

基于矩阵分解的协同过滤算法

李 改^{1,2,3}, 李 磊^{2,3}

LI Gai^{1,2,3}, LI Lei^{2,3}

1. 顺德职业技术学院, 广东 顺德 528333

2. 中山大学 信息科学与技术学院, 广州 510006

3. 中山大学 软件研究所, 广州 510275

1. Shunde Polytechnic, Shunde, Guangdong 528333, China

2. School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, China

3. Software Institute, Sun Yat-Sen University, Guangzhou 510275, China

LI Gai, LI Lei. Collaborative filtering algorithm based on matrix decomposition. *Computer Engineering and Applications*, 2011, 47(30): 4-7.

Abstract: Collaborative filtering recommendation algorithm is one of the most successful technologies in the e-commerce recommendation system. Aiming at the problem that traditional collaborative filtering algorithms generally exist sparseness resistance and extendibility, in this paper, a CF algorithm, alternating-least-squares with weighted- λ -regularization (ALS-WR) is described. That is, by using regularization constraint to the traditional matrix decomposition model to prevent model overfitting training data and using alternating-least-squares method to train the decomposition model. The experimental evaluation using two real-world datasets shows that ALS-WR achieves better results in comparison with several classical collaborative filtering recommendation algorithms not only in extendibility but also in sparseness resistance.

Key words: recommended systems; collaborative filtering; matrix decomposition; Alternating Least Square (ALS); Singular Value Decomposition (SVD)

摘 要: 协同过滤推荐算法是电子商务推荐系统中运用最成功的一种推荐技术。针对目前大多数协同过滤算法普遍存在的可扩展性和抗稀疏性问题, 在传统的矩阵分解模型(SVD)的基础上提出了一种带正则化的基于迭代最小二乘法的协同过滤算法。通过对传统的矩阵分解模型进行正则化约束来防止模型过度拟合训练数据, 并通过迭代最小二乘法来训练分解模型。在真实的实验数据集上实验验证, 该算法无论是在可扩展性, 还是在抗稀疏性方面均优于几个经典的协同过滤推荐算法。

关键词: 推荐系统; 协同过滤; 矩阵分解; 迭代最小二乘法(ALS); 矩阵奇异值分解(SVD)

DOI: 10.3778/j.issn.1002-8331.2011.30.002 文章编号: 1002-8331(2011)30-0004-04 文献标识码: A 中图分类号: TP18

1 引言

随着互联网的快速发展, 互联网上的数据量急剧增长, 从而使得如何快速而高效地从如此浩瀚的数据海洋中获取我们所需要的信息变得越来越紧迫。在此背景下推荐系统因此应运而生, 推荐系统通过收集和分析用户的各种信息来学习用户的兴趣和行为模式, 根据分析得到的用户的兴趣和行为模式, 来为用户推荐他所需要的服务。这些系统的例子包括: 卓越亚马逊(www.amazon.com)、当当网(www.dangdang.com)为用户推荐各种其可能喜欢的商品, 如书籍、音像、电器、服装等; Netflix 电影出租系统(www.netflix.com)为用户推荐各种其可能喜欢的电影。Google、Baidu、Yahoo 等为用户推荐这种个性化的新闻和搜索服务。推荐系统中运用最广泛的是基于协同过滤的推荐算法^[1-3]。

协同过滤的算法核心是分析用户兴趣, 在用户群中找到

与指定用户的相似(兴趣)用户, 综合这些相似用户对某一信息的评价, 形成系统对该指定用户对此信息的喜好程度预测。近年来协同过滤的算法在国内外得到了广泛研究。如国际上 Koren, A. Paterek, D. D. Lee 等提出了基于传统的矩阵分解模型(SVD)的协同过滤算法^[4-6]。国内的徐翔, 王煦法等对基于SVD的协同过滤算法也进行了相应研究^[7]。国内中山大学的潘嵘还将矩阵分解模型运用到了单类问题^[8-9]。但这些研究都没能很好地解决基于矩阵分解的协同过滤算法的可扩展性及抗稀疏性问题。

