}$ 中未评分用户的评分值并填充矩阵. $I_{u\cup v} = I^1_{u,v} \cup I^2_{u,v}$, $I^1_{u,v}$ 中用户对物品的评分值均为真实评分值, 而 $I^2_{u,v}$ 中用户对物品的评分值为一个真实评分值和一个预测评分值, 因此在相似度计算时 $I^1_{u,v}$ 和 $I^2_{u,v}$ 中物品的贡献大小是不同的, 设 $I^1_{u,v}$ 中物品的可信度为 1, $I^2_{u,v}$ 中物品的可信度为 C'. 本文根据评分值的可信度来区分物品的可信度, 则 $C' = C$.

定义 3. 物品可预测性 (Item predictability): 定义该物品预测评分的准确度, 设为 Pr_i. Pr_i 越大, 则此物品预测评分的准确度越高. 根据物品 i 上所有已评分用户的真实评分 $r_{u,i}$ 与预测评分 $R_{u,i}$ 的偏差来计算物品 i 的可预测性. 设物品 i 上已评分用户集合为 $U_i = \{u|r_{u,i} \neq 0, i \in I, u \in U\}$, ε 为常数, 则物品 i 的可预测性 Pr_i 如式 (7) 和 (8) 所示:

$$pr^u_i = \begin{cases} 1, & |R_{u,i} - r_{u,i}| \leq \varepsilon \\ 0, & \text{其他} \end{cases} \tag{7}$$

$$Pr_i = \frac{\sum\limits_{u\in U_i} pr^u_i}{|U_i|} \tag{8}$$

1.3 0-1 背包问题

0-1 背包问题[16] 是一个经典的组合优化问题. 给定 1 个背包和 N 个物品, 其中背包容量为 C, 物品 i 的重量为 w_i, 单价为 p_i, 如何选择装入背包的物品, 使得背包内物品的总价值 V 最大. 0-1 背包问题的数学公式如式 (9) 和 (10) 所示:

$$\begin{cases} \sum\limits_{i=1}^{N} w_i x_i \leq C \\ x_i = 0 \text{ 或 } 1 \end{cases} \tag{9}$$

$$V=\max\sum_{i=1}^{N}p_ix_i \tag{10}$$

其中, x_i 表示是否将物品放入背包, 且物品的单价和重量都是正数.

尽管 0-1 背包问题的求解本身是一个 NP-hard 问题, 但是许多研究表明利用动态规划的思想能解决这个问题并且时间复杂度为 O(NC)[17]. 0-1 背包的状态转换方程如式 (11) 所示:

$$\begin{aligned}V[i,j]=&\max\{V[i-1,j-w_i]+\\&p_i(j\geq w_i),V[i-1,j]\}\end{aligned} \tag{11}$$

其中, $V[i,j]$ 表示在前 i 件物品中选择若干件放到容量为 j 的背包中, 可以得到最大价值. 此式用来决策为使背包中物品的总价值最高, 第 i 件物品是否应该放入背包中.

2　算法设计

2.1　矩阵填充

数据稀疏性导致用户间共同评分物品特别少, 传统相似度计算结果不够准确. 为了缓解稀疏性的影响, 传统矩阵填充方法通过预测物品并集 $I_{u\cup v}$ 中用户未评分物品的评分并填充矩阵. 首先利用 Pearson 相关系数计算用户间的相似度并设置共同评分物品阈值 λ 来调节相似度值的大小, 如式 (12) 所示,

$$\begin{aligned}sim_1(u,v)=&\\&\frac{\sum\limits_{i\in I^1_{u,v}}(r_{u,i}-\bar r_u)(r_{v,i}-\bar r_v)}{\sqrt{\sum\limits_{i\in I^1_{u,v}}(r_{u,i}-\bar r_u)^2}\sqrt{\sum\limits_{i\in I^1_{u,v}}(r_{v,i}-\bar r_v)^2}}\times\\&\frac{\min(|I^1_{u,v}|,\lambda)}{\lambda}\end{aligned} \tag{12}$$

其中, $\bar r_u$ 和 $\bar r_v$ 分别表示用户 u 和用户 v 的评分均值.

然后根据邻居用户对物品的评分来预测目标用户 u 对物品并集中未评分物品的评分 $r'_{u,i}$ 并填充矩阵, 如式 (13) 所示:

$$r'_{u,i}=\bar r_u+\frac{\sum\limits_{v\in N(u)}sim_1(u,v)\times(r_{v,i}-\bar r_v)}{\sum\limits_{v\in N(u)}sim_1(u,v)} \tag{13}$$

为了提高填充评分的准确度, 本文采用文献 [18] 提出的基于用户兴趣传播的协同过滤方法 (User interests transmission based collaborative filtering approach, UIT) 进行评分预测并填充矩阵, 设预测评分为 $r''_{u,i}$, 则填充矩阵中用户 u 的评分值 $P_{u,i}$ 如式 (14) 所示:

$$P_{u,i}=\begin{cases}r''_{u,i}, & r_{u,i}=0, i\in I_{u\cup v}, v\in U\\ r_{u,i}, & r_{u,i}\neq 0\end{cases} \tag{14}$$

2.2　相似度计算

矩阵填充后, 传统相似度计算方法有以下 3 种: 余弦相似度、修正余弦相似度和 Pearson 相关系数[19−20], 本文在 Pearson 相关系数的基础上进行改进, Pearson 相关系数计算相似度时如式 (15) 所示, 其中, $\bar R_u$, $\bar R_v$ 分别是表示矩阵填充后用户 u 和用户 v 的评分均值.

