文章编号：2095-2163(2020)01-0056-06 中图分类号：TP301.6 文献标志码：A

基于 K-means 的矩阵分解推荐算法

张荣梅，陈 彬，张 琦
（河北经贸大学 信息技术学院，石家庄 050061）

摘 要：传统矩阵分解算法和基于用户画像的算法存在数据稀疏性和冷启动等问题，且多数情况下只注重于用户项目交互数据，而对用户本身的属性信息缺少借鉴，从而导致推荐准确性不高。将 K-means 与矩阵分解相结合，提出了一种基于 K-means 的矩阵分解推荐算法（Matrix Decomposition Based on K-means，KMMD）。该算法融合用户属性和用户项目交互评级数据作为输入，先将用户进行 K-means 聚类，得到近邻用户集，再将近邻用户-项目评级矩阵进行分解和重构，得到预测评级并排序推荐。将算法在 MovieLens 公开数据集上进行仿真实验，结果表明 KMMD 推荐算法在召回率和精确度上有了进一步的提高，并且对用户冷启动问题做出了很大的改善。

关键词：智能推荐；K-means；矩阵分解

K-means-based matrix decomposition recommendation algorithms

ZHANG Rongmei，CHEN Bin，ZHANG Qi
（School of Information Technology，Hebei University of Economics and Business，Shijiazhuang 050061，China）

【Abstract】 Traditional matrix decomposition algorithm and user portrait-based algorithm have some problems，such as data sparsity and cold start，and most of them only focus on user project interaction data，but lack of reference for user´s own attribute information，which leads to the inaccuracy of recommendation. Combining K-means with matrix decomposition，a matrix decomposition recommendation algorithm based on K-means is proposed. The algorithm integrates user attributes and user item interactive rating data as input. First，users are clustered by K-means to get the nearest user set，then the nearest user-item rating matrix is decomposed and reconstructed to get the predictive rating and ranking recommendation. The algorithm is simulated on the open data set of MovieLens. The results show that KMMD recommendation algorithm has further improved the recall rate and accuracy，and has greatly improved the cold start problem of users.

【Key words】 intelligent recommendation；K-means；matrix decomposition

0 引 言

当下，随着大数据时代的来临，信息资源过度膨胀，形成了“信息爆炸”的现象。为了缓解这种情况带来的信息过载、数据冗余、选择困难等问题，越来越多的专家学者已然开始关注起推荐系统领域的研究。推荐系统是通过分析用户的历史数据，以及项目等其它辅助信息，推测出用户潜在的偏好需求，进而为用户提供个性化的项目推荐。常见的传统推荐技术是基于内容的算法、基于协同过滤的算法及混合算法[1]。其中，协同过滤算法应用较为广泛，Goldberg 等人[2]于 1992 年提出了协同过滤的概念，最初应用在 Tapestry System 上用于过滤电子邮件。这是通过引入其它用户的兴趣来对当前用户进行推荐，只是涉及用户的历史交易记录，而不依赖用户和项目的属性特征。但协同过滤算法存在数据稀疏[3]和冷启动问题[4]。于洪等人[5]为了更好地解决物品冷启动问题，提出了一种附加用户时间权重的算法，对用户评论时间与项目发布时间加以计算研究，但由于很多标准数据集中缺少时间戳的属性，其作用范围有限。针对于此，为了改进协同过滤算法，本文将 K-means 聚类算法与矩阵分解技术相结合，提出一种基于 K-means 的矩阵分解推荐算法（Matrix Decomposition Based on K-means，KMMD），引入了用户属性信息，在提高推荐精度的同时，有效改善用户冷启动问题。

1 基于 K-means 的矩阵分解推荐算法（KMMD）

1.1 Funk-SVD 矩阵分解算法

基于内容的推荐和基于用户画像的推荐[6]都是聚焦于待推荐用户自身的属性信息或交易记录，并没有考虑过其它用户的数据是否会对当前用户产

基金项目：河北省重点研发计划（19210105D）。

作者简介：张荣梅（1966-），女，博士，教授，主要研究方向：人工智能、机器学习和电子商务；陈 彬（1994-），男，硕士研究生，主要研究方向：推荐系统、深度学习；张 琦（1995-），女，硕士研究生，主要研究方向：深度学习、图像识别。

