·先进计算与数据处理· 文章编号：1000-3428(2019)11-0068-06 文献标志码：A 中图分类号：TP391

基于 Coclus 联合聚类与非负矩阵分解的推荐算法

王泽华，柯新生

（北京交通大学 经济管理学院，北京 100044）

摘 要： 当前推荐系统多数存在推荐准确性低、受稀疏性影响大且稳定性差的问题，因此，在 Coclus 聚类算法的基础上，提出一种评分矩阵与联合聚类的推荐算法。通过 Coclus 联合聚类，利用图模块度最大化理论分别将评分矩阵的行与列分成 g 类，经过行列变换形成 $g \times g$ 个低秩评分子矩阵，并对低秩评分子矩阵进行矩阵分解，填充缺失值，以提高推荐质量，在矩阵分解阶段采用改进的非负矩阵分解算法，通过引入 $L1$、$L2$ 范数分别提高特征值选择能力和防止模型过拟合，并利用坐标轴下降的迭代算法进行参数更新。实验结果表明，与基线算法相比，该算法具有较高的推荐准确率，且稳定性较强。

关键词： 非负矩阵分解；联合聚类；推荐系统；坐标轴下降法；模块度

开放科学（资源服务）标志码（OSID）：

中文引用格式： 王泽华，柯新生. 基于 Coclus 联合聚类与非负矩阵分解的推荐算法[J]. 计算机工程，2019，45(11)：68-73，80.

英文引用格式： WANG Zehua，KE Xinsheng. Recommendation algorithm based on Coclus joint clustering and non-negative matrix factorization[J]. Computer Engineering，2019，45(11)：68-73，80.

Recommendation Algorithm Based on Coclus Joint Clustering and Non-negative Matrix Factorization

WANG Zehua，KE Xinsheng

（School of Economics and Management，Beijing Jiaotong University，Beijing 100044，China）

【Abstract】 Most of the current recommendation systems have many defects，such as low recommendation accuracy，being subject to sparsity and poor stability，so we propose a recommendation algorithm based on Coclus joint clustering and non-negative matrix decomposition. Firstly，through Coclus joint clustering，we use the graph modularity maximization to divide row and column of the scoring matrix into g classes respectively，forming $g \times g$ low rank scoring submatrices through row and column transformation. Then we perform matrix decomposition on each low rank scoring submatrix and fill in the missing values to improve the recommendation quality. In the matrix decomposition stage，we adopt an improved non-negative matrix decomposition algorithm，introducing $L1$ and $L2$ norms respectively to improve the feature value selecting ability and prevent the over-fitting of model. Finally，we use the iterative algorithm of coordinate descent method to update the parameters. Experimental results show that compared with the baseline algorithm，the proposed algorithm has higher recommendation accuracy and better stability.

【Key words】 non-negative matrix factorization；joint clustering；recommendation systems；coordinate axis descent method；modularity

DOI： 10.19678/j.issn.1000-3428.0055033

0 概述

随着信息技术的快速发展，推荐系统能够帮助用户更好地发现兴趣点，实现精准营销和个性化推荐。推荐系统起源于信息检索和认知科学等多个领域，并且引起了信息系统、计算机科学领域研究团体的兴趣[1]，该系统可分为基于内容的推荐系统和协同过滤推荐系统，为解决矩阵稀疏性对推荐系统的影响，一些学者提出了融合聚类的协同过滤算法。

聚类技术采用无监督方式对数据集进行分类，

基金项目： 科技部科技支撑计划项目“音乐云商业智能服务关键技术的研究”(2013BAH66F03-02)。

作者简介： 王泽华(1994—)，男，硕士研究生，主研方向为数据挖掘；柯新生，教授。

收稿日期： 2019-05-27 **修回日期：** 2019-07-01 **E-mail：** 17120624@bjtu.edu.cn

所得聚类结果要求类内数据对象间相似性尽量大,类间数据对象间相似性尽量小[2]。文献[3]首先对稀疏矩阵进行分解插值,然后利用用户余弦相似度聚类对目标用户进行推荐。文献[4]利用交替最小二乘矩阵分解,并对预测矩阵的用户进行 K-means 聚类以提高推荐的精确度。文献[5]利用 pearson 相似度对用户进行聚类,然后对聚类后的评分子矩阵进行非负矩阵分解来填充缺失值。文献[6]对评分矩阵进行矩阵分解,并分别对用户和项目进行聚类,从簇中查找用户的最近邻和项目推荐候选集。