式 (15) 在相似度计算时默认所有评分特征 (物品) 的权值相同, 然而由于预测评分与真实评分的可信度不同导致用户间共同评分物品和单一用户评分物品的意义大小不同即 $I^1_{u,v}$ 和 $I^2_{u,v}$ 意义大小不同, 相似度计算结果不够准确. 基于上述分析, 本文引入评分值可信度和物品可信度的概念并对相似度计算公式进行改进, 改进后的公式如式 (16) 所示:

$$\begin{aligned}sim_2(u,v)=&\frac{\sum\limits_{i\in I_{u\cup v}}(P_{u,i}-\bar R_u)(P_{v,i}-\bar R_v)}{\sqrt{\sum\limits_{i\in I_{u\cup v}}(P_{u,i}-\bar R_u)^2}\sqrt{\sum\limits_{i\in I_{u\cup v}}(P_{v,i}-\bar R_v)^2}}=\\&\frac{\sum\limits_{i\in I^1_{u,v}}(P_{u,i}-\bar R_u)(P_{v,i}-\bar R_v)+\sum\limits_{i\in I^2_{u,v}}(P_{u,i}-\bar R_u)(P_{v,i}-\bar R_v)}{\sqrt{\sum\limits_{i\in I^1_{u,v}}(P_{u,i}-\bar R_u)^2+\sum\limits_{i\in I^2_{u,v}}(P_{u,i}-\bar R_u)^2}\sqrt{\sum\limits_{i\in I^1_{u,v}}(P_{v,i}-\bar R_v)^2+\sum\limits_{i\in I^2_{u,v}}(P_{v,i}-\bar R_v)^2}}\end{aligned} \tag{15}$$

$$\begin{aligned}sim_2(u,v)=&\frac{\sum\limits_{i\in I^1_{u,v}}(P_{u,i}-\bar R_u)(P_{v,i}-\bar R_v)+C\sum\limits_{i\in I^2_{u,v}}(P_{u,i}-\bar R_u)(P_{v,i}-\bar R_v)}{\sqrt{\sum\limits_{i\in I^1_{u,v}}(P_{u,i}-\bar R_u)^2+C\sum\limits_{i\in I^2_{u,v}}(P_{u,i}-\bar R_u)^2}\sqrt{\sum\limits_{i\in I^1_{u,v}}(P_{v,i}-\bar R_v)^2+C\sum\limits_{i\in I^2_{u,v}}(P_{v,i}-\bar R_v)^2}}=\end{aligned}$$

$$\frac{\sum\limits_{i\in I_{u\cup v}}((P_{u,i}-\bar{R}_u)(P_{v,i}-\bar{R}_v)\times C^n)}{\sqrt{\sum\limits_{i\in I_{u\cup v}}((P_{u,i}-\bar{R}_u)^2\times C^n)}\sqrt{\sum\limits_{i\in I_{u\cup v}}((P_{v,i}-\bar{R}_v)^2\times C^n)}} \tag{16}$$

$$n=\begin{cases}0, & i\in I^1_{u,v}\\ 1, & i\in I^2_{u,v}\end{cases} \tag{17}$$

式 (16) 中, $\bar{R}_u$ 和 $\bar{R}_v$ 分别表示矩阵填充后 u 和 v 的加权评分均值. 假设填充后的矩阵中用户 u 的评分物品集合为 $I'_u=\{i|P_{u,i}\neq 0, u\in U, i\in I\}$, 则如式 (18) 所示:

$$\bar{R}_u=\frac{\sum\limits_{i\in I_u}P_{u,i}+\sum\limits_{i\in(I'_u-I_u)}P_{u,i}\times C}{|I_u|+(|I'_u-I_u|)\times C} \tag{18}$$

2.3 评分预测

选取相似度最高的 k 个用户作为用户 u 的邻居用户, 并根据邻居用户的评分来预测 u 对未评分物品的评分, 如式 (19) 所示:

$$R_{u,i}=\bar{R}_u+\frac{\sum\limits_{r\in N'(u)}sim_2(u,r)\times(P_{r,i}-\bar{R}_r)\times C^m}{\sum\limits_{r\in N'(u)}sim_2(u,r)\times C^m} \tag{19}$$

$$m=\begin{cases}0, & i\in I_r\\ 1, & i\in(I'_r-I_r)\end{cases} \tag{20}$$

其中, $N'(u)$ 表示用户 u 的近邻集合, $\bar{R}_r$ 表示矩阵填充后用户 r 的加权评分均值, I_r 表示用户 r 已评分物品, I'_r 表示填充矩阵中用户 r 已评分物品.

2.4 推荐列表

传统 Top-N 推荐方法只根据物品预测评分的高低筛选出前 N 个评分最高的物品推荐给用户, 筛选标准单一. 本文首先根据物品的预测评分选取候选推荐物品集合, 如式 (21) 所示, 然后综合考虑物品的预测评分值 $R_{u,i}$ 与物品的可预测性 Pr_i, 从而在候选集合中筛选出最佳的推荐列表 S_u, 其实质是一个组合最优化问题, 本文将其转化为 0-1 背包问题. 设物品不可预测性为 DPr_i, 则 $DPr_i=1-Pr_i$. 令背包容量为 Q, 即在物品不可预测性之和不大于 Q 的条件下, 从候选推荐物品集合 S'_u ($|S'_u|=M$) 中筛选出最佳的物品组合使得物品预测评分之和 $R_{\max}$ 最大, 如式 (22) 和 (23) 所示:

$$S'_u=\{i|R_{u,i}\geq\bar{R}_u, i\in I, u\in U\} \tag{21}$$

$$\begin{cases}\sum\limits_{i=1}^{M}DPr_ix_i\leq Q\\ x_i=0\text{ 或 }1\end{cases} \tag{22}$$

$$R_{\max}=\max\sum_{i=1}^{M}R_{u,i}x_i \tag{23}$$

最后选择装入背包的物品作为推荐列表推荐给用户.

