# 基于改进贝叶斯概率模型的推荐算法

刘付勇 高贤强 张 著
(塔里木大学信息工程学院 阿拉尔 843300)

**摘 要** 针对现有基于矩阵分解的协同过滤推荐系统预测精度与推荐精度较低的问题,提出一种改进的矩阵分解方法与协同过滤推荐系统。首先,将评分矩阵分解为两个非负矩阵,并对评分做归一化处理,使其具有概率语义;然后,采用变分推理法计算贝叶斯概率模型实部后验的分布;最后,搜索相同偏好的用户分组并预测用户的偏好。此外,基于用户向量的稀疏性设计一种低计算复杂度、低存储成本的推荐结果决策算法。基于3组公开数据集的实验结果表明,本算法的预测性能以及推荐系统的效果均优于其他预测算法与推荐算法。

**关键词** 协同过滤,贝叶斯概率模型,变分推理,矩阵分解,评分矩阵
**中图法分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.05.052

## Improved Bayesian Probabilistic Model Based Recommender System

LIU Fu-yong GAO Xian-qiang ZHANG Zhu
(College of Information Engineering, Tarim University, Alar 843300, China)

**Abstract** Aiming at the problem that matrix factorization based collaborative filtering recommender systems perform low accuracy in prediction and recommendation, a improved matrix factorization method and collaborative filtering recommender system were proposed. Firstly, the rating matrix is factorized into two non-negative matrices, and the rating results are normalized to show probabilistic meaning. Then, variational inference is used to compute the distribution of the real posterior distribution of Bayesian model. Lastly, the user groups with the same preference are searched and the preferences of each user are predicted. Besides, a recommendation result decision algorithm with low computational complexity and low storage overhead was designed based on the sparsity of the user vectors. Three public datasets based experimental results show that the proposed algorithm has better performance than other algorithms in prediction accuracy and recommendation effect.

**Keywords** Collaborative filtering, Bayesian probabilistic model, Variational inference, Matrix factorization, Rating matrix

## 1 引言

随着移动互联网的飞速发展,个人定制服务与推荐系统已广泛应用于音乐、电视、电子书与电子商务等领域,成为了网络服务商的一项关键技术[1]。现有的推荐系统主要可分为两类:基于上下文的推荐系统[2]与基于协同过滤的推荐系统[3]。前者主要通过定量分析上下文环境下的用户信息来获取用户的偏好,而未考虑不同用户对不同上下文的认知行为与用户偏好间的内在联系,导致偏好预测的准确率不高;后者则主要受困于用户间的固有差异性、预测场景的不确定性以及预测产品的可变性等外部因素,推荐的准确率不高。

基于协同过滤的推荐系统根据用户偏好预测算法的类型分为基于内存与基于模型两类方法。前者主要采用K-近邻法预测用户偏好,优点为预测精度高,且可度量预测与推荐的可靠性较高;缺点为可扩展性较差,对大型数据库的性能较差[4]。基于模型的推荐算法则具有可扩展性强、预测速度快的优点,因此应用更为广泛[5]。

在现有的诸多模型中,基于矩阵分解的预测模型获得了较高的准确率。经典的矩阵分解法已成功地应用于推荐问题中[5],然而它们没有将分解矩阵的元素约束为非负,所以难以理解评分矩阵中每个分量与预测结果的语义。非负矩阵分解是一种新的矩阵分解方法,其将一个元素非负的矩阵分解为两个非负矩阵的乘积,这种基于基向量组合的形式具有直观的语义[6-8]。许多研究将非负矩阵分解法应用于推荐系统,获得了较好的性能,例如:基于结构投影非负矩阵分解的协同过滤算法[9]、基于非负矩阵分解的个人化推荐服务[10]。但非负矩阵分解存在一些固有缺点,如:收敛速度慢、分解结果不唯一导致难以找到全局最小点等。

本文针对评分矩阵分解问题提出了一种新的非负矩阵分解方法,为每个用户与项目设置一个关联向量,将关联向量的

到稿日期:2016-05-18 返修日期:2016-07-31 本文受国家科技支撑计划(2013BAH27F00),塔里木大学校长基金项目(TDZKQN201616),新疆南疆农业信息化研究中心项目(TSAI201402)资助。