以上研究虽然从一定程度上缓解了稀疏性对推荐的影响,但是进行聚类时仅考虑了用户或项目,忽略了用户与项目间的二元关系。因此,研究人员提出用联合聚类的方法解决该问题,联合聚类是一种同时考虑矩阵行聚类和列聚类的方法,在数据分析、协同过滤等领域有着广泛应用[7]。文献[8]从统计学的角度计算每个评分属于某类的概率,选取最大的类别概率作为该评分的类别来对评分矩阵进行联合聚类。文献[9]采用基于信息论的方法进行联合聚类,然后利用混合蛙跳算法搜索项目最近临缓解稀疏性影响。文献[10]基于 Bregman 距离对矩阵进行联合聚类寻找局部低簇群,然后并发地进行矩阵分解,最后进行均值融合来提高推荐质量。

本文在联合聚类中引入模块度的概念,利用图模块最大化理论将评分矩阵分成 $g\times g$ 个低秩评分子矩阵,通过矩阵分解对低秩评分矩阵进行缺失值填充,并利用坐标轴下降法进行迭代更新,以达到推荐的目的。

1　Coclus-nmf 算法

1.1　Coclus 联合聚类算法

Coclus 联合聚类算法是一种应用图模块度最大化对文档-词汇矩阵进行联合聚类的算法[11],用户-商品的二元关系与文档-词汇的二元关系相似,因此,本文对该算法进行改进并将其应用于评分矩阵的联合聚类。假设推荐系统中有 n 个用户和 d 个商品,定义评分矩阵为 $\boldsymbol{A}\in R^{n\times d}$,用户和商品的聚类数量均设为 $g(g\geqslant 2)$,基于图模块最大化的联合聚类算法为对角线协聚算法且属于硬聚类,因此用户和商品的聚类数量一致,每个用户和商品只属于一类。定义块索引矩阵 $\boldsymbol{C}\in R^{n\times d}$,行索引矩阵 $\boldsymbol{Z}\in R^{n\times g}$,列索引矩阵 $\boldsymbol{W}\in R^{d\times g}$,$\boldsymbol{C}=\boldsymbol{Z}\times\boldsymbol{W}$,这些索引矩阵将用来计算模块度以衡量联合聚类的效果。

$$c_{ij}=\begin{cases}1, i\text{ 行与 }j\text{ 列属于一类}\\0,\text{其他}\end{cases}$$

$$z_{ik}=\begin{cases}1, i\text{ 行属于 }k\text{ 类}\\0,\text{其他}\end{cases}$$

$$w_{jk}=\begin{cases}1, j\text{ 列属于 }k\text{ 类}\\0,\text{其他}\end{cases}$$

模块度是图聚类的一种质量标准,自提出以来在多个学科得到了广泛关注[12],在联合聚类环境下,模块度公式如式(1)所示。

$$Q(\boldsymbol{A},\boldsymbol{C})=\frac{1}{a_{..}}\sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{k=1}^{g}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right)z_{ik}w_{jk}\tag{1}$$

其中,$a_{..}=\sum_{i,j}a_{ij}$是评分矩阵所有元素的和,$a_{i.}=\sum_{j}a_{ij}$是第 i 行所有元素的和,$a_{.j}=\sum_{i}a_{ij}$是第 j 列所有元素的和。模块度的大小 $Q\in[0,1]$,模块度越大表示联合聚类的效果越好,需要找到使 Q 最大时的 $\boldsymbol{Z}$ 与 $\boldsymbol{W}$ 矩阵,即求使评分矩阵联合聚类效果最好时的行列分类索引。下文用启发式算法进行求解。

$Q(\boldsymbol{A},\boldsymbol{C})$可以写成 $Q(A^{W},\boldsymbol{Z})$的形式,其中:

$$A^{W}:=\{a_{ik}=\sum_{j=1}^{d}w_{jk}a_{ij}, i=1,2,\cdots,n,k=1,2,\cdots,g\}\tag{2}$$

$$\delta^{W}:=\left\{\delta_{ik}^{W}=\frac{a_{i.}a_{.k}^{W}}{a_{..}}, i=1,2,\cdots,n,k=1,2,\cdots,g\right\}\tag{3}$$

$$a_{.k}^{W}=\sum_{j=1}^{d}w_{jk}a_{.j}\tag{4}$$

证明过程如下:

$$\begin{aligned}Q(\boldsymbol{A},\boldsymbol{C})&=\frac{1}{a_{..}}\sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{k=1}^{g}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right)z_{ik}w_{jk}=\\&\frac{1}{a_{..}}\sum_{i=1}^{n}\sum_{j=1}^{d}z_{ik}\sum_{j=1}^{d}w_{jk}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right)=\\&\frac{1}{a_{..}}\sum_{i=1}^{n}\sum_{k=1}^{g}z_{ik}\left(\sum_{j=1}^{d}w_{jk}a_{ij}-\frac{a_{i.}}{a_{..}}\sum_{j=1}^{d}w_{jk}a_{.j}\right)=\\&\frac{1}{a_{..}}\sum_{i=1}^{n}\sum_{k=1}^{g}\left(a_{ik}^{W}-\frac{a_{i.}a_{.k}^{W}}{a_{..}}\right)z_{ik}=\\&\frac{1}{a_{..}}Trace[(A^{W}-\delta^{W})'\boldsymbol{Z}]=\\&Q(A^{W},\boldsymbol{Z})\end{aligned}$$

同理,$Q(\boldsymbol{A},\boldsymbol{C})$可以写成 $Q(A^{Z},\boldsymbol{W})$的形式,其中:

$$A^{Z}:=\{a_{jk}=\sum_{i=1}^{n}z_{ik}a_{ij}, k=1,2,\cdots,g,j=1,2,\cdots,d\}\tag{5}$$

将求 $Q(\boldsymbol{A},\boldsymbol{C})$的最大值转换成求 $Q(A^{W},\boldsymbol{Z})$和 $Q(A^{Z},\boldsymbol{W})$的最大值,$\boldsymbol{Z}$、$\boldsymbol{W}$ 由初始化随机而得:

$$\boldsymbol{Z}^{*}=\operatorname{argmax}_{Z}Trace\,(A^{W}-\delta^{W})'\boldsymbol{Z}\tag{6}$$

$$\boldsymbol{W}^{*}=\operatorname{argmax}_{W}Trace\,(A^{Z}-\delta^{Z})'\boldsymbol{W}\tag{7}$$

对参数进行交替更新,直到 Q 的值不再变化即求得最大值 Q,下面举例说明算法过程,评分矩阵 $\boldsymbol{A}$ 是一个 5×4 的矩阵,分类数量 $g=2$。设:

$$\boldsymbol{A}=\begin{bmatrix}5&0&1&2\\3&0&0&4\\2&1&1&0\\0&3&0&1\\2&2&3&0\end{bmatrix}$$

$$\boldsymbol{W}=\begin{bmatrix}0&1\\1&0\\0&1\\0&1\end{bmatrix}$$

$$\boldsymbol{Z}=\begin{bmatrix}0&1\\1&0\\0&1\\1&0\\1&0\end{bmatrix}$$

此时,$Q=0.191\ 4$。

经过联合聚类后,对行列进行变换并将同类的行与列放在一起,此时的评分矩阵分成了 4 类,直观可见对角线上的两组零评分较少且评分值相近,模块度提升到了 0.3。

$$\boldsymbol{A}=\left[\begin{array}{cc:cc}5&2&0&1\\3&4&0&0\\\hdashline2&0&1&1\\0&1&3&0\\2&0&2&3\end{array}\right]$$

$$\boldsymbol{W}=\begin{bmatrix}0&1\\1&0\\1&0\\0&1\end{bmatrix}$$

$$\boldsymbol{Z}=\begin{bmatrix}0&1\\0&1\\1&0\\1&0\\1&0\end{bmatrix}$$

此时,$Q=0.3$。

1.2 改进的非负矩阵分解算法

非负矩阵分解因为 Lee 和 Seung 发表在 Nature 上的论文[13]及后续研究而被广泛关注。基础的非负矩阵分解问题可以表述如下:给定一个非负的矩阵 $\boldsymbol{R}\in R_+^{n\times m}(\boldsymbol{R}\geqslant 0)$和一个秩 $k(k\leqslant\min(m,n))$,找到 2 个非负的矩阵:$\boldsymbol{U}\in R_+^{n\times k}$称为基矩阵,$\boldsymbol{I}\in R_+^{k\times m}$称为系数矩阵,然后将 2 个矩阵的内积与 $\boldsymbol{R}$ 矩阵近似,即:

$$\boldsymbol{R}\approx\boldsymbol{U}\boldsymbol{I}^{\mathrm{T}} \tag{8}$$

此时原矩阵 $\boldsymbol{R}$ 中一列向量可以解释为对基矩阵所有列向量的加权和,而权重系数为系数矩阵中对应列向量中的元素,这种基于向量组合的表示形式具有很直观的语义解释,反映了“部分组成整体”的概念。具备这种概念的非负矩阵分解已经在现实应用中得到了体现,如人脸识别、基因分析等。

