文章编号：1004-4353(2017)04-0334-05

融合项目属性特征的 SVD 协同过滤推荐算法研究

潘峰， 怀丽波*， 崔荣一

（延边大学工学院 计算机科学与技术学科 智能信息处理研究室，吉林 延吉 133002）

摘要：针对协同过滤方法中用户-项目评分矩阵的极端稀疏性问题，提出了一种基于层次的混合推荐方法. 首先利用 TF-IDF 提取项目属性特征，并利用余弦相似度对评分矩阵的缺失值进行填充；然后通过对填充的矩阵做 SVD，寻找隐性特征，建立隐语义模型；最后将本文的算法分别与众数填充和无填充模型进行对比实验，结果表明本文提出的方法有效提高了推荐的精度.

关键词：稀疏性问题；混合推荐方法；协同过滤；隐语义模型

中图分类号：TP391 **文献标识码**：A

Research on SVD collaborative filtering recommender algorithm fused items' attribute feature

PAN Feng， HUAI Libo*， CUI Rongyi

（*Intelligent Information Processing Lab.*, *Dept. of Computer Science & Technology*, *College of Engineering*, *Yanbian University*, *Yanji* 133002, *China*）

Abstract: In this paper, a hierarchy-based hybrid recommendation method algorithm is proposed for the sparsity problem of user-item rating matrix in collaborative filtering. Firstly, this algorithm extracted the attributes of items by using TF-IDF method, and filled the missing values with cosine similarity in ratings matrix. Then, the latent factor model was established by doing SVD for the filled matrix to find the latent feature. At last, the compared experiment was carried on with mode-filling and non-filling model. The experiment results show that the proposed algorithm can improve the recommended accuracy.

Keywords: sparsity problem; hybrid recommender method; collaborative filtering; latent factor model

0 引言

推荐算法作为电子商务的核心技术，其主要任务是从用户-物品数据中找出之间的联系，根据物品之间的相关程度返回该用户感兴趣的物品[1]. 但由于稀疏性等问题，用户很难从纷繁复杂的物品种类中做出合理选择，因此如何有效利用数据，产生较高精度的推荐，成为推荐算法研究的重要内容. 目前，推荐算法的研究主要分为基于内容的方法和基于模型的方法[2-3]. 基于内容的方法是用数据的信息来提升推荐精度，如物品属性或历史行为等. 虽然基于内容的方法能够有效应对冷启动问题，但物品属性很难准确刻画. 基于模型的方法主要采用因子特征来描述和解释用户对物品的评分[4]，通过深化基础的表示和减少对任意决策的依赖方式来产生更精确的预测结果，但解决稀疏性问题仍有较多困难.

解决稀疏性问题，目前主要有两种方法：一是

收稿日期：2017-06-23 **基金项目**：吉林省科技计划项目(20140101186JC)

* **通信作者**：怀丽波(1973—)，女，副教授，研究方向为优化理论与方法、数据挖掘.

对特定问题，从算法角度改善精度. 文献[5]针对社交网络推荐问题，将用户的社交关系和行为活动融合到算法中. 另一种方法是采用填充或降维等方法减少数据稀疏性，文献[6]通过项目间相似度进行填充处理. 为更好解决稀疏性问题，在使用中经常采用混合推荐的方式. 文献[7]通过多信息融合方式，利用内容的信息提高了模型的准确率. 文献[8]通过多模型组合方式提高了推荐性能，但其只用了评分信息. 本文在评分信息的基础上，融合项目属性特征，提出了一种基于层次的混合推荐方法.

1　相关算法介绍

1.1　基于内容的推荐

“内容”的属性通常是文本，基于内容的方法目前主要使用检索模型[9]，如向量空间模型(VSM). 在 VSM 模型中，令 $I=\{\boldsymbol{i}_1,\boldsymbol{i}_2,\cdots,\boldsymbol{i}_n,\cdots,\boldsymbol{i}_N\}$ 表示一个“内容”的集合，$T=\{t_1,t_2,\cdots,t_k,\cdots,t_K\}$ 表示属性集合，T 通常从项目描述的数据中获得. 项目内容 $\boldsymbol{i}_n$ 表示为 K 维向量空间中的一个向量，即 $\boldsymbol{i}_n=\{\omega_{n1},\omega_{n2},\cdots,\omega_{nk},\cdots,\omega_{nK}\}$，其中权重 ω_{nk} 是项目 $\boldsymbol{i}_n$ 与属性特征 t_k 的关联程度.