本文的主要贡献是: 提出了基于矩阵分解的 alternating-least-squares with weighted- λ -regularization (ALS-WR) 协同过滤算法, 分析了其可扩展性及抗稀疏性问题。进而在真实的数据集上实现所提出的算法, 将其并行化, 比较了其并行化前后的运行效率。在各种稀疏度下, 比较了 ALS-WR 和 SVD

基金项目: 国家自然科学基金(the National Natural Science Foundation of China); 中山大学高性能与网格计算平台资助。

作者简介: 李改(1981—), 男, 博士研究生, 讲师, 研究方向为数据挖掘、推荐系统; 李磊(1951—), 男, 博士, 教授。E-mail: ligai999@126.com

收稿日期: 2011-05-05; 修回日期: 2011-07-18

算法的性能;实验结果表明,ALS-WR算法在各种稀疏度下均优于SVD算法。

2 基本定义和传统的基于矩阵分解的协同过滤推荐算法

2.1 基本定义

在本文中矩阵用斜体大写字母表示(如: R),标量用小写字母表示(如: i, j)。给定一个矩阵 R , R_{ij} 表示它的一个元素, R_i 表示矩阵 R 的第 i 行, R_j 表示矩阵 R 的第 j 列, R^T 表示矩阵 R 的转置。 R^{-1} 表示矩阵 R 的逆。在本文中给定的矩阵 R 表示具有 m 个用户、 n 个对象的评分矩阵,矩阵 U 、 V 分别表示用户和推荐对象的特征矩阵。

2.2 传统的基于矩阵分解的协同过滤推荐算法

传统的基于矩阵分解的协同过滤推荐算法使用SVD方法将用户评分分解为不同的特征及这些特征对应的重要程度,利用用户与项目之间潜在的关系,用初始评分矩阵的奇异值分解去抽取一些本质的特征。SVD是矩阵维数简化的一种常用方法,它将一个 $m \times n$ 的矩阵 R 分解为3个矩阵。 $R = U \times S \times V$, 其中 U 是一个 $m \times m$ 的正交矩阵, V 是一个 $n \times n$ 的正交矩阵, S 是一个 $m \times n$ 的对角矩阵,它的对角线上的元素由上往下依次递减,即 $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, 并且对角线上的元素满足: $(\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0)$, 其非对角线上的元素全为0,所有的 σ_n 从大到小排列,称为奇异值。通过这种矩阵的分解,可以近似地找到一个简化矩阵,即对角矩阵 S , 保留其 K 个最大奇异值的左奇异矩阵组成一个 k 维空间 ($k \times n$), 于是得到一个新的对角矩阵 S_k , 相应的, 矩阵 U , S , V 的维度分别变成 $m \times k$, $k \times k$, $k \times n$, 于是所得的近似矩阵 $R_k = U_k \times S_k \times V_k$, $R_k \approx R$ 。奇异值分解能够产生初始矩阵 R 的所有秩等于 k 的矩阵中与矩阵 R 最近的一个。

算法1 基于SVD的协同过滤推荐算法简化

输入: 用户的评分矩阵 R , 特征个数 d 。

输出: 矩阵 R 的逼近矩阵 X 。

(1) 数据预处理, 将原始矩阵 R 规范化为 R_{norm} ;

(2) 使用SVD对 R_{norm} 进行矩阵分解得到 U , S 和 V ;

(3) 确定适当的维度 k , 将 S 简化为维度是 k 的矩阵, 得到 S_k ;

(4) 相应得到简化矩阵 U_k, V_k ;

(5) 计算 S_k 的平方根记为 $S_k^{1/2}$;

(6) 分别计算两个相关矩阵 $U_k S_k^{1/2}$, $S_k^{1/2} V_k$;