刘付勇(1985—),男,硕士,讲师,主要研究方向为计算机网络、物联网;高贤强(1978—),男,硕士,副教授,主要研究方向为数据挖掘与计算机应用,E-mail:gaoxianqiangtlm@126.com(通信作者);张 著(1986—),男,硕士,讲师,主要研究方向为计算机教育、数字媒体。

分量值归一化处理(范围为[0,1]),使得关联向量的分量具有直观的概率语义,且有助于识别具有相同偏好的用户组。此外,本算法基于变分贝叶斯概率框架,有效地防止了过拟合问题。

## 2 基于改进矩阵分解的推荐模型

### 2.1 模型简介

首先介绍本模型的总体结构,给出算法的输入量、输出量与参数。

(1)输入。输入协同过滤推荐系统的评分矩阵 $M$,并设置以下与评分矩阵相关的符号。$N$:评分矩阵中用户的数量;$M$:评分矩阵中项目的数量;$r_{u,i}$:评分值,本文对 $r_{u,i}$ 作归一化处理,获得归一化的评分 $r^*_{u,i}$(范围为[0,1])。大多数推荐系统将用户评分范围设为1~5,此类推荐系统 $r_{u,i}\in\{1,2,3,4,5\}$ 的 $r^*_{u,i}$ 取值为(0.00,0.25,0.50,0.75,1.00)。将用户 $u$ 未对项目 $i$ 评分的情况表示为 $r_{u,i}=\cdot$。

(2)参数。考虑以下推荐系统固定的参数。1)$K\in\mathbb{O}$:表示算法的目标用户分组数量;2)$\alpha\in(0,1)$:用户组中偏好相同的用户的重叠概率,$\alpha$ 值接近0则表示用户趋向于同一个分组,$\alpha$ 值较大则表示对应用户可能属于多个分组;3)$\beta>1$:该参数对应于推理某用户偏好某个项目所需的证据数量,$\beta$ 值越大,则模型对用户是否偏好某个项目所需的证据越多。

(3)输出。输出两个矩阵:$(a_{u,k})$,$(b_{k,i})$。对于 $N\times K$ 的矩阵 $(a_{u,k})$,其中 $a_{u,k}$ 表示用户 $u$ 属于用户组 $k$ 的概率,对每个用户 $u$,可得 $\sum_{k=1}^{K}a_{u,k}=1$。对于 $M\times K$ 的矩阵 $(b_{k,i})$,其中 $b_{k,i}$ 表示用户组 $k$ 中用户偏好项目 $i$ 的概率。

通过算法的输出矩阵,可计算下面两个重要参数。

1)用户 $u$ 偏好项目 $i$ 的概率(设为 $p_{u,i}$)。其计算方法为:

$$p_{u,i}=\sum_{k=1,\cdots,K}a_{u,k}\cdot b_{k,i} \tag{1}$$

2)$u$ 对项目 $i$ 的评分(设为 $q_{u,i}$)。将 $p_{u,i}$ 转化为用户对项目 $i$ 的非归一化评分(假设评分范围为1~5),$q_{u,i}$ 的计算方法为:

$$q_{u,i}=\begin{cases}1, & \text{if } 0\leqslant p_{u,i}<0.2\\ 2, & \text{if } 0.2\leqslant p_{u,i}<0.4\\ 3, & \text{if } 0.4\leqslant p_{u,i}<0.6\\ 4, & \text{if } 0.6\leqslant p_{u,i}<0.8\\ 5, & \text{if } 0.8\leqslant p_{u,i}\leqslant 1\end{cases} \tag{2}$$

### 2.2 概率模型

图1为本文贝叶斯概率模型的示意图,其中圆形表示随机变量,箭头表示两个变量之间的依赖关系,灰色圆表示获得该随机变量值,黑色方形表示模型的参数。

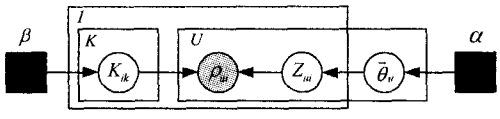


图1 本文概率方法的图形模型

根据图1,将用户对项目的评分通过以下的步骤进行仿真。

步骤1 为每个用户 $u$ 设置一个 $K$ 维向量的随机变量 $\vec{\phi}_u$(服从狄利克雷分布)的样本:$\vec{\phi}_u\sim\mathrm{Dir}(\underbrace{\alpha,\cdots,\alpha}_{K})$。

