

基于 Timed - HITS 与协同过滤的混合推荐算法

孙艳蕊, 陈 月

(东北大学 理学院, 辽宁 沈阳 110819)

摘 要: 用户间的信任关系、用户对商品的偏好兴趣及商品的时效性都会影响对商品的推荐效果. 将这些因素引入到基本的 HITS 算法中, 对 HITS 算法进行了改进. 将用户对商品的偏好兴趣矩阵进行了改进, 利用隐馈数据通过逻辑回归算法估计用户对商品的偏好兴趣, 对评分为零的情况赋予了不同的偏好兴趣度, 这样更符合实际. 将改进的 HITS 算法和协同过滤算法相结合得到一个混合推荐算法, 同时将用户分为活跃用户和非活跃用户分别进行推荐. 将提出的算法在 Movielens 数据集上进行了试验, 结果表明该算法在一定程度上缓解了数据稀疏和冷启动的问题, 推荐效果优于基于用户的协同过滤算法.

关 键 词: HITS; 信任关系; 偏好兴趣; 协同过滤; 推荐算法

中图分类号: TP 301.6

文献标志码: A

文章编号: 1005-3026(2019)04-0467-07

Hybrid Recommendation Algorithm Based on Timed-HITS and Collaborative Filtering

SUN Yan-rui, CHEN Yue

(School of Sciences, Northeastern University, Shenyang 110819, China. Corresponding author: CHEN Yue, E-mail: 18842487049@163.com)

Abstract: The product recommendation effect was affected by the trust relationship among users, the preference interest for goods and the time factor. These factors were introduced to the basic HITS algorithm, and the HITS algorithm was improved. The user preference interest matrix was also improved, which uses implicit data to estimate users' preference for goods using logistic regression algorithm. The situation with zero score gives different preference values, which is more consistent with reality. A hybrid recommendation model was proposed by combining the improved HITS algorithm with the collaborative filtering algorithm, and users were divided into active users and inactive users for recommendation. The proposed algorithm was tested using the Movielens data set, the results showed that the algorithm could generate better recommendation result in sparse data sets and cold-start situation, and it outperforms user-based collaborative filtering algorithm.

Key words: HITS (hypertext induced topic search); trust relationship; preference interest; collaborative filtering; recommendation algorithm

随着信息技术和互联网技术的不断发展, 人类逐渐从信息匮乏的时代走入了信息过载的时代, 并且信息过载问题越来越严重. 推荐系统就是解决这一矛盾的重要工具^[1]. 目前, 推荐系统已经广泛应用于各大电子商务网站, 成为其中重要的组成部分, 为用户和商家带来了极大便利, 并越来越受欢迎^[2-3]. 协同过滤是推荐系统中应用最广泛的推荐算法, 包括基于用户的协同过滤和基

于商品的协同过滤^[4-7]. 通过计算用户的历史评分相似度^[8], 得到与目标用户有相同喜好的用户, 然后将相似用户过去喜欢的物品推荐给目标用户. 虽然协同过滤算法被广泛应用, 但是目前协同过滤算法面临着数据稀疏和冷启动的问题^[9-10], 当有新的用户加入时, 由于新用户没有历史评分记录, 此时协同过滤技术表现得不是很理想.

此外,用户购买商品的情形各种各样,例如个人的喜好、朋友推荐^[11]、时尚买手的推荐等.因此,只依赖用户的历史评分进行协同过滤推荐是远远不够的.潘新等^[12]分析了对于流行度比较高的商品,用户的从众心理比较大,新用户更容易依赖系统推荐的流行商品.为解决用户冷启动问题,将 HITS 模型应用到推荐系统中^[13].此外挖掘数据的隐含信息等也是改善推荐系统比较常用的方法^[14].

本文通过引入用户之间的社交关系和用户对商品的偏好兴趣度来改进原始的 HITS 算法.此外,由于许多传统的推荐算法只依赖历史数据并没有考虑时间问题,在现实生活中无论是时尚买手推荐的还是流行商品都是具有时效性的,所以在 HITS 模型中引入时间维度,通过引入时间函数来惩罚陈旧过时的商品.最后,将改进的 HITS 算法和协同过滤算法相结合对用户进行推荐.这样对于新用户,即使没有历史数据,也可以根据改进的 HITS 算法提取流行商品进行推荐,可以有效解决用户冷启动问题.同时,根据用户的评分将用户分为活跃用户和非活跃用户,计算用户之间相似度时只需计算用户与活跃用户之间的相似度,即基于活跃用户进行协同过滤推荐.这样,不仅降低了计算的复杂度,也在一定程度上缓解数据稀疏所带来的影响.

