

DOI: 10.7652/xjtuxb201805015

加入用户对项目属性偏好的奇异值分解推荐算法

魏港明, 刘真, 李林峰, 张猛
(北京交通大学计算机与信息技术学院, 100044, 北京)

摘要: 由于目前的矩阵分解推荐算法在解决项目冷启动问题时,没有充分利用项目的属性偏好信息与用户评分行为的交互信息,因此提出了加入用户对项目属性偏好的奇异值分解推荐(UC-SVD)算法。该算法综合考虑项目属性和用户对项目的评分,不仅在矩阵分解算法中加入了项目的属性信息,同时通过对评分数据集和属性数据集的综合分析,得出用户对项目属性的偏好矩阵,将项目属性特征因子和用户对项目属性的偏好特征因子一并加入到矩阵分解中。在数据集 Movielens、HetRec2011 上进行实验,结果表明,与经典矩阵分解协同过滤算法相比,所提算法不仅在一定程度上解决了项目的冷启动问题,而且在同等条件下的均方根误差平均降低了 3.5%,平均绝对误差平均降低了 3%,尤其是在更为稀疏的 HetRec2011 数据集上,项目属性对用户评分行为的影响更加明显,改进算法在推荐精度上表现出更大的优越性。

关键词: 冷启动;奇异值分解;矩阵分解;项目属性

中图分类号: TP391.9 **文献标志码:** A **文章编号:** 0253-987X(2018)05-0101-07

Singular Value Decomposition Recommendation Algorithm Considering User's Preference for Item Attributes

WEI Gangming, LIU Zhen, LI Linfeng, ZHANG Meng
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The traditional recommendation algorithm based on matrix factorization is facing the problem that in cold start of items, the existing recommendation algorithm does not make full use of the interactive information about item attributes and user's rating behavior, therefore a singular value decomposition recommendation algorithm is proposed (UC-SVD) considering the item attributes and user's rating for items. It adds the item attribute information into the decomposition matrix, and through a comprehensive analysis on the rating data set and attribute data set to get the user preference matrix and item attribute characteristic factor, it adds the user preference characteristic factor to the matrix decomposition. Experimental results on the datasets of Movielens and HetRec2011 show that in comparison with the classic matrix factorization collaborative filtering algorithm, the proposed algorithm can not only solve the problem in cold start of items to some extent, but also under the same condition reduce the root mean square error and the mean absolute error by 3% and 4%, respectively. Especially on the more sparse HetRec2011 dataset, the impact of item attributes on user's rating behavior is more obvious, and this improved algorithm shows greater superiority in recommendation accuracy.

Keywords: cold start; singular value decomposition; matrix factorization; item attribute

基于矩阵分解的推荐算法是应用广泛的协同过滤推荐技术^[1-3],具有良好的推荐准确性,并能有效处理大规模的用户项目评分数据。在 Netflix Prize 大赛中,矩阵分解算法产生了 PMF、ASVD 和 SVD++ 等算法^[4-6],这些算法能够充分挖掘用户项目的隐类信息,具有很好的推荐效果。但是,矩阵分解推荐算法存在冷启动问题,对于评分较少的用户或新项目,不能有效挖掘其隐类信息,越来越多的社会化网络、用户的人口统计学信息、项目的描述信息等^[7-8]信息源被加入到矩阵分解的框架中,这在一定程度上解决了冷启动问题。

目前,大多数工作集中在用户冷启动的问题,而忽略了项目冷启动问题。MarCelo 提出了在推荐中应用项目元数据的 gSVD++ 算法,融入用户隐式偏好,同时充分考虑了项目的属性信息的作用,提高了推荐的准确度^[9-10];Qin 等提出了基于项目类别权重的协同过滤推荐算法,同时考虑了项目的类别信息和用户的属性信息,提高了推荐的准确^[11];Yu 等提出了在矩阵分解算法中融入项目属性信息的改进算法^[12],一定程度上解决了项目冷启动问题。矩阵分解协同过滤算法只考虑了项目属性的影响,并未考虑用户对项目属性的偏好信息。用户对项目类别的契合在某种程度上体现了用户本身性格偏好,因此需要考虑这方面的信息。

基于此,本文提出了加入项目属性和用户对项目属性偏好的 UC-SVD 矩阵分解算法,将用户的评分信息、项目的属性信息以及用户对项目属性的偏好信息应用到矩阵分解算法中,对矩阵分解中隐类的分解起到增强效果,解决项目冷启动问题,同时提高推荐的质量。