2.5 算法流程

综合本文算法关键步骤的计算, 本节给出该算法对目标用户进行评分预测以及物品推荐的流程.

输入. 用户–物品评分矩阵 R, 目标用户 u

开始.

步骤 1. 矩阵填充

1) 采用 UIT 算法预测用户对未评分物品的评分值并填充矩阵.

2) 根据定义 1 区分真实评分值与预测评分值的可信度大小.

步骤 2. 矩阵填充后的相似度计算

1) 划分物品层次, 并区分不同层次物品的可信度大小.

2) 根据式 (16) 和 (18) 改进传统的相似度计算方法.

步骤 3. 评分预测

1) 根据式 (19) 预测目标用户对未评分物品的评分值.

步骤 4. 生成推荐集合

1) 根据定义 3 计算物品可预测性.

2) 根据式 (22) 和 (23) 筛选出最终的推荐列表.

输出. 目标用户的推荐列表.

结束.

3 算法性能分析

3.1 合理性分析

传统矩阵填充方法忽略了预测评分与真实评分的差异, 影响了算法的推荐性能. 针对此问题, 提出本文的第一个创新点: 引入置信系数 C ($0<C<1$) 来区分预测评分与真实评分的可信度大小, 进而区分不同层次物品对相似度计算结果的贡献大小. 结合图 1 进行说明. 如图 1 所示, 假设存在用户 u 和用户 v. 用户 u 评价过的物品集合为 A 和 B, 用户 v 评价过的物品集合为 A 和 C, D 为它们共同未评

分物品. 本文将物品分为三个层次, 即用户共同评分物品 $I^1_{u,v} = A$, 单一用户评分物品 $I^2_{u,v} = B \cup C$ 和共同未评分物品 $I^3_{u,v} = D$. 由于数据稀疏性导致用户间共同评分物品 A 急剧减少, 基于共同评分物品的相似度计算方法不够准确. 矩阵填充方法通过填充 $I^2_{u,v}$ 中未评分用户的评分使其转化为共同评分物品, 但是填充的预测评分与真实评分的可信度是不同的, 导致相似度计算时不同层次物品之间的意义大小也不同. 假设存在物品 i, 若 $i \in I^1_{u,v}$, 则两个用户的评分都是真实评分, 相似度计算结果的可信度高; 若 $i \in I^2_{u,v}$, 则其中一个用户的评分为预测评分, 相似度计算结果的可信度较低.

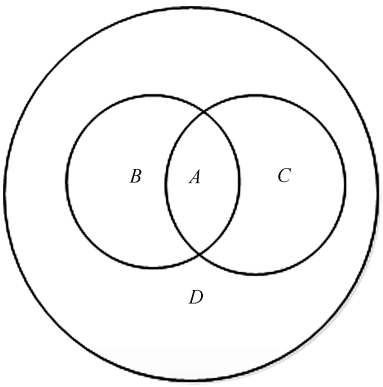


图 1　物品层次划分

Fig. 1　Hierarchy of item

传统 Top-N 推荐方法只根据物品预测评分的高低生产推荐列表, 标准单一不够准确. 针对此问题, 提出本文的第二个创新点: 引入物品可预测性对 Top-N 算法进行改进. 综合物品的预测评分与物品的可预测性, 并将其转化成最优化问题, 利用 0-1 背包算法来得到最优解即最佳的推荐列表. 由于推荐列表是在计算出预测评分之后产生, 所以改进后的推荐算法可以和其他任意评分预测算法结合, 具有良好的可扩展性.

3.2　复杂性分析

随着技术的不断进步, 存储空间对算法的影响减弱. 因此本节主要对算法的时间复杂度进行分析. 本文算法的时间开销主要来自矩阵评分填充, 用户间相似度计算, 物品可预测性计算以及推荐列表生成. 矩阵评分填充是先预测用户对未评分物品的评分, 然后对矩阵进行填充, 时间复杂度为 $\mathrm{O}(m^2)$; 用户间相似度计算需要计算 m 个用户间的相似度, 时间复杂度为 $\mathrm{O}(m^2)$; 物品可预测性计算需要遍历矩阵, 复杂度为 $\mathrm{O}(m \times n)$: 0-1 背包算法的复杂度为 $\mathrm{O}(M \times Q)$, 则推荐列表生成的复杂度为 $\mathrm{O}(m \times M \times Q)$. 所以总的时间复杂度为 $T(n) = \mathrm{O}(m^2) + \mathrm{O}(m \times n) + \mathrm{O}(n \times M \times Q)$. 在实际的推荐系统中, 物品数 n 一般是固定的且远小于用户数 m, Q 为常数且 $M < n < m$, 所以 $T(n) = \mathrm{O}(m^2)$. 传统的基于用户的协同过滤推荐算法的时间复杂度也是 $\mathrm{O}(m^2)$, 是同一个数量级, 表明本文算法是可行的.

4　实验与分析

4.1　数据集及对比算法

实验中采用 GroupLens 工作小组提供的 Movielens_100k 数据集和电影租赁网站提供的 Netflix_3m1k 数据集. Movielens_100k 存储了 943 个用户对 1 682 部电影的 100 000 条评分, 稀疏度 93.7 %. Netflix_3m1k 包含 4 427 个用户对 1 000 部电影的 56 136 条记录, 稀疏度 98.73 %. 评分取值是 1 ∼ 5, 评分越高表示用户满意度越高. 实验中将数据集随机 2-8 分割, 20 % 为测试集, 80 % 为训练集, 然后比较以下几种算法的性能.

