一种对数据集稀疏度不敏感的协同推荐新方法

蔡国永　吕瑞
(桂林电子科技大学广西可信软件重点实验室 桂林 541004)
(ccgycai@guet.edu.cn)

A Novel Collaborative Recommendation Method with Low Sensitivity to Data Sparseness

Cai Guoyong and Lv Rui
(Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology Guilin, 541004)

Abstract In the field of recommendation systems, most studies on the sparseness are based on static datasets. However, the datasets in practical application are dynamic and there exist at least two features: one is the increasing scale of User-Item matrix for users and items join into the datasets continuously; another is the increasing degree of sparseness. As a result, the accuracy of traditional recommendation methods will reduce gradually with the datasets becoming sparser. Based on the above consideration, a novel collaborative recommendation method with low sensitivity to different degrees of sparse datasets is proposed to meet the need of practical and dynamic datasets. This method incorporates tag information and factor analysis method to discover the most similar top-N users based on the similarity of users' inner idiosyncrasies. Based on the most similar top-N users discovered, an improved collaborative filtering method is designed. Extensive experiments are conducted to compare the proposed method with other state-of-the-art collaborative filtering and the matrix factorization methods. The results demonstrate that our proposed method can achieve better accuracy and has a low sensitivity to different degrees of sparse datasets.

Key words recommendation system；dataset sparseness；tag system；factor analysis； rating prediction

摘　要 在推荐系统领域中，针对数据集稀疏性问题的研究大都建立在静态数据集的基础上，而实际工业应用中的数据集则往往是动态的并且具有以下两个明显的特征：1) User-Item 矩阵维度逐渐增大；2) 稀疏程度不断增加．因此，传统的依赖固定稀疏程度数据集的推荐算法的准确率则会随着数据集稀疏度的增加而下降．基于以上考虑，针对稀疏度动态变化的工业数据集的特征，提出一种准确率高而且对数据集稀疏程度敏感性较低的方法．该方法结合了少量的标签信息并利用了因子分析的方法，通过建立一种特殊的因子模型从而为用户构建一个新的 User-Factor 向量（用户−因子向量），并基于新的 User-Factor 向量为目标用户进行“邻居发现”和评分预测．最后，我们通过大量对比实验证明了本文中的方法在处理工业应用中的数据集时算法总是能够保持较高且稳定的准确率．

关键词 推荐系统；稀疏数据集；标签系统；因子分析；评分预测

中图法分类号 TP301.6

基于用户和项的 User-Item 评分矩阵模型（UI 矩阵模型）是推荐系统领域里最为经典和有效的数据模型．众多的现实推荐问题都可以通过转化为 UI 评分矩阵进行建模和处理．然而，由于 User 和 Item 的数据规模通常处于一个较大的数量级，而 UI 矩阵中有效的评分数据往往很少（不足 5%），这就产生了评分数据集的稀疏性问题[1,2]，例如 MovieLens 数据集的稀疏度为 4.2%，Netflix 为 1.2%，Bibsonomy 为 0.35%等．而现在能够处理稀疏数据集的方法常被认为是更有应用价值的．因此，如何使推荐算法在评分数据极度稀疏的情况下仍然保持较高且稳定的准确率是推荐系统领域面临的一个非常重要的现实问题．

基金项目：国家自然科学基金项目（63061039）；广西可信软件重点实验室项目（kx201202）

近些年来，针对 UI 矩阵的稀疏性问题，不同学者从不同方面进行了研究，提出了一些改善数据稀疏性的方法．其思路主要涉及以下两个方面：（1）增加信息源，即通过在 UI 矩阵中引入更多的相关信息来弥补数据稀疏方面的不足[3,4,5]；（2）UI 矩阵挖掘，如利用已有的数据通过矩阵分解[6,7]，线性拟合[8]等方法建立对未知评分数据的预测模型以及利用扩散[9]、迭代寻优[10]、转移相似性[11]等方法发掘用户和物品潜在的关联关系等．然而，这些方法同样也存在着一些不足，例如涉及的辅助信息太多、推荐算法时间复杂度较高以及处理的数据集往往是静态的等等，而且随着数据集规模和稀疏程度的增加，这些方法的准确率会逐渐降低，对一些在实际应用中动态增长的数据集不能产生良好的效果．因此，设计一种与工业应用中数据集特征相适应的方法具有重要意义．

因子分析方法是一种在多个领域得到广泛应用的潜在因素分析方法，该方法主要通过分析某一具体行为或现象的一些外在指标信息从而发现隐藏在这些外在指标信息背后起决定性作用的内在因子．标签系统则是一种对网络资源进行管理、标记、分享以及识别的工具．用户对网络资源的理解和关注可以通过其对资源标签的选择来体现．用户对某一标签关注的越多，则说明该用户具有的此标签特征就越明显．通过将商品抽象为标签，建立用户对标签的偏爱或兴趣分布，一方面可以减弱原始数据集稀疏程度对推荐算法的影响，另一方面可以利用因子分析的方法进行深层次的挖掘，从而获得影响用户兴趣分布的内在决定性因子，并根据这些因子对用户进行邻居发现或者聚类．综合以上考虑，基于工业应用中数据集动态增长和稀疏程度不断增加的事实，本文提出了一种对数据集稀疏性敏感度较低且具有较高算法准确率的评分预测方法．该方法仅利用了少量额外的标签信息，并通过因子分析的方法为每个用户建立新的 User-Factor 向量，然后根据这些向量发现目标用户的邻居用户和基于 Top-N 邻居进行评分预测．