1 传统的矩阵分解算法

在现实中,用户和项目都有很多的偏好特征,如果用户的特征向量和项目的特征向量相吻合,那么两向量的内积就是用户的对项目的评分,从矩阵分解的角度来看,就是将评分矩阵分解为用户特征矩阵 P 和项目特征矩阵 Q 的乘积^[13],即

$$\hat{R} = P^T Q \quad (1)$$

如果考虑整体用户和项目的偏差,则可得用户 u 对项目 i 的预测评分

$$\hat{r}_{ui} = u + b_u + b_i + \sum_f^F p_{uf} q_{if} \quad (2)$$

如果能最小化训练集 T 的预测误差,那么也可以最小化测试集的预测误差。为了防止过拟合,加入正则化项 $L2, \lambda$ 为正则化参数,定义损失函数为

$$F(p, q) = \sum_{(u, i) \in T} (r_{ui} - u - b_u - b_i - \sum_f^F p_{uf} q_{if})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + \|b_u\|^2 + \|b_i\|^2) \quad (3)$$

要最小化式(3)所示的损失函数,可利用随机梯度下降最优化方法进行求解,通过调整参数,可求出预测评分。

2 项目属性和用户对项目属性的偏好

2.1 项目属性

考虑 m 个用户和 n 个项目的评分矩阵,评分矩阵 R 中包含的用户集合为 $U = \{u_1, u_2, u_3, \dots, u_m\}$,被评价的项目集合为 $I = \{i_1, i_2, i_3, \dots, i_n\}$, r_{ui} 表示用户 u 对项目 i 的评分。项目 i 的属性集合为 $G(i) = \{g_1, g_2, g_3, \dots, g_s\}$,其中 s 为项目属性个数, $G(i)$ 表示项目 i 的各个属性信息,如果 i 存在某属性,则该属性为 1,否则置为 0。

2.2 用户对项目属性的偏好

通过评分矩阵,可以得到每个用户评分过的项目集合,由于用户评分的行为本身代表着用户的偏好,因此用户对项目选择性的评分反应了一定的统计特征:如果用户评分的项目中具有某种属性的比例很高,说明用户很喜欢该类型的项目;如果项目都不具有某种属性或者比例很少,则说明用户对该类型项目不太感兴趣。因此,对用户 u 来说,如果该用户看过的所有电影集合为 $\{i_1, i_2, \dots, i_k\}$,则用户 u 对属性 g_j 的累加次数为 $\sum_{i=1}^k a_{ij}$,并对所有属性的累加次数结果进行降序排序,取前 T 种属性认为用户对其有偏好,用 1 表示,其余属性用 0 表示。用户对项目属性的偏好 $Z(u)$ 表示用户 u 对项目某属性偏好的信息,如果用户 u 对项目某属性喜欢,则置为 1,否则置为 0。所有用户对项目属性的偏好向量 $Z(u)$ 构成用户对项目属性的偏好矩阵 Z 。

3 改进后的模型和算法

3.1 加入项目属性因子

考虑到项目的属性信息对项目特征矩阵的增强作用,将项目属性信息加入到 SVD 模型当中,加入因子后的评分预测公式为

$$\hat{r}_{ui} = b_{ui} + (q_i + |G(i)|^{-\delta} \sum_{g \in G(i)} c_g)^T p_u \quad (4)$$

式中: $b_{ui} = u + b_u + b_i$ 。通过每个项目 i 的 $G(i)$ 属性集用来调整每个属性 g 的相关特征权重,并通过已知的用户评分矩阵学习得到。

3.2 加入用户对项目属性偏好因子

由于用户对项目属性的偏好并未直接在数据集文件中给出,因此首先需要构造用户对项目属性的偏好矩阵 Z ,构造如下算法 1。

算法 1 构造用户对项目属性的偏好矩阵

输入: 用户项目评分矩阵 R , 项目属性矩阵 G , 属性选择个数 T 。

输出: 用户对项目属性偏好矩阵 Z 。

(1) $I_u \leftarrow \text{GetEachUserItem}(R)$; /* 对于用户 u , 根据评分矩阵 R 取得 u 评分过的项目集合 $I_u = \{i_1, i_2, i_3, \dots, i_k\} * /$ 。

(2) $C_u \leftarrow \text{CalTimes}(I_u, G)$ /* 对于 u 评价过的所有项目 $\{i_1, i_2, \dots, i_k\}$, 将各项目针对属性 g_j 对应相加, 得到用户 u 对项目属性 g_j 的评分次数 $c_j = \sum_{i=1}^k a_{ij}$, 对所有属性进行遍历, 得到用户 u 对各属性的评分 $C_u = \{c_1, c_2, \dots, c_s\} * /$ 。

