Computer Science and Application 计算机科学与应用, 2019, 9(10), 1874-1881
Published Online October 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.910210

Hans汉斯

Research on Hybrid Recommendation Algorithm Based on Machine Learning

Jiaxing Liu, Honglie Zhang, Yanju Liu, Huiyu Zhang, Yanzhong Liu

School of Computer and Control Engineering, Qiqihar University, Qiqihar Heilongjiang
Email: 381415414@qq.com

Received: Sep. 24th, 2019; accepted: Oct. 9th, 2019; published: Oct. 16th, 2019

Abstract

In order to alleviate the dilemma of information explosion, a fused recommendation system is built to improve the accuracy of prediction and aggregate the diversity of recommendations by machine learning algorithm. According to the problems of sparse data sets and single recommendation results, the alternative-least-square optimization model of Beetle antennae search algorithm based on collaborative filtering and the user clustering model based on density-based spatial clustering of noise application are presented. And the XGBoost fusion sorting model is built to get personalized recommendation. The three models are simulated with the sales data of Apple iPhone from Amazon platform. The results show that compared with the single alternating least squares method, the new model has high expansibility, fast convergence and better practical value.

Keywords

Recommendation, Alternating Least Squares, Beetle Antennae, DBSCAN, XGBoost

基于机器学习的融合推荐算法研究

刘佳星，张宏烈，刘艳菊，张惠玉，刘彦忠

齐齐哈尔大学计算机与控制工程学院，黑龙江 齐齐哈尔
Email: 381415414@qq.com

收稿日期：2019年9月24日；录用日期：2019年10月9日；发布日期：2019年10月16日

摘　要

为了缓解信息爆炸的困境，采用机器学习算法建立一个融合的推荐系统以提高预测准确性和聚合推荐多

文章引用: 刘佳星, 张宏烈, 刘艳菊, 张惠玉, 刘彦忠. 基于机器学习的融合推荐算法研究[J]. 计算机科学与应用, 2019, 9(10): 1874-1881. DOI: 10.12677/csa.2019.910210

样性。针对稀疏的数据集及推荐结果单一的问题，提出了以协同过滤为基础的天牛须搜索优化的交替最小二乘法模型、基于密度的噪声应用空间聚类的用户聚类模型、并建立了XGBoost融合排序模型，从而得到个性化推荐。采用来自亚马逊平台的苹果手机销售数据，对三个模型进行仿真测试，结果表明：与单一的交替最小二乘法相比新模型拓展性高，收敛速度快，具有更好的实用价值。

关键词

推荐，交替最小二乘法，天牛须搜索，基于密度的噪声应用空间聚类，XGBoost

1. 引言

推荐系统现已成为诸多领域的重要工具，如信息检索、旅游、近似理论、商业和营销中的消费者选择建模以及预测理论[1]。众多个性化推荐技术中应用最广泛的是协同过滤推荐算法[2]。近年来，国内外诸多学者皆对推荐系统进行研究。文献[3]提出了一种人工免疫系统方法来进行矩阵分解，以优化学习过程中的潜在特征。在大型数据集上有良好表现，但却有分类准确度较低的不足。文献[4]提出了一种基于单线程的流形正则化非负矩阵分解模型，可以避免大规模矩阵操作。文献[5]提出了一种新的贝叶斯网络模型，通过结合基于内容和协作的功能来处理混合推荐问题。为了解决推荐系统的稀疏性和可扩展性两个主要缺点，文献[1]开发一种基于协同过滤方法的新型混合推荐方法。以上方法尽管能缓解推荐系统存在的部分问题，但并未兼顾计算速度与推荐结果个性化。鉴于此，本文基于多个弱学习者可以产生比单个强学习者更好的模型这一机器学习理论，提出一种基于机器学习的融合推荐算法，能够提高计算速度并克服冷启动问题，为用户提供个性化服务。