(7) 用户 i 对项目 j 的预测评分为:

$$P(i, j) = \bar{R}_i + U_k S_k^{1/2}(i) S_k^{1/2} V_k(j)$$

通过上述算法可以预测用户对任意项目的评分值, 其中 \bar{R}_i 是用户 i 在所有已评分项目上评分的平均值。

该算法的 k 值不容易选取, 选小了容易失去原始数据中重要的信息和结构; 如果选大了, 达不到降维的目的, 而且容易过拟合训练数据。并且该算法1中要同时对整个矩阵 R_{norm} 进行矩阵分解才能得到 U , S 和 V , 因此难以并行化。这些不足限制了算法的实际运用。

3 基于ALS的协同过滤算法介绍

本章将首先简单介绍基于ALS-WR的协同过滤推荐算法。

与传统的基于矩阵分解的协同过滤推荐算法使用SVD方法来分解矩阵 R 不同的是, 在这里, 希望找到一个低秩矩阵 X 来逼近矩阵 R , 其中 $X = UV^T$, $U \in C^{m \times d}$, $V \in C^{n \times d}$, d 表示特征个数, 一般 $d \ll r$, r 表示矩阵 R 的秩, $r \leq \min(m, n)$ 。

为了找到一个低秩矩阵 X 来最大程度地逼近矩阵 R 。最小化下面的Frobenius损失函数。

$$L(X) = \sum_{ij} (R_{ij} - X_{ij})^2 \quad (1)$$

在上面的目标函数 $L(X)$ 中, $(R_{ij} - X_{ij})^2$ 是低秩逼近中常见平方误差项。下面考虑如何有效并且快速的求解最优优化问题 $\arg \min_X L(X)$ 。

公式(1)可以改写为:

$$L(U, V) = \sum_{ij} (R_{ij} - U_i V_j^T)^2 \quad (2)$$

为了防止过拟合, 给公式(2)加上正则化项, 则公式(2)可改写为:

$$L(U, V) = \sum_{ij} (R_{ij} - U_i V_j^T)^2 + \lambda (\|U_i\|_F^2 + \|V_j\|_F^2) \quad (3)$$

固定 V , 对 U_i 求导 $\frac{\partial L(U, V)}{\partial U_i} = 0$, 得到下面求解 U_i 的公式。

$$U_i = R_i V_{ui} (V_{ui}^T V_{ui} + \lambda n_{ui} I)^{-1}, \quad i \in [1, m] \quad (4)$$

公式(4)中的 R_i 表示用户 i 评过的电影的评分组成的向量, V_{ui} 表示用户 i 评过的电影的特征向量组成的特征矩阵。 n_{ui} 表示用户 i 评过的电影的数量。

同理, 固定 U , 可以得到下面求解 V_j 的公式。

$$V_j = R_j^T U_{mj} (U_{mj}^T U_{mj} + \lambda n_{mj} I)^{-1}, \quad j \in [1, n] \quad (5)$$

公式(5)中的 R_j 表示评过电影 j 的用户的评分组成的向量, U_{mj} 表示评过电影 j 的用户的特征向量组成的特征矩阵。 n_{mj} 表示评过电影 j 的用户数量。

在公式(4)、(5)中 I 表示一个 $d \times d$ 的单位矩阵。

基于公式(4)、(5), 提出下面的基于代正则化的交叉最小二乘法(ALS-WR)的二维协同过滤推荐算法。首先用0均值, 偏差为0.01的高斯随机数初始化矩阵 V , 然后用公式(4)更新 U , 接着用公式(5)更新 V , 直到算法计算出的RMSE值收敛或迭代次数足够多而结束迭代为止。具体算法描述如下:

算法2 基于ALS-WR的二维协同过滤推荐算法

输入: 用户的评分矩阵 R , 特征个数 d 。

输出: 矩阵 R 的逼近矩阵 X 。

(1) 用一个小于1的随机数初始化 V ;

(2) 反复迭代运用公式(4)、(5)更新 U 、 V , 直到算法计算出的RMSE值收敛或迭代次数足够多而结束迭代;

(3) $X = UV^T$, 返回矩阵 X 。

为了分析算法的时间复杂度, 假定 n_r 表示矩阵 R 中评分点的个数, 一般情况下, 由于矩阵 R 高度稀疏, 故有: $n_r \ll m \times n$ 。 n_j 表示特征个数, n_i 表示算法的迭代次数; n_u 表示用户的个数; n_m 表示推荐对象的个数。能够基于标准的矩阵操作得到该算法的时间复杂度。

