文章编号:1007－6735(2023)02－0162－09　DOI: 10.13255/j.cnki.jusst.20210905002

基于潜在因子多样性的非负矩阵分解协同过滤模型

陶名康[1]，　王新利[1]，　宋　燕[2]
（1. 上海理工大学 理学院，上海 200093；2. 上海理工大学 光电信息与计算机工程学院，上海 200093）

摘要： 基于非负矩阵分解的协同过滤模型在高维稀疏数据的预测和填补上十分有效，该模型具有推荐个性化、有效利用其他相似用户回馈信息的优点，但也存在预测精度较低等不足。针对用户或项目在不同情景下的评分差异性，提出了一种改进的基于潜在因子多样性的非负矩阵分解的协同过滤模型。该模型充分考虑在不同情境下，用户和项目潜在特征矩阵的多样性，在模型的训练中，采用了单元素非负乘法更新规则和交替方向法，保证了目标矩阵的非负性，且提高了模型的收敛率。在真实的工业数据集上的实验结果表明，相比于经典的非负矩阵分解模型，该模型的预测精度有了明显提高。

关键词： 协同过滤；特征矩阵多样性；非负矩阵分解；非负乘法更新；交替方向法
中图分类号： TP 391　　**文献标志码：** A

A collaborative filtering model of non-negative matrix factorization based on diversity of latent factors

TAO Mingkang[1]，　WANG Xinli[1]，　SONG Yan[2]
(1. *College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China*; 2. *School of Optical–Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*)

Abstract: A collaborative filtering model of non-negative matrix factorization based on diversity of latent factors is effective in predicting high dimension and sparse matrix. This model can recommend personally and utilize the feedback information from other similar users effectively. However, it has the disadvantage of low prediction accuracy. Due to the diversity of ratings for users or items under different circumstances, a collaborative filtering model based on non-negative matrix factorization was proposed. The model considered the diversity of latent characteristic matrices for users and items. The single-element non-negative multiplication update rule and the principle of alternate direction method were integrated in the training of the model, which not only guaranteed the non-negativity of the target matrix, but also improved the convergence rate of the models. Finally, experiments were carried out on

收稿日期: 2021−09−05
基金项目: 国家自然科学基金资助项目(62073223)；上海市自然科学基金资助项目 (18ZR1427100)
第一作者: 陶名康(1994−)，男，硕士研究生．研究方向：机器学习、数据分析等．E-mail：13270802708@163.com
通信作者: 王新利(1975−)，女，讲师．研究方向：机器学习、数据分析、复分析等．E-mail：xlwang@usst.edu.cn

real industrial data sets. The experimental results show that the prediction accuracy of the proposed model is higher than that of the classical non-negative matrix factorization model.

Keywords: *collaborative filtering*; *diversity of latent characteristic matrix*; *non-negative matrix factorization*; *non-negative multiplication updating*; *alternate direction method*

随着互联网技术的快速发展，很多领域都产生了海量数据，在对这些大数据进行分析与挖掘的过程中，推荐系统是一种广泛采用的技术。推荐系统是一个学习系统，包括用户、项目和用户对项目的评分 3 种基本成分。它通过对已有评分数据的学习，获得用户的行为偏好，建立用户和项目之间的关系，为用户提供个性化的推荐，使得用户更快速简便地找到自己所需要的信息，从而促进点击和购买行为的发生[1-2]。协同过滤推荐[3]是推荐系统中应用最广泛的推荐算法之一，一般分为 3 种类型：基于用户（user-based）的协同过滤、基于项目（item-based）的协同过滤、基于模型（model-based）的协同过滤。基于用户的协同过滤[4]通过计算用户和用户之间的相似度，找出相似用户偏好的项目，并预测目标用户对相应项目的评分，把评分最高的若干个项目推荐给用户。而基于项目的协同过滤[5]则是通过考虑项目和项目之间的相似度，然后根据预测评分进行推荐。基于模型的协同过滤[6]是一种使用更普遍的推荐算法，它基于样本的用户喜好信息，训练一个推荐模型，然后根据实时的用户信息进行预测推荐，使用的主要方法有聚类[7]、分类[8]、回归[9]、矩阵分解[10]、神经网络[11]等。

在推荐系统中，一般情况下用户无法对每一个项目评分，而每一个项目也不能被所有的用户点击，只有部分用户和部分数据之间有评分数据，其他部分评分空白。因此，由用户对项目的评分构成的评分矩阵一般是一个高维稀疏矩阵。由于基于矩阵分解的协同过滤在处理此类矩阵的预测和填补方面具有较高的准确性和可扩展性[10]，它是目前基于模型的协同过滤广泛采用的一种方法。矩阵分解，直观上来说就是把原来的大矩阵，近似分解成两个包含潜在变量的小矩阵，其中一个代表用户的潜在特征，另一个代表项目的潜在特征。矩阵的元素表示用户或项目对各潜在因子的符合程度，这两个小矩阵被称为潜在因子特征矩阵。另一方面，为了使学习到的特征矩阵更准确地代表实际意义，并使获得的模型更具有可解释性，通常需要对特征矩阵进行非负性约束。在模型的学习中，单元素非负乘法更新规则[12]（SLF-NMU）和交替方向法[13]（ADM）是目前最常用的训练原则，它可以避免调节学习率，同时还保证了潜在因子的非负性。

传统的基于矩阵分解的协同过滤模型通过把高维矩阵映射为两个低维潜在因子矩阵，使得模型具有较高的预测精度。但是，影响评分数据的很大一部分因素和用户对物品的喜好无关，而只取决于用户或物品本身的特性。例如，对于乐观的用户来说，它的评分等级普遍偏高，而对批判性用户来说，他的评分等级普遍偏低。通常改进的基于矩阵分解的协同过滤，通过在模型中加入偏置项[14-15]来体现这些独立于用户或物品的因素，使预测精度较传统的矩阵分解有很大提升。然而这些模型并没有考虑同一个用户或项目在不同情景下的评分差异性，不能充分体现用户和项目的潜在因子矩阵客观上具有多样性的特点。通常用户或项目所在的场景不同或用户的情绪变化都会影响潜在因子特征矩阵的状态。例如，某用户在某天心情较好时可能会给某部电影的打分偏高，反之心情不好的时候可能打分偏低；同一部电影在南方上映的评分可能会比较高，而在北方上映的评分会相对偏低。这些影响用户和项目评分的信息隐藏在评分矩阵中，忽略这些重要的隐藏信息会造成模型的预测精度下降。