2. 基本理论

2.1. 交替最小二乘法 ALS

在将用户对商品的评分矩阵分解成用户对商品隐含特征的偏好矩阵和商品所包含的隐含特征矩阵这一过程中，由于存在大量的缺失评分项，传统的奇异值分解 SVD 不易处理稀疏矩阵，而交替最小二乘法可以很好的解决这个问题。对于 $\boldsymbol{R}_{m\times n}$ 的矩阵，ALS 旨在找到 $\boldsymbol{X}_{m\times k}$ 和 $\boldsymbol{Y}_{n\times k}$ 两个低秩矩阵来近似逼近 $\boldsymbol{R}_{m\times n}$，即：

$$\boldsymbol{R}_{m\times n} \approx \boldsymbol{X}_{m\times k}\boldsymbol{Y}_{n\times k}^{\mathrm{T}} \tag{1}$$

其中 $\boldsymbol{R}_{m\times n}$ 代表用户对商品的评分矩阵，$\boldsymbol{X}_{m\times k}$ 表示用户对隐含特征的偏好矩阵，$\boldsymbol{Y}_{n\times k}$ 则表示商品所包含隐含特征的矩阵，T 表示矩阵 $\boldsymbol{Y}_{n\times k}$ 的转置。Funk 函数定义为：

$$L(\boldsymbol{X},\boldsymbol{Y}) = \sum_{i,j\in r}\left(\boldsymbol{R}_{ij} - \boldsymbol{X}_i\boldsymbol{Y}_j^{\mathrm{T}}\right)^2 \tag{2}$$

引入正则化参数 λ 防止过拟合，ALS 的优化目标函数为：

$$L(\boldsymbol{X},\boldsymbol{Y}) = \sum_{i,j\in r}\left(\boldsymbol{R}_{ij} - \boldsymbol{X}_i\boldsymbol{Y}_j^{\mathrm{T}}\right)^2 + \lambda\left(\left\|\boldsymbol{X}_i\right\|_2^2 + \left\|\boldsymbol{Y}_j\right\|_2^2\right) \tag{3}$$

其中 $\boldsymbol{R}_{ij}$ 表示用户 i 对商品 j 的评分矩阵，$\boldsymbol{X}_i$ 表示用户 i 的偏好的隐含特征向量，$\boldsymbol{Y}_j$ 表示商品 j 包含的隐含特征向量，向量 $\boldsymbol{X}_i$ 和向量 $\boldsymbol{Y}_j$ 的内积 $\boldsymbol{X}_i\boldsymbol{Y}_j^{\mathrm{T}}$ 是用户 i 对商品 j 评分的近似；r 则表示矩阵 $\boldsymbol{R}$ 的秩。

由于变量 $\boldsymbol{X}_i$ 和 $\boldsymbol{Y}_j$ 耦合到一起，于是先我们随机初始化 $\boldsymbol{X}^{(0)}$，再固定 $\boldsymbol{X}^{(0)}$ 求解 $\boldsymbol{Y}^{(0)}$，然后根据公式(3)，对损失函数 $L(\boldsymbol{X},\boldsymbol{Y})$ 中的 $\boldsymbol{Y}_j$ 求偏导并令其等于 0：$\dfrac{\partial L(\boldsymbol{X},\boldsymbol{Y})}{\partial \boldsymbol{Y}_j}=-2\boldsymbol{X}_i\boldsymbol{R}_{ij}+2\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{Y}_j+2\lambda\boldsymbol{Y}_j=0$，可得 $\boldsymbol{Y}_j=\dfrac{\boldsymbol{X}_i\boldsymbol{R}_j}{\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}+\lambda\boldsymbol{I}}$，上式中 $\boldsymbol{I}$ 是单位矩阵，$\boldsymbol{R}_j$ 表示给项目 j 评过分的用户的历史评分数据组成的评分向量。

类似地，固定 $\boldsymbol{Y}^{(0)}$ 去求解 $\boldsymbol{X}^{(1)}$，同理可求得 $\boldsymbol{X}_i=\dfrac{\boldsymbol{Y}_j\boldsymbol{R}_i}{\boldsymbol{Y}_j\boldsymbol{Y}_j^{\mathrm{T}}+\lambda\boldsymbol{I}}$，$\boldsymbol{R}_i$ 代表用户 i 对项目的历史评分向量。如此这般，循环往复迭代下去，直到达到收敛状态或最大迭代次数时结束[6]。

