第 55 卷　第 4 期
2019 年 7 月

南京大学学报(自然科学版)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCE)

Vol. 55, No. 4
July, 2019

DOI:10.13232/j.cnki.jnju.2019.04.014

基于动态加权Bagging矩阵分解的推荐系统模型

何轶凡,邹海涛,于化龙*

(江苏科技大学计算机学院,镇江,212003)

摘　要:为了提升推荐模型的预测精度,传统方法通常是利用更多的附加信息参与模型的构建.然而,此类方法在提高算法精度的同时也大大增加了算法的时间开销,同时对数据集也存在一定的要求.为了解决上述问题,提出一种基于Bagging集成的矩阵分解模型.该模型根据用户、产品评分数为基学习器动态分配权重,并通过加权求和得到预测评分.在三个不同规模的真实数据集上的实验结果显示:该动态加权Bagging矩阵分解模型拥有与传统矩阵分解模型一样的时间消耗,并且在各个衡量指标上都优于传统的矩阵分解模型.

关键词:推荐系统,矩阵分解,Bagging,动态加权

中图分类号:TP391.3　　**文献标识码:**A

Recommender system model based on dynamic-weighted bagging matrix factorization

He Yifan, Zou Haitao, Yu Hualong*

(School of Computer, Jiangsu University of Science and Technology, Zhenjiang, 212003, China)

Abstract: To promote the prediction accuracy of the recommender system, the traditional methods generally use the additional information to construct model, which always increase the time consumption greatly, as well they always needs more detailed data. To solve the problem above, we propose a Bagging-based matrix factorization model which assigns dynamic weights to every base learner according to the number of users' and items' ratings, then acquires the prediction ratings by weight summation. The experimental results on three real datasets show that our dynamic-weighted bagging matrix factorization model has the same efficiency as the traditional matrix factorization model, and it is superior to the traditional matrix factorization on all measures.

Key words: recommender system, matrix factorization, Bagging, dynamic weighting

推荐算法的主要目的就是从用户和产品的历史数据中分析用户、产品的某些特征,并预测出某用户对产品的喜好程度,再向用户推荐他所可能喜欢的产品.在Shardanand and Maes[1]提出的基于用户相似度的推荐算法中,通过用户评分数据来计算用户间的相似度,选取相似

基金项目:国家自然科学基金(61305058,61572242),江苏省自然科学基金(BK20130471),中国博士后特别资助计划(2015T80481),中国博士后科学基金(2013M540404),江苏省博士后基金(1401037B),江苏省研究生科研与实践创新计划(KYCX19_1698)

收稿日期:2019－05－12

*通讯联系人,E-mail:yuhualong@just.edu.cn

度较大的作为邻居用户,再通过邻居用户对某产品评分的加权求和来得到预测评分.该算法被运用于国外的音乐网站Ringo为用户推荐他可能喜欢的音乐,取得了不错的效果.由于产品数量的爆发式增加,Deshpande and Karypis[2]提出了一种基于产品相似度的推荐算法.该算法首先计算出产品之间的相似度,然后通过用户已购买的产品列表计算出用户对各个产品的喜好程度,在产品数较多的情况下具有优于基于用户相似度推荐算法的算法表现.上述算法都是基于相似度的推荐算法,此类算法在较为稀疏的数据集上无法精准地构造用户或产品间的相似度关系,导致算法预测精度较差.

在解决上述问题的过程中,矩阵分解模型[3]进入了研究者的视线.研究者们发现在隐含特征空间中某用户对产品的偏好(评分)可以根据用户特定的系数对产品特征向量进行线性组合得到.Salakhutdinov and Mnih[4]对矩阵分解模型给出了概率层次的解释,并提出一种受限概率矩阵分解模型(Constraint-Probabilistic Matrix Factorization),在构造用户特征矩阵时加入代表用户之间关系的隐含特征矩阵,大大提高了对稀疏、不平衡数据的预测精度.Koren et al[5]提出矩阵分解算法的几个变种,向原始矩阵分解模型中加入偏好、置信度等概念,提升算法的预测精度.此外,还有不少学者利用其他外部信息与矩阵分解模型构成混合模型,针对特定的数据集获得精准的推荐.如Zou et al[6]利用社交网络中社区成员之间的信任度作为用户间的相似度协同产生预测评分,Cheng et al[7]通过用户评分与评论共同协作产生推荐等.