(3) $b_{uk} \leftarrow \text{SortAndTakeT}(C_u)$ /* 对步骤(2)中评分次数 $c_i (i=1, 2, \dots, s)$ 进行降序排序, 取评分次数靠前的 T 个属性, 将其结果置为 1, 其余置为 0 (如果不为 0 项小于 T , 则全部取为 T), 得到用户 u 的项目属性偏好向量 $b_{uk} (k=1, 2, 3, \dots, s) * /$ 。

(4) 遍历所有用户, 重复步骤(1)~(3), 得到所有用户对项目属性的偏好矩阵 Z 。

在算法 1 中, 用户对项目属性评分次数在统计意义上反映了用户 u 对项目各属性的偏好信息。由于整个评分矩阵中, 用户间对项目评分数量差距较大, 不能统一按照某个阈值进行属性选择处理, 因此只利用用户对项目属性的二元偏好而不使用对电影属性评分次数, 这样有效避免了不同用户评分对项目各属性次数相差过大的问题, 从而得到了用户 u 对项目属性的偏好向量。把用户对项目属性的偏好信息加入到式(4)中, 用户对相关属性偏好的权重可以通过用户评分信息学习到, 加入用户偏好后的评分预测公式为

$$\hat{r}_{ui} = b_{ui} + (q_i + |G(i)|^{-\delta} \sum_{g \in G(i)} c_g)^T (p_u + |Z(u)|^{-\phi} \sum_{j \in Z(u)} p_j) \quad (5)$$

在式(5)中, 与传统矩阵分解算法相比, 分别对

q_i 、 p_u 进行改进, 加入项目属性偏好特征和用户对项目属性偏好特征, 算法改进示意图如图 1 所示。

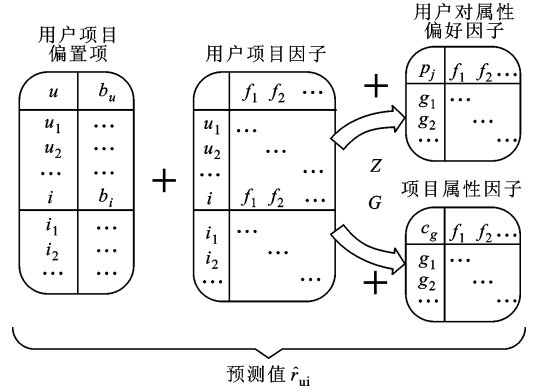


图1 算法改进示意图

模型建立后, 可得评分预测损失函数

$$\min_{b, p, q, c} \sum (r_{ui} - \mu - b_u - b_i - (q_i + |G(i)|^{-\delta} \sum_{g \in G(i)} c_g)^T (p_u + |Z(u)|^{-\phi} \sum_{j \in Z(u)} p_j))^2 + \lambda(b_u^2 + b_i^2 + \|p_u\|^2 + \|q_i\|^2 + \sum_{g \in G(i)} c_g^2 + \sum_{j \in Z(u)} p_j^2) \quad (6)$$

通过算法更新可以得到用户的特征矩阵 P 、项目的特征矩阵 Q 、用户对项目属性偏好特征矩阵 P_F 及项目属性特征矩阵 Q_F 。 δ, ϕ 根据属性信息进行调整: 如果项目存在相关属性信息, 则 δ 设置为 1, 否则为 0; 如果用户存在对项目属性偏好信息, 则 ϕ 设置为 1, 否则设置为 0, 其余的相关参数将通过在训练集中最小化损失函数学习到, 参数的更新算法如算法 2 所示。

算法 2 参数更新算法

输入: 用户对项目的评分矩阵 R , 项目的属性矩阵 G , 用户对项目的属性偏好矩阵 Z ;

输出: 返回 List 列表, 包括用户和项目偏差 b_u 和 b_i , 用户和项目特征矩阵 P 和 Q , 用户对项目属性偏好特征矩阵 P_F 和项目属性特征矩阵 Q_F 。

(1) $\text{List}[b_u, b_i, W, H] \leftarrow \text{Initialize}(\text{Double})$;

$P_F, Q_F \leftarrow \text{Initialize}(\text{Gaussian})$; /* 以 Double 数据类型初始化 b_u, b_i, P 和 Q , 并以高斯分布(加速收敛)初始化用户对项目属性偏好特征矩阵 P_F 和项目属性特征矩阵 Q_F */。