为了解决这一问题，本文考虑了用户和项目的潜在因子矩阵多样性这一重要的隐含特征，提出了一种基于潜在因子多样性的非负矩阵分解的协同过滤模型（non-negative matrix-factorization-diversity-based，NMFD），刻画了用户和项目的情景差异性，并且在训练模型参数时集成了单元素非负乘法更新规则和交替方向法训练原则，保证了矩阵元素的非负性，使得模型具有更强的解释性。同已有的模型相比较，该模型在减少计算和存储成本的同时，推荐的精度也有明显提升。

1 模型与学习方法

1.1 非负矩阵分解模型（NMF）

协同过滤的推荐算法通常将用户关于项目的评分矩阵作为基本的数据输入源。用 M 表示用户组成的集合，N 代表项目组成的集合，用户对项目的评分构成的评分矩阵，记为 $\boldsymbol{R}^{|M|\times|N|}$。其中：$|\cdot|$ 表示集合中所有元素的个数；$\boldsymbol{R}^{|M|\times|N|}$ 中的元素记为 $r_{m,n}$，它表示第 m 个用户对第 n 个项目的评分（偏好）。通常，某一个用户不能点击所有的项目，某个项目也不能被所有的用户点击，所以，评分矩阵 $\boldsymbol{R}^{|M|\times|N|}$ 通常是高维稀疏（HiDS）的。

在经典的基于矩阵分解的协同过滤模型中[10]，给定的评分矩阵 $\boldsymbol{R}$ 被分解为两个秩为 s 的矩阵 $\boldsymbol{P}^{|M|\times s},\boldsymbol{Q}^{|N|\times s}$。一般情况下，令 $s\ll\min\{|M|,|N|\}$，则 $\boldsymbol{R}\approx\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}$。这里的 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 被称为用户和项目的潜在因子矩阵，代表用户和项目隐藏在评分数据中的特征。基于潜在因子矩阵分解协同过滤模型的基本方法是寻找最优的 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$，使得 $\boldsymbol{R}$ 中有评分的位置（实际值）与 $\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}$ 得到的相应评分位置（预测值）之间的误差最小，目标函数为

$$\underset{\boldsymbol{P},\boldsymbol{Q}}{\operatorname{argmin}}\ Loss=\left\|\boldsymbol{R}-\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}\right\|_F^2 \tag{1}$$

式中：$Loss$ 表示损失函数；$\|\cdot\|_F$ 表示矩阵的 F 范数。

由于现实世界的评分不会出现负值，在式（1）中加入非负性约束，可以使所涉及的参数 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 在训练过程中保证非负性。在形式上可以得到

$$\begin{cases}\underset{\boldsymbol{P},\boldsymbol{Q}}{\operatorname{argmin}}\ Loss=\left\|\boldsymbol{R}-\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}\right\|_F^2\\ \text{s.t. }\boldsymbol{P},\boldsymbol{Q}\geqslant 0\end{cases} \tag{2}$$

为了防止模型在训练过程中发生过拟合，在目标函数中加入 Tikhonov 正则化，可以提高模型的性能[16]，结合式（2）可以得到最终优化的目标函数为

$$\begin{cases}\underset{\boldsymbol{P},\boldsymbol{Q}}{\operatorname{argmin}}\ Loss=\|\boldsymbol{R}-\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}\|_F^2+\dfrac{\lambda_1}{2}\|\boldsymbol{P}\|_F^2+\dfrac{\lambda_2}{2}\|\boldsymbol{Q}\|_F^2\\ \text{s.t. }\boldsymbol{P},\boldsymbol{Q}\geqslant 0\end{cases} \tag{3}$$

式中，λ_1，λ_2 为正则化系数。

为了方便后续推导更新公式，将式（3）改写成矩阵的元素形式为

$$\begin{cases}\underset{\boldsymbol{P},\boldsymbol{Q}}{\operatorname{argmin}}\ Loss=\dfrac{1}{2}\displaystyle\sum_{r_{m,n}\in\Lambda}(r_{m,n}-\hat{r}_{m,n})^2+\\ \qquad\dfrac{\lambda_1}{2}\displaystyle\sum_{k=1}^{s}p_{m,k}^2+\dfrac{\lambda_2}{2}\sum_{k=1}^{s}q_{n,k}^2\\ \text{s.t. }\{p_{m,k},q_{n,k}\}\geqslant 0;m\in M,n\in N;\\ \qquad k\in\{1,2,\cdots,s\}\end{cases} \tag{4}$$

式中：Λ 表示 $\boldsymbol{R}$ 中已知元素的集合；$p_{m,k}$，$q_{n,k}$ 分别为 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 中的元素；$\hat{r}_{m,n}=\displaystyle\sum_{k=1}^{s}p_{m,k}q_{n,k}$。