1 传统的推荐算法

介绍与本文直接相关的两个经典算法:基于用户的协同过滤算法(UCF)^[5]和 HITS 算法^[15].

1.1 基于用户的协同过滤算法

假设推荐系统中有 m 个用户, n 个商品,用户集合、商品集合及用户对商品的评分矩阵分别为 $U = \{u_1, u_2, \dots, u_m\}$, $I = \{i_1, i_2, \dots, i_n\}$ 和 $R = [r_{ui}]_{m \times n}$. 其中 r_{ui} 代表用户 u 对商品 i 的评分,评分的取值为 1~5 的整数.评分矩阵中的缺失项代表用户未对商品评分,暂时用零填充.评分矩阵中隐含了用户对商品的潜在偏好信息及用户之间的潜在信任关系,是推荐算法的主要数据.

传统的基于用户的协同过滤算法首先要获取用户的历史偏好,通常由评分矩阵给出.当给目标用户 u 做推荐时,只要找出和 u 有相似行为的用户,把他们购买的商品推荐给目标用户 u . 基于用户的协同过滤算法包括两个步骤.

步骤 1 确定用户的近邻用户.计算目标用户 u 和其他用户间的相似度,将相似度大的前 k

个用户确定为目标用户 u 的近邻用户,记为 $N(u)$.

采用文献[5]中的修正余弦相似度公式计算用户 u 和 v 之间的相似度 $\text{sim}(u, v)$, 即

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_i)(r_{vi} - \bar{r}_i)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_i)^2 (r_{vi} - \bar{r}_i)^2}}. \quad (1)$$

式中: I_{uv} 是用户 u 和 v 同时评过分的商品集合; \bar{r}_i 是所有用户对商品 i 的平均评分.

步骤 2 计算目标用户对项目的预测评分值.预测评分式为

$$\tilde{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_i(u)} \text{sim}(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in N_i(u)} \text{sim}(u, v)}. \quad (2)$$

式中: \tilde{r}_{ui} 是目标用户 u 对项目 i 的预测评分值; \bar{r}_u 是目标用户 u 的平均评分; $N_i(u)$ 是对项目 i 评过分的用户 u 的近邻用户集合.

根据式(2),对目标用户未评分的项目进行预测评分,将评分值高的前 k 个项目推荐给目标用户.

1.2 HITS 算法

HITS 算法是网页搜索的经典算法之一.它通过分析页面之间的超链接,找出页面集合中的 authority 页面和 hub 页面. authority 页面是与查询主题最为相关并具有高质量、权威性的网页,而 hub 页面是包含指向查询主题最为重要的站点链接.

构造一个有向图 $G = (V_1, V_2, E)$, 表示网页的链接结构. 其中: V_1 是 hub 网页顶点构成的集合; V_2 是 authority 网页顶点构成的集合,若 V_1 和 V_2 中的顶点之间存在链接关系则连边; E 是所有边构成的集合.

对于 V_1 中的任一顶点 v , 它的 hub 值 $h(v)$ 是 v 所指向的 authority 页面的 authority 值之和,对于 V_2 中的任一顶点 u 的 authority 值 $a(u)$ 是所有 u 指向的 hub 网页的 hub 值之和. 计算 authority 值和 hub 值的具体步骤如下.

步骤 1 初始化:

对所有的 $v \in V_1, u \in V_2, h(v) = a(u) = 1$.