(2) $b_u \leftarrow b_u - \gamma_1 \nabla b_u, b_i \leftarrow b_i - \gamma_1 \nabla b_i; P \leftarrow P - \gamma_1 \nabla P$;

$Q \leftarrow Q - \gamma_1 \nabla Q; P_F \leftarrow P_F - \gamma_2 \nabla P_F; Q_F \leftarrow Q_F - \gamma_2 \nabla Q_F$;
/* 利用式(5)进行评分计算初始的预测评分, 然后

根据式(6)及评分矩阵 $\mathbf{R}$,项目属性矩阵 $\mathbf{G}$,用户对项目属性偏好矩阵 $\mathbf{Z}$,利用 SGD 随机梯度下降法更新矩阵 $\mathbf{P},\mathbf{Q},\mathbf{P}_F$ 和 $\mathbf{Q}_F$,直至预测误差基本不再变化 * /。

(3)Return List $[b_u, b_i, \mathbf{P}, \mathbf{Q}, \mathbf{P}_F, \mathbf{Q}_F]$ / * 参数列表 * /。

3.3 UC-SVD 算法流程

综合算法 1、2,得到整体的 UC-SVD 算法框架如图 2 所示。

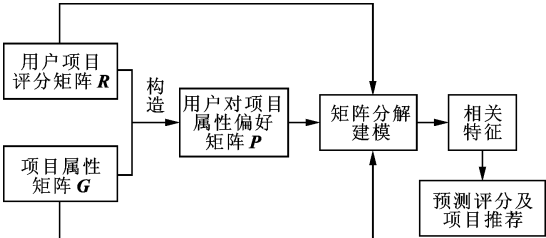


图 2 UC-SVD 算法框架

算法 3 UC-SVD 算法

输入:用户对项目的评分矩阵 $\mathbf{R}$,属性矩阵 $\mathbf{G}$ 。

输出:用户的推荐列表 List[Item]。

(1) $\mathbf{Z} \leftarrow \text{Process}(\mathbf{R}, \mathbf{G})$; / * 对用户对项目评分矩阵 $\mathbf{R}$ 和项目的属性偏好处理,得到用户对项目属性偏好矩阵 * /。

(2)List $[b_u, b_i, \mathbf{P}, \mathbf{Q}, \mathbf{P}_F, \mathbf{Q}_F] \leftarrow \text{OverallUpdate}(\mathbf{R}, \mathbf{G}, \mathbf{Z}, b_u, b_i, \mathbf{P}, \mathbf{Q}, \mathbf{P}_F, \mathbf{Q}_F)$; / * 利用 SGD 进行全局参数更新,直到参数最终达到收敛 * /。

(3) $\mathbf{R}' \leftarrow \text{PreRate}(b_u, b_i, \mathbf{P}, \mathbf{Q}, \mathbf{P}_F, \mathbf{Q}_F)$ / * 使用最优参数进行评分预测,预测评分矩阵 $\mathbf{R}'$ * /。

(4)List[Item_K] $\leftarrow \text{RecommendK}(\mathbf{R}')$ / * 利用预测评分矩阵,对用户未评分过项目的预测评分进行排序,将前 K 个项目推荐给用户 * /。

4 实验和结果分析

4.1 实验数据及实验环境

实验采用 Movielens^[14] 的两个数据集, Movielens100KB 的数据集包括 943 个用户对 1 682 个项目的 100 000 条评分数据和相关电影的属性信息,该数据集的评分稀疏度为 93.70%; Movielens1MB 的数据集包括 6 040 个用户对 3 952 个项目的 1 000 209 条评分数据和相关电影的属性信息,该评分数据集的稀疏度为 95.53%。首先对 Movielens100KB、Movielens1MB 数据集的评分数据和电影类别数据进行处理,由于 Movielens100KB 数据集中存在无效类别,为防止其对结果产生干扰,将其剔

除,这样可以得到的电影类别为 18 项。通过实验,选取用户的类别偏好属性个数为 7,其余 11 个属性置为 0,处理后的电影类别文件 Category 和用户对电影属性的偏好文件 Preference 存储在数据文件中,可提高算法效率。实验中采用的基准算法为 RSVD 算法和 ALS 推荐算法,数据集的 80% 为训练集,20% 为测试集,结果取平均值。

本文实验环境为:Win8.1 操作系统,8 G 内存, Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz,实验程序使用 Scala2.11.7 编写。