为了估计基矩阵 $\boldsymbol{U}$ 和系数矩阵 $\boldsymbol{I}$,需要考虑成本函数来量化矩阵之间的差异,使 $\boldsymbol{U}$ 和 $\boldsymbol{I}$ 的内积与原矩阵近似,比较简单的方法是用 Frobenius 范数:

$$F=\frac{1}{2}\|\boldsymbol{R}-\boldsymbol{U}\boldsymbol{I}\|_{\mathrm{Fro}}^2 \tag{9}$$

该成本函数的作用是使原始矩阵 $\boldsymbol{R}$ 和预测矩阵之间的差异最小化,因此,这个函数的下界为零,当且仅当 $\boldsymbol{R}=\boldsymbol{U}\boldsymbol{I}^{\mathrm{T}}$。为了防止模型的过拟合以及增强特征选择能力,本文将传统的非负矩阵分解算法加以改进,引入 *L*2 范数防止过拟合,引入 *L*1 范数增强特征选择能力,目标函数如下:

$$F=\frac{1}{2}\|\boldsymbol{R}-\boldsymbol{U}\boldsymbol{I}\|_{\mathrm{Fro}}^2+\alpha\rho\|\boldsymbol{U}\|_1+\alpha\rho\|\boldsymbol{I}\|_1+\frac{\alpha(1-\rho)}{2}\|\boldsymbol{U}\|_{\mathrm{Fro}}^2+\frac{\alpha(1-\rho)}{2}\|\boldsymbol{I}\|_{\mathrm{Fro}}^2 \tag{10}$$

其中,α 为 *L*1&*L*2 正则化参数,而 ρ 为 *L*1 正则化占总正则化项的比例,$\|*\|_1$ 为 *L*1 范数。传统的非负矩阵分解算法目标函数使用梯度下降法求解,然而引入 *L*1 范数后,因为有绝对值的存在,所以目标函数的导函数不连续,不能使用梯度下降[14],本文采用坐标轴下降的方法进行参数求解。坐标轴下降法是一种迭代法,通过启发式的方法一步步地迭代求解函数的最小值,和梯度下降法不同的是,坐标轴下降法是沿着坐标轴的方向下降,而不是采用梯度的负方向下降。对于目标函数式(10),有 $n\times k+k\times m$ 个参数,即 $\boldsymbol{U}$ 和 $\boldsymbol{I}$ 矩阵中的所有元素,在每次迭代中,固定 $n\times k+k\times m-1$ 个参数,对 1 个参数进行求导优化。

在初始化的 $\boldsymbol{U}$ 和 $\boldsymbol{I}$ 矩阵中,待优化参数 $\theta_1\sim\theta_{n\times k+k\times m}$,在第 t 轮迭代,从 θ_1^t 开始计算,到 $\theta_{n\times k+k\times m}^t$ 为止,F 为式(10)目标函数,计算公式如下:

$$\begin{aligned}\theta_1^t&=\arg_{\theta_1}\min F(\theta_1^t,\theta_2^{t-1},\theta_3^{t-1},\cdots,\theta_{n\times k+k\times m}^{t-1})\\\theta_2^t&=\arg_{\theta_2}\min F(\theta_1^t,\theta_2^{t-1},\theta_3^{t-1},\cdots,\theta_{n\times k+k\times m}^{t-1})\\&\vdots\\\theta_{n\times k+k\times m}^t&=\arg_{\theta_{n\times k+k\times m}}\min F(\theta_1^{t-1},\theta_2^{t-1},\theta_3^t,\cdots,\theta_{n\times k+k\times m}^{t-1})\end{aligned} \tag{11}$$

检查 θ^t 和 θ^{t-1} 参数在各个纬度的变化情况,如果所有维度的变化情况都较小,那么认为结束迭代,否则继续 $t+1$ 轮的迭代。结束迭代则求得最终的 $\boldsymbol{U}$ 和 $\boldsymbol{I}$ 矩阵,2 个矩阵的内积所得的新矩阵是一个没有缺失值的矩阵,因此,完成了预测评分的推荐任务。