1　相关研究

与数据稀疏性密切相关的推荐方法主要包括传统的协同过滤方法和基于模型的方法．

协同过滤方法是一种典型的基于集体智慧的方法．它利用已有用户群过去的行为或意见预测当前用户可能感兴趣的内容，其核心是定义和发现相似的用户．定义用户相似性的方法除了传统的计算用户向量间的距离外（如皮尔逊距离，Spearman 相关系数[12]，曼哈顿距离，马氏距离等[13]）还包括诸如用户聚类[14]，图分割技术[15]等数据挖掘领域的方法．在 User-Item 评分矩阵中，评分数据的极度稀疏导致了不同用户之间仅有少量的共同评分项，基于较少共同评分的用户相似度计算一方面准确率不高，另一方面可能导致推荐结果的不确定性．近些年来，学者们开展了大量的研究工作，其关注点在于通过一些其它途径获取一些额外的相关信息，从而弥补用户和物品数据信息的不足．由此也产生了一些新的推荐方法，如基于用户社交关系的商品推荐[16]，基于话题模型建模的跨领域交叉推荐[17]，基于上下文感知[18]和自适应能力[19]的动态推荐以及其它的如借助信任模型[20]，传播模型[21]，线性回归模型[22]，贝叶斯模型[23]，Markov 决策过程[24]，Gibbs 抽样[25]，标签系统[26]，关联规则[27]，知识推理[28]等进行相似用户关系挖掘的推荐方法．

基于模型的方法主要有二部图网络结构模型[29]和矩阵因子分解模型[30]，这两种模型方法主要从 UI 矩阵挖掘的角度改善数据的稀疏性．二部图网络结构模型通过在二部图上模拟物质扩散[31]、热传导[32]等复杂网络动力学过程来对用户进行个性化推荐．此类算法及其改进算法不仅在物品推荐的准确性和复杂性上明显优于经典的协同过滤算法，而且在推荐的多样性、长尾发掘、个性化程度等方面均有较大的提升[33]．矩阵分解模型是近些年来非常成功和主流的推荐算法模型．其主要思想是从评分模式中抽取一组少量的潜在特征因子，并根据这些因子建立用户和物品的特征向量，然后使用梯度下降[34]等方法求解特征向量并实现原稀疏评分矩阵中未知评分项的评分预测．由于矩阵分解方法在 2009 年 Netflix 竞赛中取得的优异成绩，人们对矩阵分解方法在推荐系统中的应用给予了极大关注，也提出了一些效果良好的算法及改进，如 SGD[35]，SVD++[36]，PMF[37]，Bias LFM[38]以及开发了一些专门的矩阵分解分析工具，如上海交通大学 Apex 实验室的 SVDFeature[39]和 Steffen Rendle 等人开发的 LibFM[40]．

本文提出的方法将结合上述两类方法的思想，一方面将结合物品的标签等额外信息，另一方面将结合一种特殊的潜在因子抽取方法（因子分析方法），从而进一步提高推荐算法在评分预测方面的准确性和应对不同稀疏程度的数据集时算法效果的稳定性．

2　预备知识

2.1 标签系统

标签系统是一种应用非常广泛和有效的网络资源注释，组织和分享工具．通过标签系统我们可以为网络资源定义各种不同的标签，所有的这些标签又可以构成一种非结构化的协同分类策略（即大众分类法 folksonomy），不同的网络资源根据用户定义的标签

的不同而被划归为不同的类别. 分析用户的标签行为以及物品的标签构成可以帮助我们发现相似的用户以及物品. 标签系统已经被应用在推荐算法设计的多个方面，例如在基于内容的推荐方法中通常利用关键词（标签）对物品进行建模与分析. 标签系统按照标签的来源可以分为由用户自主标记的用户标注系统和由领域专家预先定义的专家标注系统. 用户标注系统应用更加广泛，标签更加多样化，更能反映用户的长尾兴趣，但缺点是质量不高，处理过程复杂. 其中较为著名的用户标注系统有 Delicious[1]，Flickr[2] 等. 专家标注系统则由相关领域的专家对网络资源的各项特征进行预先细致的描述以供用户选择，该类系统的标签对资源的描述更准确但灵活性不足，不能完全反应用户的兴趣，其代表系统有 Pandora[3] 音乐推荐系统和 Jinni[4] 电影推荐系统.

一般而言，人们通过对标签的理解选择自己喜爱的资源. 对标签的关注度不仅能够影响用户对资源的选择，也蕴含了用户潜在的兴趣特征. 在推荐系统里，通过将稀疏的 User-Item 关系映射为 User-Tag 关系，可以帮助我们快速了解用户关于不同标签的兴趣分布（即关注度分布）. 例如，对某一用户曾经观看过的电影按电影的标签进行分类，可以建立用户和所有电影标签的 User-Tag 向量，该向量反映了用户对每一个标签的兴趣度.

2.2 因子分析

因子分析是主成分分析[41]的推广和发展，是多元统计分析中降维的一种方法，它主要是用来分析隐藏在表面现象背后因子作用的一类统计模型. 因子分析研究相关阵或协方差阵的内部依赖关系，将多个变量综合为少数几个因子，以再现原始变量与因子之间的相关关系. 因子分析在心理学，社会学，经济学等学科中取得了成功的应用[42]. 例如，对体育竞技比赛中运动员十项全能成绩的得分进行分析，可以得出短跑速度，爆发性臂力，爆发性腿力和耐力等因子对运动员的比赛成绩起到了决定性作用，从而可以将不同的运动员划分到不同的类别中进行有针对性的指导.