2.2. 天牛须搜索算法 BAS

2017 年李帅等人提出一种基于天牛觅食原理的仿生优化算法天牛须搜索(BAS)算法[7]，在优化任务中有出色表现。该算法不同于拟牛顿法 L-BFGS、非线性共轭梯度 NCG 等算法，BAS 不需要具体函数形式与梯度。它基于天牛的行为：使用两个触角随机探索附近区域并调整到具有更高浓度气味的位置[8]。通过天牛的检测行为，可以在以下迭代形式的多维空间中获得全局最优：

$$x^t=x^{t-1}+\delta t\vec{\boldsymbol{b}}sign\left(f(x_r)-f(x_l)\right) \tag{4}$$

其中 x 表示甲虫在第 t 次迭代时的位置，δ 表示每次迭代的步长，$\vec{\boldsymbol{b}}$ 是归一化的随机单位矢量，代表搜索行为；$sign(\,)$ 表示符号函数；$f(x)$ 是适应度函数，而 x_r 和 x_l 是右侧位置和左侧位置的气味浓度。

2.3. 基于密度的噪声应用空间聚类 DBSCAN

DBSCAN 是一种基于密度的聚类方法，其原理是找到被低密度区域分离的稠密区域，要求聚类空间中的一定区域内所包含对象的数目不小于某一给定的阈值[9]。DBSCAN 涉及扫描半径 Eps 和邻域内点最少个数 MinPts 两个参数，并基于中心的密度将点分为核心点、边界点和噪声点三类。其优点是：聚类速度快，且能够有效处理噪声点，能发现任意形状的空间聚类；此外，不需要输入要划分的聚类个数。算法流程如下：

(1) 将所有点标记为核心点、边界点或噪声点；
(2) 删除噪声点；
(3) 为距离在 Eps 之内的所有核心点之间赋予一条边；
(4) 每组连通的核心点形成一个簇；
(5) 将每个边界点指派到一个与之关联的核心点的簇中。

3. 融合推荐算法

3.1. ALS + BAS 模型的建立

正如 2.1 节所介绍的，ALS 自身存在迭代次数与运行时间成正比、最大迭代次数可以自己设置、收敛速度慢等一些问题，本文使用天牛须搜索算法 BAS 对非线性的约束问题进行独立优化，减小运算量并且提高寻优速度。模型建立步骤如下：

(1) 随机初始化式(3)中的 $\boldsymbol{Y}^{(0)}$，此时认为 $\boldsymbol{Y}^{(0)}$ 是已知常量，欲求未知量 $\boldsymbol{X}^{(0)}$，损失函数为：

$$L(\boldsymbol{X},\boldsymbol{Y})=\sum_{(i,j)\in k}\left\{\left(\boldsymbol{R}_{ij}-\boldsymbol{X}_i\boldsymbol{Y}_j^{(0)\mathrm{T}}\right)^2+\lambda\left(\left\|\boldsymbol{X}_i\right\|^2+\left\|\boldsymbol{Y}_j^{(0)\mathrm{T}}\right\|^2\right)\right\} \tag{5}$$

(2) 初始化天牛个体。创建天牛须朝向的随机向量 $\vec{\boldsymbol{b}}$，定义空间维度 k，设置步长因子 δ。

$$\vec{\boldsymbol{b}} = \frac{rands(k,1)}{\left\| rands(k,1) \right\|} \tag{6}$$

在式(6)中，$rands(\ )$ 为随机函数。

(3) 确定适应度函数。本文使用测试数据的平均绝对误差 MAE 作为适应度评价函数。

$$fitness = \text{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left| y_t' - y_t \right| \tag{7}$$

在式(7)中：N 为观测次数；y_t' 为第 t 个样本的模型输出值；y_t 为第 t 个样本的真实值。故，模型迭代停止时，适应度函数值最小的对应位置即为问题所求的最优解。

(4) 设置天牛左右须空间坐标。

$$X_l^t = X^t + d_0 \times \vec{\boldsymbol{b}}/2, X_r^t = X^t + d_0 \times \vec{\boldsymbol{b}}/2 \tag{8}$$