定理1 对于ALS-WR, 每次更新 U 需要的时间为 $O(n_j^2(n_r + n_j n_u))$, 每次更新 V 需要的时间为 $O(n_i^2(n_r + n_i n_m))$, 如果算法总共迭代 n_i 次后停止, 则其运行时间为

$O(n_f^2(n_r + n_f n_u + n_f n_m)n_i)$ 。

分析算法2可知,该算法的关键是第二步,即反复迭代运用公式(4)、(5)更新 U 、 V 。分析公式(4)、(5)可知,每次调用公式(4)、(5)只是计算更新矩阵 U 、 V 的一行值。故可以对矩阵 U 、 V 进行分割,分成多个等列长的子矩阵来进行并行运算。故本文所提出的ALS-WR协同过滤算法完全可以并行化运算,从而解决了传统的基于矩阵分解的协同过滤算法难以并行化、可扩展性差的问题。并且实验证实本文所提出的ALS-WR算法,在推荐性能上也优于传统的SVD算法。另一方面,分析 $O(n_f^2(n_r + n_f n_u + n_f n_m)n_i)$ 可知,在特征个数 n_f 、迭代次数 n_i 、用户数 n_u 及推荐对象 n_m 一定的情况下,时间复杂度取决于矩阵 R 中评分点的个数 n_r ,即算法运行总的时间复杂度与 n_r 成正比,这也进一步说明了本算法的可扩展性。

4 实验结果及分析

首先介绍本文实验所采用的数据集及评价标准。接着给出了所提出的ALS-WR算法并行化前后的实验结果。最后比较了所提出的ALS-WR算法和传统的SVD算法,给出了在各个数据稀疏度下的实验结果。

4.1 实验数据集

在实验中,使用了两个数据集,一个是Netflix数据集,一个是MovieLens数据集^[2]。

Netflix数据集是Netflix对外发布的一个电影评分数据集^[1,10]。这个数据集包括了480 189个用户在对17 770部电影的103 297 638个评分。所有的评分值都是1到5中的整数值,其中分数越高表示客户对相应电影的评价越高(越喜欢)。这个数据集非常稀疏,有将近99%的评分值未知。从这个数据集中随机抽取140万条评分记录作为测试集TestSet,其余作为训练集TrainSet。

MovieLens数据集是由美国Minnesota大学的GroupLens研究小组创建并维护的^[11,12]。其中包括943个用户对1 682部电影的100 000条评分记录。所有的评分值也都是1到5中的整数值,其中分数越高表示客户对相应电影的评价越高(越喜欢)。这个数据集也非常稀疏,稀疏度为6.305%,即6.305%的项有评分。从该数据集中随机抽取80%的评分数据作为第一个训练集,记为Train80;把Train80中评分数据的7/8抽取出来组成另一个训练集,记为Train70。依次构造训练集Train60、Train50、Train40。评分数据中除Train80以外的数据集构成测试集,记为Test20。

为了更好地比较并行化前后ALS-WR的运行效率,在较大规模的Netflix数据集上运行所提出的并行算法。由于SVD难以并行化,难以处理大数据集,故在MovieLens数据集,比较各个数据稀疏度下ALS-WR算法和传统的SVD算法的性能优劣。

4.2 实验的评价标准

本文实验采用RMSE作为评价标准^[2],RMSE通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性。RMSE为推荐质量提供了直观的度量方法,是最常用的一种推荐质量度量方法。推荐算法整体的RMSE越小,意味着

推荐的质量越高。假设算法对 N 个项目预测的评分向量表示为 $\{p_1, p_2, \dots, p_N\}$,对应的实际用户评分集合为 $\{r_1, r_2, \dots, r_N\}$,则算法的RMSE表示为:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_i - r_i)^2}{N}} \quad (6)$$