同样，在推荐系统领域中，利用因子分析的方法对某一具体行为下用户的某种兴趣分布进行分析，可以获得一些因子. 这些因子往往反映了用户外在行为表现和偏好的区别于他人的一些内在特质. 例如用户的情感、认知、气质、特殊情结等个性化的内在特质决定了什么样类型的电影可能是用户的首选. 一般而言，具有极为相似的内在特质的人往往具有更为接近的偏好选择，例如双胞胎兄弟（姐妹）. 因此，找出和目标用户具有相同或相似内在特质的邻居用户，然后基于这些邻居用户进行协同推荐可能会有良好的推荐效果.

[1]http://www.delicious.com/
[2]http://www.flickr.com/
[3]http://www.pandora.com/
[4]http://www.jinni.com/

3 推荐算法框架

结合了物品的标签信息和潜在因子分析方法，本文中的方法首先采用一个特殊的过程为每个用户构建一个关于物品所有标签的兴趣分布，然后利用因子分析的方法分析多个用户的兴趣分布从而建立一个特殊的因子模型，最后利用该因子模型为用户生成新的基于潜在因子的 User-Factor 向量并基于此进行“邻居发现”.

算法的具体过程如图 1 所示：步骤①构建 User-Tag 矩阵：User-Tag 矩阵是一个以用户为行、标签为列的二维矩阵，它由原始的 User-Item 矩阵变换而来，矩阵中的每一行表示一个用户对所有标签的兴趣分布. User-Tag 矩阵也可看成是以每个行向量为个体构成的一个样本；步骤②学习因子模型：通过对步骤①获得的 User-Tag 矩阵（一个样本）进行因子分析，挖掘出样本中不同兴趣分布背后隐藏的一些潜在因子并建立一种描述潜在因子与原标签关系的因子模型. 该因子模型是一个经过数据除噪、降维后的 Tag-Factor 矩阵. 矩阵以标签为行、因子为列，每一列表示一个因子（*Factor*）与原始标签（*Tag1* 到 *Tagm*）的关系，该关系可以用式子（1）所示的多元齐次线性方程进行描述：

$$Factor = \lambda_1 Tag1 + \lambda_2 Tag2 + ... + \lambda_m Tagm . \quad (1)$$

其中 λ_1 到 λ_m 为权重系数，对应于因子模型中某一列的值；*Tag1* 到 *Tagm* 表示 m 个标签. 步骤③则利用因子模型 Tag-Factor 矩阵为测试集中的所有用户建立新的 User-Factor 向量，这些向量的集合构成 User-Factor 矩阵；步骤④计算不同用户向量之间的相似度，即 User-Factor 矩阵中相应两个行向量的距离，从而找出和目标用户最相似的 *N* 个邻居用户.

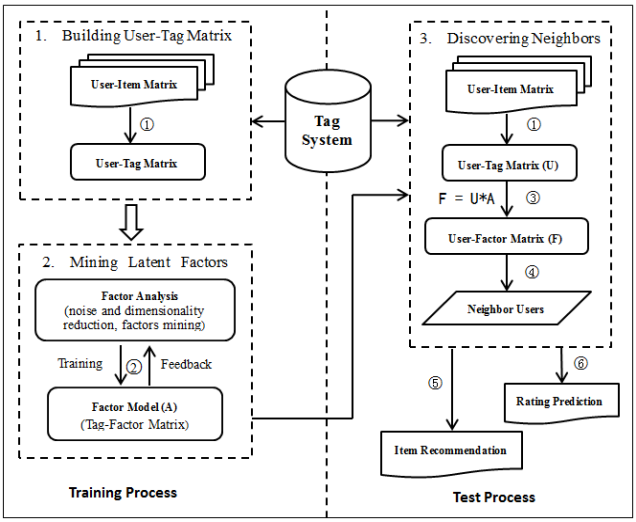

Fig. 1 The flow diagram of algorithm.

图 1 算法流程图

基于获得的目标用户的“邻居”，我们可以将“邻居”曾经打分较高且目标用户没有评分过的的物品对目标用户进行推荐（步骤⑤），也可以定义一个基于目标用户“邻居”的评分预测方法来对目标用户未评分的物品进行评分预测（步骤⑥）. 式子（2）描述了一种常用的根据目标用户“邻居”预测目标用户未知评分的计算方法：

$$\tilde{R}_{ui}=\bar{r}(u)+\frac{1}{k}\sum_{v\in N_k(u,i)} sim(u,v)(r(v,i)-\bar{r}(v)) . \quad (2)$$

其中，$\tilde{R}_{ui}$ 表示用户 u 对物品 i 的预测评分；$\bar{r}(u)$ 为用户 u 对其所评分过的所有物品的平均分；$sim(u,v)$ 为用户 u 和 v 的相似度；$r(v,i)$ 为用户 v 对物品 i 的评分；$N_k(u,i)$ 为目标用户的“邻居”中对 i 有评分的用户的集合；k 为 $N_k(u,i)$ 中元素的个数.

4 潜在因子挖掘和邻居发现

本节对上节算法框架中的核心步骤的具体实施过程进行详细说明. 对于图 1 中的过程①-④可以用如下 π_1 到 π_4 的矩阵变换过程进行描述：

$$\pi_1: U\times I \xrightarrow{I\times T} U\times T$$
$$\pi_2: U\times T \to T\times F$$
$$\pi_3: (U\times T)*(T\times F) \to U\times F$$
$$\pi_4: sim(U\times F) \to Top\text{-}N$$

其中，U，I，T，F 分别为用户集合、物品集合、物品标签集合和潜在因子集合；运算符“×”表示由两个集合笛卡尔积构成的二维矩阵，例如 $I\times T$ 表示物品为行、标签为列的 Item-Tag 矩阵；“*”表示两个矩阵的乘积；函数 $sim(X)$ 计算矩阵 X 中任意两个行向量的相似度从而为用户找出 N 个最相似的“邻居”.