在式(8)中，X_l^t 和 X_r^t 分别表示在第 t 次迭代时，天牛左须和右须的坐标；X^t 为天牛质心坐标，而两须之间的距离用 d_0 表示。

(5) 更新天牛位置。将天牛左右须的坐标带入式(4)中来确定当前天牛的空间位置和气味强度测定适应度函数，以求出天牛个体位置的气味强度：

$$f(X) = f\left(X^{t+1}\right) \tag{9}$$

(6) 找出最新气味强度。找出天牛须中最新的气味强度 f 及位置 $\vec{\boldsymbol{X}}$，即

$$\max\left(\vec{\boldsymbol{X}}\right) \to \left[f, \vec{\boldsymbol{X}} \right] \tag{10}$$

(7) 找寻最强的气味强度，并飞向该位置。

$$f = \max\left(f_{best}\right), \boldsymbol{X} = \max\left(\boldsymbol{X}_{best}\right) \tag{11}$$

(8) 最优解生成。重复执行步骤(4)~(7)，迭代优化来寻找最优解，在适应度函数值达到设定的精度 0.0001 时迭代结束，$\boldsymbol{X}_{best}$ 中的解为训练的最优解。

3.2. XGBoost

XGBoost (extreme gradient boosting)是一种基于梯度提升的集成学习算法，其原理是通过弱分类器的迭代计算来实现准确分类[10]。XGBoost 使用树集合模型，是一组分类和回归树木。梯度增强则是通过构造新的回归树以最大程度地与损失函数的梯度的负相关，进一步增强了增强算法的灵活性[11]。在处理预测问题上具有以下优势：(1) XGBoost 使用所有的并行方式构建树本身计算机在训练期间的 CPU 核心，拥有更大计算速度[12]。(2) XGBoost 是一种通用的监督机器学习方法，在诸多实际应用中实现高精度预测[13]，其高精度可归因于机器学习理论，即多个弱学习者可以产生比单个强学习者更好的模型[12]。

3.3. 整体框架图

整个融合模型主要由：天牛须搜索优化的交替最小二乘法模块、DBSCAN 推荐模块和 XGBoost 三个模块组成如图 1。具体步骤如下：

Step1：将数据集按照 8：2 的比例划分成训练集和测试集。

Step2：建立改进的 ALS 模型，计算目标用户的候选集 A。

Step3：运用 DBSCAN 算法将用户聚类，找到相类似的用户簇，再计算出目标用户的候选集 B。

Step4：将 Step2 和 Step3 各自求得的推荐结果输入 XGBoost 模型中融合排序，从而获得针对用户的个性化 TopK 推荐。

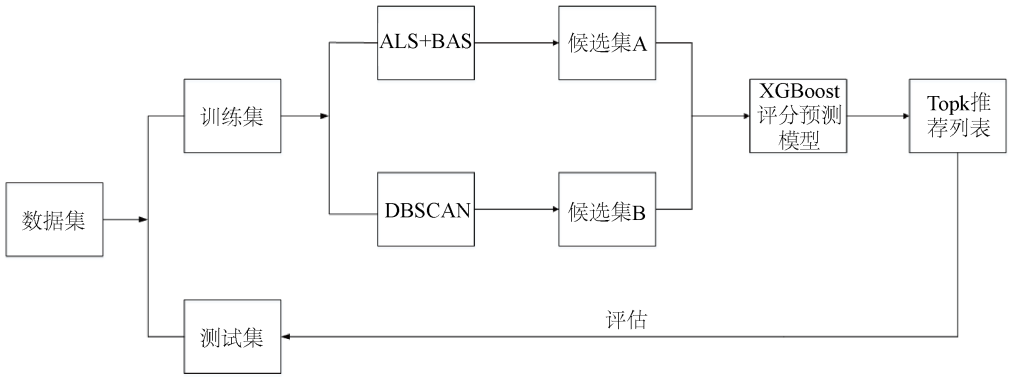

Figure 1. The overall recommendation framework proposed in this paper
图 1. 本文提出的整体推荐框架

4. 实验结果与分析

4.1. 实验环境和数据

本文实验使用的是亚马逊销售苹果手机评分数据。数据集包括 3582 条评分数据，包含产品名称、品牌、价格、评分(1~5)、评价、实际情绪(正面和负面)、更新投票和投票评级，后两项不在考虑范围之内。实验计算机配置：Intel 的 Core I7 处理器，内存 8G，Windows10 操作系统，程序在 R 语言平台实现。