1.3 Coclus-nmf 融合推荐算法

基于 Coclus 联合聚类与非负矩阵分解的推荐算法是一种 2 个阶段的混合推荐算法,第 1 个阶段为联合聚类阶段,采用图模块度最大化理论将评分矩阵分成 $g\times g$ 个子评分矩阵,子评分矩阵内部评分值近似,将整体的稀疏矩阵转化为 $g\times g$ 个相对密集的子矩阵,完成对稀疏矩阵的初步降维。通过 Coclus 联合聚类算法得出对评分矩阵的联合聚类索引矩阵 $\boldsymbol{Z}$、$\boldsymbol{W}$,通过索引矩阵可以从原始评分矩阵中抽取某类评分子矩阵。对每个评分子矩阵进行改进的非负矩阵分解并填充缺失值,具体算法如下:

算法 1 Coclus-nmf 算法

第 1 阶段:Coclus 联合聚类

输入 评分矩阵 $\boldsymbol{A}$,聚类数量 g

输出 索引矩阵 $\boldsymbol{Z}$、$\boldsymbol{W}$

1. 随机初始化 $\boldsymbol{W}$ 矩阵

循环:

2. 根据式(2)计算 A^W

3. 根据式(6)计算使 $Q(A^W,\boldsymbol{Z})$ 最大时的 $\boldsymbol{Z}$

4. 根据式(5)计算 A^Z

5. 根据式(7)计算使 $Q(A^Z,\boldsymbol{W})$ 最大时的 $\boldsymbol{W}$

6. 根据式(1)计算模块度 $Q(\boldsymbol{A},\boldsymbol{C})$

直到模块度 $Q(\boldsymbol{A},\boldsymbol{C})$ 的值不再变化

第 2 阶段:nmf 非负矩阵分解

输入　评分子矩阵 $\boldsymbol{R}$,正则化参数 α、ρ,潜在因子数 k

输出　填充后的矩阵 $\boldsymbol{R}^*$

1. 随机初始化 $\boldsymbol{U}$、$\boldsymbol{I}$ 矩阵

2. 提取 $\boldsymbol{U}$、$\boldsymbol{I}$ 矩阵所有元素 $\theta_1 \sim \theta_{n\times k+k\times m}$

循环:

3. 根据式(11)更新参数

4. 根据式(10)计算目标函数

直到目标函数 F 的值不再变化

5. 根据式(8)计算填充后的矩阵 $\boldsymbol{R}^*$

算法的参数选择将直接影响算法的结果,本文算法涉及的参数有 4 个,分别是联合聚类数量 g、正则化系数 α、ρ 和潜在因子数 k。对于联合聚类数量,在社区划分中,社区网络的复杂程度直接影响社区划分的数量与模块度的大小[15]。相似地,将模块度引入评分矩阵的划分时,评分的分值范围与稀疏度影响联合聚类的数量划分。将本文联合聚类算法应用到不同的数据集时,根据评分范围与评分的复杂程度选取聚类数量的区间,再结合具体实验结果确定具体的聚类数量。在本文实验所用数据集中,虽然评分矩阵规模宏大,但较小的评分范围与稀疏的数据使得联合聚类数量可以在较小时($g<10$)取得比较好的聚类效果。

矩阵分解的正则化系数 α、ρ 是 2 个超参数,α 主要控制模型的拟合程度,增强模型的泛化能力,取值越大则正则化惩罚越强,而 ρ 增强模型的特征选择能力,使模型更好地拟合训练数据[16]。在本文算法中,正则化系数用于控制矩阵分解的推荐功能,使推荐结果精确的同时降低样本训练数据的影响,提高泛化能力。正则化系数受训练数据的选取、数据结构等因素影响,一般用启发式的方式确定参数值,通过固定其他参数实验得到。

基于矩阵分解的推荐算法是潜在因子模型的一种,模型假设用户根据潜在因子对商品进行打分,潜在因子 k 的数量决定用户对商品考量的维度[17]。潜在因子数量的选取与商品特性有关,数量过多会使模型过拟合且计算和存储的复杂度增大。

2　实验结果与分析

本文进行一系列实验,从推荐质量和稳定性方面检验该算法的质量,选用推荐常用数据集来描述性能。

2.1　数据集

推荐系统的数据集多种多样,本文选用 MovieLens 电影评分数据集对非负矩阵分解算法进行评价。从 MovieLens 网站上收集 GroupLens 提供的数据集,MovieLens 由 100K、1M 和 10M 3 个数据集组成,选择 100K 作为数据集,数据集的统计数据如表 1 所示。

表 1　数据集信息统计

用户数量	电影数量	评分数	评分范围	最少评分	稀疏度/%
943	1 682	100 000	1 ~ 5	20	93.7

2.2　评价指标

评价协同过滤推荐系统的方法有多种[18],本文采用常用的度量方法、平均绝对误差(MAE)、均方根误差(RMSE)、覆盖率(Coverage)和稳定性(Stability)来度量推荐质量。