4.1 构建 User-Tag 矩阵

UI 矩阵中用户对物品的评分反映了用户对物品的关注. 评分的高低是物品被关注后用户对物品的判断，这不仅跟用户自身有关也跟物品的好坏有关. 为了准确描述用户本身对物品标签的兴趣分布，用户对物品的评分值应该被忽略. 因此，一个用户对所有标签的兴趣分布可以用如下的方法获得.

假设用户 u 对物品 i 有过评分且物品 i 对应有 n 个标签，则物品 i 对应的每个标签将获得 $1/n$ 个关注度. 从而，用户 u 对标签 t 的关注度可以用式子（3）表示：

$$r_{ut}=\frac{\sum_{i\in D_k(u)}\frac{sgn(u,i,t)}{N(i)}}{k} . \quad (3)$$

其中，$sgn(u,i,t)=\begin{cases}1 & t \text{ 是物品 } i \text{ 的一个标签}\\ 0 & \text{其它}\end{cases}$，

r_{ut} 为用户 u 对标签 t 的关注度；$N(i)$ 为物品 i 的标签个数；$D_k(u)$ 为用户 u 有过评分的物品的集合；k 为集合 $D_k(u)$ 中元素的个数. 假设共有 p 个标签，则用户 u 对所有标签的关注度可用 User-Tag 向量 U_u 表示：

$$U_u=(r_{u1},r_{u2},...,r_{up}) . \quad (4)$$

多个 User-Tag 向量的集合构成了 User-Tag 矩阵.

4.2 潜在因子挖掘

因子分析方法对一些可观测的统计变量（如 4.1 节中的 U_u）进行分析并将这些原始可观测的统计量映射到潜在因子空间，以达到降维，除噪，潜在因子挖掘的目的. 因子分析方法和推荐算法中的矩阵分解方法存在着明显的差别：前者主要通过研究多个统计变量的相关阵或者协方差阵来发掘变量与变量之间、变量与潜在因子之间的关系，从而为用户建立新的关于潜在因子的特征向量. 而矩阵分解方法则是从数据拟合的角度出发将原始评分矩阵分解为两个矩阵的乘积，通过对两个矩阵中的数值进行调整使乘积后的矩阵逼近原始的评分矩阵并实现对未知评分的预测.

（1）因子模型

设 $X=(X_1,X_2,...,X_p)^T$ 是可观测的 p 维随机变量，则因子模型可用一个二维矩阵 A 表示，其中 $A=(a_{ij})_{p*m}$ 且 A 满足式子（5）：

$$\begin{cases} X_1-\mu_1=a_{11}f_1+a_{12}f_2+...+a_{1m}f_m+\varepsilon_1\\ X_2-\mu_2=a_{21}f_1+a_{22}f_2+...+a_{2m}f_m+\varepsilon_2\\ \vdots\\ X_p-\mu_p=a_{p1}f_1+a_{p2}f_2+...+a_{pm}f_m+\varepsilon_p \end{cases} . \quad (5)$$

其中，$\mu_1,\mu_2,\cdots,\mu_p$ 是随机变量 $X_1,X_2,\cdots,X_p$ 的数学期望；$f_1,f_2,\cdots,f_m$ (m<p)为公共因子，$\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_p$ 为特殊因子，它们都是不可观测的随机变量. 公共因子 $f_1,f_2,\cdots,f_m$ 出现在每一个原始变量 X_i (i=1,2,⋯, p)的表达式中，可理解为原始变量共同具有的公共因素，每个公共因子 f_j (j=1,2,⋯,m)一般至少对两个原始变量有作用，否则它将归入特殊因子. 每个特殊因子 ε_i (i=1,2,⋯, p)仅仅出现在与之对应的第 i 个原始变量的表达式 X_i 中，它只对这个原始变量有作用，式子（5）也可以写成如式子（6）的矩阵表示形式：

$$X=\mu+AF+\varepsilon\approx\mu+AF . \quad (6)$$

其中，$F=(f_1,f_2,\cdots,f_m)^T$ 为公共因子向量；$\varepsilon=(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_p)^T$ 为特殊因子向量，$\mu=(\mu_1,\mu_2,\cdots,\mu_p)^T$ 为随机变量 X 的数学期望.

因此，求解载荷矩阵 A 是获得因子模型的关键.

（2）求解载荷矩阵 A

设 $X_{(1)},X_{(2)},...,X_{(n)}$ 是一组 p 维个体组成的样本，其中 $X_{(i)}=(x_{i1},x_{i2},...,x_{ip})^T$，则数学期望 μ 的估计量为

$$\mu=\overline{X}=\frac{1}{n}\sum_{i=1}^{n}X_{(i)}, \qquad (7)$$

为了求解载荷矩阵 A，样本协方差矩阵的估计量为：

$$\Sigma=\frac{1}{n-1}\sum_{i=1}^{n}(X_{(i)}-\overline{X})(X_{(i)}-\overline{X})^T. \qquad (8)$$

由式子（5）的性质[42]可知：

$$\Sigma\approx AA^T. \qquad (9)$$

设样本的协方差阵 S 的特征值为 $\lambda_1\geq\lambda_2\geq...\geq\lambda_p\geq0$，相应的单位正交特征向量为 $l_1,l_2,...,l_p$，从而当前 m 个特征值的总和远大于最后 p-m 个特征值的总和时有：

$$\begin{aligned}\Sigma&=\lambda_1l_1l_1^T+...+\lambda_ml_ml_m^T+\lambda_{m+1}l_{m+1}l_{m+1}^T+...+\lambda_pl_pl_p^T\\&\approx\lambda_1l_1l_1^T+...+\lambda_ml_ml_m^T\\&=AA^T\end{aligned} \qquad (10)$$

则 $A=(a_{ij})_{p*m}=(\sqrt{\lambda_1}l_1,\sqrt{\lambda_2}l_2,...,\sqrt{\lambda_m}l_m)$ 即为因子模型的解.