通讯作者：陈 彬 Email：787834305@qq.com

收稿日期：2019-10-11

生推荐影响,在召回率和精确度上不能进一步提高。而矩阵分解[7]作为协同过滤的一种重要方法,不仅将待推荐用户自身的属性考虑在内,还吸收借鉴了其它用户的数据信息,来实施推荐。且矩阵分解算法将庞大的用户-项目评级矩阵分解为多个矩阵存储,大大减缓了磁盘存储压力。

矩阵分解采用 Funk-SVD, 是通过将用户 - 项目评级矩阵 $\boldsymbol{R}_{m\times n}$ 进行分解,得到可以表示用户和项目特征的 2 个低维的抽象隐因子矩阵:$\boldsymbol{U}_{m\times k}$ 表示 m 个用户的 k 维隐因子矩阵,$\boldsymbol{V}_{n\times k}{}^{\mathrm{T}}$ 表示 n 个项目的 k 维隐因子矩阵,使得 $\boldsymbol{U}_{m\times k}\cdot\boldsymbol{V}_{n\times k}{}^{\mathrm{T}}\approx\boldsymbol{R}_{m\times n}$。使用重构后的评级矩阵来预测用户对未知项目的评分。对于矩阵中的缺失项,在参数更新时选择忽略不计,不对其进行操作,而只针对有效数据进行参数训练。

假设原始评级矩阵为 $\boldsymbol{R}_{m\times n}$,重构后的评级矩阵为 $\hat{\boldsymbol{R}}_{m\times n}$,而 $\hat{r}_{i,j}=\boldsymbol{u}_i\boldsymbol{v}_j{}^{\mathrm{T}}=\sum_{k=1}^{k}u_{i,k}v_{k,j}$,矩阵分解的标准是减小预测评级 $\hat{r}_{i,j}$ 与原始评级 $r_{i,j}$ 的误差。此时将用到如下数学公式:

$$\min loss=\sum_{i=1}^{m}\sum_{j=1}^{n}e_{i,j}^2+L=(r_{i,j}-\hat{r}_{i,j})^2+\lambda\sum_{k=1}^{k}(\|\boldsymbol{U}\|^2+\|\boldsymbol{V}\|^2). \tag{1}$$

其中,损失函数由误差平方和正则化项组成,引入正则化项可以防止训练结果的过拟合。

而训练过程是使用梯度下降降低损失函数,其数学公式可表示为:

$$u_{i,k}^{'}=u_{i,k}+\alpha\frac{\partial}{\partial u_{i,k}}e_{i,j}^2, \tag{2}$$

$$v_{j,k}^{'}=v_{j,k}+\alpha\frac{\partial}{\partial v_{j,k}}e_{i,j}^2. \tag{3}$$

其中, α 是学习率参数,表示每次更新的快慢。

传统的矩阵分解算法多采用传统的 SVD 矩阵分解方式[8]。传统 SVD 分解后会得到 3 个矩阵: $\boldsymbol{U}$、$\sum$ 和 $\boldsymbol{V}$。其中, $\sum$ 是一个对角矩阵,表示 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 矩阵在每个维度上的重构重要度可通过该对角矩阵进行降维,但存在对矩阵求逆等操作,致使计算复杂度高。而本文实验采用的是 Funk-SVD 分解方式,借鉴线性回归的思想,通过最小化均方误差来寻求最优的用户和项目的隐含向量表示,其维度可直接调整。Funk-SVD 用 2 个矩阵就可以实现 SVD 三个矩阵的重构效果,在一定程度上提高了运算效率。

1.2　K-means 聚类

由于传统矩阵分解算法只使用用户对项目的评级信息,其推荐精度上限难以进一步提升,所以引入用户属性信息,经 K-means 聚类分析再进行矩阵分解运算,提高算法推荐能力。K-means 是一种将数据按策略划分为某几种类别的聚类算法[9],其中 K 表示聚类的类别种数,即质心的数量,means 表示均值。K-means 聚类之前的预处理过程可分述如下。