1.2 NMFD 模型

考虑到用户和项目的潜在因子矩阵多样性的特点，在 NMF 模型的基础上本文提出了基于潜在因子多样性的非负矩阵分解模型（NMFD）。在式（3）中，令 $\boldsymbol{P}=\boldsymbol{A}+\boldsymbol{B}$，其中，$\boldsymbol{A}$，$\boldsymbol{B}$ 表示受不同影响的用户潜在因子矩阵；$\boldsymbol{Q}=\boldsymbol{C}+\boldsymbol{D}$，其中，$\boldsymbol{C}$，$\boldsymbol{D}$ 表示处于不同外部环境下的项目潜在因子矩阵，因此 NMFD 模型的目标函数如下：

$$\begin{cases}\underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}}{\operatorname{argmin}}\ Loss=\dfrac{1}{2}\displaystyle\sum_{r_{m,n}\in\Lambda}\Big(r_{m,n}-\\ \qquad\displaystyle\sum_{k=1}^{s}(a_{m,k}+b_{m,k})(c_{n,k}+d_{n,k})\Big)^2+\\ \qquad\dfrac{\lambda_1}{2}\displaystyle\sum_{k=1}^{s}\left(a_{m,k}^2+b_{m,k}^2\right)+\dfrac{\lambda_2}{2}\sum_{k=1}^{s}\left(c_{n,k}^2+d_{n,k}^2\right)\\ \text{s.t. }\{a_{m,k},b_{m,k},c_{n,k},d_{n,k}\}\geqslant 0;\\ \qquad m\in M,n\in N;k\in\{1,2,\cdots,s\}\end{cases} \tag{5}$$

式中，$a_{m,k},b_{m,k},c_{n,k},d_{n,k}$ 分别表示 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$ 中的元素。

1.3 NMFD 模型的求解

首先使用加性梯度下降法（AGD），对式（5）各个变量进行更新。

$$(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D})=\underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}}{\operatorname{argmin}}\ Loss\begin{cases}a_{m,k}\leftarrow a_{m,k}-\eta_1\left(\displaystyle\sum_{r_{m,n}\in\Lambda(m)}(r_{m,n}-\hat{r}_{m,n})(-(c_{n,k}+d_{n,k}))+\lambda_1 a_{m,k}\right)\\ b_{m,k}\leftarrow b_{m,k}-\eta_2\left(\displaystyle\sum_{r_{m,n}\in\Lambda(m)}(r_{m,n}-\hat{r}_{m,n})(-(c_{n,k}+d_{n,k}))+\lambda_1 b_{m,k}\right)\\ c_{n,k}\leftarrow c_{n,k}-\eta_3\left(\displaystyle\sum_{r_{m,n}\in\Lambda(n)}(r_{m,n}-\hat{r}_{m,n})(-(a_{m,k}+b_{m,k}))+\lambda_2 c_{n,k}\right)\\ d_{n,k}\leftarrow d_{n,k}-\eta_4\left(\displaystyle\sum_{r_{m,n}\in\Lambda(n)}(r_{m,n}-\hat{r}_{m,n})(-(a_{m,k}+b_{m,k}))+\lambda_2 d_{n,k}\right)\end{cases} \tag{6}$$

式中：$\eta_1,\eta_2,\eta_3,\eta_4$分别为$a_{m,k},b_{m,k},c_{n,k},d_{n,k}$的学习率；$\hat{r}_{m,n}=\sum_{k=1}^{s}(a_{m,k}+b_{m,k})(c_{n,k}+d_{n,k})$为$r_{m,n}$的估计；$\Lambda(m)$为有评分记录的用户构成的集合；$\Lambda(n)$为被评分的项目构成的集合。

由式(6)可知，AGD更新公式中有负项存在，可能导致更新过程中出现负值，而现实世界的项目评分不会有负项出现，为了解决这个问题，根据SLF-NMU，将学习率设置成

$$\begin{cases}\eta_1=a_{m,k}\Big/\left(\sum\limits_{r_{m,n}\in\Lambda(m)}\hat{r}_{m,n}(c_{n,k}+d_{n,k})+\lambda_1 a_{m,k}\right)\\ \eta_2=b_{m,k}\Big/\left(\sum\limits_{r_{m,n}\in\Lambda(m)}\hat{r}_{m,n}(c_{n,k}+d_{n,k})+\lambda_1 b_{m,k}\right)\\ \eta_3=c_{n,k}\Big/\left(\sum\limits_{r_{m,n}\in\Lambda(n)}\hat{r}_{m,n}(a_{m,k}+b_{m,k})+\lambda_2 c_{n,k}\right)\\ \eta_4=d_{n,k}\Big/\left(\sum\limits_{r_{m,n}\in\Lambda(n)}\hat{r}_{m,n}(a_{m,k}+b_{m,k})+\lambda_2 d_{n,k}\right)\end{cases}\tag{7}$$

将式(7)代入式(6)得到最终的更新方法。

$$(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D})=\underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}}{\operatorname{argmin}}\,Loss\begin{cases}a_{m,k}\leftarrow a_{m,k}\dfrac{\sum\limits_{r_{m,n}\in\Lambda(m)}r_{m,n}(c_{n,k}+d_{n,k})}{\sum\limits_{r_{m,n}\in\Lambda(m)}\hat{r}_{m,n}(c_{n,k}+d_{n,k})+\lambda_1 a_{m,k}}\\ b_{m,k}\leftarrow b_{m,k}\dfrac{\sum\limits_{r_{m,n}\in\Lambda(m)}r_{m,n}(c_{n,k}+d_{n,k})}{\sum\limits_{r_{m,n}\in\Lambda(m)}\hat{r}_{m,n}(c_{n,k}+d_{n,k})+\lambda_1 b_{m,k}}\\ c_{n,k}\leftarrow c_{n,k}\dfrac{\sum\limits_{r_{m,n}\in\Lambda(n)}r_{m,n}(a_{m,k}+b_{m,k})}{\sum\limits_{r_{m,n}\in\Lambda(n)}\hat{r}_{m,n}(a_{m,k}+b_{m,k})+\lambda_2 c_{n,k}}\\ d_{n,k}\leftarrow d_{n,k}\dfrac{\sum\limits_{r_{m,n}\in\Lambda(n)}r_{m,n}(a_{m,k}+b_{m,k})}{\sum\limits_{r_{m,n}\in\Lambda(n)}\hat{r}_{m,n}(a_{m,k}+b_{m,k})+\lambda_2 d_{n,k}}\end{cases}\tag{8}$$

值得注意的是，如果$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，$\boldsymbol{D}$的初始值是非负的，那么经过SLF-NMU更新之后它们的值也为非负。