上述模型都以预测评分和真实评分的差的平方为目标函数,通过最优化方法迭代求解用户、产品的隐含特征.Park et al[8]以目标函数的重构为出发点,提出以成对损失替代原始目标函数,最小化用户之间在共同购买产品上预测评分大小关系与真实评分大小关系的差,能获得更好的预测精度,但也带来了更高的时间复杂度.Xu et al[9]构造了一种不断缩小预测方差的高阶评分距离模型.在构造目标函数时,高阶评分距离模型不仅考虑用户对产品的预测评分与实际评分间的差异(称为一阶评分距离),还增加了同一用户对不同产品的预测评分与实际评分差异的差异(称为二阶评分距离).二阶评分距离的引入能不断缩小预测结果的方差,提高预测的精度,但由于模型过于复杂,算法的时间复杂度随数据集规模的增长呈指数增长.

集成学习[10-11]的方法也被运用于提高推荐算法的精度.Sun et al[11]通过线性回归对多个算法分配权值,最终把各个算法的预测评分加权求和作为预测评分.方育柯等[12]集成基于用户相似度的推荐算法,使用不同的相似度度量产生不同的推荐模型,并加权求和得到最终的预测评分,提高了模型的预测精度.崔岩等[13]结合基于用户和基于产品的预测评分差与真实评分构造出新的数据集,再通过XGboost模型进行训练与预测.上述的集成方法,都以基于内容的推荐算法为基础,该算法本身就存在时间复杂度高、预测精度较低的缺点.此类算法用来计算相似度的时间与用户、产品个数的平方成正比,而且在面临稀疏数据时,甚至会出现相似度均为0的用户或产品,导致此类算法在预测精度方面也有不足.

为解决上述问题,本文以矩阵分解模型为基础,提出一种动态加权Bagging[14]的矩阵分解模型.首先,通过自助采样产生不同的训练集,再利用矩阵分解模型作为基学习器为每个训练集训练出用户、产品特征向量,最后构造基学习器间的动态权值,结合所有基学习器产生预测评分.对基学习器的并行计算使该模型在时间消耗相同时,算法的预测精度得到了提高.

1　相关工作

1.1　矩阵分解　矩阵分解(Matrix Factorization,MF),是在推荐领域广泛被应用的一种模型.该模型用一定维度的向量来表示用户和产品的特征,而某用户对某产品的评分就可由它们的特征向量的内积来预测.

假设有 m 个用户,n 个产品,对应存在一个 $m \times n$ 的评分矩阵 $\boldsymbol{R}$,其中的元素 r_{ij} 表示用户 i 对产品 j 的评分. 通过矩阵分解的方法,把 $\boldsymbol{R}$ 分解为两个较小的规模为 $m \times d$ 的矩阵 $\boldsymbol{U}$ 和规模为 $n \times d$ 的矩阵 $\boldsymbol{V}$. $\boldsymbol{U},\boldsymbol{V}$ 的每一行都代表着一个用户或一个产品的特征向量,$\boldsymbol{u}_i$ 为 $\boldsymbol{U}$ 的第 i 行,代表用户 i 的特征向量,$\boldsymbol{v}_j$ 为 $\boldsymbol{V}$ 的第 j 行,代表产品 j 的特征向量,d 为特征向量的维度($d \ll m,n$),则第 i 个用户对第 j 个产品的预测评分表示为:$\hat{r}_{ij} = \boldsymbol{u}_i^T \boldsymbol{v}_j$. 给定一组训练数据 D,其中每条数据形如 (i,j,r_{ij}),通过训练数据构建优化问题如式(1):

$$\arg\min_{\boldsymbol{U},\boldsymbol{V}} \sum\nolimits_{(i,j,r_{ij})\in D} (r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j)^2 \tag{1}$$

使得式(1)中目标函数最小的 $\boldsymbol{U},\boldsymbol{V}$ 就是期望的用户、产品特征矩阵. 为了防止过拟合,特征向量二范数的平方被引入到目标函数中作为正则项,式(1)变为:

$$\arg\min_{\boldsymbol{U},\boldsymbol{V}} \sum\nolimits_{(i,j,r_{ij})\in D} (r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j)^2 + \lambda(\left|\left|\boldsymbol{u}_i\right|\right|^2 + \|\boldsymbol{v}_j\|^2) \tag{2}$$

其中,系数 λ 是一个通过实验获得的常量.