4.3 生成“邻居”用户

假设数据集的因子模型为 A，为 UI 评分矩阵建立的 User-Tag 矩阵为 U，则 User-Factor 矩阵 F 可表示为式子（11-12）：

$$F=U*A=\begin{bmatrix}\sigma_{11}&\sigma_{12}&\cdots&\sigma_{1m}\\\sigma_{21}&\sigma_{22}&\cdots&\sigma_{2m}\\\vdots&\vdots&\ddots&\vdots\\\sigma_{n1}&\sigma_{n2}&\cdots&\sigma_{nm}\end{bmatrix}, \qquad (11)$$

$$\sigma_{ij}=\sum_{k=1}^{p}r_{ik}a_{kj}. \qquad (12)$$

其中，$U=\begin{bmatrix}r_{11}&r_{12}&\cdots&r_{1p}\\r_{21}&r_{22}&\cdots&r_{2p}\\\vdots&\vdots&\ddots&\vdots\\r_{n1}&r_{n2}&\cdots&r_{np}\end{bmatrix}$，$A=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1m}\\a_{21}&a_{22}&\cdots&a_{2m}\\\vdots&\vdots&\ddots&\vdots\\a_{p1}&a_{p2}&\cdots&a_{pm}\end{bmatrix}$

矩阵 F 中每一个行表示一个 User-Factor 向量.针对 User-Factor 矩阵 F，我们可以利用不同的方法来计算任意两个行向量的距离（即用户的相似度），如 CityBlock 距离[43]，Pearson 距离[44]等. 根据到不同用户间的距离获得和目标用户最相似的 N 个邻居用户.

5 实验结果与分析

为了验证本文中提出的方法在评分预测方面的准确性以及对稀疏数据集的低敏感性，选取了推荐系统领域经典的 MovieLens 1M[5] 数据集进行实验. 该数据集包含 2000 年 3900 个匿名用户对 6040 部电影的 1,000,209 个评分（稀疏度为 4.2%），标签的种类数共有 18 种，每个电影有一个或多个标签构成. 数据集被分为两个部分：训练集包含 3000 个用户，测试集包含剩余的 3040 个用户. 训练集用来学习因子模型，然后用训练集获得的因子模型对测试集中的用户进行 User-Factor 向量生成，并基于生成的 User-Factor 向量计算用户的 Top-N 个邻居以及利用 Top-N 邻居对目标用户进行评分预测.

5.1 实验一：因子个数选择

在建立因子模型的过程中，为了确定应当从多少个样本用户中抽取多少个因子来建立因子模型，进行了以下实验. 实验结果如图 2：

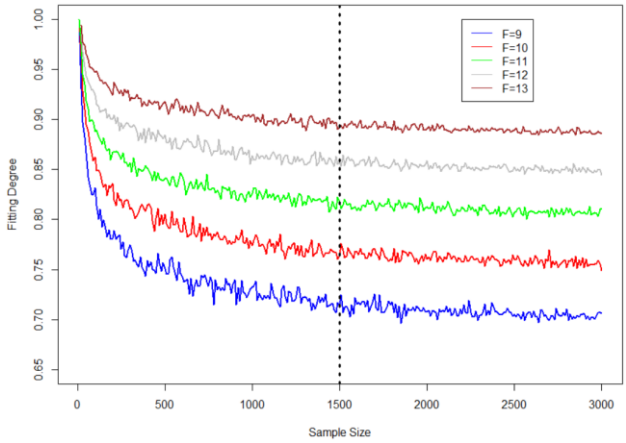

Fig. 2 Relation between fitting degree and sample sizes.

图 2 拟合程度和样本数量关系图

图 2 中横轴代表从训练集中随机选择的用户数，纵轴表示得到的因子模型对训练集中选择的样本数据的拟合程度，F 值表示指定的因子个数. 从图中可知，拟合程度随着样本数的增多而下降，其原因是当因子个数一定时，随着数据量的增大，拟合所有数据的难度随之增加. 当样本数增加到 1500 个时因子的拟合程度基本收敛，表示从 1500 个样本数据中抽取的因子模型已经可以用来描述数据集中其它的用户. 当样本数量一定时，随着因子个数增加，因子拟合程度也随之增加，因为更多的因子有更强的拟合原

[5] http://www.datatang.com/data/44521

数据的能力．一般而言，为了防止过拟合，通常选择拟合程度为 80%．综合以上分析，本文中使用的训练样本为从 3000 个用户中随机选择 1500 个，挖掘的因子个数选定为 F=11．此时得到的因子模型如表 1 示：