由于$\boldsymbol{P}$和$\boldsymbol{Q}$都进行了分解，则模型的解空间扩大，最优解容易陷入鞍点或者局部最优，从而导致模型收敛的速度变得缓慢。根据文献[13]，将交替方向法加入训练过程中可以提高模型的收敛率。训练规则为，首先将原始问题分解成几个子问题，然后按顺序解决每一个子问题。其中，每一个子问题的解依赖于上一个子问题的解，即

$$\begin{cases}\forall m\in M,k=1,2,\cdots,s,\boldsymbol{A}^{(n+1)}\overset{\text{SLF-NMU}}{\longleftarrow}\underset{\boldsymbol{A}}{\operatorname{argmin}}\,Loss\times\left(\boldsymbol{A}^{(n)},\boldsymbol{B}^{(n)},\boldsymbol{C}^{(n)},\boldsymbol{D}^{(n)}\right)\\ \forall m\in M,k=1,2,\cdots,s,\boldsymbol{B}^{(n+1)}\overset{\text{SLF-NMU}}{\longleftarrow}\underset{\boldsymbol{B}}{\operatorname{argmin}}\,Loss\times\left(\boldsymbol{A}^{(n+1)},\boldsymbol{B}^{(n)},\boldsymbol{C}^{(n)},\boldsymbol{D}^{(n)}\right)\\ \forall n\in N,k=1,2,\cdots,s,\boldsymbol{C}^{(n+1)}\overset{\text{SLF-NMU}}{\longleftarrow}\underset{\boldsymbol{C}}{\operatorname{argmin}}\,Loss\times\left(\boldsymbol{A}^{(n+1)},\boldsymbol{B}^{(n+1)},\boldsymbol{C}^{(n)},\boldsymbol{D}^{(n)}\right)\\ \forall n\in N,k=1,2,\cdots,s,\boldsymbol{D}^{(n+1)}\overset{\text{SLF-NMU}}{\longleftarrow}\underset{\boldsymbol{D}}{\operatorname{argmin}}\,Loss\times\left(\boldsymbol{A}^{(n+1)},\boldsymbol{B}^{(n+1)},\boldsymbol{C}^{(n+1)},\boldsymbol{D}^{(n)}\right)\end{cases}\tag{9}$$

分别求出$\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$的解作为第$n$次迭代的初始，$\boldsymbol{A}^{(n)},\boldsymbol{B}^{(n)},\boldsymbol{C}^{(n)},\boldsymbol{D}^{(n)},\boldsymbol{A}^{(n+1)},\boldsymbol{B}^{(n+1)},\boldsymbol{C}^{(n+1)},\boldsymbol{D}^{(n+1)}$表示它们第$n$次和第$n$+1次迭代值。

2　算法设计与分析

基于以上讨论，本文设计了NMFD算法，如

算法1所示。值得注意的是，算法1依赖4个过程，分别为Update_$\boldsymbol{A}$，Update_$\boldsymbol{B}$，Update_$\boldsymbol{C}$，Update_$\boldsymbol{D}$。由于训练过程非常相似，本文只给出了Update_$\boldsymbol{A}$的训练细节，如算法2所示。从算法1可知，基于ADM训练规则，$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，$\boldsymbol{D}$按顺序进行了更新，后一次的更新依赖于前一次的结果。

算法1 NMFD模型

设置潜在因子维数s，最大迭代次数T，收敛阈值δ，n=1。具体计算步骤如下：

Step 1 输入原始评分矩阵$\boldsymbol{R}$以及已知的元素集合Λ和M，N，$\boldsymbol{A}^{(n)}$，$\boldsymbol{B}^{(n)}$，$\boldsymbol{C}^{(n)}$，$\boldsymbol{D}^{(n)}$；

Step 2 初始化$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，$\boldsymbol{D}$，λ_1，λ_2；

Step 3 根据Update_$\boldsymbol{A}$，Update_$\boldsymbol{B}$，Update_$\boldsymbol{C}$，Update_$\boldsymbol{D}$分别计算出$\boldsymbol{A}^{(n+1)}$，$\boldsymbol{B}^{(n+1)}$，$\boldsymbol{C}^{(n+1)}$，$\boldsymbol{D}^{(n+1)}$；

Step 4 输入迭代终止条件为目标函数达到收敛阈值δ或者达到最大迭代次数T；

Step 5 令n=n+1，继续循环；

Step 6 输出潜在因子矩阵$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，$\boldsymbol{D}$。

算法2 Update_$\boldsymbol{A}$

设置最大迭代次数T，收敛阈值δ，n=1。算法具体步骤如下：

Step 1 输入原始的评分矩阵$\boldsymbol{R}$中已知的元素集合Λ和M，N，$\boldsymbol{A}^{(n)}$，$\boldsymbol{B}^{(n)}$，$\boldsymbol{C}^{(n)}$，$\boldsymbol{D}^{(n)}$；

Step 2 对任意Λ中$r_{m,n}$，根据公式$\boldsymbol{R}$=$\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}$，$\boldsymbol{P}$=$\boldsymbol{A}$+$\boldsymbol{B}$，$\boldsymbol{Q}$=$\boldsymbol{C}$+$\boldsymbol{D}$进行计算；

Step 3 结合式(8)与式(9)更新$\boldsymbol{A}^{(n+1)}$；

Step 4 输入终止迭代条件为目标函数达到收敛阈值δ或者达到最大迭代次数T；

Step 5 令n=n+1，继续循环；

Step 6 输出$\boldsymbol{A}^{(n+1)}$。

接下来考虑NMFD算法的复杂度，可以知道，在高维稀疏矩阵中，通常有$|\Lambda| \gg \max\{|M|,|N|\}$。因此，通过忽略低阶项和常数可以得到Update_$\boldsymbol{A}$的计算复杂度为

$$\Theta_{\mathrm{NMFD}} = \Theta(|M|s+|\Lambda|s+|M|s) \approx \Theta(|\Lambda|s) \tag{10}$$

而NMFD算法依赖4个过程，所以最终得到NMFD的计算复杂度为

$$C_{\mathrm{NMFD}} = \Theta((|M|+|N|)s+4n|\Lambda|s) \approx \Theta(n|\Lambda|s) \tag{11}$$