(1) 提取用户基本属性信息, $User=(Age, Gender, Occupation)$,构建用户属性矩阵 $\boldsymbol{U}$。

(2)对于非数值的离散型数据,进行 one-hot 编码[10]数值化处理。one-hot 编码是将原某类型属性按照其种类进行编码,编码结果是只有一位为有效值 1,其余位都是 0。年龄属性已经是数值型属性,所以不做处理。性别是二元属性,分为“男”和“女”,所以分别编码为[0]和[1]。职业属于标称属性,包含多种类别, 按照类别总数 S 构建长度为 S 的编码串, 并由先后出现顺序为其置“1”编码,在 MovieLens-100K 数据集中共出现有 21 种职业类型,数据集中最先出现的“technician”职业的 one-hot 编码为[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1],紧接着出现的“writer”职业的 one-hot 编码为[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0],以此类推。然后所有属性拼接组合成稀疏的用户属性向量,对于“年龄=24,性别=男,职业=技术员”的用户 User_a=(24, man, technician),经如上处理后得到的用户属性向量为 User_a′=[24 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]。

(3)embedding 处理。经由 one-hot 编码后的数据必然是稀疏的,所以将其进行 embedding 处理,即是将原稀疏向量与固定转换矩阵做内积变换,转化为稠密向量,即:

$$\boldsymbol{User\_a}''=\boldsymbol{User\_a}'\cdot\boldsymbol{W}_{h\times k}$$

$$(\boldsymbol{User\_a}''\in R^k,\ \boldsymbol{User\_a}'\in R^h, k<<h). \tag{4}$$

得到由用户稠密属性向量构建的用户属性矩阵,就可以进行 K-means 聚类分析了。算法 1 的设计步骤见如下。

算法 1　K-means。用于聚类划分的 K-均值算法,其中每个簇的中心(质心)都用簇中所有对象元素的均值来表示[11]

输入: K 表示簇的数目;D 为包含 n 个对象元素的数据集

输出: K 个簇的集合

(1) 从 D 中任意选取 K 个对象作为初始簇的中心;

(2)repeat

(3)根据簇中对象元素的均值,将每个对象分配到最相似的簇;

(4)更新簇均值,即重新计算每个簇中对象的均值;

(5)until 不再发生变化。

采用 K-means 聚类算法提前对用户进行聚类分析,可减小构建的用户–项目评级矩阵规模,若不做预处理,则需要对全部用户和全部项目信息构建评级矩阵,这样在矩阵分解时会严重降低算法效率,占用大量内存。而通过聚类构建小规模的近邻用户–项目矩阵,可加快矩阵分解计算,利于算法模型的整体效率,且额外的聚类代价远远小于大规模矩阵分解的消耗代价。

K-means 聚类处理为当前用户构建了近邻集合,由于兴趣偏好相似的用户倾向于购买相同项目的思想,所以在近邻用户中进行数据分析,可有效提高推荐的准确度,实验证明这样的设计思路的确对推荐中召回率和精确度有提升效果。

1.3 KMMD 算法设计

本文将 K-means 算法与 Funk-SVD 矩阵分解算法结合,提出了 KMMD 算法。算法 2 的设计流程详见如下。

算法 2 KMMD, 基于 K-means 的矩阵分解算法

输入: *User* 表示包含 m 个用户的用户属性(*User*)数据集;U_a 表示当前待推荐用户的属性信息; *Ratings* 表示包含 m 个用户对 n 个项目的部分评级数据集;L 表示推荐列表长度;K 表示聚类簇数

输出: 针对用户 U_a 的推荐列表 *List*

(1)对 *User* 进行 K-means 聚类分析,划分为 K 个簇: $C_1, C_2, \ldots, C_K$;

(2)if U_a == 老用户 then

(3)提取 U_a 所在簇 C_a 的所有对象元素,并从 *Ratings* 中筛选出这些对象元素的评级数据,构建近邻用户–项目评级矩阵 $\boldsymbol{R}$;

(4)将 $\boldsymbol{R}$ 进行 Funk-SVD 分解,得到 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 两个低秩矩阵;

(5)矩阵重构: $\boldsymbol{R}' = \boldsymbol{U} \cdot \boldsymbol{V}$;

(6)在重构评级矩阵 $\boldsymbol{R}'$ 中找到对 U_a 的重构预测向量 $\boldsymbol{r}_a$;

(7)else 将 U_a 与簇中心求相似度,找到 U_a 归属的簇 C_a;

(8)从 *Ratings* 中筛选出 C_a 簇中对象元素的评级数据,构建近邻用户 – 项目评级矩阵 $\boldsymbol{R}$;