Table 1　Factor Model

表 1 因子模型

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
X1	0.04	0.05	-0.42	0.79	-0.04	-0.28	-0.67	0.85	-0.08	0.25	-0.03
X2	0.67	-0.03	0.05	-0.01	-0.19	0.07	0.00	0.00	0.08	0.57	-0.02
X3	-0.02	0.04	-0.13	-0.02	-0.14	-0.64	-0.04	-0.05	-0.02	0.11	-0.04
X4	-0.83	0.12	0.00	-0.01	-0.13	-0.52	0.10	0.01	-0.14	0.45	0.10
X5	-0.04	-0.18	-0.13	0.07	0.02	0.85	0.03	-0.04	-0.13	-0.01	0.02
X6	0.03	-0.04	-0.02	-0.07	0.04	-0.02	0.04	-0.01	0.09	0.02	-0.06
X7	-0.08	0.01	0.93	-0.02	0.44	0.00	0.09	-0.01	-0.07	0.01	0.92
X8	0.00	0.10	-0.02	0.00	-0.04	0.02	0.01	-0.08	-0.03	-0.07	0.02
X9	0.41	-0.01	0.07	-0.05	-0.11	-0.07	0.02	-0.63	-0.25	-0.08	-0.12
X10	0.10	0.85	0.33	0.01	-0.07	0.01	-0.02	-0.06	0.04	0.02	0.03
X11	-0.01	0.00	-0.07	0.04	0.00	-0.01	-0.01	-0.03	0.02	-0.03	0.92
X12	0.01	-0.07	0.01	0.47	0.00	-0.01	0.00	-0.06	0.00	-0.01	0.06
X13	0.00	-0.02	0.02	-0.02	0.08	0.00	-0.01	-0.02	-0.01	-0.02	0.01
X14	0.99	0.01	0.91	0.10	0.29	-0.01	-0.25	-0.10	-0.18	-0.19	-0.35
X15	-0.16	0.00	0.03	-0.10	0.03	0.03	-0.03	0.00	-0.01	0.11	0.04
X16	0.09	0.00	0.53	-0.11	-0.17	-0.85	-0.22	0.17	0.15	0.02	0.03
X17	0.02	-0.07	-0.04	0.00	0.95	0.02	-0.09	0.00	0.12	0.08	-0.03
X18	0.11	0.03	0.02	0.14	-0.12	0.05	-0.03	0.04	-0.04	-0.18	-0.01

X1-X18 表示 18 个原标签，F1-F11 代表 11 个潜在因子．每一个因子是关于 18 个原标签的一个线性组合关系，即 18 个原标签对挖掘出的每一个因子有着不同程度的影响．

5.2　实验二：算法准确性测试

为了对比本文中的方法和其他方法在未知评分预测方面的差异，本文采用了 RMSE 评估方法．RMSE 的思想是通过对测试集中对用户已打过分的项进行评分预测，然后比较预测值和真实值之间的差距，差距越小，则准确率越高．RMSE 的表达式如式子（13）：

$$RMSE=\sqrt{\frac{\sum_{1}^{n}(r_i-\hat{r}_i)^2}{n}}. \qquad (13)$$

其中，r_i 是物品的真实评分；$\hat{r}_i$ 是对物品的预测评分；n 为实验中预测的物品的样本数．

将本文中的方法（记为 Native）和传统的协同过滤方法（CityBlock 距离相似度[43]，Pearson 距离相似度[44]）、矩阵分解方法（SGD 方法[35]，SVD++方法[36]）以及为目标用户随机产生邻居的方法（记为 Random）进行对比．其中传统的协同过滤方法通过计算用户的 User-Item 向量的距离来寻找邻居；矩阵分解方法则通过建立拟合已有评分数据的矩阵模型来预测未知的评分；Random 方法提供了一种基于邻居用户进行评分预测的方法的基础对照．

本实验中用到的其它参数和方法如表 2：

Table 2　Other Parameters in Experiments

表 2 其他实验参数

Parameters (methods)	Value
sim(u,v)	CityBlockSimilarity[43]
Ratings prediction in step ⑥	Formula（2）
Value of *N* in step ④	5

实验中采用的其它方法的实现均来自 Apache Mahout[6]，其中协同过滤方法以及 Random 方法参考的邻居用户数和本文中的方法相同，矩阵分解方法 SGD 和 SVD++的迭代次数为 5000 次．每一种方法重复计算 10 次并取平均值作为最后的结果．

不同实验方法的结果如图 3 所示：

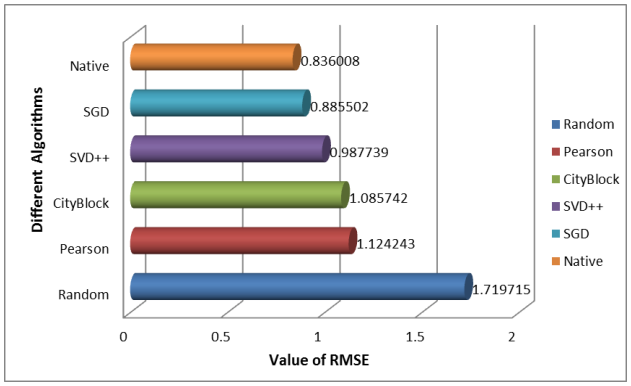

Fig. 3　Comparison of different algorithms.