由于n和s都是正常数，因此在高维稀疏矩阵中，NMFD模型算法的复杂度与观察到的元素个数呈线性关系，所以在现实中容易实施。

3 实验设计与结果分析

3.1 评估指标及对比模型

由于本文关注的是模型预测生成数据的准确性，故采用平均绝对误差(MAE)和均方根误差(RMSE)来作为评价指标。具体表达式如下：

$$\xi_{\mathrm{MAE}} = \sum_{r_{m,n}\in\Psi} \left|r_{m,n}-\hat{r}_{m,n}\right| / |\Psi| \tag{12}$$

$$\xi_{\mathrm{RMSE}} = \sqrt{\sum_{r_{m,n}\in\Psi} (r_{m,n}-\hat{r}_{m,n})^2 / \|\Psi\|} \tag{13}$$

式中，Ψ表示验证集，并且$\Psi \cap \Lambda = \varnothing$。

从以上两个评价标准表达式可以得知，MAE和RMSE数值越小代表模型的预测精度越高。

实验对比模型为：M1(带有正则化的NMFD)和M2(不带正则化的NMFD)分别表示本文提出的带有正则化项和不带正则化项的NMFD模型，M3(带有正则化的DF NMF)和M4(不带正则化的DF NMF)分别表示基于二分解(DF)的带正则化和不带正则化的NMF模型[10]。

3.2 实验数据集

为了验证本文模型的有效性，针对以下4个数据集进行实验：

a. D1(MovieLens)。该数据集是电影MovieLens采用的0.5～5的评分等级，包含610个用户对9 724部电影的评价，创建时间从1995年1月至2015年3月，其密度为1.70%[17]。

b. D2(Filmtrust)。这是2011年从Film Trust网站获取的不完整数据集，其中包含1 508名用户和2 071部电影，评分等级为0.5～4，评分矩阵密度为1.14%[18]。

c. D3(Learning-from-sets-2019)。该数据集是2016年2月至4月从MovieLens网站收集得到的，包含854个用户对6 473部电影的评分，评分等级为0.5～5，其评分矩阵密度为0.53%[19]。

d. D4(Eachmovie)。该数据集是1997年DEC Systems Research Center运行EachMovie推荐系统18个月，在此期间记录的72 916个用户对1 628部电影的评分，评分等级为1～6。它的评分矩阵密度为9.89%[20]。

为了易于比较，在实验之前本文将数据集D1—D4映射到[0, 5]。为了消除偏差和主观因素，本文

采用 20%～80% 的测试集–训练集划分，并使用 5 折交叉验证。为了说明 NMFD 模型的有效性，训练过程重复 50 次。

3.3　对比分析

由于正则化系数对于模型性能影响较大[21]，所以在对比实验进行之前，需要对上述模型中的参数利用交叉验证方法进行敏感度实验。在实验时发现λ_1,λ_2取不同值时，实验的精度影响不明显。所以为了简化实验，在 M1 和 M3 中令 $\lambda_1=\lambda_2=\lambda$，即仅对单一的参数进行训练。

表 1 给出了模型 M1，M3 在 D1—D4 上λ的最优值。表 2 和图 1 给出了 M1—M4 在 D1—D4 上训练的 RMSE 结果和收敛曲线。表 3 和图 2 给出了 M1—M4 在 D1—D4 上训练的 MAE 结果和收敛曲线。从表 2 和图 1 可得如下结论：

a. M1 在 D1—D4 上整体表现较好，但是在不同的数据集之间存在着差异，M1 在 D1—D4 上最终收敛的 RMSE 分别为 0.2680，0.3561，0.0717，

表 1　M1, M3 在 D1—D4 上最优的正则化系数λ

Tab.1　Optimal λ of M1 and M3 on D1—D4

数据集	M1	M3
D1	0.012	0.053
D2	0.026	0.039
D3	0.018	0.069
D4	0.035	0.037

表 2　M1—M4 在 D1—D4 上训练的 RMSE

Tab.2　RMSE of M1—M4 training on D1—D4

数据集	M1	M2	M3	M4
D1	0.2680	0.3405	0.4007	0.4419
D2	0.3561	0.4170	0.4779	0.5210
D3	0.0717	0.1719	0.2304	0.2682
D4	0.9141	0.9410	0.9726	1.0015