向量 $(\phi_{u,1},\cdots,\phi_{u,K})$ 表示用户 $u$ 属于各分组的概率。根据狄利克雷分布模型,$\vec{\phi}_u$ 没有实数值或整形值,并且 $\vec{\phi}_u$ 是建模 $z_{u,i}$ 所使用分类分布的共轭先验分布。

步骤2 将每个项目 $i$ 与因子 $k$ 设置为服从一个贝塔分布的随机变量 $\mathcal{K}_{i,k}$:$\mathcal{K}_{i,k}\sim\mathrm{Beta}(\beta,\beta)$。$\mathcal{K}_{i,k}$ 值代表用户组 $k$ 中用户喜爱项目 $i$ 的概率。$\mathcal{K}_{i,k}$ 是建模 $\rho_{i,u}$ 所使用二项分布的共轭先验分布。

步骤3 对每个用户 $u$ 与项目 $i(r_{u,i}\neq\cdot)$,设置下面两个随机变量。

(1)服从分类分布的随机变量 $z_{u,i}$:$z_{u,i}\sim\mathrm{Cat}(\vec{\phi}_u)$。$z_{u,i}$ 中的值 $k$ 表示用户 $u$ 对项目 $i$ 的评分。

(2)服从二项分布的随机变量 $\rho_{u,i}$(取值范围为0~$R$,实验中将 $R$ 设为4):$\rho_{u,i}\sim\mathrm{Bin}(R,\mathcal{K}_{i,z_{u,i}})$。本文使用 $\rho_{u,i}$ 值计算归一化的评分结果,计算方法为:$r^*_{u,i}=\frac{\rho_{u,i}}{R}$。

不同于其他概率模型[11],本文采用二项分布建模评分系统,主要有以下几个原因:1)二项分布与评分体系都是离散的;2)分类分布未考虑评分值之间的顺序[11];3)二项分布只有一个未知参数;4)二项分布的先验分布是贝塔分布,该分布在贝叶斯框架中代表某个用户偏好某个项目的置信度(先验知识),并且贝塔分布的取值范围为[0,1]。

使用上述算法可获得随机变量值 $\rho_{u,i}=r^*_{u,i}\cdot R$(图1的灰色圆),而其他的变量(图1的白色圆)是未知的,因此本模型的目标则变为搜索未知随机变量的分布。

### 2.3 模型推理(变分推理)

基于上文的概率模型以及评分矩阵 $M$,决定未知随机变量的条件概率分布。

本文采用变分推理法计算实部后验分布的方法分布,原因在于本模型的结构为共轭指数形式(变量服从分布指数族的分布),所以使用文献[12]的框架将实部的后验分布 $p(\vec{\phi}_u,\mathcal{K}_{i,k},z_{u,i}|\rho_{u,i})$ 近似为分布 $q(\vec{\phi}_u,\mathcal{K}_{i,k},z_{u,i})$。

$$q(\vec{\phi}_u,\mathcal{K}_{i,k},z_{u,i})=\prod_{u=1}^{N}q_{\vec{\phi}_u}(\vec{\phi}_u)\prod_{i=1}^{M}\prod_{k=1}^{K}q_{\mathcal{K}_{i,k}}(\mathcal{K}_{i,k})\prod_{r_{u,i}\neq\cdot}q_{z_{u,i}}(z_{u,i})$$

其中,各项描述如下:

(1)$q_{\vec{\phi}_u}(\vec{\phi}_u)$,$q_{\mathcal{K}_{i,k}}(\mathcal{K}_{i,k})$,$q_{z_{u,i}}(z_{u,i})$ 具有以下的分布[12]。

a)在已知评分矩阵 $M$ 的情况下,$\vec{\phi}_u$ 的近似条件概率分布同时服从狄利克雷分布:

$$q_{\vec{\phi}_u}(\vec{\phi}_u)\sim\mathrm{Dir}(\gamma_{u,1},\cdots,\gamma_{u,K}) \tag{3}$$

其中,$\gamma_{u,1},\cdots,\gamma_{u,K}$ 是需学习的参数。

b)在已知评分矩阵 $M$ 的情况下,$\mathcal{K}_{i,k}$ 的近似条件概率分布同时服从贝塔分布:

$$q_{\mathcal{K}_{i,k}}(\mathcal{K}_{i,k})\sim\mathrm{Beta}(\in^{+}_{i,k},\in^{-}_{i,k}) \tag{4}$$