4.2 算法评价指标

在实验中,采用常用的均方根误差和平均绝对误差,计算公式为

$$E_m = \left(\frac{1}{|T_{\text{est}}|} \sum_{R_{uj} \in T_{\text{test}}} (\hat{R}_{uj} - R_{uj})^2 \right)^{1/2} \quad (7)$$

$$E_r = \frac{\sum_{R_{uj} \in T_{\text{est}}} |\hat{R}_{uj} - R_{uj}|}{|T_{\text{est}}|} \quad (8)$$

式中: R_{uj} 为测试集中的真实评分; $\hat{R}_{uj}$ 为预测评分; T_{est} 为测试集。 E_r, E_m 越小,则预测的值与真实值的差距越小,推荐的结果越准确,故将其作为改进算法的衡量指标。

4.3 实验结果及分析

对相关参数进行选取,选择使 UC-SVD 算法 E_r, E_m 最优的参数值。更新过程中的主要参数为梯度下降速率 γ_1, γ_2 ,对速率进行实验并选取最优值,结果如表 1、2 所示。对式(6)中的正则项 λ 进行实验并选取最优值,结果如表 3 所示。

表 1 当 γ_2 不变、 γ_1 变化时的 E_m, E_r 值

γ_1	E_m	E_r
0.000 5	0.726 0	0.927 9
0.001 0	0.720 4	0.920 4
0.002 0	0.721 2	0.921 7
0.005 0	0.721 5	0.921 4
0.010 0	0.722 7	0.923 8

表 2 当 γ_1 不变、 γ_2 变化时的 E_m, E_r 值

γ_2	E_m	E_r
0.000 5	0.726 3	0.939 4
0.001 0	0.723 1	0.923 1
0.002 0	0.720 4	0.920 4
0.005 0	0.726 0	0.927 8
0.010 0	0.728 5	0.929 8

表 3 λ 变化时的 E_m 、 E_r 值

λ	E_m	E_r
0.01	0.724 6	0.921 4
0.05	0.720 0	0.920 1
0.10	0.727 3	0.923 5
0.15	0.728 1	0.922 4
0.20	0.727 7	0.923 2

由表 1、2 可知,当 $\gamma_1=0.001$ 、 $\gamma_2=0.002$ 时,改进的 UC-SVD 算法 E_r 、 E_m 取得最优。由表 3 可知,当 $\lambda=0.05$ 时,改进的 UC-SVD 算法的 E_r 、 E_m 取得最优。

对带有正则化项的矩阵分解推荐(RSVD)算法^[15]、基于交替最小二乘的协同过滤推荐(ALS)算法^[16]、本文算法等进行了对比实验,并在 Movielens 的两个数据集上运行这 3 种算法,各算法 E_r 、 E_m 对比结果如图 3 所示。

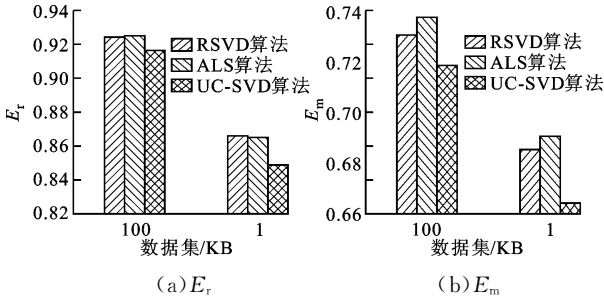


图 3 各算法 E_r 、 E_m 对比

由图 3 可知,本文所提算法的 E_r 值低于 RSVD 以及 ALS 推荐算法的,在 Movielens1MB 数据集上,UC-SVD 算法比 RSVD 算法下降了 2%,比 ALS 算法下降了约 1.8% 左右,说明改进算法在预测准确度上优于对比算法的。尤其是在 Movielens 1MB 数据集实验下,当迭代次数为 400、特征因子为 20 时, E_r 降低到 0.848 7。本文所提算法的 E_m 比 RSVD 算法的下降了 3.1%,比 ALS 算法的下降了约 4%。

为了验证在用户和项目特征因子固定的情况下,各算法的 E_r 、 E_m 随迭代次数变化的情况,在两个数据集下实验对比结果如图 4、5 所示。由图 4 可知,当迭代次数小于 100 时,推荐效果较差;当迭代次数在 100~200 时,3 种对比算法大致相当,当迭代次数大于 200 时,推荐的效果要好于两种对比算法,当迭代次数大于 100 时,UC-SVD 算法的 E_r 好于两种对比算法。由图 5 可知:RSVD、ALS 算法随迭代次数的增加基本保持不变;UC-SVD 算法在迭