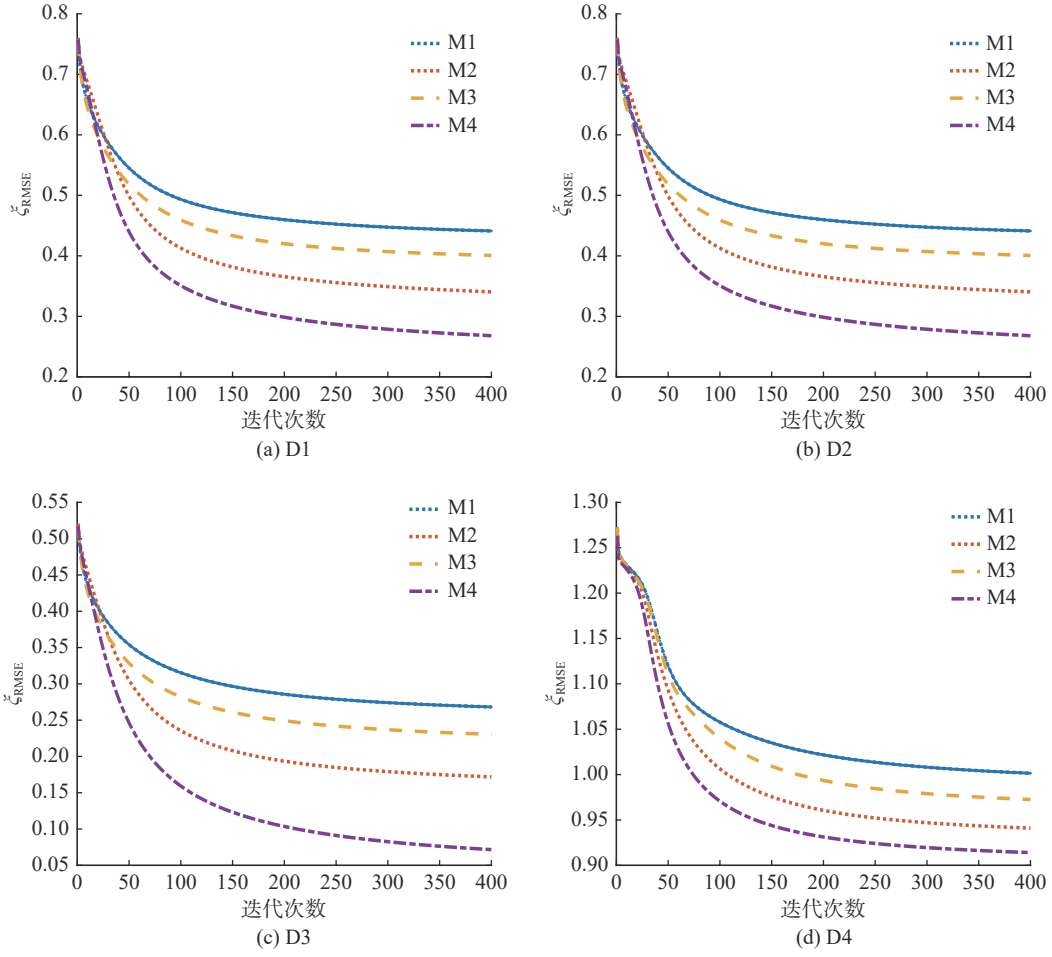

图 1　M1—M4 在 D1—D4 上训练的 RMSE 收敛曲线（$s=20$）

Fig.1　RMSE convergence curve of M1—M4 training on D1—D4 (s=20)

表 3 M1—M4 在 D1—D4 上训练的 MAE

Tab.3 MAE of M1—M4 training on D1—D4

数据集	M1	M2	M3	M4
D1	0.1631	0.2376	0.2936	0.3293
D2	0.2293	0.2944	0.3518	0.3905
D3	0.0463	0.1241	0.1744	0.2052
D4	0.7011	0.7283	0.7576	0.7833

0.9141。整体而言，M1 在 D1—D4 上的表现最好，这是因为 M1 中拥有更多的自由变量，使得模型最终的解空间扩大，从而导致精度提升。

b. 在加入 Tikhonov 正则化项之后，模型对未知数据的预测精度比不带正则化的模型有明显的提升。比如：在 D1 上，M1 最终收敛的 RMSE 为 0.2680，而 M2 最终收敛的 RMSE 为 0.3405；在 D2 上，M1 最终收敛的 RMSE 为 0.3561，而 M2 在 D1 上最终收敛的 RMSE 为 0.4170；在 D3 上，M1 最终收敛的 RMSE 为 0.0717，而 M2 最终收敛的 RMSE 为 0.1719；在 D4 上，M1 最终收敛的 RMSE 为 0.9141，而 M2 最终收敛的 RMSE 为 0.9410。相同的结论可以从图 2 和表 3 可以得到。

由于潜在因子维数 s 对算法的存储和计算效率具有比较大的影响，所以接下来将 s 从 5 增加到 100 来训练模型，训练过程中的收敛曲线由图 3 给出。从图 3 易看出，M1 的 RMSE 在 4 个数据集中都是最小的。但是由前面算法复杂度式(10)可知，s 越大，算法的复杂度就越高，训练时间随之变长，并且随着 s 的增加，模型的精度不再有所提升，如图 3(c)所示。所以本文选取 $s=20$ 来平衡预测精度和计算效率。

表 4 记录了模型 M1—M4 在数据集 D1—D4 上关于 RMSE 的收敛时间。从表 2、表 3 和表 4 可以看出，M1 和 M2 虽然增加了一定的计算负担，但是预测精度相比 M3 和 M4 有了大幅提高，更加适合在线实时进行和对计算精度要求比较高的实验。