其中,$\in^{+}_{i,k}$,$\in^{-}_{i,k}$ 是需学习的参数。

c)在已知评分矩阵 $M$ 的情况下,$z_{u,i}$ 的条件概率分布同时服从分类分布:

$$q_{z_{u,i}}(z_{u,i})\sim\mathrm{Cat}(\lambda_{u,i,1},\cdots,\lambda_{u,i,K}) \tag{5}$$

其中，$\lambda_{u,i,k}$是需学习的参数，因为 $q_{z_{u,i}}(z_{u,i})$为分类分布，所以这些参数必须满足：$\lambda_{u,i,1}+\cdots+\lambda_{u,i,K}=1$。

(2)上述参数 $\gamma_{u,k}$，$\in^{+}_{i,k}$，$\in^{-}_{i,k}$，$\lambda_{u,i,k}$必须满足以下条件[7]：

$$\gamma_{u,k}=\alpha+\sum_{\{i|r_{u,i}\neq\cdot\}}\lambda_{u,i,k} \tag{6}$$

$$\in^{+}_{i,k}=\beta+\sum_{\{u|r_{u,i}\neq\cdot\}}\lambda_{u,i,k}\cdot r^{+}_{u,i} \tag{7}$$

$$\in^{-}_{i,k}=\beta+\sum_{\{u|r_{u,i}\neq\cdot\}}\lambda_{u,i,k}\cdot r^{-}_{u,i} \tag{8}$$

$$\lambda'_{u,i,k}=\exp(\Psi(\gamma_{u,k})+r^{+}_{u,i}\cdot\Psi(\in^{+}_{i,k})+r^{-}_{u,i}\cdot\Psi(\in^{-}_{i,k})-R\cdot\Psi(\in^{+}_{i,k}+\in^{-}_{i,k}))$$

$$\lambda_{u,i,k}=\frac{\lambda'_{u,i,k}}{\lambda'_{u,i,1}+\cdots+\lambda'_{u,i,K}} \tag{9}$$

其中，$\Psi$定义为伽马函数的对数导数：$\Psi(x)=(\ln\Gamma(x))'=\frac{\Gamma'(x)}{\Gamma(x)}$，$r^{+}_{u,i}=\rho_{u,i}=R\cdot r^{*}_{u,i}$，$r^{-}_{u,i}=R-\rho_{u,i}=R\cdot(1-r^{*}_{u,i})$。

## 2.4　模型计算

上文方法计算了$\{\gamma_{u,k},\in^{+}_{i,k},\in^{-}_{i,k},\lambda_{u,i,k}\}$，决定自由参数的值，即可直接计算用户的偏好。

(1)搜索相同偏好的用户分组

a)根据式(3)条件分布的期望值，用户 $u$ 属于分组 $k$ 的概率为 $a_{u,k}$，则 $a_{u,k}$的计算公式为：

$$a_{u,k}=P_{(u\in k)}=\frac{\gamma_{u,k}}{\sum_{j=1,\cdots,K}\gamma_{u,j}} \tag{10}$$

可将 $a_{u,k}$的值视为用户 $u$ 所属分组 $k$ 的软隶属函数，因此可直观地看出，对于每个用户 $u$，$\sum_{k=1}^{K}a_{u,k}=1$。

b)根据式(4)条件分布的期望值，分组 $k$ 中偏好项目 $i$ 的用户比例 $b_{k,i}$为：

$$b_{k,i}=P(u\Rightarrow i|u\in k)=\frac{\in^{+}_{i,k}}{\in^{+}_{i,k}+\in^{-}_{i,k}} \tag{11}$$

(2)用户偏好的预测

首先需要获得条件概率分布 $\rho_{u,i}$，再根据推荐系统所支持评分的数量设置该随机变量的取值区间为$[0,R-1]$。将 $\rho_{u,i}$归一化处理获得随机变量$\frac{1}{R}\rho_{u,i}$，其取值范围为$[0,1]$，表示用户偏好一个项目的概率。可按如下方法计算随机变量$\frac{1}{R}\rho_{u,i}$的期望值：

$$p_{u,i}=E(\frac{1}{R}\rho_{u,i}|M)=\frac{1}{R}E(\rho_{u,i}|M)=\sum_{k=1}^{K}a_{u,k}b_{k,i}$$