步骤 2 求 $a(u)$ 和 $h(v)$ 的值:

$$\forall v \in V_1: h(v) = \sum_{u \in V_2} a(u), \quad (3)$$

$$\forall u \in V_2: a(u) = \sum_{v \in V_1} h(v). \quad (4)$$

步骤 3 对 $a(u)$ 和 $h(v)$ 进行标准化:

$$h(v) = \frac{h(v)}{\sqrt{\sum_{v \in V_1} [h(v)]^2}}, \tag{5}$$

$$a(u) = \frac{a(u)}{\sqrt{\sum_{u \in V_2} [a(u)]^2}}. \tag{6}$$

步骤 4 重复执行步骤 2 和步骤 3,直到达到指定迭代次数或 $a(u)$ 和 $h(v)$ 收敛. 这里 $a(u)$ 和 $h(v)$ 收敛是指:对于给定的 $\varepsilon > 0$,有

$$\|a_t - a_{t-1}\| + \|h_t - h_{t-1}\| < \varepsilon. \tag{7}$$

式中: a_t 和 h_t 分别为迭代 t 次后的 authority 值和 hub 值.

步骤 5 返回 authority 值和 hub 值.

HITS 算法中 authority 和 hub 的概念引起了一些学者的注意,并且已将 HITS 算法应用到位置社交网络中,在位置社交网络中通过 HITS 算法提取受欢迎的地点推荐给用户^[13]. 由于位置社交网络只有签到数据,没有评分数据,改进的 HITS 算法不能直接应用到商品推荐中. 本文在文献[13]的基础上,针对商品推荐问题及 HITS 算法存在的问题对其进行改进,然后将改进的算法应用到推荐系统中.

2 改进的 HITS 算法:Timed - HITS

在传统的 HITS 算法中,任一个网页的 authority 值和 hub 值,只是单纯考虑从该网页链出或链入该网页的网页数量,并未考虑其他网页对该网页影响程度的大小. 然而在实际情况下,与该主题最为相关的、优质的网页对其影响更重要. 文献[13]中的 HITS 算法没有考虑用户对景点的偏好程度对景点推荐的影响,也没有考虑时效性,导致一些时间久远的景点,由于链接次数多而被作为流行的景点. 针对 HITS 算法的这些问题,对其进行改进.

用 hub 值和 authority 值分别代表推荐系统中时尚买手和流行商品,一个商品的 authority 值由购买过该商品的所有用户的 hub 值决定. 但是,这些用户对该商品的喜好程度是不同的,那些差评用户的 hub 值的权重应该较小,评分较高的用户的 hub 值的权重应该较大. 在推荐系统中,信任关系也在影响着用户的行为,用户信任的“朋友”购买的商品,该用户也可能会购买,这样一个用户的 hub 值,不仅由他所购买商品的 authority 值决定,还与他的“信任好友”的 hub 值有关. 此外,对于推荐流行产品,它的时效性很重要. 因为在实际生活中当前流行的商品一直在随时间推移而不断更

换,推荐系统环境实际上是动态的,它处在持续的变化中,过去流行的商品在当前和未来未必受欢迎. 综合以上分析,本文把信任关系、兴趣偏好程度及时间因素加入到原始的 HITS 模型中.

本文改进的 HITS 算法仅应用在活跃用户上,来提取“专家”用户和流行商品,这样就可以避免因新用户的加入而造成 HITS 算法结构不稳定的问题. 同时,降低了整体算法的时间复杂度,提高算法的效率.

2.1 信任关系

通常的数据集并没有直接提供真实的用户间信任关系数据. 从用户的行为角度考虑,认为购买相同商品越多的用户,兴趣越可能相同,越容易成为朋友,他们彼此之间的推荐越值得信任. 信任关系大小定义为两个用户共同评分的商品数. 设 I_u 和 I_v 分别表示用户 u 和用户 v 评过分的商品集合,则用户 u 和 v 间信任度大小为

$$t(u, v) = |I_u \cap I_v|, \tag{8}$$

信任度矩阵 $T = [t(u, v)]_{m \times m}$.

2.2 时间函数

推荐系统环境是动态的,它处在持续的变化中. 考虑流行商品的时效性,在 HITS 模型基础上增加了一个时间维度,引入一个时间函数 $f(t)$ ($0 < f(t) \leq 1$) 来惩罚陈旧的商品. 这样就能突出用户最新兴趣的权重,降低了先前兴趣的权重,从而能更准确地推荐. 根据“牛顿冷却公式”,给出时间函数:

$$f(t) = \frac{1}{\ln(\delta + t_{\max} - t)}. \tag{9}$$

式中: δ 是调节惩罚力度的参数,通过实验确定; t_{\max} 是数据集中的最大时刻值,代表最近时刻,是衡量商品陈旧的一个标准; $t_{\max} - t$ 是距离最近时刻的时间间隔,间隔越小,说明商品越新,反之代表商品越陈旧.