代次数小于 100 时,推荐效果较差;在迭代次数大于 100 时,推荐效果好于其余两种对比算法;在大于 50 时,UC-SVD 算法的 E_m 误差结果好于其余两种对

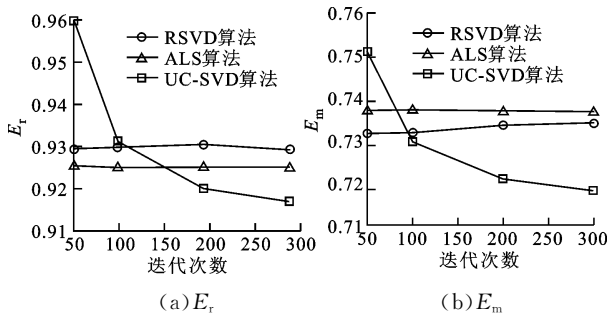


图 4 Movielens100KB 数据集 E_r 、 E_m 随迭代次数变化

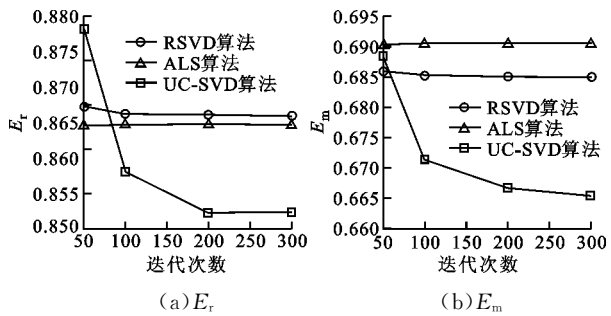


图 5 Movielens1MB 数据集 E_r 和 E_m 随迭代次数变化

比算法。迭代次数固定时,在两个数据集下各算法的 E_r 值随特征因子 k 变化的情况如图 6、7 所示。由图 6 可知:随着 k 的增加,UC-SVD 算法的 E_r 不断降低,当特征因子大于 100 后, E_r 值保持稳定,这说明此时特征因子继续增加不会对推荐产生更积极的影响,整体上 UC-SVD 算法的 E_r 低于其余两种对比算法;UC-SVD 算法的 E_m 误差随着因子的增加逐渐减低,均低于两种对比算法。

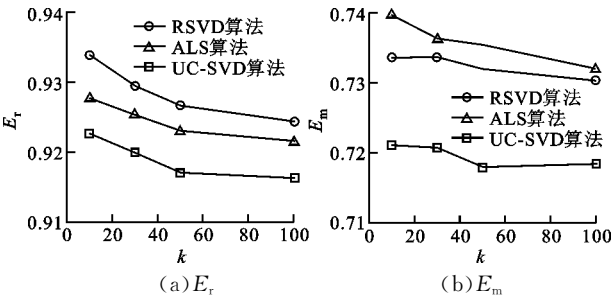


图 6 Movielens100KB 数据集 E_r 、 E_m 随特征因子 k 的变化情况

由图 7 可知:随着 k 的增加,UC-SVD 算法当 k 大于 30 时, E_r 有所增加,但改进算法的 E_r 在整体上优于其余两种对比算法;UC-SVD 算法的 E_m 当 k 大于 30 时有所升高,这可能是由于因子的过大对推

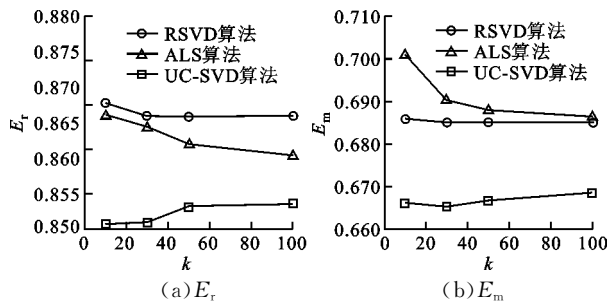


图 7 Movielens1MB 数据集 E_r 、 E_m 随特征因子 k 的变化情况

荐产生负影响,但在整体上远低于两种对比算法。

使用 HetRec2011 数据集来验证本文所提算法在不同数据集下的表现,该数据集数据稀疏度为 96.03%,较 MovieLens100KB、MovieLens 1MB 数据集更为稀疏。由于属性信息中多了 Short、IMAX 两项属性,考虑到该属性对类别的影响较小,将其剔除,只保留项目的 18 个属性,与 Movielens100KB 数据集保持一致,其余数据处理与上述数据集相同,对比结果如表 4 所示。由表 4 可知,在新的稀疏数据集下,UC-SVD 算法表现良好,当 $k=10$ 时,改进算法的 E_r 较 RSVD、ALS 算法的降低了约 3.3%,当 $k=50$ 时,较 RSVD、ALS 算法降低了约 3.7%,UC-SVD 算法在整体上好于其余两种对比算法。