1) 传统的基于 Pearson 相关系数的协同过滤推荐算法 (Collaborative filtering recommendation algorithm based on Pearson correlation coefficient, Per-CF)[19];

2) 文献 [12] 提出的 SingCF 算法 (Collaborative filtering algorithm based on singular ratings);

3) 本文提出的基于矩阵填充和物品可预测性的协同过滤算法 (Collaborative filtering recommendation algorithm based on rating matrix filling and item predictability, MfP-CF);

4) 本文提出的基于矩阵填充的协同过滤算法 (Collaborative filtering recommendation algorithm based on rating matrix filling, Mf-CF);

5) 融合本文物品可预测性的 Per-CF 算法 (Collaborative filtering recommendation algorithm based on Pearson correlation coefficient and item predictability, PerP-CF)[19];

6) 融合本文物品可预测性的 SingCF 算法 (Collaborative filtering algorithm based on singular ratings and item predictability, SingPCF)[12].

4.2　评价标准

实验中采用平均绝对偏差 (Mean absolute error, MAE) 和精确率 Precision 作为衡量算法准确率的标准[21], 采用覆盖率 (Coverage) 衡量推荐算法对物品长尾发掘的能力, 并采用时间指标来衡量算法训练时间的长短.

1) MAE 值越小, 用户对物品预测评分的准确度越高, 计算公式如式 (24) 所示:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{24}$$

其中, p_i 表示算法的预测评分, q_i 表示测试集中的

真实评分, N 表示测试集中的物品个数.

2) Precision 值越大, 推荐列表的准确度越高, 计算公式如式 (25) 所示:

$$\text{Precision} = \frac{1}{|U|}\sum_{u\in U}\frac{|A_u \cap B_u|}{|A_u|} \tag{25}$$

其中, A_u 表示用户 u 推荐列表中的物品集合, B_u 表示测试集中用户 u 的评分值大于其评分均值的物品集合即测试集中用户 u 喜欢的物品集合.

3) 覆盖率[22] 是指算法给用户推荐的物品占系统总物品的比例, 覆盖率越高, 推荐给用户的物品种类就越多, 推荐多样新颖的可能性就越大, 故覆盖率也能间接反映推荐的多样性和新颖性, 覆盖率的计算公式如 (26) 所示:

$$\text{Coverage} = \frac{|U_{u\in U}I(u)|}{|I|} \tag{26}$$

4) 算法的训练时间越短, 推荐速率越快, 实时性越强.

4.3 实验结果及分析

实验 1. 参数的选取

1) 置信系数 C 的选取

实验中设置常数 $\varepsilon = 1.2$, 由于背包量 Q 不会影响预测评分准确度 MAE 的值, 所以不用考虑. 调整置信系数 C 的值使得算法效果最好, 设置 $C = 0.10$, 0.20, 0.25, 0.30, 0.35, 0.40, 令近邻数 $k = 40$, 在 Movielens_100k 和 Netflix_3m1k 两个数据集下比较 MAE 的大小, 如图 2 所示.

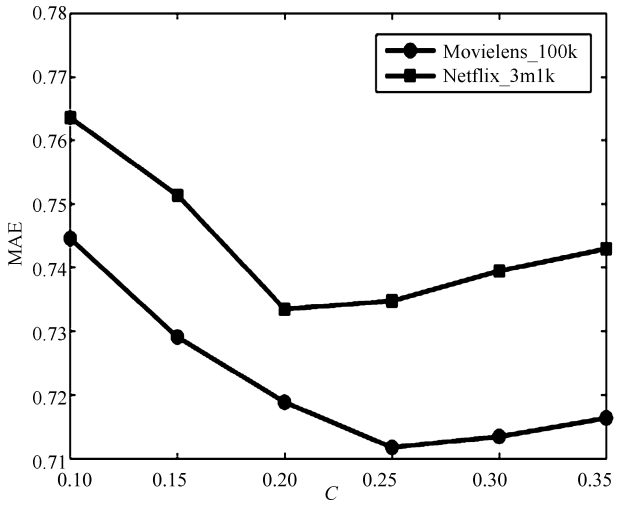


图 2 C 与 MAE 的关系

Fig. 2 The relationship between C and MAE

如图 2 所示, Movielens_100k 数据集中, $C = 0.25$ 左右时算法的 MAE 值最小, Netflix_3m1k 数据集中, $C = 0.2$ 左右时算法的 MAE 值最小, 多次实验后得到在两个数据集中 C 分别为 0.28 和 0.19 时, 本文算法效果最佳, 所以后续实验分别取 $C = 0.28$ 和 $C = 0.19$.

2) 背包量 Q 的选取

实验中通过设置不同的背包量来观察推荐列表的准确度, 设置背包量 $Q = 6$, 10, 11, 12, 13, 14, 15, 在 Movielens_100k 和 Netflix_3m1k 两个数据集下比较算法 Precision 值的大小, 如图 3 所示.

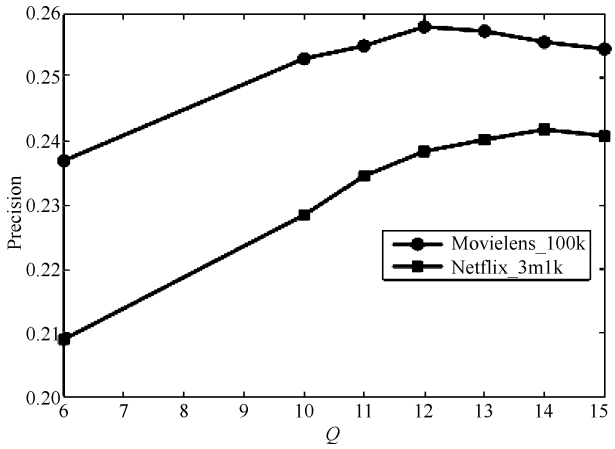


图 3 Q 与 precision 的关系

Fig. 3 The relationship between Q and precision

如图 3 所示, Movielens_100k 数据集中, $Q = 12$ 以及 Netflix_3m1k 数据集中 $Q = 14$ 时, Precision 的值最大, 本文算法效果最佳, 所以后续实验中分别取 $Q = 12$ 和 $Q = 14$.

