计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(5):198−202 [doi: 10.15888/j.cnki.csa.006338]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

BC-AW 协同过滤推荐算法研究①

张志强[1,2], 李　改[3]

[1](顺德职业技术学院 电子与信息工程学院, 顺德 528300)
[2](华中科技大学 计算机科学与技术学院, 武汉 430074)
[3](中山大学 信息科学与技术学院, 广州 510006)
通讯作者: 张志强, E-mail: gd229@126.com

摘　要: 针对现有协同过滤算法普遍存在数据稀疏、可扩展性低、计算量大的缺点, 提出一种基于 BC-AW 的协同过滤推荐算法, 引入联合聚类 (BlockClust, BC) 和正则化迭代最小二乘法 (Alternating least squares with Weighted regularization, AW), 首先对原评分矩阵进行用户—项目双维度的联合聚类, 接着产生具有相同模式评分块的多个子矩阵, 通过分析得出这些子矩阵规模远小于原评分矩阵, 从而有效降低预测阶段的计算量. 然后分别对每个子矩阵应用正则化迭代最小二乘法来预测子矩阵的未知评分, 进而实现推荐. 经仿真实验表明, 本文算法与传统的协同过滤算法比较, 能有效改善稀疏性、可扩展性和计算量的问题.

关键词: 协同; 过滤; 联合聚类; 正则化迭代最小二乘法

引用格式: 张志强,李改.BC-AW 协同过滤推荐算法研究.计算机系统应用,2018,27(5):198–202. http://www.c-s-a.org.cn/1003-3254/6338.html

Study on BC-AW Collaborative Filtering Recommendation Algorithm

ZHANG Zhi-Qiang[1,2], LI Gai[3]

[1](School of Electronics and Information Engineering, Shunde Polytechnic, Shunde 528300, China)
[2](School of Computer Science and Technology, Huazhong University of Science &Technology, Wuhan 430074, China)
[3](School of Information Science &Technology, Sun Yat-Sen University, Guangzhou 510006, China)

Abstract: Aiming at the weaknesses of sparse data, low scalability and large computing existing in the current collaborative filtering algorithm, a BlockClust-Alternating least squares with Weighted regularization (BC-AW) collaborative filtering recommendation algorithm is proposed. Firstly, the user and the item of the original scoring matrix are jointly clustered and several submatrixes with the same scoring mode are generated. According to the research, the scale of these submatrixes is far less than the original scoring matrix which effectively decreases the computational complexity in the prediction process. Then, the regularized iterative least-square method is applied to each submatrix to predict its score. Hence recommendation is realized. The simulation results reveal that the proposed algorithm can effectively improve sparsity, expand scalability, and reduce computing compared with the traditional one.

Key words: collaborative; filtering; joint cluster; regularized iterative least-square method

当今社会, 基于互联网技术的电子商务不断普及, 大数据分析和大数据挖掘已成为迫切要解决的问题. 推荐系统是指根据用户的兴趣特点和购买行为, 向用户推荐用户感兴趣的信息和商品的系统. 其中协同过滤推荐 (collaborative filtering recommendation) 是推荐系统中应用最早和最为成功的技术之一, 它一般采用最近邻技术, 利用用户的历史喜好信息计算用户之间的距离, 然后 利用目标用户的最近邻居用户对商品评

① 基金项目: 国家自然科学基金(41072247); 广东省自然科学基金(2016A030310018); 广东省哲学社会科学项目(GD16XJY36); 顺德职业技术学院重点教研项目 (2014-SZJGXM06)
收稿时间: 2017-06-21; 修改时间: 2017-07-17; 采用时间: 2017-09-25; csa 在线出版时间: 2018-04-23

价的加权评价值来预测目标用户对特定商品的喜好程度, 系统从而根据这一喜好程度来对目标用户进行推荐. 协同过滤最大优 点是对推荐对象没有特殊的要求, 能处理非结构化的复杂对象, 如音乐、电影等媒体数据[1]. 其中的潜在因素模型 (latent factor models) 的核心思想是: 通过分析历史数据来预测事物的发展趋势, 矩阵分解算法 (matrix factorization) 是其中最为成功的算法之一[2]. 但随着互联网数据的不断扩大, 应用传统的矩阵分解算法构造大数据潜在因素模型, 算法性能会大大下降, 难以实现有效的协同过滤, 其关键问题体现在数据稀疏性和算法可扩展性的改进上.