2.2.1　平均绝对误差

平均绝对误差计算公式如下:

$$M_{\mathrm{MAE}}=\frac{1}{R_{\mathrm{test}}}\sum_{u,i}\left|R_{ui}-R_{ui}^{*}\right| \tag{12}$$

其中,R_{ui}^{*} 表示通过模型预测出的用户 u 对项目 i 的评分,R_{test} 表示评分的数量,MAE 表示的是经过模型预测后的评分与原始评分的差异。

2.2.2　均方根误差

均方根误差计算公式如下:

$$R_{\mathrm{RMSE}}=\sqrt{\left(R_{ui}-R_{ui}^{*}\right)^{2}} \tag{13}$$

从定义可以看出,MAE 和 RMSE 的值越小,表示推荐的质量越好。

2.2.3　覆盖率

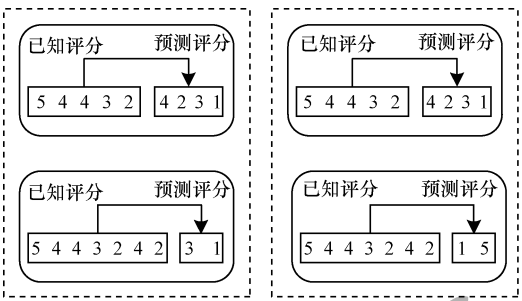
除了推荐的质量外,推荐的覆盖率也极为重要,MAE 和 RMSE 并没有完全衡量推荐系统的有用性,还需要计算推荐系统的覆盖率,即推荐系统可为用户推荐的项占用来构建模型的项的比例,在有些情况下用户评价的条目很少,推荐系统虽然为其做出了精确的推荐,但推荐的项目个数不满足要求,此时覆盖率的评价指标变得比较重要。

$$coverage=\frac{\left|R_i \mid R_i\in R_{\mathrm{test}}\right|}{\left|R_{\mathrm{test}}\right|} \tag{14}$$

其中,R_i 表示经过模型推荐的项,一个推荐系统的准确率和覆盖率必须放在一起考虑,准确率和覆盖率的均衡是推荐系统的关键。

2.2.4 稳定性

推荐系统的稳定性能够考量推荐系统预测的一致性,稳定性影响用户对推荐系统的使用体验[19],推荐系统通过某用户的历史评分对其进行预测并推荐项目,当用户使用了推荐项目中的某个或某几个项目并对其进行评分时,此时推荐系统根据现有的评分集合(历史评分加新评分)预测的项目评分与第一次预测的分数相近,则系统稳定性好,反之则稳定性差。推荐系统稳定性如图 1 所示。



(a)稳定的系统 (b)不稳定的系统

图 1 推荐系统的稳定性

稳定性测量方法[20]如下:1)对基于已知评分的推荐算法进行训练,对所有未知评分进行预测;2)选择并添加预测评分的子集传入原始集合;3)对基于新评分集合的推荐算法进行训练,并预测未知评分;4)比较第 1)步和第 3)步中的结果。

2.3 实验结果

本节重点比较本文算法和其他方法的性能,所有的实验都在 python3.7 环境下运行,为保证实验结果的准确性,在进行对比实验时,相关参数的设置不变。经过实验得出聚类与矩阵分解的最优参数设置如下:$g=8$,$\alpha=0.5$,$\rho=0.5$,$k=5$。

2.3.1 推荐质量评价

为展示本文算法的有效性,将该算法与以下基础推荐算法进行对比:

1)用户均值(User Mean):该方法使用用户的平均评分来填充缺失值。

2)项目均值(Item Mean):该方法使用项目的平均评分来填充缺失值。

3)基于用户的协同过滤(CF User):该方法通过计算用户相似度为用户的新项目评分。

4)基于项目的协同过滤(CF Item):该方法通过计算项目相似度为项目的新用户评分。

5)非负矩阵分解(NMF):传统的非负矩阵分解,该方法通过矩阵分解降维近似原始矩阵达到推荐目的。

本文实验采用五折交叉实验的方法,每一折包含 80% 的数据作为训练集,剩余的 20% 作为测试集,因为每一次选取的数据集是固定的,重复 5 次后数据的每 20% 都曾作为测试集,将最后的结果取平均作为最终结果。