2.3 偏好兴趣度估计

兴趣度矩阵 $F = [f(u, i)]_{m \times n}$ 中的 $f(u, i)$ 最初定义为

$$f(u, i) = \begin{cases} 1, r_{ui} > 0; \\ 0, r_{ui} = 0. \end{cases} \tag{10}$$

这样定义的兴趣度只能反映用户对商品是否喜欢,不能反映用户对商品的偏好程度. 在文献[12]中定义了一个随着评分递增而递增的函数,用来计算用户对商品的偏好兴趣度. 该函数只给出了已评分用户对不同商品的偏好兴趣度,但是对于没有评分的项,偏好兴趣度的值都是一样的. 事实上,用户对没有评分的商品的偏好程度未必

都是一样的. 对于未评分的商品存在以下几种情况: 可能是没注意到该商品; 可能是商品价格较高; 可能是时间上的冲突; 也可能是不喜欢等多种可能. 鉴于存在这些情况, 对偏好兴趣度矩阵进行改进, 对于评分为零的商品也有不同的偏好兴趣度. 本文基于逻辑回归的思想估计偏好程度, 给出兴趣度矩阵.

首先, 兴趣度矩阵中 0 和 1 本身体现了用户对物品的喜欢和不喜欢信息. 把评分 0 和 1 看成用户对物品偏好的两个维度, 估计出用户对项目偏好在 1 这个维度上的可能性有多大, 利用逻辑回归的思想估计用户喜欢一个商品的概率大小, 作为偏好兴趣度的大小. 逻辑回归函数为

$$g(z) = \frac{1}{1 + e^{-z}}. \quad (11)$$

设训练集为 $[Z_{m \times n}, F_{m \times n}]$, 利用矩阵分解方法^[16]得, $Z_{m \times n} = P_{m \times k} Q_{k \times n}$, 其中 $P_{m \times k}$ 和 $Q_{k \times n}$ 分别表示 m 个用户的隐因子特征向量构成的矩阵和 n 个商品的隐因子特征向量构成的矩阵. 令 p_u 表示用户 u 的隐因子特征向量, q_i 表示商品 i 的隐因子特征向量. 由逻辑回归模型得兴趣度矩阵:

$$F = g(P_{m \times k} Q_{k \times n}). \quad (12)$$

将每个样本代入式(11), 有

$$g(p_u q_i) = \frac{1}{1 + e^{-p_u q_i}}. \quad (13)$$

最小化如下似然函数:

$$L(p_u q_i) = \sum (-f(u, i) p_u q_i + \ln(1 + e^{p_u q_i})). \quad (14)$$

利用随机梯度下降算法求得 p_u 和 q_i , p_u 和 q_i 的更新法则为

$$\left. \begin{aligned} p_u &= p_u - \eta(g(p_u q_i) - f(u, i)) q_i, \\ q_i &= q_i - \eta(g(p_u q_i) - f(u, i)) p_u. \end{aligned} \right\} \quad (15)$$

算法如下:

输入: 兴趣度矩阵 F , 学习率 η , 迭代次数 k .

输出: 偏好兴趣度矩阵 W_{ui} .

步骤 1 随机初始化 p_u 和 q_i .

步骤 2 对于每个样本 $(u, i) \in (U, I)$, 利用式(15)更新 p_u 和 q_i .

步骤 3 当达到指定迭代次数 k , 停止迭代.

步骤 4 求得 p_u 和 q_i , 将每对 p_u 和 q_i 代回逻辑函数, 即得到用户 u 对商品 i 的偏好兴趣度 $w(u, i)$, 所有偏好兴趣度 $w(u, i)$ 构成偏好兴趣度矩阵 W_{ui} .