针对上述问题, 国内外学者专家们提出了一些改进想法: Pilaszy I 最先提出基于 ALS 的协同过滤算法, 相对传统算法而言, 能有效提高过滤质量和推荐速度, 但该算法只考虑用户的相似性[3]; 李改等对前者进行改进, 提出了 ALS-WR 协同过滤算法, 该算法考滤到可扩展性方面, 但需要专门设计迭代式算法[4]; 王辉等把用户聚类思想引入到协同过滤算法中, 改进了传统算法的数据稀疏性和可扩展性, 但精确度较低[5] .

本文提出一种基于 BC-AW 的协同过滤推荐算法, 在传统算法的基础上, 引入联合聚类 BC(BlockClust) 和正则化迭代最小二乘法 (Alternating least squares with Weighted regularization, AW). 首先分解原始评分矩阵的用户项目双维度, 然后使用联合聚类计算出多个同类评分块的子矩阵, 最后使用正则化迭代最小二乘法预测各个子矩阵的未知评分, 从而计算出推荐结果. 使用联合聚类所求得的子矩阵远比初此矩阵规模小, 可以大大减少过滤推荐的计算量. 通过与 BaseMF 算法、RSTE 算法、TidaTrust 算法、MoleTrust 算法进行对比实验, 分析结果表明, 本文算法可以有效缓解传统算法并行化、可扩展性差的问题.

1 AW 算法

设定矩阵R代表m个用户、n个对象的评分矩阵、矩阵U代表用户特征矩阵、矩阵V代表推荐对象特征矩阵. $R_{i,j}$代表R中的某个元素, R_i代表R的第i行, R_j代表R的第j列, R^{T}代表R的转置. R^{-1}代表R的逆. 传算法一般使用 SVD 算法来分解R[6], 但本文算法是通过计算低秩矩阵X来逼近R, 使得$\mathrm{X}=\mathrm{UV}^{\mathrm{T}}$, $\mathrm{U}\in C^{m\times d}$, $\mathrm{V}\in C^{n\times d}$(其中$d$代表R的特征数), 一般状态下, $d\leqslant r$, r代表R的秩, $r\leqslant\min(m,n)$.

Step1: 我们通过最小化 Frobenius 的损失函数来算出一个低秩矩阵X, 令X的得尽量接近R, 如公式 (1) 所示:

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

上述公式 (1) 中, $(R_{ij}-X_{ij})^2$为低秩逼近中一般平方误差项.

Step 2: 求解$L(\mathrm{X})$的最优化问题$\arg\min_x L(\mathrm{X})$, 于是改写公式 (1) 如下:

$$L(rmU,V)=\sum_{ij}(R_{ij}-U_iV_j^{\mathrm{T}})^2 \tag{2}$$

上述式 (2) 会产生过于拟合的问题, 因此需要通过正则化项来解决这个问题, 则改写如下:

$$L(\mathrm{U},\mathrm{V})=\sum_{ij}\left(R_{ij}-U_iV_j^{\mathrm{T}}\right)^2+\lambda\left(U_{iF}^2+V_{jF}^2\right) \tag{3}$$

Step3: 对上面公式 (3), 设定V不变, 对U_i求导$\partial L(\mathrm{U},\mathrm{V})/\partial U_i=0$, 求解$U_i$, 如公式 (4) 所示:

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

公式 (4) 中, R_i为用户所评过影片的向量矩阵, 由评分组组成, V_{ui}为用户i所评过影片的特征矩阵, 由特征向量组成, n_{ui}为用户i所评过的影片数, I为$d\times d$阶的单位矩阵.