确定了目标函数之后,该模型采用随机梯度下降法[15]迭代更新 $\boldsymbol{U},\boldsymbol{V}$ 来最小化目标函数,如式(3)和式(4):

$$\boldsymbol{u}_i \leftarrow \boldsymbol{u}_i - \alpha \cdot \left(\left(r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j\right) \boldsymbol{v}_j + \lambda \boldsymbol{u}_i \right) \tag{3}$$

$$\boldsymbol{v}_j \leftarrow \boldsymbol{v}_j - \alpha \cdot \left(\left(r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j\right) \boldsymbol{u}_i + \lambda \boldsymbol{v}_j \right) \tag{4}$$

其中,α 为学习率,当满足一定迭代次数或目标函数变化小于一定阈值时迭代停止,最后通过训练出的 $\boldsymbol{U},\boldsymbol{V}$ 特征矩阵来预测评分.

矩阵分解模型比基于相似度的推荐算法在效率和预测精度上已有很大提升,但由于推荐领域数据本身的稀疏性和模型构建中随机初始值的设置导致模型的不稳定,使得产生的预测评分存在较大的方差,影响了推荐的准确性.

1.2 Bagging Bagging(Bootstrap aggregating)[14]是并行式集成学习方法最著名的代表,是一种基于自助采样法(bootstrap sampling)[16]的集成学习算法,能获得相较于基学习器更低的预测方差,可用于提高推荐算法的精度. 该方法的总体思想如下:给定某包含 k 个样本的数据集 D,随机取出一个样本放入采样集,再把该样本放回原数据集,使得下次采样该样本仍有可能被选中,这样,在 k 次随机采样后,就能得到含有 k 个样本的采样集 D'. 值得注意的是,初始训练集中存在样本多次出现在采样集中. 由于初始训练集中的样本每次被抽取的概率为 $1/k$,则样本在 k 次采样仍然不被采到的概率为 $(1-1/k)^k$,取极限为:

$$\lim_{k\to\infty} \left(1 - \frac{1}{k}\right)^k \to \frac{1}{e} \approx 0.368 \tag{5}$$

由式(5)可知,通过自主采样所得的采样集能包含初始训练集大约 63.2% 的样本. 利用上述方法,可采样出 G 个包含 k 个样本的采样集 $\{D'_1,\cdots,D'_G\}$,基于每个采样集训练出一个基学习器,再集成基学习器产生模型预测,如对于分类问题常采用简单投票法,对于回归问题则采用简单平均法.

2 动态加权 Bagging 矩阵分解构建

2.1 算法思想 对于一个回归任务,设 $(\boldsymbol{x},y)$ 为数据集 D 中的一条数据,其中 $\boldsymbol{x}$ 为特征向量,y 为真实值. 通过数据集 D 训练出多个回归模型,再把特征向量放入回归模型产生对应的预测值 $\phi(\boldsymbol{x},D)$. 集成的预测值是多个模型在数据集 D 上预测的平均值,即:

$$\phi_A(\boldsymbol{x}) = E_D \phi(\boldsymbol{x},D) \tag{6}$$

取 $\boldsymbol{x}$ 为固定输入值,y 为输出值,则:

$$E_D(y - \phi(\boldsymbol{x},D))^2 = y^2 - 2yE_D\phi(\boldsymbol{x},D) + E_D\phi^2(\boldsymbol{x},D) \tag{7}$$

应用式(6)和不等式 $EZ^2 \geqslant (EZ)^2$ 于式(7)的末项可得:

$$E_D(y - \phi(\boldsymbol{x},D))^2 \geqslant (y - \phi_A(\boldsymbol{x}))^2 \tag{8}$$

由式(8)可知,集成产生的预测值 $\phi_A(\boldsymbol{x})$ 的均方误差小于 $\phi(\boldsymbol{x},D)$ 均方误差的平均值,而且 $\phi(\boldsymbol{x},D)$ 越不稳定,集成对模型的提升越大.

Bagging 是基于自助采样的集成算法,自助采样采用的是概率论里的 bootstrap 思想,原始

稀疏样本不能反映数据集的真实分布,通过k次随机采样拟合数据的真实分布.针对矩阵分解模型由于数据稀疏性和随机初始值设置导致的预测评分不稳定,存在较大方差的问题,本文利用G个矩阵分解模型作为基学习器构造Bagging集成框架,并为每个基学习器分配动态权重,最后通过对基学习器的预测结果加权求和产生最终预测结果.算法流程如图1所示.