## 2.5　自由参数计算算法

从式(6)－式(9)可以看出，$\{\gamma_{u,k},\in^{+}_{i,k},\in^{-}_{i,k},\lambda_{u,i,k}\}$中自由参数之间具有相互依赖的关系，所以无法直接计算获得。本文使用坐标上升算法计算这些参数，本算法迭代地更新这些自由参数的值直至收敛(根据文献[13]的理论)。

**算法 1**　基于坐标上升算法的自由参数计算方法

输入：M，α，β

输出：$\gamma_{u,k}$，$\in^{-}_{i,k}$，$\in^{+}_{i,k}$

1. 随机地初始化 $\gamma_{u,k}$，$\in^{-}_{i,k}$，$\in^{+}_{i,k}$；
2. WHILE (变化明显) DO
3.　FOREACH u DO
4.　　FOREACH i DO //训练集中 u 所评分的项目
5.　　FOREACH k DO
6.　　计算式(9)；
7.　FOREACH u DO
8.　　FOREACH k DO
9.　　计算式(6)；
10.　FOREACH i DO //训练集中 u 所评分的项目
11.　　FOREACH k DO
12.　　计算式(7)和式(8)；
13. 计算式(10)和式(11)；

## 2.6　模型参数的效果

下面对本模型的参数做详细分析。1)参数 $K$：该参数表示预测评分所使用的用户分组总数量。2)参数 $\alpha\in(0,1)$：该参数与用户向量$\vec{a}_u$ 的稀疏程度有关。本模型的目标为：$\alpha$ 值越低，向量$\vec{a}_u$ 越稀疏。本文考虑用户向量的稀疏程度，通过交叉验证法决定参数 $\alpha$ 的值。3)参数 $\beta$：该参数表示模型中某用户是否偏好某项目所需的证据数量，$\beta$ 值越大，所需证据越多。$\beta$ 值也使用交叉验证法决定。

## 2.7　模型性能和复杂度的分析与改进方案

本模型中用户偏好的预测按如下方法计算：

$$p_{u,i}=\sum_{k=1,\cdots,K}a_{u,k}\cdot b_{k,i} \tag{12}$$

因为需对每个用户与每个项目计算用户偏好，所以对于大型数据库的计算成本极大。结合本模型的特点，本文设计了 $p_{u,i}$的快速计算法。

假设一个阈值 $S$，如果某个用户 $u$ 对项目 $i$ 的偏好 $p_{u,i}>S$，则认为 $i$ 对 $u$ 是可推荐的。对于经典的矩阵分解，$p_{u,i}$代表用户 $u$ 对项目 $i$ 的预测评分，对于评分范围$\{1,\cdots,5\}$的评分系统，一般将 $S$ 设为 4，所以归一化的 $S$ 值一般设为 0.8。本模型的 $p_{u,i}$与 $S$ 具有概率语义：如果估算的概率 $p_{u,i}$高于 $S$，则认为项目 $i$ 对用户 $u$ 是可推荐的。

因为本模型中 $0\leqslant a_{u,k}\leqslant 1$，$0\leqslant b_{k,i}\leqslant 1$，$\sum_{k=1}^{K}a_{u,k}=1$，同时$\vec{a}_u$ 应较为稀疏，所以本文设计了一种新方法来检测 $i$ 对用户 $u$ 是否为可推荐的。算法 2 输出一个二元值，表示项目 $i$ 对用户 $u$ 是否为可推荐的。

**算法 2**　改进的推荐结果输出算法

输入：用户 u，项目 i，阈值 S

输出：布尔值(表示是否可推荐)

1. sum=0，p=0；
2. FOREACH k=1～K DO
3.　sum=sum+$a_{u,k}$；p=p+$a_{u,k}\cdot b_{k,i}$；
4.　IF (p>S) THEN return 可推荐；
5.　IF (p+(1−sum)<S) THEN return 不可推荐

算法 2 的改进主要体现在第 4 行与第 5 行。1)第 4 行：因为 $a_{u,k}\geqslant 0$，$b_{u,k}\geqslant 0$，所以式(12)的和值随着 $k$ 的改变而提高；如果式(12)的和值高于 $S$，则认为 $p_{u,i}>S$。2)第 5 行：因为 $a_{u,k}\geqslant 0$ 且 $b_{u,k}\leqslant 1$，可直接获得 $p_{u,i}$的上界：

$$\begin{aligned}p_{u,i}&=\underbrace{a_{u,1}\cdot b_{1,i}+a_{u,2}\cdot b_{2,i}+\cdots}_{p}+a_{u,k+1}\cdot b_{k+1,i}+\cdots+a_{u,K}\cdot b_{K,i}\\&\leqslant p+a_{u,k+1}\cdot 1+\cdots+a_{u,K}\cdot 1\\&=p+(1-\sum_{t=1}^{k}a_{u,k})\\&=p+(1-sum)\end{aligned}$$