4.3 实验结果

4.3.1 ALS-WR算法并行化前后的实验结果

在此采用Hadoop平台下的MapReduce技术来对提出的ALS-WR算法进行并行化运算^[13,15]。进行了三个实验,分别是ALS算法在单节点的实现,在一台Master、2台Slave的Hadoop集群中的实现和在一台Master、5台Slave的Hadoop集群中的实现,所有的机器都是HP计算机,每台计算机配置为4颗Intel® Core™ i7处理器,8 GB内存。这些机器都处于同一个局域网内。实验中,设定迭代次数为10,特征个数分别为10,20,30,40,50。最终实验结果如图1所示。

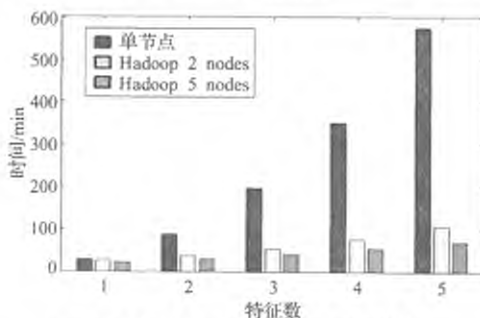


图1 ALS并行化前后的运行时间比较

横坐标为特征个数,纵坐标为时间(min)。从图1可以看出并行运算节点数和ALS-WR算法中特征矩阵的特征个数越多,ALS-WR算法并行化前后的运算效率差别越明显。

4.3.2 ALS-WR算法和几个经典的CF算法的性能及抗稀疏性比较

在本节将比较所提出的ALS-WR算法和几个经典的协同过滤(CF)算法的性能及抗稀疏性。比较的算法有传统的SVD算法、基于用户的KNN算法(KNN-USER)^[1,11,16]和基于项目的KNN算法(KNN-ITEM)^[1,11,17,18]。在基于用户的KNN算法中,用户对产品的评分预测是基于与此用户相似的用户所给予的评分获得的,这类KNN模型假设如果两个用户过去有相似的兴趣,那么他们现在也同样保持相近的兴趣。基于项目的KNN算法假设一个用户现在保持着和过去相近的兴趣,也就是如果某个产品和用户过去喜欢的产品相似,那么此产品现在也很有可能被用户所喜欢。在本实验中基于用户的KNN算法和基于项目的KNN算法中的相似度计算均采用Pearson相关系数。

图2显示的是ALS-WR算法和几个经典的协同过滤算法的在采用RMSE作为性能评价指标时的性能对比。各个算法均在Train80训练集上训练,在Test20上测试得到RMSE值。横轴表示特征矩阵的特征个数,从1依次加2变化到40,ALS-WR算法在每个特征数下均迭代40次直到收敛。纵轴表示RMSE值。在本实验中,给出的是使得基于用户的KNN算法和基于项目的KNN算法取得最优性能的RMSE值。由于两个KNN算法均与特征数无关,故在Train80训练集上训练,在

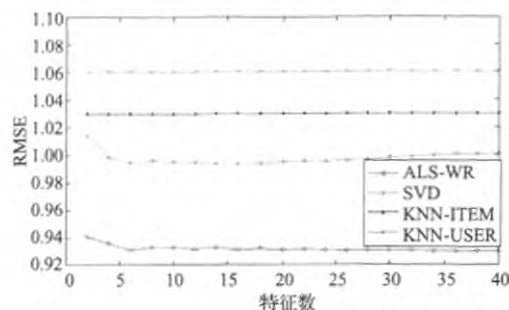


图2 ALS-WR算法和几个经典的CF算法的性能比较

Test20上测试得到的最优RMSE值在各个特征数下均一致。从图2中可以看出在各个特征数下,ALS-WR算法的RMSE值均大幅度优于几个经典的协同过滤算法。而且ALS-WR算法在6个特征后RMSE值即趋于稳定,并随特征数增加RMSE值单调下降,单调下降的原因是ALS-WR算法在矩阵分解的过程中加入了正则化,以对特征矩阵 U 、 V 进行约束,防止了过拟合现象发生;而传统的SVD算法没有加入这个约束,随着特征数增加到一定数值后RMSE反而上升,也即在训练集上出现了过拟合现象。