图 3 不同算法对比图

从图 3 的实验结果可知，Random 方法效果最差，矩阵分解方法 SGD、SVD++要优于采用 CityBlock、Pearson 相似性度量的传统协同过滤方法，而本文中提出的方法 Native 则具有最低的 RMSE 值，相比其它几种比较的方法具有更高的算法准确率．

5.3　实验三：算法敏感性测试

算法的敏感性主要是指数据集的稀疏程度对算法评分预测准确性的影响．好的推荐算法应当能够适应不同稀疏程度的数据集．为了验证本文中的方法在处理稀疏程度不断增加的真实数据集时的优越性，进行了以下对比实验．首先对原始数据集中的评分数据进行随机抽样，使原始数据集的稀疏程度分别降低到 0.5%，1.0%，1.5%，2.0%，2.5%，3.0%，3.5%，4.2%，然后对比不同算法在不同数据稀疏度情况下准确率的变化情况．实验结果如图 4 所示：

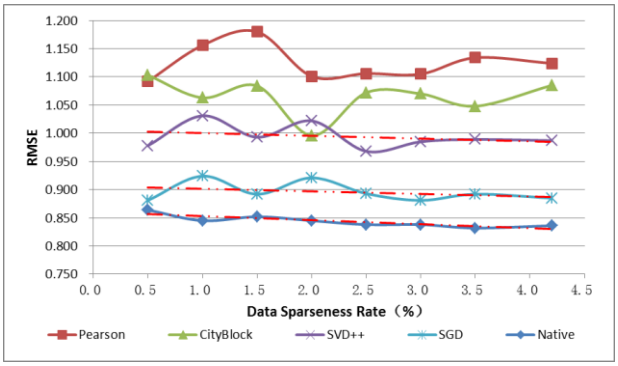

Fig. 4　Comparison of Data Sensitivity.

图 4 数据敏感性对比

图 4 中横轴代表了数据集的稀疏程度（从左到右稀疏程度逐渐减弱），纵轴表示在相应稀疏度条件下

[6] http:// mahout.apache.org/

算法的 RMSE 值，其中三条红色虚线分别代表了对 SVD++，SGD 和 Native 方法结果的线性拟合．从图 4 中可以看出，相比协同过滤方法和矩阵分解方法 Native 方法在不同稀疏度条件下均有具有更低的 RMSE 值，即更好的准确率．矩阵分解方法随着数据稀疏性的增加算法的准确率逐渐下降（由 SVD++和 SGD 方法的拟合曲线可以看出），说明矩阵分解方法是一种对数据集稀疏程度敏感的方法．而在实际应用中，随着用户和商品数量的不断增加，UI 矩阵中有效评分数增长的速度远小于矩阵所能表示的评分的增长速度，从而，UI 评分矩阵会变得越来越稀疏（例如淘宝数据集的稀疏度已达百万分之一），但矩阵中大部分用户的评分数则会随着时间的延长而有所增加．基于文中构建用户关于标签兴趣的方法可知，用户关注的物品越多计算用户关于标签的兴趣分布与用户的真实兴趣分布也越接近，从而，利用因子分析方法计算的邻居用户也越精确．因此，算法的 RMSE 值也应当越来越低（当用户关于标签的兴趣分布未达到真实分布时）或者保持基本稳定（当用户关于标签的兴趣分布已逼近真实的兴趣分布时）．针对上面的分析，实验中 Native 方法在用户具有较多的评分情况下（对应于图 4 中数据稀疏度较低的情况，如 4.0%，而在实际动态增长的数据集中则对应稀疏度较高的情况）具有较低的 RMSE 值验证了这一点．所以在实际动态增长的数据集中，本文中的方法将会随着数据集的稀疏程度的增加，准确率略有提高或者保持基本稳定．

由于传统的协同过滤方法及矩阵分解方法和本文中的方法在评分预测的原理上存在的本质差别，即前两种方法直接基于稀疏的 UI 矩阵进行评分预测，而文中的方法则主要与用户的评分数相关，从而决定了前两种方法是一种对数据集稀疏程度敏感性较高的算法，而本文中的方法则只对用户的评分数敏感，与数据集的稀疏程度无关．因此本文中的方法更能适应在工业应用中的动态增长的数据集．

5.4 算法时间复杂度分析

时间复杂度是衡量一个推荐算法好坏的非常重要的指标，与用户的体验息息相关．本文中方法的时间复杂度为： $O(kn+n^2)\approx O(n^2)$，其中 k 为常量系数；$O(kn)$ 用来为用户计算基于标签的兴趣分布的向量 User-Tag．$O(n^2)$ 的开销则在于计算不同用户之间的相似度．由于在本文中对 User-Item 向量进行了降维，所以与其它直接基于 User-Item 向量进行相似度计算的方法相比速度有大幅度提升，其加速比为 κ：

$$\kappa=\frac{N(Item)}{N(Factor)}. \qquad (14)$$

其中，$N(Item)$ 和 $N(Factor)$ 分别表示物品 Item 和潜在因子 Factor 的个数．

6 总结与展望

本文针对实际工业应用中数据集稀疏度动态增加的特点，进行了深入的分析和研究，提出了一种准确率较高的协同推荐新方法．这种方法假设用户的一些内在特质在短时间内（如一年）保持稳定．最后，通过实验证实了文中提出的方法在处理真实的动态增长的数据集时具有较低的算法敏感性．

当数据集涉及的时间跨度较大时，用户的内在特质也会发生变化，相应的兴趣也会变化，文中提出的基于标签的兴趣建模方法的准确性也会逐渐降低，从而影响算法的准确性，这时就需要对用户的兴趣变化进行跟踪并基于较短的时间片进行兴趣建模，因此，定义兴趣随时间变化的模型将是扩展本文方法的未来方向之一．

参 考 文 献

[1] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for e-commerce[C]//Proceedings of the 2nd ACM conference on Electronic commerce. ACM, 2000: 158-167.