在对推荐算法的准确性进行实验时,选取的 80% 的数据相当于从原始评分矩阵中抽取了 80% 的评分放入原始大小的矩阵作为训练集(每个用户抽取 80% 的评分确保比例均衡),剩下的 20% 的评分放入原始大小的矩阵作为测试集并记下这些评分的位置,此时提取经过模型预测后得到的预测矩阵中预测评分的响应位置,计算 MAE 和 RMSE 衡量推荐算法的准确性,结果如表 2 所示。从表 2 可以看出,本文算法在数据集上的表现均优于其他基线算法,且融合联合聚类的 NMF 比单纯的 NMF 推荐准确性高,证明了联合聚类能够提升推荐效果。

表 2 推荐算法的准确性

算法	MAE	RMSE
User Mean 算法	0.847	1.055
Item Mean 算法	0.840	1.051
CF User 算法	0.809	1.009
CF Item 算法	0.815	1.016
NMF 算法	0.839	1.046
本文算法	0.798	1.005

由于 MovieLens 数据集中每个用户至少对20 个项目评分,因此 User Mean 算法具有 100% 的覆盖率,但是针对其他数据集并不能保证 100%,而矩阵分解在所有数据集中均能满足 100% 覆盖率,结果如表 3 所示。

表 3 推荐算法的覆盖率 %

算法	覆盖率
User Mean 算法	100
Item Mean 算法	97
CF User 算法	97
CF Item 算法	98
NMF 算法	100
本文算法	100

2.3.2 稀疏性

由前面数据集的信息得知,数据集 100K 的密度为 6.3%,为衡量稀疏性对各个算法推荐质量的影响,抽取数据集的 1%、2%、3%、4% 用来进行实验,其中每个用户抽取的评分比例一致,确保评分矩阵中不出现全零的行,通过对比 User Mean、Item Mean、CF User、CF Item、NMF 和本文算法在不同稀疏度下的推荐准确性,分析 MAE 和 RMSE 随矩阵的密度变化而变化。

图 2 和图 3 显示了 MovieLens 100K 稀疏度变化时推荐算法的准确性,结果显示,评分矩阵越稠密,算法对其推荐的效果越好,而本文算法 MAE 和 RMSE 曲线低于其他曲线,说明稀疏性对它的影响最小。

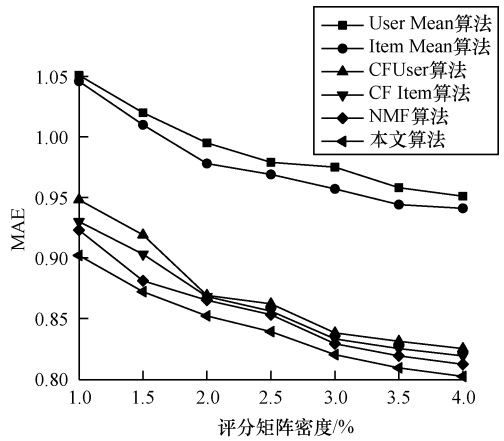


图 2　不同密度对 MAE 的影响

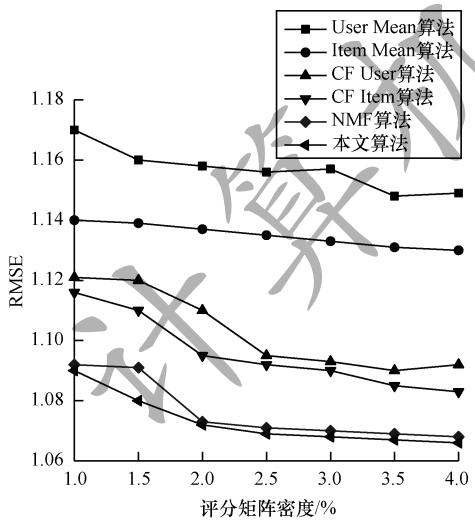


图 3　不同密度对 RMSE 的影响

2.3.3　稳定性

为评价各算法的稳定性表现,在 MovieLens 100K 数据集中,按照每个用户相同的比例选取 80K 的评分来预测剩余 20K 的未知评分,然后从这 20K 的评分中随机选取 10K 的评分与之前的 80K 组成 90K 评分,用来预测剩下的 10K 评分,将 2 次预测的评分对比计算稳定性,结果如表 4 所示。

表 4　推荐算法的稳定性

算法	MAE	RMSE
User Mean 算法	0.006 6	0.009 1
Item Mean 算法	0.004 5	0.007 3
CF User 算法	0.284 9	0.413 7
CF Item 算法	0.210 4	0.325 4
NMF 算法	0.103 9	0.173 7
本文算法	0.044 7	0.067 1