实验 2. 近邻数对算法准确度的影响

设置邻居数 $k = 10$, 20, 30, 40, 50, 在 Movielens_100k 和 Netflix_3m1k 两个数据集下比较以下 4 种算法的准确度, 实验结果如图 4 和图 5 所示.

从图 4 和图 5 可以看出: 1) 图 4 中, MfP-CF 算法和 Mf-CF 算法的曲线重合, 即两者的 MAE 值始终相同.

2) 随着 k 的增加, 4 种算法的准确度都逐渐上升, 当 $k > 40$ 时, 算法的准确度趋于稳定.

3) 在两个数据集下本文 MfP-CF 算法的准确度一直最佳, 说明本文算法的稳定性.

实验结果分析:

1) MfP-CF 算法与 Mf-CF 算法的区别在于推荐列表生成环节的优化, 两者的预测评分值是相同的, 因此 MfP-CF 算法和 Mf-CF 算法的 MAE 值始终相同, 所以两者的曲线重合.

2) 随着 k 的增加, 有用信息也随之增加, 4 种算法的准确度都得到提高. 当邻居数高于 40 时, 算法准确度趋于稳定, 表明 40 个邻居用户就可以提供足够高质量的推荐, 因此后续实验取 $k = 40$.

3) Movielens_100k 和 Netflix_3m1k 都是稀疏数据集, 用户间的共同评分物品非常少, 导致传统的

Per-CF 算法相似度计算不够准确, 算法准确率低. SingCF 算法利用预测评分填充矩阵, 增加了共同评分物品, 缓解了稀疏性的影响, 在一定程度上提高了算法的准确度, 但忽略了预测评分与真实评分之间的可信度差异, 影响了算法的准确度. MfP-CF 算法和 Mf-CF 算法首先利用预测评分填充矩阵缓解稀疏性, 然后引入置信系数 C 区分预测评分与真实评分之间的差异, 进而区分不同层次物品之间的意义大小, 保证了相似度计算的准确性, 因此算法的准确度高于其他两种算法. 同时 MfP-CF 算法在推荐环节引入了物品可预测性的概念, 综合考虑物品的预测评分与物品的可预测性, 从而筛选出最优化的推荐列表, 虽然不会改变物品的预测评分值, 但优化了生成的推荐列表. 因此虽然这两种算法的 MAE 值始终相同, 但是 MfP-CF 算法的 Precision 值一直比 Mf-CF 算法的高.