图3显示的是ALS-WR算法和几个经典的协同过滤算法在Train40—Train80这5个不同稀疏度的数据集上训练模型,在Test20上测试得到的试验结果。在本次实验中ALS-WR算法和传统的SVD算法中特征矩阵的特征个数均取20个,基于用户的KNN算法和基于项目的KNN算法的实验结果取在各个数据集上运用不同的近邻数所计算出的最优值。比较4个算法在各种稀疏度下的性能。为了避免随机误差,每个实验均运行10次,然后取均值作为最终实验结果。本实验中ALS-WR算法的参数 λ 在各个数据集上的取值通过交叉验证确定,取使得ALS-WR算法性能最好的 λ 值。如在Train80数据集上最好的 λ 值取0.125。

图3中X轴上的数字表示Train40—Train80中的某个训练集,如X轴的坐标为40,则表示训练集选择的是Train40。Y轴表示在Test20上测试得到的RMSE值。从图3中可以看出ALS-WR算法在各个稀疏度下均优于传统的SVD算法、基于用户的KNN算法和基于项目的KNN算法。随着训练数据稠密度的提高,ALS-WR算法的性能提高得比几个经典的CF算法都快,并且数据越稠密,性能提高的幅度越明显。

5 总结

本文在传统的矩阵分解模型SVD算法的基础上提出了一种新的基于矩阵分解的Alternating-Least-Squares with Weighted- λ -Regularization(ALS-WR)协同过滤算法,并分析了其可扩展性及抗稀疏性问题。进而在真实的数据集上实现所提出的算法,将其并行化,比较了其并行化前后的运行效率。并在各种稀疏度下,比较了ALS-WR和几个经典的CF算法的性能;实验结果表明,ALS-WR算法在各种稀疏度下均优于几个经典的CF算法。在以后的工作中还将考虑ALS-WR算法的冷启动问题以及与其他算法结合提出更加高性能的混合模型。

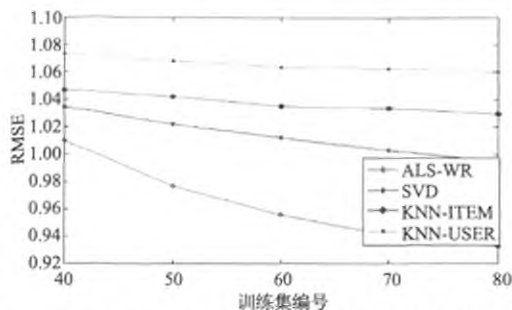


图3 ALS-WR算法和几个经典的CF算法的抗稀疏性比较

参考文献:

- [1] Wu J L. Collaborative filtering on the Netflix prize dataset[D/EB]. <http://dsec.pku.edu.cn/~jinlong/>.
- [2] Ricci F, Rokach L, Shapira B, et al. Recommender system handbook[M]. [S.l.]: Springer, 2011.
- [3] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. TKDE, 2005, 17(6): 734-749.
- [4] Bell R, Koren Y, Volinsky C. The bellkor 2008 solution to the Netflix prize[R]. 2007.
- [5] Paterek A. Improving regularized singular value decomposition for collaborative filtering[C]//KDD-Cup and Workshop. [S.l.]: ACM Press, 2007.
- [6] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 2001, 401: 788-791.
- [7] 徐翔, 王煦法. 基于SVD的协同过滤算法的欺诈攻击行为分析[J]. 计算机工程与应用, 2009, 45(20): 92-95.
- [8] Pan R, Zhou Y, Cao B, et al. One-class collaborative filtering[C]//IEEE International Conference on Data Mining (ICDM), 2008.
- [9] Pan R, Martin S. Mind the Gaps: weighting the unknown in large-scale one-class collaborative filtering[C]//International Conference on Knowledge Discovery and Data Mining (KDD), 2009.
- [10] Netflix. Netflix prize[EB/OL]. <http://www.netflixprize.com>.
- [11] 罗辛, 欧阳元新, 熊璋, 等. 通过相似度支持度优化基于K近邻的协同过滤算法[J]. 计算机学报, 2010, 33(8).
- [12] 汪静, 印鉴, 郑利荣, 等. 基于共同评分和相似性权重的协同过滤推荐算法[J]. 计算机科学, 2010, 37(2).
- [13] Hadoop[EB/OL]. <http://hadoop.apache.org/>.
- [14] Apache MapReduce Architecture[EB/OL]. http://hadoop.apache.org/mapreduce/docs/current/cn/mapred_tutorial.html.
- [15] White T. Hadoop 权威指南[M]. 周敏, 曾大刚, 周傲英, 译. [S.l.]: [s.n.], 2010.
- [16] Herlocker J, Konstan J, Borchers A, et al. An algorithmic framework for performing collaborative filtering[C]//Proceedings of the 1999 Conference on Research and Development in Information Retrieval (SIGIR-99), 1999.
- [17] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering[C]//IEEE Internet Computing, 2003.
- [18] Sarwar B, Karypis G, Konstan J. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of 10th World Wide Web (WWW10) Conference, Hong Kong, 2001.