Step4: 上述对公式 (4) 中, 设定U不变, 可以通过公式 (5) 求解V_j:

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

上述公式 (5) 中, R_j为用户i所评过影片j的评分所形成的向量, U_{mj}为用户i评过影片j的特征矩阵, 它由用户群中的特征向量组成, n_{mj}为用户i评过影片j的用户数量.

2 BC 算法

BC 算法能够将用多个较小且高相似度的评分矩阵来替代原始数据[7]. 通过扫描该矩阵中的评分, 使用聚类方法计算行和列, 计算一次后, 重新评估每个子矩阵中用户、项目的概率, 如此类推, 直到产生收敛为止[8]. 然后将评分匹配到最大概率的子矩阵中. 其算法思想如下:

首先扫描评分矩阵, 通过计算各评分的所属概率$p(k|u,v,r)$来判断该评分是否属于某个子矩阵, 其参数包括: 用户—项目属于某子矩阵的概率$p(k|u)$和$p(k|v)$、该评分值属于某子矩阵中的概率$p(r|k)$. 其算法公式如下 (其中$\alpha=0.000\,0001$, $\beta=0.000\,0001$, $\theta=0.000\,0001$):

$$p(k|u,v,r)=\frac{(p(k|u)+\alpha)\times(p(k|v)+\beta)\times(p(r|k)+\theta)}{\sum_{k'\in k}(p(k'|u)+a)\times(p(k'|v)+\beta)\times(p(r|k'+\theta))} \quad (6)$$

$$p(k|u)=\frac{\sum_{V\in V(u)}p(k|u,v,r)}{\sum_{k'\in k}\sum_{V\in V(u)}p(k'|u,v,r)} \quad (7)$$

$$p(k|v)=\frac{\sum_{u\in U(V)}p(k|u,v,r)}{\sum_{k'\in k}\sum_{u\in U(V)}p(k'|u,v,r)} \quad (8)$$

$$p(r|k)=\frac{\sum_{p}(k|u,v,r)}{\sum_{r'}\sum_{V\in U(V)}p(k'|u,v,r)} \quad (9)$$

以上公式中, u表示当前用户, v表示当前项目, r表示评分值, r'表示 1 到 5 的整数, k表示当前聚类, k'表示当前累加聚类, $U(v)$表示对v评了分的所有用户集合, $V(u)$表示u已评过分的所有项目集合. BC 算法把原矩阵分为k个具有高密度、高相似度的子矩因此, 该算法具有一定的降维作用[9].

3 BC-AW 算法

以下通过联合聚类和 AW 协同过滤两阶段来对评分矩阵的未知项进行预测.

1) 联合聚类

输入: 子矩阵个数k, 用户—项目评分矩阵R.

输出: 子矩阵数k.

Step1: u表示用户量, v表示项目数, $p(k|u,v,r)$表示评分r属于聚类k的概率, 分别初始化u、v、$p(k|u,v,r)$, 使得$\sum' kp(k'|u,v,r)=1$.

Step2: 应用公式 (7) 计算用户u属于聚类k的概率$p(k|u)$; 应用公式 (8) 计算用户v属于聚类k的概率$p(k|v)$; 应用公式 (9) 算出分值概率$p(r|k)$; 应用公式 (6) 算出$p(k|u,v,r)$.

Step3: 最后选择概率k的最大值作为该评分的子矩阵.

Step4: 跳回步骤 2, 直到出现收敛, 结束循环.

2) AW 协同过滤

首先用偏差小于等于的高斯随机数初始化矩阵 V, 接着分别用公式 (4) 和公式 (5) 更新U和V, 重复进行多次迭代, 直到得出的 *RMSE* 值出现收敛为止, 结束迭代.

输出: 用户评分矩阵R, 特征个数d.

输出: R的逼近矩阵X.

Step1: 用偏差小于等于 0.01 的高斯随机数初始化 V;

Step2: 分别用公式 (4) 和公式 (5) 更新U和V;

Step3: 重复迭代公式 (4)、公式 (5), 判断得出的 *RMSE* 值是否收敛, 如果是, 则迭代结束;

Step4: 令$X=UV^T$, 返回矩阵X.