所以，如果 $p+(1-sum)<S$，则认为 $p_{u,i}<S$。

算法 2 无需对所有项目计算式(12),所以计算速度快于传统计算方法。

此外,本文针对一般情况下 $p_{u,i}$ 值的计算提出快速计算方法。因为向量 $a_u$ 较为稀疏,所以 $a_{u,k}$ 的大多数分量值接近 0。为每个用户 $u$ 设置两个列表:1)$k_{u,n}$,即 $\vec{a}_u$ 的 $n$ 个最高分量对应的因子 $k$;2)$a'_{u,n}$,即 $\vec{a}_u$ 的 $n$ 个最高分量,$a'_{u,n}=a_{u,k_{u,n}}$。

因为 $a_u$ 具有稀疏性,所以仅需保存 $a_{u,k}$ 中重要的分量,最终决定项目对用户是否为可推荐的算法如算法 3 所示。

**算法 3** 考虑稀疏性的推荐结果输出算法

输入:用户 u,项目 i,阈值 S

输出:布尔值(表示是否可推荐)

1. sum=0,p=0;
2. FOREACH n=1~$N_u$
3. sum=sum+$a'_{u,n}$;k=$k_{u,n}$;p=p+$a'_{u,n}$ · $b_{k,i}$;
4. IF (p>S) THEN return 可推荐;
5. IF (p+(1−sum)<S) THEN return 不可推荐

算法 2 和算法 3 极为相似,但算法 3 考虑了向量 $\vec{a}_u$ 的稀疏特点,该算法仅需保存 $N_u-K$ 个分量,并且最多迭代 $N_u$ 次,而非 $K$ 次。综上所述,算法 3 明显降低了算法的计算成本与存储成本。

## 3 实验与结果分析

为了横向地评估本算法的推荐系统,将本算法与文献[14-16]中的方法进行比较,统计预测准确率与推荐准确率两项指标。使用数据集 MoveLens 1M,MovieLens 10M 与 NetFlix 作为实验数据库,将数据集分为两个子集:80%作为训练集,20%作为测试集合。每组实验重复 20 次,计算各性能指标的均值作为实验结果。本算法选择文献[5]的参数:$\alpha=0.8,\beta=5,K=6$。

### 3.1 预测准确率

采用 MAE 作为预测准确率性能指标。MAE 计算训练数据的平均绝对误差:

$$MAE=\frac{\sum_{(u,i)\in T}|r_{u,i}-q_{u,i}|}{|T|}$$

考虑评分中的相关误差:相关误差出现在用户预测其偏好的项目($q_{u,i}\geqslant 4$)以及用户真实偏好的项目($r_{u,i}\geqslant 4$)中。在推荐系统中,当 $r_{u,i}=2$ 时,如果系统预测 $q_{u,i}=1$,该错误并不明显;当 $r_{u,i}=3$ 时,如果系统预测 $q_{u,i}=4$,则为一个重大的错误。为了纳入该情况,定义以下的度量指标(约束的平均绝对误差):

$$CMAE=\frac{\sum_{(u,i)\in T'}|r_{u,i}-q_{u,i}|}{|T'|}$$

其中,$T'=\{(u,i)\in T|q_{u,i}\geqslant 4\vee r_{u,i}\geqslant 4\}$。

此外,实验采用 0-1 损失函数评估各算法的预测准确率,即根据是否为可推荐的预测来为项目分类。0-1 损失指标评价了可推荐预测与不可推荐预测之间的错误数量。0-1 损失函数的计算方法为:

$$0\_1\_loss=\frac{\sum_{(u,i)\in T}h(r_{u,i},q_{u,i})}{|T|}$$

其中,$h(r_{u,i},q_{u,i})=\begin{cases}0, & r_{u,i}\geqslant 4\wedge q_{u,i}\geqslant 4\\ 0, & r_{u,i}<4\wedge q_{u,i}<4\\ 1, & \text{其他情况}\end{cases}$