作者: 李改, 李磊, LI Gai, LI Lei
作者单位: 李改, LI Gai (顺德职业技术学院, 广东顺德528333; 中山大学信息科学与技术学院, 广州510006; 中山大学软件研究所, 广州510275), 李磊, LI Lei (中山大学信息科学与技术学院, 广州510006; 中山大学软件研究所, 广州510275)
刊名: 计算机工程与应用 **ISTIC PKU**
英文刊名: Computer Engineering and Applications
年, 卷(期): 2011, 47(30)
被引用次数: 1次

参考文献(18条)

1. [Wu J L Collaborative filtering on the Nefifix prize dataset](#)
2. [Ricci F. Rokach L. Shapira B Recommender system handbook](#) 2011
3. [Adomavicius G. Tuzhilin A Toward the next generation of recommender systems:a survey of the state-of-the-art and possible extenstions](#) 2005(06)
4. [Bell R. Koren Y. Volinsky C The bellkor 2008 solution to the Netflix prize](#) 2007
5. [Paterek A Improving regularized singular value decomposition for collaborative filtering](#) 2007
6. [Lee D D. Seung H S Leaming the parts of objects by non-negative matrix factorization](#)[外文期刊]
7. 徐翔. 王煦法 基于SVD的协同过滤算法的欺诈攻击行为分析[期刊论文]-计算机工程与应用 2009(20)
8. [Pan R. Zhou Y. Cao B One-class collaborative filtering](#) 2008
9. [Pan R. Martin S Mind the Gaps:weighting the unknown in largescale one-class collaborative filtering](#) 2009
10. [Netflix Netflix prize](#)
11. 罗辛. 欧阳元新. 熊璋 通过相似度支持度优化基于K近邻的协同过滤算法[期刊论文]-计算机学报 2010(08)
12. 汪静. 印鉴. 郑利荣 基于共同评分和相似性权重的协同过滤推荐算法[期刊论文]-计算机科学 2010(02)
13. [Hadoop\[E B/OL\]](#)
14. [Apache MapReduce Architecture](#)
15. [White T. 周敏. 曾大聘. 周傲英 Hadoop权威指南](#) 2010
16. [Herlocker J. Konstan J. Borchers A An algorithmic framework for performing collaborative filtering](#) 1999
17. [Linden G. Smith B. York J Amazon.com recommendations:Itemto-item collaborative filtering](#)[外文期刊] 2003
18. [Sarwar B. Karypis G. Konstan J ltem-based collaborative filtering recommendation algorithms](#) 2001

引证文献(1条)

1. [沈韦华. 陈洪涛. 沈锦丰 基于最佳匹配算法的精密零件检测研究](#)[期刊论文]-科技通报 2013(5)

本文链接: http://d.wanfangdata.com.cn/Periodical_jsjgcyty201130002.aspx