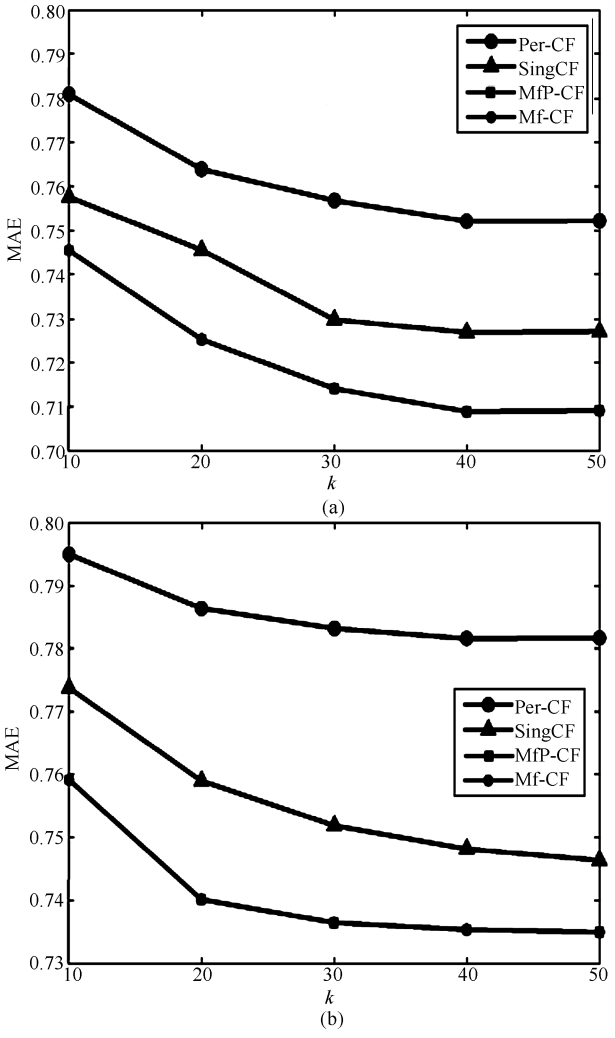


图 4　Movielens_100k 中 k 与 MAE 的关系

Fig. 4　The relationship between k and MAE in Movielens_100k

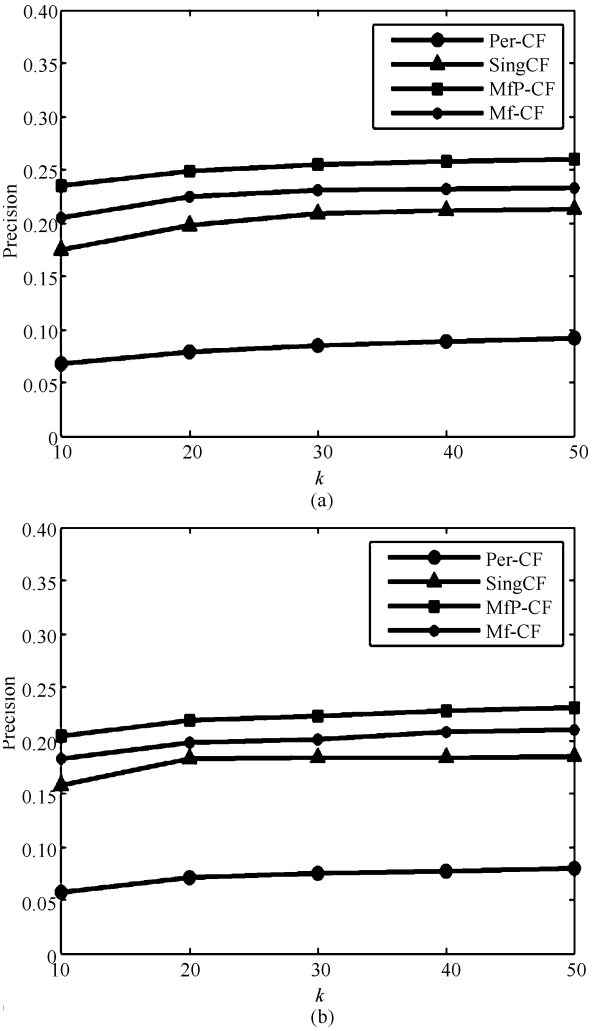


图 5　Movielens_100k 中 k 与 precision 的关系

Fig. 5　The relationship between k and precision in Movielens_100k

实验 3. 近邻数对算法覆盖率的影响

设置邻居数 $k=10$, 20, 30, 40, 50, 在 Movielens_100k 和 Netflix_3m1k 两个数据集下比较以下三种算法的覆盖率, 实验结果如图 6 所示.

从图 6 可以看出: 随着 k 的增加, 三种算法的覆盖率都逐渐降低. 这是因为 k 决定了推荐时近邻用户的个数. 那么 k 越大, 近邻用户越多, 推荐时就越趋向于热门物品, 从而对长尾物品的推荐越来越少, 造成覆盖率的降低. 同时本文算法的覆盖率一直优于其他对比算法, 能更好地对长尾物品进行推荐.

实验 4. 稀疏度对算法准确度的影响

为了进一步分析稀疏性对 MfP-CF 算法性能的影响, 实验中随机减少用户–物品评分矩阵中的评分来设置不同的稀疏度, 令 $k=40$, 在 Movielens_100k 数据集下比较以下三种算法预测评分的准确度, 实验结果如图 7 所示.

从图 7 可以看出:

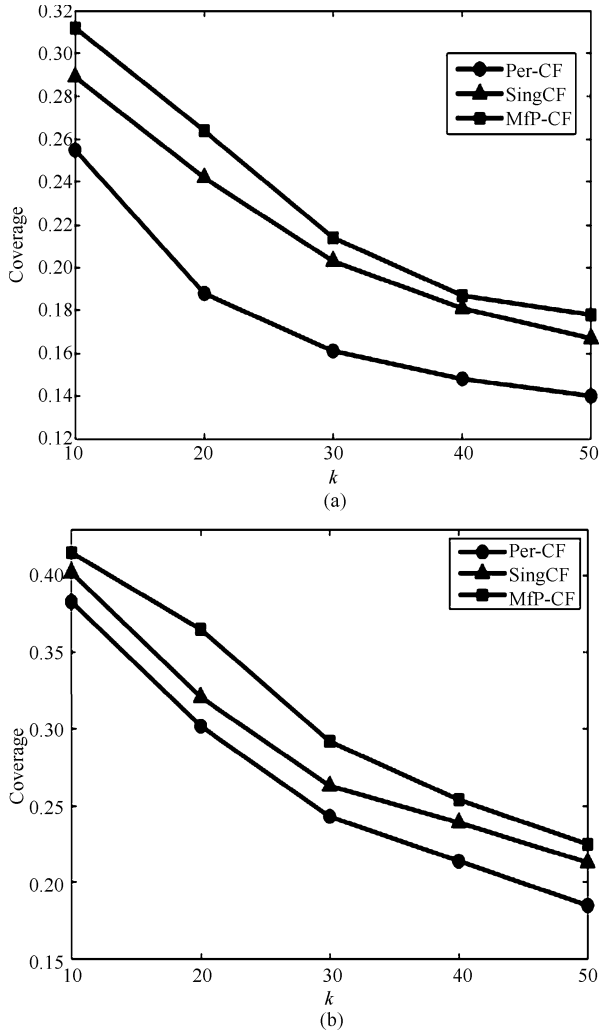


图 6 Movielens_100k 中 k 与 Coverage 的关系

Fig. 6 The relationship between k and Coverage in Movielens_100k

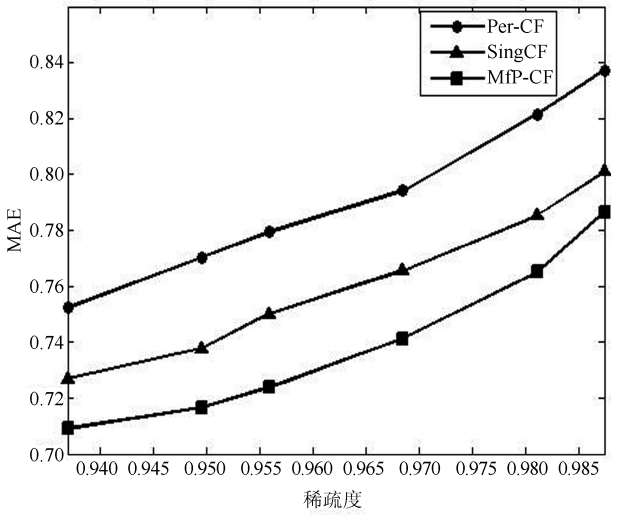


图 7 Movielens_100k 中稀疏度与 MAE 的关系

Fig. 7 The relationship between sparsity and MAE in Movielens_100k

1) 随着稀疏性的增大, 三种算法的准确度都逐渐下降:

2) 本文算法 MfP-CF 算法的准确度虽然也下降但一直高于其他两种算法.