(9)将 $\boldsymbol{R}$ 进行 Funk-SVD 分解,得到 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 两个低秩矩阵;

(10)矩阵重构: $\boldsymbol{R}' = \boldsymbol{U} \cdot \boldsymbol{V}$;

(11)求得 U_a 与 C_a 中各对象元素的相似度 $Sim_{a,i}(i \in C_a)$;

(12)加权计算求得 U_a 的预测评级:

$$r_a = \frac{\sum_{i \in C_a} sim_{a,i} \cdot r_i}{\sum_{i \in C_a} sim_{a,i}}. \tag{5}$$

(13)end if

(14)对 r_a 中各个项目预测值排序,选出前 L 个项目组成推荐列表 *List*。

2 实验与结果分析

本实验采用数据集 MovieLens-100K,使用召回率和精确度作为模型评价指标,进行参数影响及参数确定的实验,得到效果最佳的 KMMD 算法模型,并将其与 2 种传统算法 CBbyPortrait 和 MFbySVD 做了对比实验。研究可得解析详述如下。

2.1 数据集

基于 MovieLens 数据集[12]是由明尼苏达大学的 Grouplens 研究项目收集整理。数据集获取地址: http://files.grouplens.org/datasets/movielens/。实验具体选用的是 MovieLens-100K。MovieLens-100K 数据集包含 943 名用户对 1 682 部电影的 10 万条评分记录,评分采用 5 分制(1,2,3,4,5)。该数据集主要包含 3 个文件:用户数据文件(u.user)、项目数据文件(u.item)和评分文件(u.data),其中 u.user 包含用户的人口统计信息,字段有:用户标识(user id)、年龄(age)、性别(gender)、职业(occupation)和邮编(zip code);u.item 包含电影项目的信息,字段有:电影标识(movie id)、电影标题(movie title)、上映日期(release date)、视频发布日期(video release date)、数据源链接(IMDb URL)和类别属性(genres);u.data 包含完整的评级数据,字段有:用户标识(user id)、项目标识(item id)、评分(rating)和时间戳(timestamp)。

2.2 模型评估指标

本文采用混淆矩阵[13]中的精确度(*Precision*)和召回率(*Recall*)进行算法模型评估。涉及到的参数和计算方法如下: *TP*(True Positive)表示将正类预测为正类;*TN*(True Negative)表示将负类预测为负类;*FP*(False Positive)表示将负类预测为正类,也称为误报;*FN*(False Negative)表示将正类预测

为负类,也称为漏报。

其中, 召回率(*Recall*) 和精确度(*Precision*) 的计算公式为:

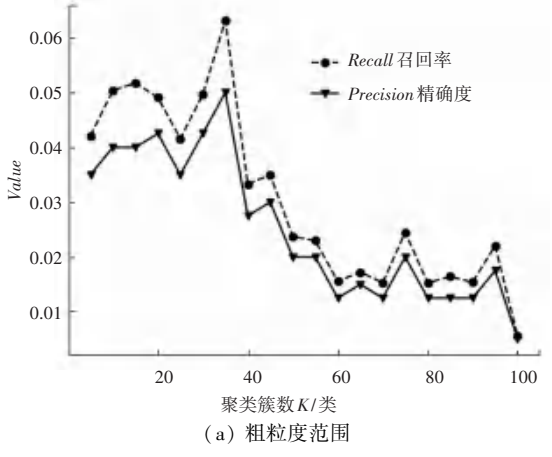
$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