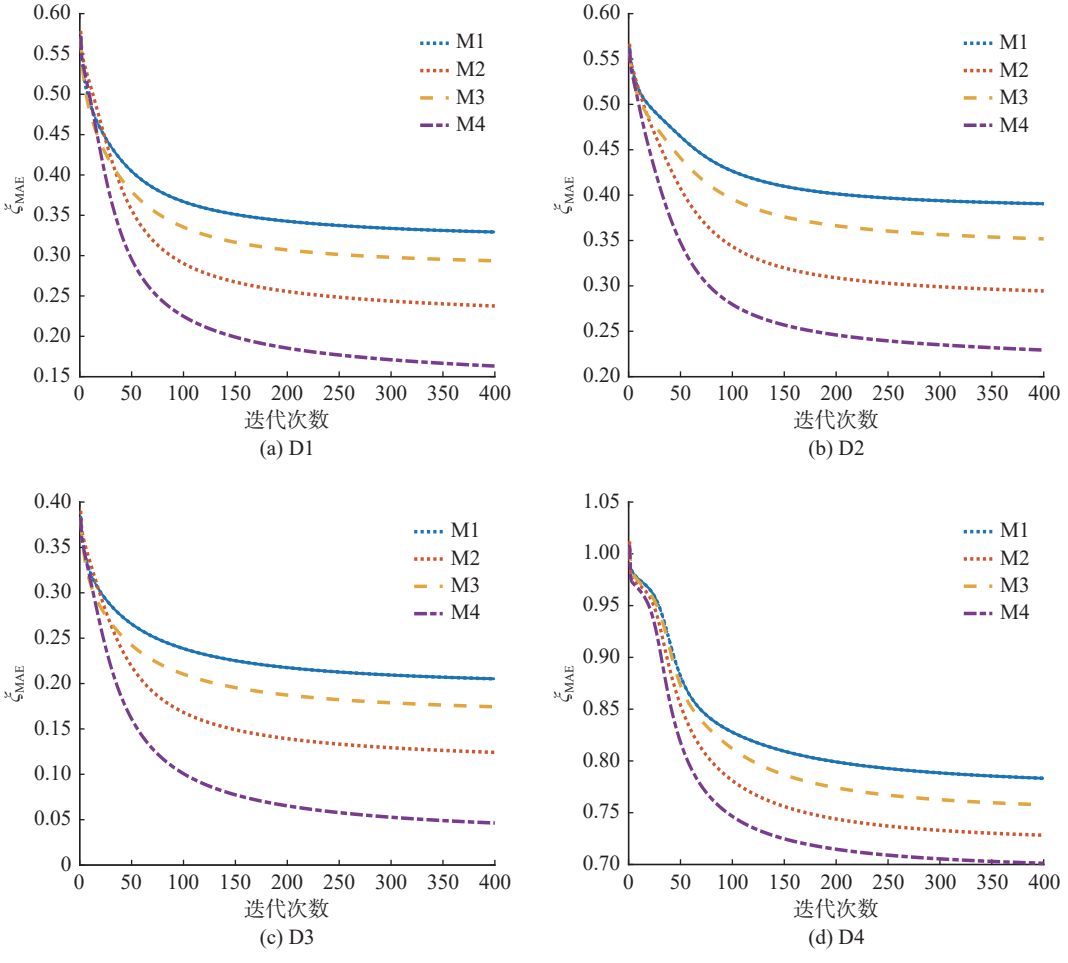

图 2 M1—M4 在 D1—D4 上训练的 MAE 收敛曲线($s=20$)

Fig.2 MAE convergence curve of M1—M4 training on D1—D4 ($s=20$)

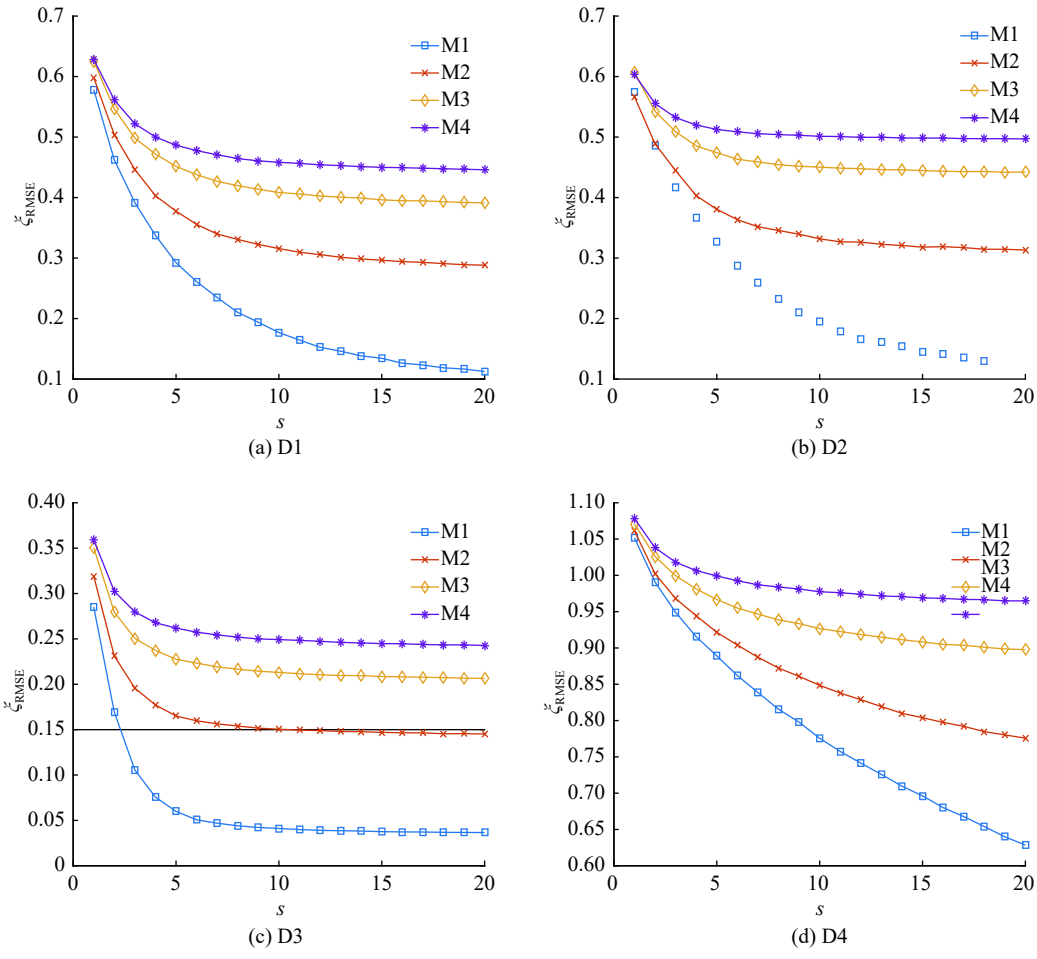

图 3　潜在因子维数 *s* 从 5 增加到 20 时 M1—M4 在 D1—D4 上训练的 RMSE 收敛曲线

Fig.3　RMSE convergence curve of M1—M4 training on D1—D4 as *s* increasing from 5 to 20

表 4　M1—M4 在 D1—D4 上训练的平均收敛时间

Tab.4　Average converge time of M1—M4 training on D1—D4

ms

数据集	M1	M2	M3	M4
D1	55614	55701	36424	36895
D2	41226	40021	28059	32483
D3	50929	58515	32920	33736
D4	66385	71139	42626	56159