实验结果分析:

随着稀疏性的增大, 有用信息不断减少, 相似度计算结果的准确性逐渐下降, 因此三种算法的准确度都下降. 传统的 Per-CF 算法是利用 Pearson 相关系数计算相似度的, 而 Pearson 相关系数是基于用户间共同评分物品的. 当稀疏性升高时, 共同评分物品会急剧下降, 导致相似度计算结果不可靠, 因此 Per-CF 算法的性能很差, 不适合高稀疏数据集. SingCF 算法和本文 MfP-CF 算法首先利用预测评分填充矩阵, 增加了共同评分物品数, 保证了相似度计算的可靠性, 有效地缓解稀疏性的影响, 因此算法准确度得到提高. 同时本文 MfP-CF 算法将预测评分与真实评分的可信度进行了区分, 进一步提高了算法的准确度.

实验 5. 基于物品可预测性算法的可扩展性

为了验证本文推荐环节优化算法的可扩展性, 在传统的 Per-CF 算法和 SingCF 算法基础上融合本文推荐环节的优化算法 — 基于物品可预测性算法进行实验. 设置邻居数 $k = 10$, 20, 30, 40, 50, 在 Movielens_100k 数据集中比较以下两组对比算法的准确度, 实验结果如图 8 所示:

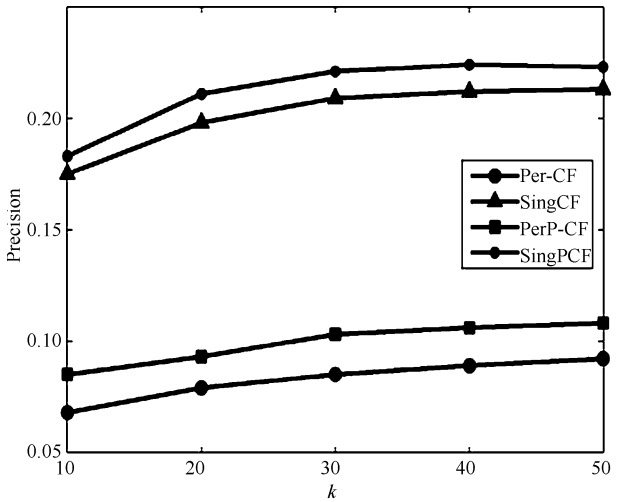


图 8 基于物品可预测性算法可扩展性对比

Fig. 8 Scalability comparison of algorithms

从图 8 可以看出:

PerP-CF 和 SingPCF 算法的 Precision 值分别比 Per-CF 和 SingCF 算法的高. 从这两组对比算法可以看出: 基于物品可预测性的算法具有良好的可扩展性.

实验 6. 算法时间复杂度分析

为验证 MfP-CF 算法的时间复杂度, 比较以下三种算法的运行时间, 实验结果如图 9 所示:

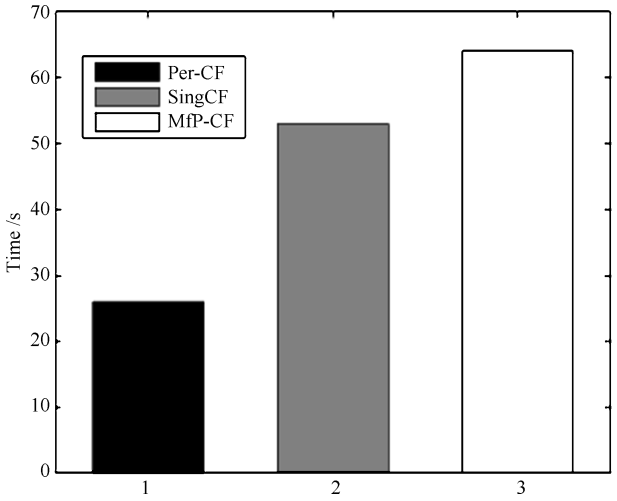


图 9 三种算法的运行时间对比

Fig. 9 Comparing the running time of the three algorithms

从图 9 可以看出:

MfP-CF 算法和 SingCF 算法的运行时间都高于传统的 Per-CF 算法. 同时 MfP-CF 算法的运行时间略高于 SingCF 算法, 说明本文 MfP-CF 算法在提高准确度的同时增加了时间复杂度.

5 结论

本文针对传统矩阵填充算法没有区分预测评分与真实评分的差异和推荐环节标准单一的问题, 首先引入置信系数区分预测评分与真实评分的可信度大小, 然后提出物品可预测性的概念, 利用 0-1 背包算法综合考虑物品预测评分与物品可预测性, 从而得到最优化的推荐列表. 基于以上两个创新点, 提出基于矩阵填充和物品可预测性的协同过滤算法, 并通过实验证明了算法的性能以及基于物品可预测性算法的可扩展性.

本文工作计划, 首先针对本文算法复杂度较大的问题进行算法改进, 降低复杂度; 其次改进矩阵填充环节的相关计算, 使得填充评分更加合理.

References

1 Chen Y, Tsai W T. *Service-Oriented Computing and Web Software Integration: From Principles to Development* (Fourth edition). Dubuque, IA, USA: Kendall Hunt Publishing, 2014.