下面我们分析该算法的时间复杂度: 设n_r为在矩阵R里的评分点个数, $n_r\leqslant m\times n$(一般情况下, 矩阵R稀疏). n_f为特征个数, n_t为该算法的迭代数; 设n_u为用户数; 设n_m为对象推荐数. 变量U每次更新的时间为$O(n_f^2(n_r+n_fn_m))$, 变量V每次更新的时间为$O(n_f^2(n_r+n_fn_m))$, 如此类推, 在迭代n_t次后的时间则为$O(n_f^2(n_r+n_fn_u+n_fn_m)n_t)$. 由此可见, 在$n_f$、$n_r$、$n_u$以及$n_t$恒定的情况下, $O(n_f^2(n_r+n_fn_u+n_fn_m)n_t)$取决于$n_r$, 也就是说, 算法的总时间复杂度与$n_r$成正比, 可证该算法具有一定的可扩展性. 在 AW 协同过滤中, Step 2 是最关键的步骤, 在这个步聚里, 每次应用公式 (4) 和公式 (5) 只能更新U和V中的一行值, 效率较低. 因此可以先将U和V分解为n个列相等的子矩阵, 然后, 应用公式 (4) 和公式 (5) 对每个子矩阵并行更新, 这样可以大大提高运算效率.

综上所述, 本文基于矩阵分解的 BC-AW 协同过滤推荐算法改进传统算法的串行运算方式, 实现并行运算, 可以有效地解决传统算法中存在的可扩展性差问题.

4 实验分析

4.1 数据准备

本实验目的是通过分析比较本文算法和传统算法的性能, 硬件方面选用基于云计算的分布式硬件实验平台, 该平台由 20 台电脑组成 20 个运算节点, 每个节点配有 4.0 Hz 的 Intel 处理器和 32 G 的内存, Linux 操作系统. 实验数据来自美国 Minnesota 大学 GroupLens 项目组的 MovieLens 数据集. MovieLens 数据集是 GroupLens 项目组开发的一个互联网研究型推荐系统数据集. MovieLens 数据集中含有 943 个用户对 1682 部电影的 100 000 条评分数据, 平均每个用户评分的电影大于等于 20 部[10].

4.2 评价指标

本次实验采用均方根误差 *RMSE* 和评分覆盖率 COV_R (Rating Coverage) 作为算法推荐质量的评价指标.

$$RMSE=\sqrt{\sum_{(u,i)|R_{\text{test}}}(r_{ui}-\hat{r}_{ui})^2/R_{\text{test}}} \quad (10)$$

其中r_{ui}代表用户的实际评分, $\hat{r}_{ui}$代表用户对影片的预测评分, R_{test}代表影片的评分数. *RMSE* 的值越小表示算法的精度越高. Rating Coverage 是指在测试集中, 推荐系统可预测的评分数量占总评分的百分比[11].

$$COV_R = \frac{\text{可预测的评分数}}{\text{整个测试集的评分个数}} \tag{11}$$

4.3 实验结果及分析

本文算法分别与 BaseMF (基于矩阵分解算法) [12]; RSTE (概率矩阵分解算法) [13]; TidaTrust (基于 TidaTrust 模型推荐算法) [14]; MoleTrust (基于 MoleTrust 模型推荐算法) [15]这 4 种推荐方法通过分析 RMSE 和COV_R的结果值进行分析比较. 实验开始, 首先随机选取两组训练集 GROUP1 和 GROUP2: GROUP1 为 80% 的 Epinions 数据集, GROUP1 为 90% 的 Epinions 数据集. 然后分别把本文算法、BaseMF 算法、RSTE 算法、TidaTrust 算法、MoleTrust 算法所建立的模型应用到训练集上进行学习, 指定y、λ和f为潜在维度 ($y = 0.005$, $\lambda = 0.003$, $f = 0.001$), k为实验基础 ($k = 4$). 反复实验 5 次, 求平均值作为最终实验结果. 图 1 为这 5 种算法得出的 *RMSE* 值比较图.