4.2. 评价指标

本文实验分别从运行时间、预测精确度和分类准确率三个方面来衡量所提出的推荐模型。预测精确度使用平均绝对误差 MAE，MAE 数值越小，证明预测的精度越高，其数学定义如式(12)：

$$\mathrm{MAE}=\sum_{t=1}^{N}\left|y_t'-y_t\right|\Big/N \tag{12}$$

分类准确度则用来衡量算法预测的 TopK 结果的好与坏，本文使用的评价指标是召回率 Recall、准确率 Precision 和 F1-Score，三个指标的数学定义公式如下：

$$\mathrm{Recall}=\sum_{i\in User}\left|R(i)\cap T(i)\right|\Big/\sum_{i\in User}\left|T(i)\right| \tag{13}$$

$$\mathrm{Precision}=\sum_{i\in User}\left|R(i)\cap T(i)\right|\Big/\sum_{i\in User}\left|R(i)\right| \tag{14}$$

$$\mathrm{F1}=2\times\mathrm{Recall}\times\mathrm{Precision}/\mathrm{Recall}+\mathrm{Precision} \tag{15}$$

4.3. 结果分析

4.3.1. 时间性能对比

将原始的交替最小二乘法和基于天牛须搜索算法改进的交替最小二乘法分别作用于不同大小的数据集上，其运行时间(以期望的收敛值 10^{-3} 为停止标准)如图 2 所示。

从图中可以观察到：两种算法的运行时间随评分矩阵大小的变化趋势，即矩阵越大，运行时间长；原始的 ALS 在每个试验数据集上的运行时间都要比改进后 ALS 算法的运行时间长；改进后 ALS 算法越是处理更大的矩阵，其加速效果越是显著。

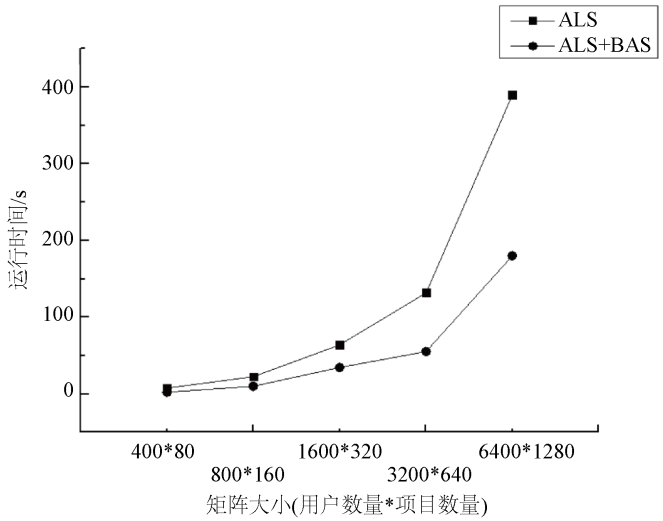

Figure 2. Curve: system result of standard experiment
图 2. 运行时间性能比较

4.3.2. 预测精确度

ALS 的平均绝对误差最大，为 0.7496，DBSCAN 的平均绝对误差次之，为 0.7443。而融合了改进的 ALS 算法与 DBSCAN 算法的 XGBoost 模型的 MAE 值仅为 0.7268。由图 3 可见，融合推荐算法提高了预测评分的精度。