[2] Lam X N, Vu T, Le T D, et al. Addressing cold-start problem in recommendation systems[C]//Proceedings of the 2nd international conference on Ubiquitous information management and communication. ACM, 2008: 208-211.

[3] Basu C, Hirsh H, Cohen W. Recommendation as classification: Using social and content-based information in recommendation[C]//AAAI/IAAI. 1998: 714-720.

[4] Bao J, Zheng Y, Mokbel M F. Location-based and preference-aware recommendation using sparse geo-social networking data[C]//Proceedings of the 20th International Conference on Advances in Geographic Information Systems. ACM, 2012: 199-208.

[5] Zhang Z K, Zhou T, Zhang Y C. Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs[J]. Physica A: Statistical Mechanics and its Applications, 2010, 389(1): 179-186.

[6] Sarwar B, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system-a case study[R]. Minnesota Univ Minneapolis Dept of Computer Science, 2000.

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

[8] Jacobs D W. Linear fitting with missing data for structure-from-motion[J]. Computer Vision and Image Understanding, 2001, 82(1): 57-81.

[9] Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 116-142.

[10] Ren J, Zhou T, Zhang Y C. Information filtering via self-consistent refinement[J]. EPL (Europhysics Letters), 2008, 82(5): 58007.

[11] Sun D, Zhou T, Liu J G, et al. Information filtering based on transferring similarity[J]. Physical Review E, 2009, 80(1): 017101.

[12] Zar J H. Significance testing of the Spearman rank correlation coefficient[J]. Journal of the American Statistical Association, 1972, 67(339): 578-580.

[13] Mark H L, Tunnell D. Qualitative near-infrared reflectance analysis using Mahalanobis distances[J]. Analytical Chemistry, 1985, 57(7): 1449-1456.

[14] Xue G R, Lin C, Yang Q, et al. Scalable collaborative filtering using cluster-based smoothing[C]//Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2005: 114-121.

[15] Bellogin A, Parapar J. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering[C]//Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012: 213-216.

[16] Konstas I, Stathopoulos V, Jose J M. On social networks and collaborative recommendation[C]//Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2009: 195-202.

[17] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 448-456.

[18] Karatzoglou A, Amatriain X, Baltrunas L, et al. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 79-86.

[19] Lin W, Alvarez S A, Ruiz C. Collaborative recommendation via adaptive association rule mining[C]//Proceedings of the International Workshop on Web Mining for E-Commerce (WEBKDD. 2000.

[20] Walter F E, Battiston S, Schweitzer F. A model of a trust-based recommendation system on a social network[J]. Autonomous Agents and Multi-Agent Systems, 2008, 16(1): 57-74.

[21] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 135-142.

[22] McCullagh P. Generalized linear models[J]. European Journal of Operational Research, 1984, 16(3): 285-292.

[23] Zhang Y, Koren J. Efficient bayesian hierarchical user modeling for recommendation system[C]//Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007: 47-54.

[24] White D J. Real applications of Markov decision processes[J]. Interfaces, 1985, 15(6): 73-83.

[25] Chib S. Bayes regression with autoregressive errors: A Gibbs sampling approach[J]. Journal of Econometrics, 1993, 58(3): 275-294.

[26] Sigurbjörnsson B, Van Zwol R. Flickr tag recommendation based on collective knowledge[C]//Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 327-336.

[27] Kim C, Kim J. A recommendation algorithm using multi-level association rules[C]//Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on. IEEE, 2003: 524-527.

[28] Fan B, Liu L, Li M, et al. Knowledge recommendation based on social network theory[C]//Advanced Management of Information for Globalized Enterprises, 2008. AMIGE 2008. IEEE Symposium on. IEEE, 2008: 1-3.

[29] Fouss F, Pirotte A, Renders J M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. Knowledge and Data Engineering, IEEE Transactions on, 2007, 19(3): 355-369.

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

[31] Liu J G, Wang B H, Guo Q. Improved collaborative filtering algorithm via information transformation[J]. International Journal of Modern Physics C, 2009, 20(02): 285-293.

[32] Zhang Y C, Blattner M, Yu Y K. Heat conduction process on community networks as a recommendation model[J]. Physical review letters, 2007, 99(15): 154301.

[33] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007, 76(4): 046115.

[34] Baird L, Moore A W. Gradient descent for general reinforcement learning[J]. Advances in neural information processing systems, 1999: 968-974.

[35] Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent

[36] Umbrath A S R W W, Hennig L. A hybrid PLSA approach for warmer cold start in folksonomy recommendation[J]. Recommender Systems & the Social Web, 2009: 10-13.

[37] Ma H, Yang H, Lyu M R, et al. Sorec: social recommendation using probabilistic matrix factorization[C]//Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008: 931-940.

[38] Agarwal D, Chen B C. Regression-based latent factor models[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 19-28.

[39] Chen T, Zhang W, Lu Q, et al. SVDFeature: a toolkit for feature-based collaborative filtering[J]. The Journal of Machine Learning Research, 2012, 13(1): 3619-3622.

[40] Rendle S. Factorization machines with libFM[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 57.

[41] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1): 37-52.

[42] Yi X et al. statistical modeling and R software[J]. Tsinghua University Press, 2006.

[43] Melter R A. Some characterizations of city block distance[J]. Pattern recognition letters, 1987, 6(4): 235-240.

[44] Stigler S M. Francis Galton's account of the invention of correlation[J]. Statistical Science, 1989: 73-79.

Cai Guoyong, born in 1971. Professor and PhD. His major research interests include deep analysis of social media, trusted computing.

Lv Rui, born in 1988, Master Degree Candidate, His major research interests include data mining of social media, recommendation systems.