表 1 比较了本算法与其他几种推荐算法的性能,可看出本算法的 MAE 值低于经典矩阵分解法获得的 MAE 值,此外,本技术的优势在于本算法的计算结果具有语义。

表 1 几种预测算法的预测准确率结果

| 算法 | MoveLens 1M | | | MovieLens 10M | | | NetFlix | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE | CMAE | 0-1 损失 | MAE | CMAE | 0-1 损失 | MAE | CMAE | 0-1 损失 |
| KNN 皮尔森相关性 | 0.821 | 0.720 | 0.423 | 0.842 | 0.741 | 0.433 | 0.835 | 0.735 | 0.469 |
| KNN JMSD[14] | 0.741 | 0.690 | 0.392 | 0.752 | 0.701 | 0.403 | 0.742 | 0.695 | 0.421 |
| 文献[15] | 1.241 | 1.100 | 0.462 | 1.352 | 1.234 | 0.483 | 1.282 | 1.106 | 0.438 |
| 文献[16] | 0.794 | 0.764 | 0.392 | 0.822 | 0.787 | 0.403 | 0.802 | 0.761 | 0.468 |
| 经典矩阵分解[5] | 0.684 | 0.574 | 0.384 | 0.692 | 0.587 | 0.396 | 0.692 | 0.571 | 0.378 |
| 本算法 | 0.664 | 0.526 | 0.272 | 0.676 | 0.547 | 0.284 | 0.666 | 0.531 | 0.248 |

在 CMAE 指标方面,本方法明显优于经典的矩阵分解法,说明本方法预测的相关评分远好于经典的矩阵分解法。此外,本方法在 0-1 损失指标方面获得了理想的结果。

### 3.2 推荐准确率

IMULT 算法[17]是另一种基于矩阵分解的协同过滤推荐算法,已证明其性能优于多个主流的矩阵分解协同过滤推荐系统。本文将 IMULT 作为对比算法来横向评估本算法的性能。通过精度与召回率对推荐系统的推荐结果质量进行评价。推荐系统中精度与召回率的意义可描述为:假设 $N$ 是为每个用户推荐的数量;假设 $R$ 是训练集中未被 $u$ 评分的项目集合,$R$ 表示推荐系统为每个用户提供的 $N$ 个推荐集合;假设 $S$ 是测试集中各用户高评分值(高于 4)的项目集合,集合 $S$ 代表应该推荐给用户的项目集合。

召回率指标定义为正确推荐的项目占应被推荐项目总数的百分比:

$$Recall=\frac{|R\cap S|}{|S|}\cdot 100$$

精度指标定义为正确推荐的项目占项目推荐总数的百分比:

$$precision=\frac{|R\cap S|}{|R|}\cdot 100$$

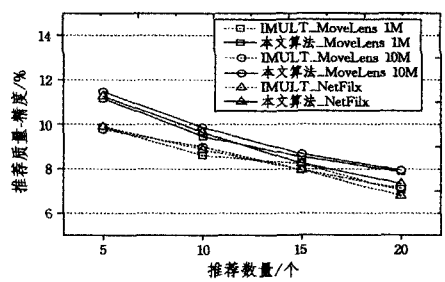


图 2 不同推荐数量下两种推荐算法的精度结果

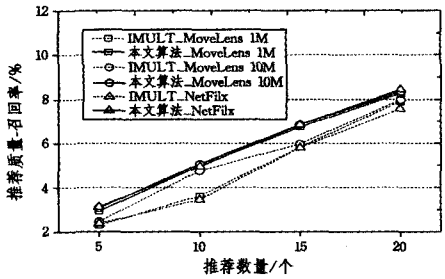


图 3　不同推荐数量下两种推荐算法的召回率结果

图 2 与图 3 所示为不同推荐数量下两种推荐算法的精度与召回率指标结果，可以看出本算法的精度与召回率结果优于 IMULT 算法，主要原因在于本方法对于中级别评分（评分 3）的预测效果好于 IMULT 算法。

**结束语**　针对现有基于矩阵分解的协同过滤推荐系统的预测精度与推荐精度较低的问题，本文提出了一种新的非负矩阵分解方法，该方法为每个用户与项目设置一个关联向量，并将关联向量的分量值进行归一化处理（范围为[0,1]），使得关联向量的分量具有直观的概率语义，且有助于识别具有相同偏好的用户组。此外，本算法基于变分贝叶斯概率框架，有效地防止了过拟合问题。最终利用用户向量的稀疏性，为推荐系统设计了低计算复杂度与低存储开销的推荐决策算法。

## 参 考 文 献

[1] XU H L, WU X, LI X D, et al. Comparison Study of Internet Recommendation System[J]. Journal of Software, 2009, 20(2): 350-362. (in Chinese)
许海玲，吴潇，李晓东，等. 互联网推荐系统比较研究[J]. 软件学报，2009，20(2)：350-362.