表 4 各算法在 HetRec 数据集下对比

算法	$k=10$		$k=50$	
	E_m	E_r	E_m	E_r
RSVD	0.593 7	0.781 3	0.592 3	0.779 3
ALS	0.607 5	0.781 4	0.597 2	0.779 6
UC-SVD	0.571 0	0.755 3	0.567 4	0.750 7

5 结 论

本文考虑了项目属性的重要作用,通过对项目属性、用户评分的综合考虑,得出用户对项目属性的偏好信息,提出基于用户对项目属性偏好的 UC-SVD 协同过滤推荐算法。先介绍算法的改进思路和算法步骤,通过更新得出算法的最优参数及模型,最后在 Movielens100KB、Movielens1MB 数据集上,将改进的 UC-SVD 算法和传统的经典协同过滤算法进行对比。实验结果表明,加入项目属性因子和用户对项目属性偏好信息使得准确度有所提升, E_r 平均降低 3.5%。为了验证模型在更加稀疏数据集上的作用,本文在 HetRec2011 数据集上进行了算法的对比实验,表明改进算法在更为稀疏的数据

集下表现良好。在实际过程数据比较稀疏的情况下,利用项目的属性信息和用户对项目属性的偏好信息,能够解决一定的稀疏性问题,同时提高推荐的准确率。

参考文献:

[1] 赵长伟,彭勤科,张志勇. 混合因子矩阵分解推荐算法[J]. 西安交通大学学报, 2016, 50(12): 87-91.
ZHAO Changwei, PENG Qinke, ZHANG Zhiyong. A matrix factorization algorithm with hybrid implicit and explicit attributes for recommender systems[J]. Journal of Xi'an Jiaotong University, 2016, 50(12): 87-91.

[2] PAN T, LIU Q, CHANG L I U. Ratings distribution recommendation model-based collaborative filtering recommendation algorithm[C]// The 2nd International Conference on Software, Multimedia and Communication Engineering. [s. n.]: SMCE, 2007: 378-380.

[3] 燕彩蓉,张青龙,赵雪,等. 基于广义高斯分布的贝叶斯概率矩阵分解方法[J]. 计算机研究与发展, 2016, 52(12): 2793-2800.
YAN Cairong, ZHANG Qinglong, ZHAO Xue, et al. A method of Bayesian probabilistic matrix factorization based on generalized Gaussian distribution[J]. Journal of Computer Research and Development, 2016, 52(12): 2793-2800.

[4] 吴金龙. Netflix Prize 中的协同过滤算法[D]. 北京: 北京大学, 2010: 25-34.

[5] BAO Y, FANG H, ZHANG J. TopicMF: simultaneously exploiting ratings and reviews for recommendation[C]// Proceedings of the 28th AAAI Conference on Artificial Intelligence. New York, USA: ACM, 2014: 2-8.

[6] KOREN Y. Factor in the neighbors: Scalable and accurate collaborative filtering[J]. ACM Transactions on Knowledge Discovery from Data, 2010, 4(1): 1-24.

[7] GUO G, ZHANG J, YORKE-SMITH N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings[C]// Proceedings of the 29th AAAI Conference on Artificial Intelligence. New York, USA: ACM, 2015: 123-129.

[8] JAMALI M, ESTER M. A matrix factorization technique with trust propagation for recommendation in social networks[C]// ACM Conference on Recommender Systems. New York, USA: ACM, 2010: 135-142.