2.4 用户商品网络图

一个推荐系统可以用一个赋权的无向图 $G(U, I, E, W)$ 来表示, 其中 $U = \{u_1, u_2, \dots, u_m\}$ 代

表所有用户构成的节点集合, $I = \{i_1, i_2, \dots, i_n\}$ 代表所有的商品构成的节点集合. E 为边集合, 由 E_u 和 E_{ui} 两部分组成, E_u 表示用户间的边集合, 代表用户间的信任关系, 若两个用户 u_i 和 u_j 对应的信任度值 $t(i, j) > 0$, 那么用户 u_i 和 u_j 之间存在连边; E_{ui} 表示用户和商品之间的连边, 代表用户对商品的购买行为, 如果用户对商品评过, 用户和商品之间连边. W 表示权值矩阵, 与 E_u 和 E_{ui} 对应的权值矩阵分别记为 W_u 和 W_{ui} , 即 W_u 表示用户和用户之间边的权值矩阵, $W_u = T = (t(u_i, u_j))_{m \times m}$; W_{ui} 表示用户和商品之间边的权值矩阵, $W_{ui} = (w(u_k, i_j))_{m \times n}$.

2.5 Timed - HITS 模型

引入用户间的信任关系、用户对商品的偏好兴趣度及时间函数对原始的 HITS 算法进行改进. 将改进后的 HITS 算法称为 Timed - HITS 算法. Timed - HITS 算法的 authority 值和 hub 值的更新公式分别为

$$a(i_j) = \beta f(t_{ij}) \sum_{k=1}^m w(u_k, i_j) h(u_k), \quad (16)$$

$$h(u_k) = \beta \sum_{j=1}^n w(u_k, i_j) a(i_j) + (1 - \beta) \cdot$$

$$f(t_{uk}) \sum_{l=1}^m t(u_k, u_l) h(u_l). \quad (17)$$

式中: β 是参数, 通过实验确定; $f(t_{ij})$ 代表商品 i_j 对应的时间函数的值; $f(t_{uk})$ 代表用户 u_k 对应的时间函数的值. 将式(16), (17)应用到基本 HITS 算法中, 通过迭代过程得到流行商品和“专家”用户.

3 混合推荐算法: Timed - HCF

3.1 活跃用户

对于每个用户 u_i , 定义其活跃值 $\text{pos}(u_i)$ 为用户 u_i 评分向量中非零元素所占的比例. 对于 $\text{pos}(u_i)$ 大于评分稀疏度 λ 的用户 u_i 定义为活跃用户, 活跃用户的集合为

$$\bar{U} = \{u_i | u_i \in U, \text{pos}(u_i) > \lambda\}, \quad (18)$$

式中, λ 值根据具体数据集而定.

由所有的活跃用户的评分向量构成的矩阵称为密集子矩阵. 后面的算法在密集子矩阵上提取“专家”和“流行商品”, 这样可以提高算法的效率. 同时可以缓解推荐算法面临的数据稀疏性的问题.

3.2 Timed - HCF 推荐算法

下面给出基于 Timed - HITS 与协同过滤的

混合推荐算法(记为 Timed - HCF).

为解决用户冷启动问题,本文将用户分为活跃用户和非活跃用户两部分分别进行推荐. 对活跃用户基于 Timed - HITS 与协同过滤的混合推荐算法 Timed - HCF 进行推荐. 应用改进的 Timed - HITS 算法获取专家用户和流行商品. 将 Timed - HITS 算法得到的“专家”对商品评分的均值和协同过滤算法预测的评分加权,作为用户对商品的预测值. 为更准确描述用户间的相似性,在协同过滤算法中引入信任度. 为降低算法的复杂度,本文改进的 Timed - HITS 算法和协同过滤算法中寻找近邻集合的过程均在密集子矩阵上进行.

对于非活跃用户,考虑到非活跃用户的历史评分信息很少,并且很容易依赖推荐系统推荐的流行商品,本文通过 Timed - HITS 算法提取的流行商品对用户进行推荐. 下面分别给出针对活跃用户和非活跃用户的预测公式.

针对活跃用户:

$$r_{cf}(u,i) = \bar{r}_u + \frac{\sum_{v \in N_i(u)} \text{sim}(u,v)t(u,v)(r_{ui} - \bar{r}_v)}{\sum_{v \in N_i(u)} \text{sim}(u,v)t(u,v)}, \tag{19}$$

$$r_{thcf}(u,i) = \alpha \cdot \bar{r}_{hub}(i) + (1 - \alpha) \cdot r_{cf}(u,i). \tag{20}$$

式中: $r_{cf}(u,i)$ 代表通过协同过滤算法求得的用户 u 对商品 i 的预测值; \bar{r}_u 代表用户的平均评分; $t(u,v)$ 代表用户之间的信任关系; $\bar{r}_{hub}(i)$ 代表“专家”对商品 i 的平均评分; $r_{thcf}(u,i)$ 代表利用 Timed - HCF 算法得到的用户 u 对商品 i 的预测值; $\alpha(0 < \alpha < 1)$ 为权重因子,权重参数 α 通过交叉验证得到.