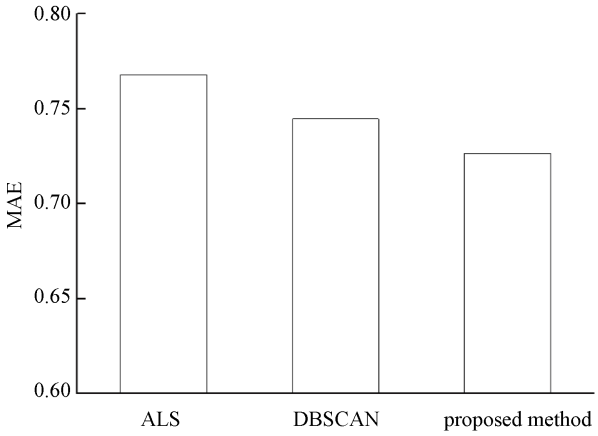

Figure 3. MAE value comparison chart
图 3. MAE 值对比图

4.3.3. 分类准确度

最后，本文将对提出的融合推荐模型的有效性进行评估验证，图 4~6 分别展示不同算法在 3 种衡量标准上的实验结果：

从图 4 种能够看出，融合推荐模型的召回率最大，为 15.20。证明推荐的 TopK 结果和数据集中所有相关项目的比率最大；我们希望推荐结果 precision 越高越好，同时 recall 也越高越好，但事实上这两者在某些情况下存在矛盾，同图 5 不难观察到，融合推荐算法的准确率稍有逊色；F1 综合对分类准确度综合度量，F1 值可视作召回率和准确率的调和平均值。ALS、DBSCAN 和融合推荐算法的 F1 值分别为：0.1886、0.2007 和 0.2135，融合模型优于其他单一算法。

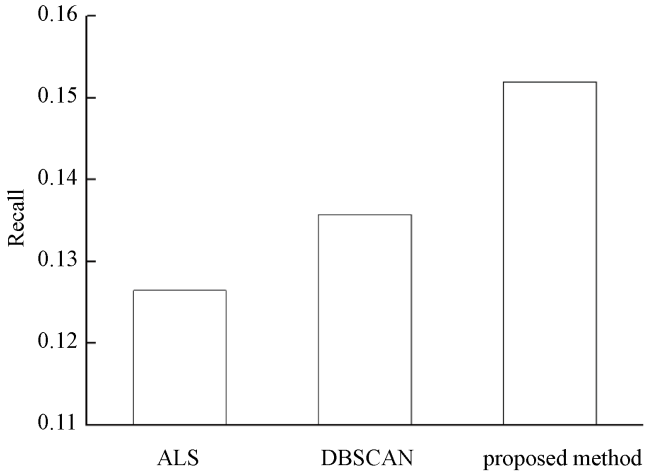

Figure 4. Recall rate comparison chart
图 4. 召回率对比图

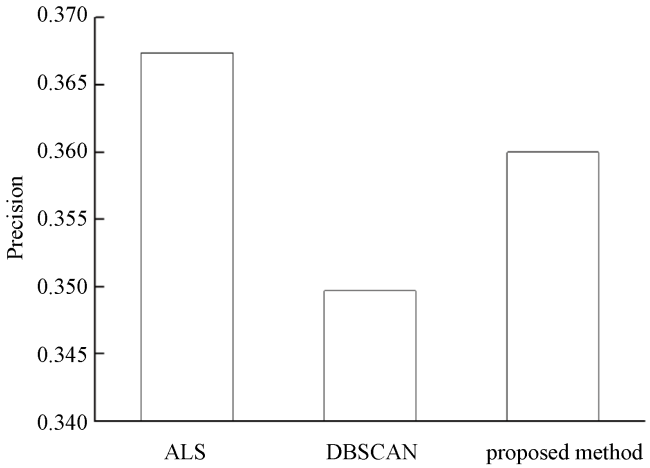

Figure 5. Accuracy comparison chart
图 5. 准确率对比图

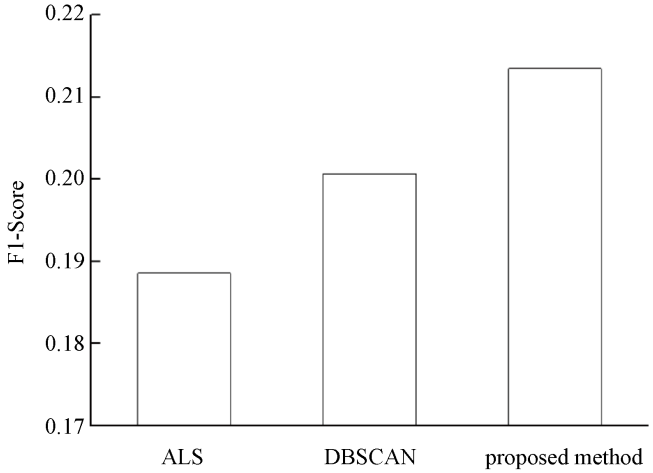

Figure 6. F1-Score comparison chart
图 6. F1-Score 对比图

5. 结语

本文在协同过滤的基础上，使交替最小二乘法与天牛须搜索算法结合，得到收敛速度更快的 ALS + BAS 算法。将 DBSCAN 与 ALS + BAS 算法所生成的候选集组合输入 XGBoost 模型进行融合排序，以获取个性化 TopK 推荐。实验结果表明，1) 数据集越大，ALS + BAS 算法加速效果越显著；2) XGBoost 融合模型在保证预测精确性的前提下，F1 评估指标提高 0.0249，可以为用户提供更加个性化服务。