4　结　论

本文从矩阵的二分解出发，通过考虑用户和项目潜在因子矩阵的多样性，提出了 NMFD 模型。针对现实工业界评分数据的非负性特点，引入了 AGD 和 SLF-NMU 算法来保证训练过程中用户和项目潜在因子的非负性。由于训练的参数较多，为了防止模型在迭代过程中陷入局部最优，又将 ADM 原则加入到算法中，提高了模型的收敛性。最后在 4 个真实的数据集上做了对比实验。实验结果表明，本文提出的带有 Tikhonov 正则化项的 NMFD 在适当牺牲一些时间的情况下，预测缺失数据精度比经典的矩阵二分解模型有显著提升。未来可以将 NMFD 模型与带偏置的 NMF 结合，对该模型进行进一步的拓展研究。

参考文献：

[1] 姚静天, 王永利, 侍秋艳, 等. 基于联合物品搭配度的推荐算法框架 [J]. 上海理工大学学报, 2017, 39(1): 45–53.
[2] 王国霞, 刘贺平. 个性化推荐系统综述 [J]. 计算机工程与应用, 2012, 48(7): 66–76.
[3] 姚曜, 赵洪利, 杨海涛, 等. 协同过滤技术研究综述 [J]. 装备指挥技术学院学报, 2011, 22(5): 81–88.
[4] 王成, 朱志刚, 张玉侠, 等. 基于用户的协同过滤算法的推荐效率和个性化改进 [J]. 小型微型计算机系统, 2016, 37(3): 428–432.
[5] 傅鹤岗, 王竹伟. 对基于项目的协同过滤推荐系统的改进 [J]. 重庆理工大学学报(自然科学), 2010, 24(9): 69–74.
[6] 王硕. 基于模型的协同过滤推荐算法研究 [D]. 北京: 北京邮电大学, 2019.
[7] 卢志茂, 冯进玫, 范冬梅, 等. 面向大数据处理的划分聚

类新方法 [J]. 系统工程与电子技术, 2014, 36(5): 1010–1015.
[8] 邹斌, 董雪梅, 付丽华, 等. 基于算法稳定的分类机器学习泛化能力的研究 [J]. 模式识别与人工智能, 2004, 17(4): 430–433.
[9] 李振波, 杨晋琪, 岳峻. 基于协同回归模型的矩阵分解推荐 [J]. 图学学报, 2019, 40(6): 983–990.
[10] 王鹏. 基于矩阵分解的推荐系统算法研究 [D]. 北京: 北京交通大学, 2015.
[11] YI B L, SHEN X X, LIU H, et al. Deep matrix factorization with implicit feedback embedding for recommendation system[J]. IEEE Transactions on Industrial Informatics, 2019, 15(8): 4591–4601.
[12] LUO X, ZHOU M C, XIA Y N, et al. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems[J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1273–1284.
[13] LUO X, ZHOU M C, LI S, et al. A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(3): 579–592.
[14] 王建芳, 刘冉东, 刘永利. 一种带偏置的非负矩阵分解推荐算法 [J]. 小型微型计算机系统, 2018, 39(1): 69–73.
[15] 毕华玲, 周微, 卢福强. 引入偏置的矩阵分解推荐算法研究 [J]. 计算机应用研究, 2018, 35(10): 2928–2931.
[16] 张航, 叶东毅. 一种基于多正则化参数的矩阵分解推荐算法 [J]. 计算机工程与应用, 2017, 53(3): 74–79.
[17] HARPER F M, KONSTAN J A. The MovieLens datasets: history and context[J]. ACM Transactions on Interactive Intelligent Systems, 2016, 5(4): 1068-1074.
[18] GUO G B, ZHANG J, YORKE-SMITH N. A novel bayesian similarity measure for recommender systems[C]//Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. Beijing: AAAI Press, 2013: 2619–2625.
[19] SHARMA M, HARPER F M, KARYPIS G. Learning from sets of items in recommender systems[J]. ACM Transactions on Interactive Intelligent Systems, 2019, 9(4): 1-26.
[20] MCJONES P. EachMovie collaborative filtering data set[J]. Dec Systems Research Center, 1997, 24(9): 57–68.
[21] LUO X, ZHOU M C, XIAO Y N, et al. Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(3): 524–537.

（编辑：丁红艺）

[上接第 136 页]

[54] SU C J, CHEN S R, JIANG H Y, et al. Ankle joint torque prediction based on surface Electromyographic and angular velocity signals[J]. IEEE Access, 2020, 8: 217681–217687.
[55] KELEŞ A D, YUCESOY C A. Development of a neural network based control algorithm for powered ankle prosthesis[J]. Journal of Biomechanics, 2020, 113: 110087.
[56] SIU H C, SLOBODA J, MCKINDLES R J, et al. Ankle torque estimation during locomotion from surface electromyography and accelerometry[C]//2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics. New York: IEEE, 2020: 80-87.
[57] COKER J, CHEN H, SCHALL M C, et al. EMG and joint angle-based machine learning to predict future joint angles at the knee[J]. Sensors, 2021, 21(11): 3622.
[58] KIM I, LEE T, LEE S H. Ankle intention detection algorithm using electromyography signal[J]. Journal of Computational Design and Engineering, 2021, 8(5): 1234–1242.
[59] ZABRE-GONZALEZ E V, RIEM L, VOGLEWEDE P A, et al. Continuous myoelectric prediction of future ankle angle and moment across ambulation conditions and their transitions[J]. Frontiers in Neuroscience, 2021, 15: 709422.
[60] CAULCRICK C, HUO W G, HOULT W, et al. Human joint torque modelling with MMG and EMG during lower limb human-exoskeleton interaction[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7185–7192.
[61] ZANGENE A R, ABBASI A, NAZARPOUR K. Estimation of lower limb kinematics during squat task in different loading using sEMG activity and deep recurrent neural networks[J]. Sensors, 2021, 21(23): 7773.
[62] LI W T, LIU K P, SUN Z B, et al. A neural network-based model for lower limb continuous estimation against the disturbance of uncertainty[J]. Biomedical Signal Processing and Control, 2022, 71: 103115.
[63] FOROUTANNIA A, AKBARZADEH-T M R, AKBARZADEH A. A deep learning strategy for EMG-based joint position prediction in hip exoskeleton assistive robots[J]. Biomedical Signal Processing and Control, 2022, 75: 103557.

（编辑：石 瑛）