针对非活跃用户:将利用 Timed - HITS 算法得到的流行商品推荐给非活跃用户.

Timed - HCF 算法如下:

输入:评分矩阵 R ,信任矩阵 T .

输出:用户 u 对商品 i 的评分 $r_{thcf}(u,i)$.

步骤 1 将用户分为活跃用户和非活跃用户.

步骤 2 通过 Timed - HITS 算法得到流行商品集 $N_{authority}$ 和“专家”用户集合 N_{hub} .

步骤 3 计算“专家”用户对商品 i 的平均评分 $\bar{r}_{hub}(i)$:

$$\bar{r}_{hub}(i) = \frac{1}{|N_{hub}|} \sum_{v \in N_{hub}} r(v,i). \tag{21}$$

其中, N_{hub} 表示专家用户集合.

步骤 4 利用式(19)预测用户 u 对商品 i 的评分 $r_{cf}(u,i)$.

步骤 5 对于活跃用户 u ,利用式(20)预测用户 u 对商品 i 的评分 $r_{thcf}(u,i)$.

步骤 6 对于非活跃用户,将流行商品推荐给非活跃用户.

4 实验分析

4.1 数据集

在 Movielens 数据集中 100 K 的部分对本文所提出的 Timed - HCF 算法进行实验. 该数据集记录了 943 位用户对 1 682 部电影的 100 000 条评分数据,数据集中的所有评分值是 1 ~ 5 的整数(1 表示“很不喜欢”,5 表示“很喜欢”). 实验在 Movielens 数据集上进行,采用五折交叉验证的方式进行测试. 表 1 是 MovieLens - 100 K 数据集的统计特性.

表 1 MovieLens - 100K 数据集的统计特性
Table 1 Characteristics of MovieLens-100 K databases

数据集	Movielens
用户数量	943
电影数量	1 682
评分记录	100 000
用户平均评分商品数	106
电影平均被评分数量	60
评分稀疏度	0.063

4.2 误差度量

为了评估本文算法的性能,选择两个常用的精确度指标,平均绝对误差(mean absolute error, MAE)和均方根误差(root mean squared error, RMSE). 对于测试集中的一个用户 u 和物品 i , r_{ui} 是用户 u 对商品 i 的实际评分, \hat{r}_{ui} 是 Timed - HITS 推荐算法给出的预测评分,Test 代表测试集. 平均绝对误差 MAE 和均方根误差 RMSE 的计算公式分别为

$$RAE = \frac{\sum_{(u,i) \in \text{Test}} |r_{ui} - \hat{r}_{ui}|}{|\text{Test}|}, \tag{22}$$

$$RMSE = \frac{\sum_{(u,i) \in \text{Test}} (r_{ui} - \hat{r}_{ui})^2}{|\text{Test}|}. \tag{23}$$

MAE 和 RMSE 值越低,算法预测的准确度越高.

4.3 实验结果与分析

将本文算法和基于用户的协同过滤算法进行

预测精度的比较. 为了说明本文最终得到的 Timed - HCF 算法的有效性, 根据算法的改进过程, 实验中作了 4 个算法的比较. 首先在基于用户的协同过滤算法(UFC)的基础上引入信任矩阵 T (算法记为 TUFC), 再结合改进的 HITS 算法(算法记为 HCF), 最后引入时间因素得 Timed - HITS 算法(记为 Timed - HCF). 所提到的算法均采用 MAE 和 RMSE 指标来评估预测准确度.

算法中的参数调节惩罚力度的系数 δ 、综合权重因子 α 和 β , 通过交叉验证法得到. 对于每种模型, 选用的邻居大小 k 均等于 20, 参数 $\delta = 5$, $\beta = 0.95, \alpha = 0.46$. 表 2 给出了逐步改进的各推荐算法的误差.