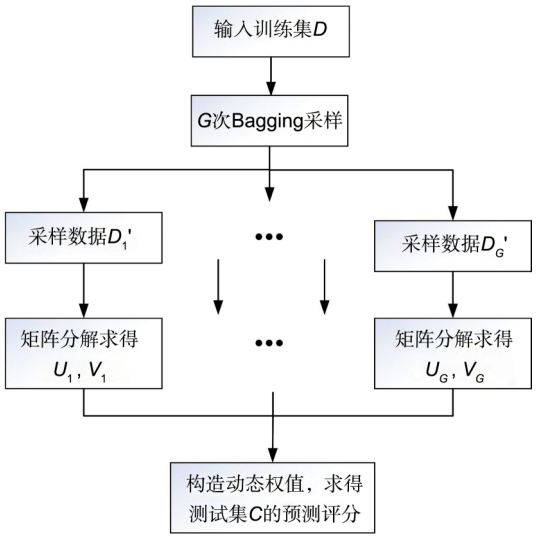


图1　算法流程图

Fig. 1　Flow chart of the algorithm

2.2　算法具体描述　首先,对有k个样本的训练数据集D进行自助采样,采取G个包含k个样本的采样集$\{D'_1,\cdots,D'_G\}$.不同的是,为了保证每个用户和产品在每个采样集都拥有训练样本,每次采样首先在每个用户和产品所参与的评分数据中随机选取一个作为样本,共计$(m+n)$个样本$((m+n)\ll k)$,然后再对整体训练集进行自助采样得到包含k个样本的采样集.

接着,针对每个采样集D'_g,构造矩阵分解模型,根据式(3)和式(4)训练出用户、产品特征矩阵$\boldsymbol{U}_g$和$\boldsymbol{V}_g$.

最后,考虑到对于每个基学习器每个用户产品对的预测评分的置信度不同,给每个基学习器分配动态权重.通常,训练样本越多,则越能精确地拟合当前用户和产品的评分行为.因此,本文采用各个采样集中用户和产品的评分个数的归一化统计量作为各个学习器的权重,通过式(9)得到集成预测评分:

$$p_{ij}=\sum_{g=1}^{G}\frac{\Omega_{g_{i,j}}}{\sum_{g=1}^{G}\Omega_{g_{i,j}}}\boldsymbol{u}_{g_i}^{T}\boldsymbol{v}_{g_j} \tag{9}$$

其中$\Omega_{g_{i,j}}$为第g个采样集中用户i的评分样本个数和产品j的评分样本个数的和.倘若在每个采样集中,$\Omega_{g_{i,j}}$都相等,则加权方式与简单平均法一致.在实验中,本文将上述算法称为动态加权Bagging矩阵分解.算法的具体步骤参考算法1.

算法1　动态加权Bagging矩阵分解.

输入:训练集D,测试集C,最大迭代次数$MaxIter$;
输出:预测评分矩阵$\boldsymbol{P}$.
①自助采样G个采样集$\{D'_1,\cdots,D'_G\}$;
②统计用户、产品的评分个数Ω;
③for $g=1$ to G
④　随机初始化特征矩阵$\boldsymbol{U}_g,\boldsymbol{V}_g$;
⑤　for $Iter=1$ to $MaxIter$
⑥　while $(i,j,r_{ij})\in D_g$
⑦　根据公式(3)更新$\boldsymbol{u}_{g_i}$;
⑧　根据公式(4)更新$\boldsymbol{v}_{g_j}$;
⑨　end while
⑩　end for
⑪end for
⑫for $(i,j)\in C$
⑬　根据公式(9)得到预测评分p_{ij};
⑭end for
⑮return $\boldsymbol{P}$.

3　实验结果与讨论

3.1　实验数据　使用从MovieLens(https://grouplens.org/datasets/movielens/)抽取的三个不同的数据集进行了测试,数据集的基本描述如表1所示.三个数据集都具有稀疏性的共同特点,其中MovieLens1m稀疏度为4.19%,MovieLens10m和MovieLens20m的稀疏度分别为0.21%和0.11%.不同的是,MovieLens1m的评分为1~5的整数,而MovieLens10m和MovieLens20m的评分区间则是在0.5到5之间,每次增量为0.5.