[9] MANZATO M G. gSVD++: supporting implicit

- feedback on recommender systems with metadata awareness [C]//ACM Symposium on Applied Computing. New York, USA: ACM, 2013: 908-913.
- [10] MANZATO M G. Discovering latent factors from movies genres for enhanced recommendation [C] // Proceedings of the 6th ACM Conference on Recommender Systems. New York, USA: ACM, 2012: 249-252.
- [11] QIN J, CAO L, PENG H. Collaborative filtering recommendation algorithm based on weighted item category [C]// Control and Decision Conference. Piscataway, NJ, USA: IEEE, 2016: 2782-2786.
- [12] YU Y, WANG C, WANG H, et al. Attributes coupling based matrix factorization for item recommendation [J]. Applied Intelligence, 2016, 46(3): 1-13.
- [13] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012: 186-195.
- [14] HARPER F M, KONSTAN J A. The MovieLens datasets: history and context [J]. ACM Transactions on Interactive Intelligent Systems, 2016, 5(4): 19.
- [15] CARAGEA C, SILVESCU A, MITRA P, et al. Can't see the forest for the trees?: a citation recommendation system [C] // Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries. New York, USA: ACM, 2013: 111-114.
- [16] WINLAW M, HYNES M B, CATERINI A, et al. Algorithmic acceleration of parallel ALS for collaborative filtering: speeding up distributed big data recommendation in spark [C]//21st IEEE International Conference on Parallel and Distributed Systems. Piscataway, NJ, USA: IEEE, 2015: 682-691.

(编辑 赵炜)

(上接第74页)

- [11] BJORNBERG N, KUTTERGAT A N I, VASTRA F A. Screw pump: WO 82/03428 [P]. 1982-10-14.
- [12] JOHANSSON A. Screw pump: WO 91/14869 [P]. 1991-10-03.
- [13] Foilex Engineering AB. Introduction of the Foilex products [EB/OL]. [2017-07-25]. <http://www.foilex.com>.
- [14] JOHANSSON A. Screw pump: WO 00/05503 [P]. 2000-02-03.
- [15] 戴国强, 王清. 基于高黏度介质的单螺杆泵试制研究与应用 [J]. 科技创新与生产力, 2012(4): 105-106.
- DAI Guoqiang, WANG Qing. Trial production and application of single screw pump based on high viscosity medium [J]. Sci-Tech Innovation and Productivity, 2012(4): 105-106.
- [16] 金光熹, 林强. 单螺杆压缩机的泄漏通道和泄漏量计算 [J]. 流体工程, 1986(1): 19-24.
- JIN Guangxi, LIN Qiang. Leakage channel and leakage calculation of single screw compressor [J]. Fluid Engineering, 1986(1): 19-24.
- [17] 李增亮, 石白妮, 鲁佳琪, 等. 圆弧齿形单螺杆式水力机械的啮合原理及应用 [J]. 中国石油大学学报(自然科学版), 2014, 38(4): 132-137.
- LI Zengliang, SHI Baini, LU Jiaqi, et al. Meshing theory and application of single screw hydraulic machinery with arc tooth profile [J]. Journal of China University of Petroleum (Natural Science Edition), 2014, 38(4): 132-137.
- [18] 安永生, 宋扬, 张德实, 等. 螺杆泵转子三维运动仿真分析及型线优化设计 [J]. 中国石油大学学报(自然科学版), 2012, 36(3): 155-158.
- AN Yongsheng, SONG Yang, ZHANG Deshi, et al. Three dimensional motion simulation and locus optimization design of progressive cavity pump rotor [J]. Journal of China University of Petroleum (Natural Science Edition), 2012, 36(3): 155-158.
- [19] 吴伟烽. 单螺杆压缩机啮合副的多圆柱包络啮合理论及加工技术的研究 [D]. 西安: 西安交通大学, 2010.
- [20] 吴伟烽, 冯全科, 徐健. 单螺杆压缩机齿型的多圆柱包络原理 [J]. 西安交通大学学报, 2007, 41(11): 1271-1274.
- WU Weifeng, FENG Quanke, XU Jian. Principle of multi-column envelope couple of single screw compressor [J]. Journal of Xi'an Jiaotong University, 2007, 41(11): 1271-1274.
- [21] 李增亮, 石白妮, 鲁佳琪, 等. 全金属单螺杆泵结构优化设计 [J]. 石油机械, 2013, 41(12): 97-101.
- LI Zengliang, SHI Baini, LU Jiaqi, et al. Optimum design of all-metal screw pump [J]. China Petroleum Machinery, 2013, 41(12): 97-101.
- [22] 杜秀华, 刘双新, 宋玉杰, 等. 短幅内摆线型螺杆-衬套副滑动速度分析 [J]. 化工机械, 2015, 42(4): 522-526.
- DU Xiuhua, LIU Shuangxin, SONG Yujie, et al. Analysis of sliding speed of a short cycloidal screw bushing pair [J]. Chemical Engineering & Machinery, 2015, 42(4): 522-526.

(编辑 荆树蓉 苗凌)