表 2 改进算法的误差比较		
Table 2 Errors comparison of improved algorithm		
推荐算法	RMSE	MAE
UFC	1.297 2	0.957 9
TUFC	1.291 0	0.954 6
HCF	1.179 0	0.861 0
Timed - HCF	1.126 4	0.824 3

由表 2 可知, 本文提出的 Timed - HCF 算法相比 UFC 算法的 MAE 和 RMSE 值分别降低了 17% 和 13%, 改进的过程中, 4 个推荐算法的 MAE 值和 RMSE 值逐渐降低, 并且本文的 Timed - HCF 算法表现更好.

Timed - HCF 算法和 UFC 算法的 RMSE 和 MAE 值随 k 值变化如图 1, 2 所示.

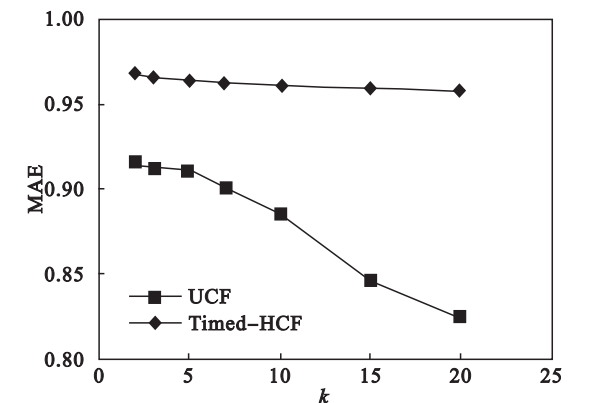


图 1 MAE 随 k 的变化
Fig. 1 Changes of MAE with k

图 1 和图 2 表明, 本文提出的 Timed - HCF 算法的 MAE 和 RMSE 值均低于 UFC 算法的.

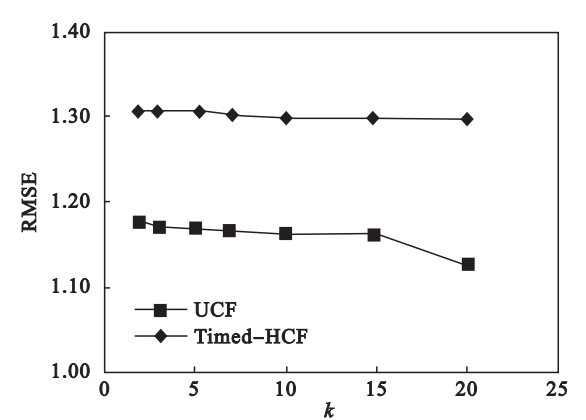


图 2 RMSE 随 k 的变化
Fig. 2 Changes of RMSE with k

5 结 论

1) 利用逻辑回归函数和矩阵分解改进了兴趣度矩阵, 使评分为零的情况也能估计出兴趣度, 更准确地反映了用户的兴趣, 同时缓解了数据的稀疏性问题.

2) 对 HITS 算法进行了改进, 使其更符合实际情况并应用到商品推荐中.

3) 将改进的 HITS 算法与协同过滤算法结合得到一个混合推荐算法. 算法不仅缓解了数据稀疏性和用户冷启动问题, 同时提高了推荐精度, 降低了推荐的时间复杂度, 是一个有效的推荐算法.

参考文献:

[1] Lu L, Medo M, Yeung C H, et al. Recommender system[J]. *Physics Reports*, 2012, 519(1): 1 - 49.

[2] Wang X, Wang Y. Improving content-based and hybrid music recommendation using deep learning [C]//Proceedings of the ACM International Conference on Multimedia. Orlando, 2014: 627 - 636.

[3] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles [C]// Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining. Sab Diego, 2001: 448 - 456.

[4] Adomavicius G, Tuzhilin A. Towards the next generation of recommend systems; a survey of the state-of-the-art and possible extensions [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(6): 734 - 749.

[5] Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering [J]. *Knowledge-Based Systems*, 2013, 37(1): 146 - 153.

[6] APirasteh P, Jung J J, Hwang D. Item-based collaborative filtering with attribute correlation: a case study on movie recommendation [M]. Berlin: Springer-Verlag, 2014: 245 - 252.