此外，在我们未来的研究中，计划进一步改进提出的方法并对其使用多样性和新颖性等附加指标数据集进行评估。

基金项目

黑龙江省教育厅基本业务专项理工面上项目(135209234)；齐齐哈尔市基金项目(GYGG-201913)。

参考文献

[1] Mehrbakhsh, N., Othman, I. and Karamollah, B. (2018) A Recommender System Based on Collaborative Filtering Using Ontology and Dimensionality Reduction Techniques. *Expert Systems with Applications*, **92**, 507-520. https://doi.org/10.1016/j.eswa.2017.09.058

[2] 毕曦文, 纪明宇, 吴鹏, 等. 个性化高校新闻分类推荐的应用研究[J]. 计算机应用与软件, 2019, 36(7): 218-223.

[3] Mlungisi, D. and Bhekisipho, T. (2018) Optimising Latent Features Using Artificial Immune System in Collaborative Filtering for Recommender Systems. *Applied Soft Computing*, **71**, 183-198. https://doi.org/10.1016/j.asoc.2018.07.001

[4] Li, H., Li, K.Q., An, J.Y., *et al*. (2018) An Efficient Manifold Regularized Sparse Non-Negative Matrix Factorization Model for Large-Scale Recommender Systems on GPUs. *Information Sciences*, **496**, 464-484. https://doi.org/10.1016/j.ins.2018.07.060

[5] De, C., Fernández-Luna, L.M., *et al.* (2010) Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. *International Journal of Approximate Reasoning*, **51**, 785-799. https://doi.org/10.1016/j.ijar.2010.04.001

[6] 张宏烈, 刘佳星, 刘艳菊, 等. 加速交替最小二乘法推荐系统优化设计[J]. 科学技术与工程, 2019, 19(14): 257-261.

[7] 骆正山, 姚梦月, 骆济豪, 等. 基于 kpca-bas-grnn 的埋地管道外腐蚀速率预测[J]. 表面技术, 2018, 47(11): 173-180.

[8] Zhang, J.F., Ma, G.W., Huang, Y.M., *et al.* (2019) Modelling Uniaxial Compressive Strength of Lightweight Self-Compacting Concrete Using Random Forest Regression. *Construction and Building Materials*, **210**, 713-719. https://doi.org/10.1016/j.conbuildmat.2019.03.189

[9] 郭智鹏. 个性化聚类下基于 DBSCAN 的密度聚类算法研究[D]: [硕士学位论文]. 武汉: 华中科技大学, 2018.

[10] 陈明华, 刘群英, 张家枢, 等. 基于 XGBoost 的电力系统暂态稳定预测方法研究[J/OL]. 电网技术: 1-10 [2019-04-02]. https://doi.org/10.13335/j.1000-3673.pst.2018.1649

[11] Debaditya, C. and Hazem, E. (2019) Early Detection of Faults in HVAC Systems Using an XGBoost Model with a Dynamic Threshold. *Energy and Buildings*, **185**, 326-344. https://doi.org/10.1016/j.enbuild.2018.12.032

[12] João, N. and Rui, F.N. (2019) Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to Trade in the Financial Markets. *Expert Systems with Applications*, **125**, 181-194. https://doi.org/10.1016/j.eswa.2019.01.083

[13] Andreja, S., Nenad, S., Gordana, V., *et al.* (2019) Explainable Extreme Gradient Boosting Tree-Based Prediction of Toluene, Ethylbenzene and Xylene Wet Deposition. *Science of the Total Environment*, **653**, 140-147. https://doi.org/10.1016/j.scitotenv.2018.10.368