2 Yu F, Zeng A, Gillard S, Medo M. Network-based recommendation algorithms: a review. *Physica A: Statistical Mechanics and its Applications*, 2016, **452**: 192−208

3 Sun Guang-Fu, Wu Le, Liu Qi, Zhu Chen, Chen En-Hong. Recommendations based on collaborative filtering by exploiting sequential behaviors. *Journal of Software*, 2013, **24**(11): 2721−2733
(孙光福, 吴乐, 刘淇, 朱琛, 陈恩红. 基于时序行为的协同过滤推荐算法. 软件学报, 2013, **24**(11): 2721−2733)

4 Hernando A, Bobadilla J, Ortega F. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems*, 2016, **97**: 188−202

5 Lv G, Hu C L, Chen S B. Research on recommender system based on ontology and genetic algorithm. *Neurocomputing*, 2016, **187**: 92−97

6 Mashal I, Alsaryrah O, Chung T Y. Performance evaluation of recommendation algorithms on internet of things services. *Physica A: Statistical Mechanics and its Applications*, 2016, **451**: 646−656

7 Zhang J, Peng Q K, Sun S Q, Liu C. Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Physica A: Statistical Mechanics and its Applications*, 2014, **396**: 66−76

8 Kim H N, Ji A T, Ha I, Jo G S. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, 2010, **9**(1): 73−83

9 Li Cong, Luo Zhi-Gang. Detecting shilling attacks in recommender systems based on non-random-missing mechanism. *Acta Automatica Sinica*, 2013, **39**(10): 1681−1690
(李聪, 骆志刚. 基于数据非随机缺失机制的推荐系统托攻击探测. 自动化学报, 2013, **39**(10): 1681−1690)

10 Leng Ya-Jun, Liang Chang-Yong, Ding Yong, Lu Qing. Method of neighborhood formation in collaborative filtering. *Pattern Recognition and Artificial Intelligence*, 2013, **26**(10): 968−974
(冷亚军, 梁昌勇, 丁勇, 陆青. 协同过滤中一种有效的最近邻选择方法. 模式识别与人工智能, 2013, **26**(10): 968−974)

11 Deng Ai-Lin, Zhu Yang-Yong, Shi Bo-Le. A collaborative filtering recommendation algorithm based on item rating prediction. *Journal of Software*, 2013, **14**(9): 1621−1628
(邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法. 软件学报, 2003, **14**(9): 1621−1628)

12 Xu R Z, Wang S Q, Zheng X W, Chen Y N. Distributed collaborative filtering with singular ratings for large scale recommendation. *Journal of Systems and Software*, 2014, **95**: 231−241

13 Chen Gang, Liu Fa-Sheng. Method for data mining based on BP neural network. *Computer and Modernization*, 2006, (10): 20−22
(陈刚, 刘发升. 基于 BP 神经网络的数据挖掘方法. 计算机与现代化, 2006, (10): 20−22)

14 Jang S, Yang J, Kim D K. Minimum MSE design for multiuser MIMO relay. *IEEE Communications Letters*, 2010, **14**(9): 812−814

15 Eldar Y C. Universal weighted MSE improvement of the least-squares estimator. *IEEE Transactions on Signal Processing*, 2008, **56**(5): 1788−1800

16 Kaleli C. An entropy-based neighbor selection approach for collaborative filtering. *Knowledge-Based Systems*, 2014, **56**: 273−280

17 Zou D X, Gao L Q, Li S, Wu J H. Solving 0-1 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing*, 2011, **11**(2): 1556−1564

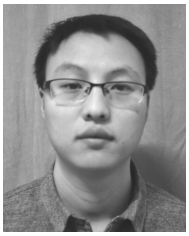
18 Gao Jian-Huang, Chen En-Hong, Liu Qi. User interests transmission based collaborative filtering approach. *Electronic Technology*, 2010, **47**(6): 1−4
(高建煌, 陈恩红, 刘淇. 基于用户兴趣传播的协同过滤方法. 电子技术, 2010, **47**(6): 1−4)

19 Javari A, Gharibshah J, Jalili M. Recommender systems based on collaborative filtering and resource allocation. *Social Network Analysis and Mining*, 2014, **4**: 234

20 Hu Y C. Recommendation using neighborhood methods with preference-relation-based similarity. *Information Sciences*, 2014, **284**: 18−30

21 Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems*, 2013, **37**: 146−153

22 Zhu Yu-Xiao, Lv Lin-Yuan. Evaluation metrics for recommender systems. *Journal of University of Electronic Science and Technology of China*, 2012, **41**(2): 163−175
(朱郁筱, 吕琳媛. 推荐系统评价指标综述. 电子科技大学学报, 2012, **41**(2): 163−175)



潘涛涛 国家数字交换系统工程技术研究中心硕士生. 主要研究方向为人工智能和数据挖掘. 本文通信作者.
E-mail: pan_taotao@126.com
(**PAN Tao-Tao** Master student at the China National Digital Switching System Engineering and Technological R&D Center. His research interest covers artificial intelligence and data mining. Corresponding author of this paper.)

文 锋 江南计算技术研究所高级工程师. 主要研究方向为计算机应用.
E-mail: wensinliu@163.com
(**WEN Feng** Senior engineer at the Jiangnan Computing Technology Research Institute. His main research interest is computer application.)

刘勤让 国家数字交换系统工程技术研究中心研究员. 主要研究方向为片上网络设计. E-mail: qinrangliu@sina.com
(**LIU Qin-Rang** Researcher at the China National Digital Switching System Engineering and Technological R&D Center. His main research interest is network-on-chip.)