表 1 实验数据集描述

Table 1 Description of datasets

数据集	用户数	产品数	打分数
MovieLens1m	6040	3952	1000209
MovieLens10m	71567	65133	10000054
MovieLens20m	138493	131262	20000263

3.2 度量标准 本文选取平均绝对误差(*MAE*)、均方根误差(*RMSE*)、归一化折损累计(*NDCG@K*)作为算法评估的三个指标,*MAE*和*RMSE*的定义如下:

$$MAE=\frac{\sum_{r_{ij}\in C}|\hat{r}_{ij}-r_{ij}|}{|C|}$$

$$RMSE=\sqrt{\frac{\sum_{r_{ij}\in C}(\hat{r}_{ij}-r_{ij})^2}{|C|}}$$

从*MAE*和*RMSE*的定义可发现,*MAE*能很好地反映预测值误差,而*RMSE*对误差较大的异常值更敏感.*NDCG@K*常用于作为列表排序的评价指标.本文把测试集中每个用户评价的产品按预测评分降序排列,并使用*NDCG@K*对上述排序进行衡量,定义如下:

$$NDCG@K=\frac{DCG@K}{iDCG@K}$$

$$DCG@K=\sum_{i=1}^{K}\frac{2^{r(i)}-1}{\log_2(i+1)}$$

其中,K为序列中元素的个数,$r(i)$为第i个元素的值,*DCG@K*是预测出的序列的前K个元素的*DCG*值,*iDCG@K*代表理想最优序列的*DCG*值.本文采用*NDCG*@10作为衡量指标,*NDCG*@10越大则说明列表排序质量越好,即产生的预测评分越好.

3.3 参数设置 对于矩阵分解模型,实验中使用 Xu et al[9]的参数$\lambda_u=0.1$,$\lambda_v=0.1$,学习率$\alpha=0.01$.实验证明,Bagging集成表现随G的增大而趋于稳定.基于时间效率的考量,实验中将G设置为4.这样,当实验在四核CPU上并行时,只有和单个矩阵分解一样的时间消耗.此外,实验总迭代次数设置为80次,这个迭代次数已经可以保证算法收敛.前期实验表明,算法的预测精度会随着用户、产品特征矩阵维度d的增长而提高,但在维度d超过45时这种提高几乎消失,所以本文所有算法的特征矩阵维度d都设置为45.

3.4 实验结果及讨论 实验中使用动态加权Bagging矩阵分解(Dynamic Weighted Bagging Matrix Factorization)的英文首字母缩写DWBMF来代指该算法.实验展示了DWBMF与MF在三个数据集上完成最大迭代次数后的性能表现,表2展示了在MovieLens1m数据集上的性能,表3和表4则分别为在MovieLens10m和MovieLens20m数据集上的性能,黑体字表示性能最优的数据.实验结果表明,本文提出的DWBMF模型在所有情况下都要优于MF模型.

表 2 两种算法在MovieLens1m数据集上的性能

Table 2 Performance of two algorithms on MovieLens1m

算法	*MAE*	*RMSE*	*NDCG@10*
MF	0.6918	0.8678	0.8225
DWBMF	**0.6858**	**0.8603**	**0.8296**

表 3 两种算法在MovieLens10m数据集上的性能

Table 3 Performance of two algorithms on MovieLens10m

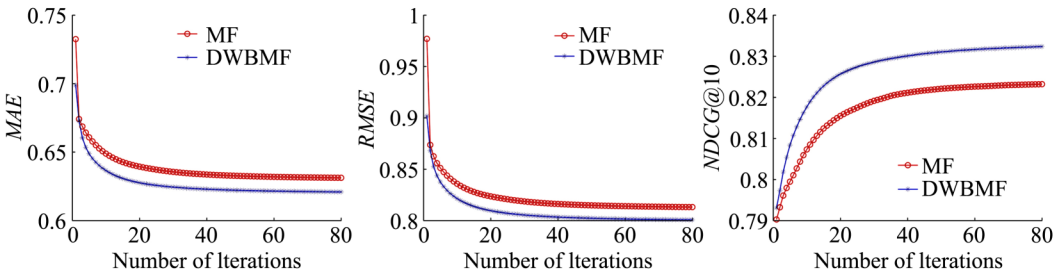
算法	*MAE*	*RMSE*	*NDCG@10*
MF	0.6384	0.8187	0.8250
DWBMF	**0.6295**	**0.8064**	**0.8342**