[2] GAO Q L, GAO L, YANG J F, et al. A Preference Elicitation Method Based on Users' Cognitive Behavior for Context-Aware Recommender System[J]. Chinese Journal of Computers, 2015 (9): 1767-1776. (in Chinese)
高全力，高岭，杨建锋，等. 上下文感知推荐系统中基于用户认知行为的偏好获取方法[J]. 计算机学报，2015(9)：1767-1776.

[3] LI X, LI S W. Research on collaborative filtering algorithm of bipartite network oriented to personal recommendation system [J]. Application Research of Computers, 2013, 30(7): 1946-1949. (in Chinese)
李霞，李守伟. 面向个性化推荐系统的二分网络协同过滤算法研究[J]. 计算机应用研究，2013，30(7)：1946-1949.

[4] XIA L M, ZHAO Y D, PENG D L, et al. Recommendation Research Based on Improved URP Model and K Nearest Neighbors [J]. Computer Science, 2013, 40(6): 276-278. (in Chinese)
夏利民，赵业东，彭东亮，等. 基于改进 URP 模型和 K 近邻的推荐研究[J]. 计算机科学，2013，40(6)：276-278.

[5] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. IEEE Computer, 2009, 42 (8): 30-37.

[6] WU J, CHEN L, FENG Y, et al. Predicting Quality of Service for Selection by Neighborhood-Based Collaborative Filtering [J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2013, 43(2): 428-439.

[7] GUAN N, TAO D, LUO Z, et al. Online Nonnegative Matrix Factorization With Robust Stochastic Approximation[J]. IEEE Transactions on Neural Networks & Learning Systems, 2012, 23(7): 1087-1099.

[8] WU Q, TAN M, LI X, et al. NMFE-SSCC: Non-negative matrix factorization ensemble for semi-supervised collective classification[J]. Knowledge-Based Systems, 2015, 89: 160-172.

[9] JU B, QIAN Y T, YE M C. Collaborative filtering algorithm based on structured projective nonnegative matrix factorization [J]. Journal of Zhejiang University(Engineering Science), 2015, 49(7): 1319-1325. (in Chinese)
居斌，钱沄涛，叶敏超. 基于结构投影非负矩阵分解的协同过滤算法[J]. 浙江大学学报(工学版)，2015，49(7)：1319-1325.

[10] YU Q. CloudRec: a framework for personalized service Recommendation in the Cloud[J]. Knowledge & Information Systems, 2014, 43(2): 1-27.

[11] GHAHRAMANI Z, BEAL M J. Propagation Algorithms for Variational Bayesian Learning[J]. Advances in Neural Information Processing Systems, 2001, 13: 507-513.

[12] BISHOP C M. Pattern Recognition and Machine Learning (Information Science and Statistics)[M]. New York: Springer-Verlag, 2006.

[13] BEN-TAL A, NEMIROVSKI A. Robust Convex Optimization [J]. Mathematics of Operations Research, 2013, 23(4): 769-805.

[14] BOBADILLA J, SERRADILLA F, BERNAL J. A new collaborative filtering metric that improves the behavior of recommender systems[J]. Knowledge-Based Systems, 2010, 23(6): 520-528.

[15] WANG C, BLEI D M. Collaborative topic modeling for recommending scientific articles[C] // ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011: 448-456.

[16] KANG J H, LERMANK K. LA-CTR: a limited attention collaborative topic regression for social media[C] // AAAI Conference on Artificial Intelligence. 2013.

[17] RANJBAR M, MORADI P, AZAMI M, et al. An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems[J]. Engineering Applications of Artificial Intelligence, 2015, 46(PA): 58-66.

[18] FU H G, WANG Z W. Improvem ent of Item-Based Collaborative Filtering Algorithm[J]. Journal of Chongqing University of Technology(Natural Science), 2010, 24(9): 69-74(in Chinese)
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(自然科学版)，2010，24(9)：69-74.