$$Precision = \frac{TP}{TP + FP}. \tag{7}$$

2.3　参数选取

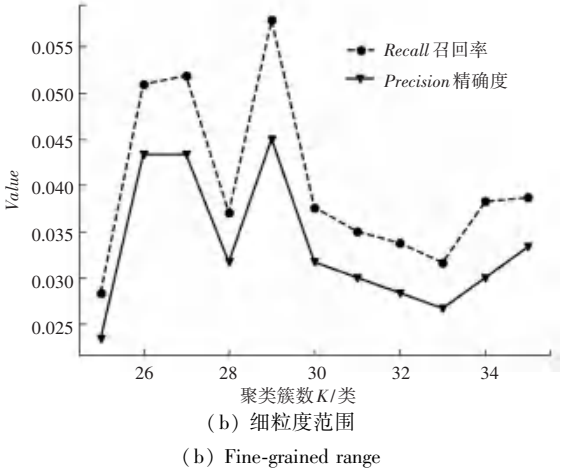
为了得到 KMMD 算法的最佳效果,首先进行控制变量实验,测试重要参变量的取值变化对算法效果的影响。KMMD 算法的重要参变量有 3 个: K-means 聚类的聚类种数 K; 矩阵分解的训练步数 *Steps*; 推荐列表长度 L。而 Funk-SVD 的学习率参变量 α 依据经验设定为 0.000 2。实验随机选取 MovieLens-100K 数据集的 80% 作为训练集,余下 20%作为测试集。研究中,将分 3 组进行控制变量实验。研究内容详见如下。

(1) *Steps* 固定取 50, L 固定取 10。观察参数 K 的变化影响,结果如图 1 所示。



(a) 粗粒度范围

(a) Coarse granularity range



(b) 细粒度范围

(b) Fine-grained range

图 1　*K* 值变化影响

Fig. 1　The influence of *K* value change

图 1 中,实线实心倒三角表示精确度的变化趋势,虚线实心圆圈表示召回率的变化趋势。图 1(a) 是在[5,100]范围内粗粒度验证 K 值变化对实验精度的影响,可以看到,随着 K 值逐渐增大,召回率和精确度都是先增后降的趋势,其极值在 K 取 25 - 35 的范围内取得。为进一步获取最佳 K 值的推荐效果, 图 1(b)给出了在[25,35]闭区间范围内细粒度实验仿真。由实验结果可知,当 K 值取 29 时,算法的当前推荐效果最佳。

(2) K 固定取 29,L 固定取 10。观察参数 *Steps* 的变化影响,结果如图 2 所示。

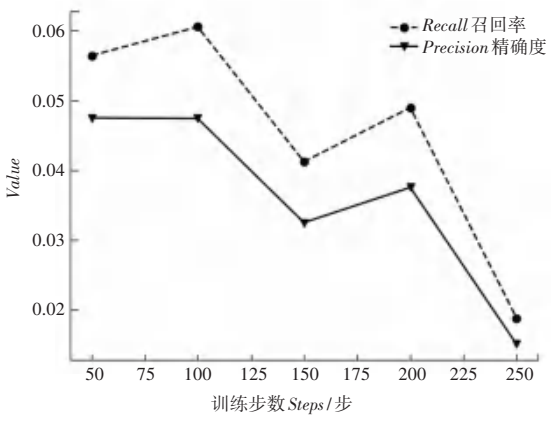


图 2　*Steps* 值变化影响

Fig. 2　The influence of *Steps* value change

由实验结果可知, 随着 *Steps* 值的增大,召回率和精确度总体趋势都是下降的, 当 *Steps* 值取[80,110] 区间值时,其实验效果较好,为方便实验进行,将 *Steps* 值取为 100。

(3)*Steps* 固定取 100,K 固定取 29。观察参数 L 的变化影响,结果如图 3 所示。

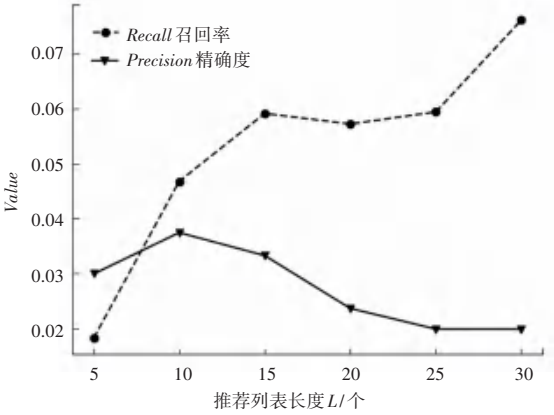


图 3　*L* 值变化影响

Fig. 3　The influence of *L* value change

由实验结果可知,随着 L 值的增大,召回率会呈现递增趋势,而精确度总体是呈现递减趋势,最终逐

渐收敛平稳。为确定 L 取具体何值时,可使推荐的整体效果较好,研究中还对图 3 实验结果进行 F 度量表示,即:

$$F = \frac{2 \times precision \times recall}{precision + recall}. \tag{8}$$

F 度量赋予了召回率和精确度相等的权重,是两者的调和均值。其结果如图 4 所示。可知在 L 值取 15 时,F 值最高,综合推荐效果最好。

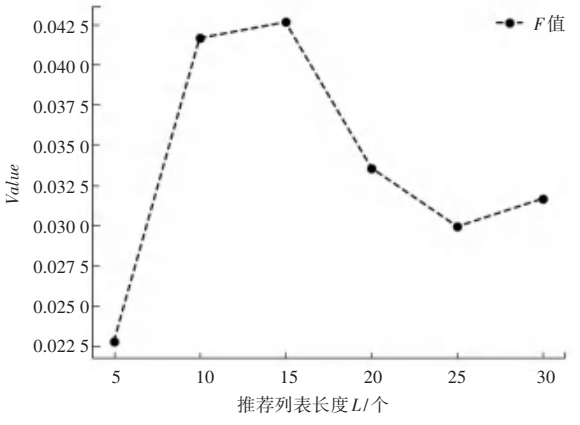


图 4 F 度量值变化

Fig. 4 Change of F value

2.4 对比实验

为了验证 KMMD 算法模型的真实效果,选取了 2 种推荐算法模型与本文提出算法进行比较。这 2 种算法的设计表述如下。

(1) CB(Content-based Recommendation):基于内容的推荐模型,通过召回用户日志文件获取与用户有过交互的项目信息,再根据项目的属性特征学习用户的偏好兴趣,可构建用户画像,并以此计算用户与待推荐项目匹配度,推荐与其过去已购买过物品相似度高的商品。

(2) MF(Matrix Factorization):传统基于矩阵分解的推荐模型,通过用户对项目的显式评分数据构建用户-项目评分矩阵,使用 SVD 奇异值分解矩阵后再进行重构,预测用户对未购买过的商品的评分,进行推荐。

对比实验在 MovieLens-100K 数据集下进行,使用了十折交叉验证法。根据控制变量实验结果,对 KMMD 算法的参数选取了 $K = 29$,$Steps = 100$,$L = 15$。实验结果对比见表 1。对表 1 分析后可知,其研究结果的重点陈述见如下。

(1) 召回率表现的是推荐效果的灵敏性,从该值的角度来分析,KMMD 算法的效果最好,其相较于 CB 算法提升了 15.64%,相较于 MF 算法提升了 154%。说明在灵敏性上,KMMD 算法有了很大的提升。

(2) 从精确度来看,KMMD 算法的效果远高于其它两个算法,比 CB 提升了 30.43%,比 MF 提升了 103%。可见,KMMD 算法在推荐的精确度上也有很不错的表现。

由实验可得,KMMD 算法在召回率和精确度上都有很大提升,能为用户提供更准确的推荐。

表 1 召回率和精确度对比

Tab. 1 Comparison of recall rate and accuracy

模型	召回率	精确度
CB	0.040 9	0.018 4
MF	0.018 6	0.011 8
KMMD	0.047 3	0.024 0

2.5 模型预测

该部分通过实验验证提出的 KMMD 算法可以对新用户进行有效推荐。实验是在包含 943 名用户的 MovieLens-100K 数据集基础上,仿照其数据格式,构造了 2 名用户基本属性数据作为新用户,作为用户冷启动实验对象,数据如下:

(1) 用户 a:年龄 = 24,性别 = 男(M),职业 = 技术员(technician);

(2) 用户 b:年龄 = 50,性别 = 女(F),职业 = 作家(writer)。

实验结果见表 2。

表 2 用户冷启动预测

Tab. 2 Prediction of user cold start

Attribute	Recommended projects (ID)	Predictive ratings
Ua = (*Age* =	59	4.393
24,	134	4.380
Gender = *M*,	657	4.364
Occupation =	483	4.309
technician)	1 137	4.300
Ub = (*Age* =	302	4.521
50,	272	4.294
Gender = *F*,	313	4.177
Occupation =	898	4.142
writer)	315	4.138

实验结果说明,KMMD 算法可以对新用户进行项目推荐,结果以评级高低排序。KMMD 算法有效改善了用户冷启动问题。

3 结束语

将聚类算法与协同过滤思想相结合,提出了一种基于 K-means 的矩阵分解推荐算法。首先通过实验分析不同参数对KMMD算法的召回率和精确度

(下转第 66 页)