从表 4 可以看出,User Mean 和 Item Mean 算法的稳定性均低于其他算法,因为用平均值填充了矩阵,所以添加新的评分不会改变平均值,因此,这 2 种算法考量稳定性没有实际意义。相对于 CF User、CF Item、NMF 算法,本文算法在稳定性上有显著优势。

3　结束语

本文通过引入图模块最大化的理论,计算评分矩阵与最佳分类索引的模块度,完成对评分矩阵的联合聚类,并通过引入 $L1$、$L2$ 范数增强特征选择能力,防止过拟合。对低秩的评分子矩阵进行缺失值填充,从而达到推荐的目的。实验结果表明,本文算法相较于基线算法推荐准确率高、受稀疏性影响小且稳定性强。但本文联合聚类属于硬聚类,与实际中用户、项目的分类有出入,下一步研究将软聚类与矩阵分解融合进行推荐,以提高分类的准确性。

参考文献

[1] AGHDAAM M H, ANALOUI M, KABIRI P. Modelling trust networks using resistive circuits for trust-aware recommender systems [J]. Journal of Information Science, 2017, 43(1): 135-144.

[2] LU Jie, WU Dianshuang, MAO Mingsong, et al. Recommender system application developments: a survey[J]. Decision Support Systems, 2015, 74: 12-32.

[3] 王娟,熊巍. 基于矩阵分解的最近邻推荐系统及其应用[J]. 统计与决策,2019(6):17-20.

[4] 董立岩,王宇,任怡,等. 基于矩阵分解和聚类的协同过滤算法[J]. 吉林大学学报(理学版),2019,57(1):111-116.

[5] 骆孜,龙华,邵玉斌,等. 基于聚类的非负矩阵分解推荐算法研究[J]. 通信技术,2018,51(11):153-157.

[6] 段元波,高茂庭. 基于项目评分与类型评分聚类的推荐算法[J]. 现代计算机,2017(16):6-11.

[7] LIN Renjie, WANG Shiping, GUO Wenzhong. An overview of co-clustering via matrix factorization[J]. IEEE Access, 2019(99): 1.

[8] 李翔,朱全银. 基于联合聚类和评分矩阵共享的协同过滤推荐[C]//全国高性能计算学术年会论文集. 无锡:[出版者不详],2013:576-583.

[9] 喻金平,张勇,廖列法,等. 基于混合蛙跳联合聚类的协同过滤算法[J]. 微电子学与计算机,2016,33(1):65-71.

[10] 郭蕊,孙福振,王绍卿,等. 基于 Bregman 联合聚类与加权矩阵分解的融合推荐算法[J]. 科学技术与工程,2019,19(8):176-181.

(下转第 80 页)

(上接第 73 页)

[11] AILEM M, ROLE F, NADIF M. Co-clustering document-term matrices by direct maximization of graph modularity[C]//Proceedings of the 24th ACM International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2015: 1807-1810.

[12] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[EB/OL]. [2019-04-20]. http://dx. doi. org/10. 1103/PhysRevE. 69. 026113.

[13] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788.

[14] YU Xianghao, SHEN Juei Chin, ZHANG Jun, et al. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems[J]. IEEE Journal of Selected Topics in Signal Processing, 2016, 10(3): 485-500.

[15] GILARRANZ L J, RRYFIELD B. Effects of network modularity on the spread of perturbation impact in experimental metapopulations[J]. Science, 2017, 357(6347): 199-201.

[16] GUNASEKAR S, WOODWORTH B E, BHOJANAPALLI S, et al. Implicit regularization in matrix factorization[EB/OL]. [2019-04-20]. https://www. doc88. com/p-4522872867906. html.

[17] WEI Jie, HE Jie, CHEN Ke, et al. Collaborative filtering and deep learning based recommendation system for cold start items[J]. Expert Systems with Applications, 2017, 69: 29-39.

[18] MELVILLE P, SINDHWANI V. Recommender systems[M]. Berlin, Germany: Springer, 2017: 1056-1066.

[19] WU C Y, AHMED A, BBUTEL A, et al. Recurrent recommender networks[C]//Proceedings of the 10th ACM International Conference on Web Search and Data Mining. New York, USA: ACM Press, 2017: 495-503.

[20] ADOMACICIUS G, ZHANG Jingjing. Stability of recommendation algorithms[J]. ACM Transactions on Information Systems, 2012, 30(4): 47-54.

编辑　索书志