表 4 两种算法在MovieLens20m数据集上的性能

Table 4 Performance of two algorithms on MovieLens20m

算法	*MAE*	*RMSE*	*NDCG@10*
MF	0.6321	0.8131	0.8233
DWBMF	**0.6209**	**0.8007**	**0.8323**

为了更直观地比较DWBMF与MF算法的性能,图2绘制了各个衡量指标随迭代次数的变化曲线.尽管三个数据集在数据规模、评分区间、数据稀疏度上都存在差异,但DWBMF



(a)*MAE*　　(b)*RMSE*　　(c)*NDCG*@10

图2　两种算法在MovieLens20m上表现对比

Fig. 2　Performance of two algorithms on MovieLens20m

和MF在三个数据集上的实验结果均有类似的表现趋势,因此,此处只列举各个指标在数据集MovieLens20m上的变化曲线.实验结果显示,随着迭代次数的增加,在*MAE*,*RMSE*和*NDCG*@10三个衡量指标中,本文提出的DWBMF模型总能持续明显优于MF.

4　结　论

针对推荐领域数据稀疏性和矩阵分解随机参数设置所导致的预测评分不稳定,预测精度不足的缺点,本文提出了一种动态加权Bagging矩阵分解算法,利用用户、产品评分个数的统计量作为Bagging集成的权重,加权求和产生最终预测评分.对不同规模的三个MovieLens数据集的实验结果分析表明,该模型在各个指标上都明显优于传统矩阵分解模型.

参考文献

[1] Shardanand U, Maes P. Social information filtering: algorithms for automating "Word of Mouth"//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Denver,CO,USA:ACM,1995:210－217.

[2] Deshpande M, Karypis G. Item-Based Top-*N* recommendation algorithms. ACM Transactions on Information Systems,2004,22(1):143－177.

[3] Rennie J D M, Srebro N. Fast maximum margin matrix factorization for collaborative prediction// Proceedings of The 22th International conference on Machine learning. New York, NY, USA: ACM,2005:713－719.

[4] Salakhutdinov R, Mnih A. Probabilistic matrix factorization//Proceedings of the 20th International Conference on Neural Information Processing Systems. Vancouver,Canada:ACM,2008:1257－1264.

[5] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009,42(8):30－37.

[6] Zou H T, Gong Z G, Zhang N, et al. Adaptive ensemble with trust networks and collaborative recommendations. Knowledge and Information Systems,2015,44(3):663－688.

[7] Cheng Z Y, Ding Y, Zhu L, et al. Aspect-aware latent factor model: Rating prediction with ratings and reviews. 2018,arXiv:1802.07938.

[8] Park D, Neeman J, Zhang J, et al. Preference completion: large-scale collaborative ranking from pairwise comparisons//Proceedings of the 32th International Conference on Machine Learning. Lille,France:Springer,2015:1907－1916.

[9] Xu J W, Yao Y, Tong H H, et al. HoORaYs:high-order optimization of rating distance for recommender systems//Proceedings of the 23th International Conference on Knowledge Discovery and Data Mining. Halifax, Canada: ACM, 2017: 525－534.

[10] Wang S, Minku L L, Yao X. Resampling-based ensemble methods for online class imbalance

learning. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1356－1368.

[11] Sun Z B, Song Q B, Zhu X Y, et al. A novel ensemble method for classifying imbalanced data. Pattern Recognition, 2015, 48(5): 1623－1637.

[12] 方育柯,傅彦,周俊临. 基于集成学习的个性化推荐算法. 计算机工程与应用, 2011, 47(10): 1－4. (Fang Y K, Fu Y, Zhou J L. Boosting algorithm for personalized recommendation. Computer Engineering and Applications, 2011, 47(10): 1－4.)

[13] 崔岩,祁伟,庞海龙等. 融合协同过滤和XGBoost的推荐算法. 计算机应用研究, 2018, 37(1).(Cui Y, Qi W, Pang H L, et al. Extreme gradient boosting recommendation algorithm with collaborative filtering. Application Research of Computers, 2018, DIO: 10.19734/j. issn. 1001－3695.2018.06.0463).

[14] Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123－140.

[15] Oh J, Han W S, Yu H, et al. Fast and robust parallel SGD matrix factorization//Proceedings of the 21th International Conference on Knowledge Discovery and Data Mining. Sydney, Australia: ACM, 2015: 865－874.

[16] Johnson R W. An introduction to the bootstrap. Teaching Statistics, 2001, 23(2): 49－54.

(责任编辑 杨可盛)