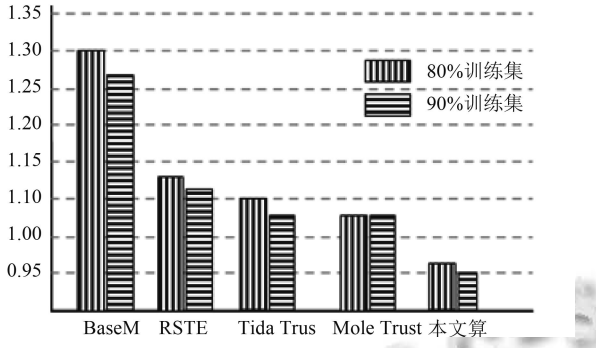

图 1 5 种算法的 RMSE 值比较图

从上图可以看出, 针对 *RMSE* 值, 当训练集为 80% 时, 本文算法比 BaseMF 算法降低了 22%, 比 RSTE 算法降低了 11%, 比 TidaTrust 算法降低了 8%, 比 MoleTrust 算法降低了 7%; 当训练集为 90%, 本文算法比 BaseMF 算法降低了 21%, 比 RSTE 算法降低了 10%, 比 TidaTrust 算法降低了 7%, 比 MoleTrust 算法降低了 7%; 由此可知, 本文算法的误差值明显低于传统的 4 种推荐算法.

图 2(a)、图 2(b) 分别表示本文算法在 GROUP1、GROUP2 这两组训练集中, *RMSE* 在不同k值下的变化情况. k值越大, *RMSE* 值越小, 证明推荐精度越高. 当 $k \in [0,4]$时, *RMSE* 减小的幅度最大, 迭代时间的增幅最小; 当$k \geqslant 4$时, *RMSE* 的变化幅度趋于 0. 从而得出当 $k = 4$时, 算法的推荐性能最佳.

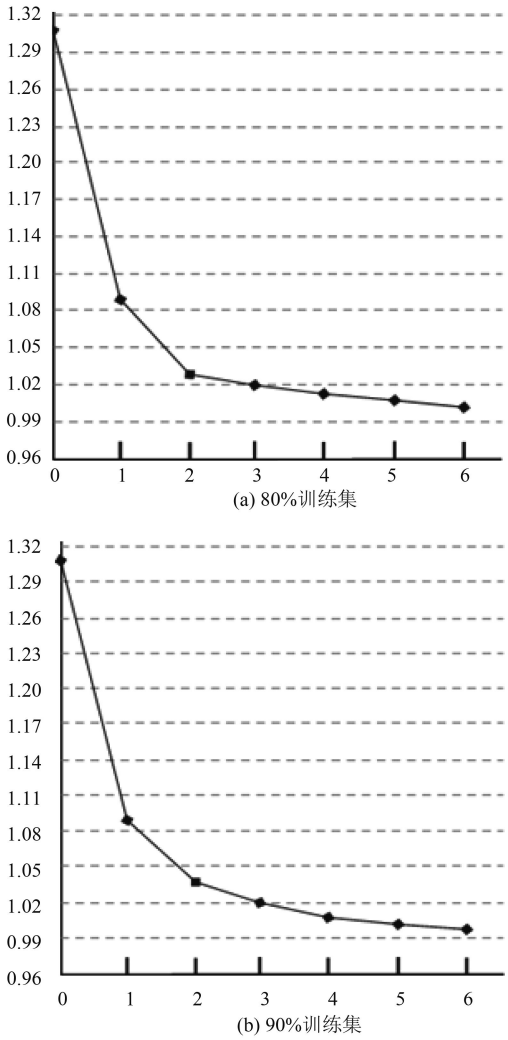

图 2 RMSE 随参数的变化曲线

下面分析 5 种算法的COV_R. 我们知道, COV_R直接影响用户对影片的可选择范围, COV_R越高, 用户对影片的可选择范围越广, 从而用户满意度越高. 采用 GROUP1 数据集 (即 80% 的训练集), 实验结果如表 1 所示.

表 1 5 种算法的 COV_R

方法	COV_R(%)
BaseMF	20.19
RSTE	51.62
TidaTrust	84.87
MoleTrust	82.13
本文算法	98.71

分析表 1 可以看出, 当训练集为 80% 时, 本文算法的cov_R比 BaseMF 算法提高了 68.52%,比 RSTE 算法提高了 38.09%, 比 TidaTrust 算法提高了 4.84%, 比 MoleTrust 算法提高了 6.58%. 由此可见, 本文算法比传统 4 种算法具有更广的覆盖范围.

4 结论与展望

本文针对现有协同过滤算法中普遍存在的数据稀疏、可扩展性低这两个核心问题, 提出 BC-AW(基于联合聚类和迭代最小二乘法) 两阶段协同过滤算法. 首先由该算法通过对原评分矩阵进行用户—项目双维度联合聚类后得到的若干个子矩阵 (这些子矩阵均为模式相同的评分块), 其规模比原矩阵小得多, 因此可以有效减少预测工作量; 再者, 采用正则化迭代最小二乘法预测子矩阵的未知评分可以优化推荐效率. 该算法在模拟大数据实验中 (美国 Minnesota 大学 GroupLens 项目组的 MovieLens 数据集), 通过与几个经典的协同过滤算法 (BaseMF 算法、RSTE 算法、TidaTrust、MoleTrust 算法) 作比较. 实验结果表明, 本文算法能有效改进推荐系统的稀疏性、可扩展性问题, 系统预测评分与用户实际评分接近, 并能为用户提供良好的使用体验.

参考文献

1 Shuo LX, Chai BF, Zhang XD. Collaborative filtering algorithm based on improved nearest neighbors. Computer Engineering and Applications, 2015, 51(5): 137–141.

2 Zhang HJ. The research and application of distributed matrix factorization algorithm in recommend system. Bulletin of Science and Technology, 2013, 29(12): 151–153.

3 Wang QM, Miao Y, He M, *et al*. Parallelized research on collaborative filtering algorithm based on matrix factorization. Computer Technology and Development, 2015, 25(2): 55–59.

4 Zhu X, Song AB, Dong F, *et al*. A collaborative filtering recommendation mechanism for cloud computing. Journal of Computer Research and Development, 2014, 51(10): 2255–2269.

5 Bi XR. Collaboration filtering recommendation algorithm of sub-similarity integration between items. Computer Systems & Applications, 2015, 24(1): 147–150.

6 Wang L, Fu XF, Wang XM. Hybrid dynamic collaborative filtering algorithm based on big data sets. Journal of Guangdong University of Technology, 2014, 31(3): 44–48.

7 Chen W, Shi QL. An adaptive algorithm for collaborative filtering recommendation. Journal of Xianyang Normal University, 2014, 29(6): 47–49.

8 Li G, Li L. One-class collaborative filtering based on matrix factorization. Application Research of Computers, 2012, 29(5): 1662–1665.

9 Ke LW, Wang J. Collaborative filtering recommendation based on user feature transfer. Computer Engineering, 2015, 41(1): 37–43.

10 Yi ZA, Mu CM. Collaborative filtering algorithm based on subtractive clustering and genetic fuzzy. Computer & Digital Engineering, 2014, 42(8): 1363–1367.

11 Xu W, Duan F. Combining clustering and collaborative filtering for implicit recommender system. Computer Engineering and Design, 2014, 35(12): 4181–4185.

12 Liu L. A collaborative filtering recommender algorithm based on iterative kernel method. Information Technology & Informatization, 2014, (12): 76–81.

13 Zha J, Li ZB, Xu GQ. An optimised collaborative filtering algorithm based on combined similarity. Computer Applications and Software, 2014, 31(12): 323–328.

14 Wu HC, Wang XJ, Cheng Y, *et al*. Advanced recommendation based on collaborative filtering and partition clustering. Journal of Computer Research and Development, 2011, 48(S3): 205–212.

15 Luo Q, Miao XJ, Wei Q. Further research on collaborative filtering algorithm for sparse data. Computer Science, 2014, 41(6): 264–268.