内容的属性特征要具有描述性和分辨性，因此要将描述性强的特征突显出来. 在整个集合 I 中，如果一个属性 t_k 只在个别项目中频繁出现，在其它的项目中很少出现，那么该属性为强特征. 在 VSM 中，这些强特征被加权表示. 常用的加权模式是 TF-IDF，它可以使项目的强特征更具有描述性和分辨力，并且可以归一化属性权重. TF-IDF 定义如下：

$$\text{TF-IDF}(\boldsymbol{i}_n,t_k)=\text{TF}(\boldsymbol{i}_n,t_k)\cdot\lg\frac{N}{count(t_k)},$$

$$\text{TF}(\boldsymbol{i}_n,t_k)=\omega_{nk}/\sum_{n=1}^{N}\omega_{nk},\tag{1}$$

其中 $count(t_k)$ 表示在整个属性集中属性 t_k 出现的总数.

1.2　基于 SVD 的协同过滤算法

基于模型的方法主要是利用 SVD 方法揭示用户和项目的特征，这些隐藏的特征能够解释用户的评分，常见的有 pLSA、CTM、隐语义模型(LFM) 等. 隐语义模型是把用户-项目信息映射到一个维度为 F 的联合隐语义空间中. 每个用户 u 都与一个 F 维向量 $\boldsymbol{p}_u\in\boldsymbol{R}^F$ 相关，每个项目 i 都与一个 F 维向量 $\boldsymbol{q}_i\in\boldsymbol{R}^F$ 相关. 点积 $\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u$ 记录了用户与项目之间的交互，即用户对项目的总体兴趣度. 因此，可通过下面的规则得到评分：

$$\text{Preference}(u,i)=\hat{r}_{ui}=\boldsymbol{p}_u^{\mathrm{T}}\boldsymbol{q}_i=\sum_{f=1}^{F}p_{uf}q_{if},\tag{2}$$

其中 p_{uf} 表示用户 u 在隐语义因子 f 方面的特征，q_{if} 表示项目 i 在因子 f 方面的特征. 为了学习模型中的 $\boldsymbol{p}_u$ 和 $\boldsymbol{q}_i$，通过最小化损失函数来近似估计，其计算公式为

$$\min_{\boldsymbol{q},\boldsymbol{p}}\sum_{(u,i)\in D}(r_{ui}-\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u)+\lambda(\boldsymbol{q}_i^2+\boldsymbol{p}_u^2),\tag{3}$$

其中 D 为用户-项目集，常量 λ 是正则化项参数，用来防止过拟合. 最小化方法通过随机梯度下降算法实现[10]. 对于给定的评分 r_{ui}，预测评分记为 $\hat{r}_{ui}$，相关的误差记为 $e_{ui}=r_{ui}-\hat{r}_{ui}$. 优化过程如公式(4) 所示：

$$\begin{cases}\boldsymbol{q}_i\leftarrow\boldsymbol{q}_i+\gamma(e_{ui}\boldsymbol{p}_u-\lambda\boldsymbol{q}_i),\\ \boldsymbol{p}_u\leftarrow\boldsymbol{p}_u+\gamma(e_{ui}\boldsymbol{q}_i-\lambda\boldsymbol{p}_u),\end{cases}\tag{4}$$

其中 γ 为学习率，γ 越大迭代下降越快.

2　融合项目属性特征的 SVD 推荐算法

2.1　融合项目属性特征的评分矩阵填充

度量项目之间的相似度是推荐算法的核心，一般先选择可信的近邻做预测评分，然后给予不同近邻在预测中的权重. 相似度权重的计算直接影响推荐准确性和执行的性能[11]. 在评分数据中，令 $N(u)$ 表示用户 u 评分的项目集合，则用户 u 表示为 $u=\{r_{u1},r_{u2},\cdots,r_{ui},\cdots,r_{uL}\mid L=|N(u)|\}$，其中 r_{ui} 为用户 u 对项目 i 的评分. 用户 u 和 v 之间的相似度通过公式(5) 来计算：

$$\text{sim}(u,v)=\frac{\sum_{i\in N(u)\cap N(v)}r_{ui}r_{vi}}{\sqrt{\sum_{i\in N(u)}r_{ui}^2}\cdot\sqrt{\sum_{j\in N(v)}r_{vj}^2}},\tag{5}$$

其中 $N(u)\cap N(v)$ 表示同时被用户 u 和用户 v 评分的项目集合.

2.1.1　项目属性特征提取　1) 从原始项目属性信息中提取数据. 在 MovieLens 的电影数据集中，电影属性有标签(ID)、电影名称(Title) 和类型(Genres)，其中每个 ID 对应一个名称和多个类型. 类型属性中有 Thriller、Drama 等共 18 个类型. 首先向量化每部电影：对于电影 $\boldsymbol{i}_n$，若属于类

别 t_k，则 $\omega_{nk}=1$，否则 $\omega_{nk}=0$，由此可以得到一个关于电影和类型关系的布尔矩阵，其部分数据如表 1 所示.

表 1 项目属性矩阵

ID	Thriller	Drama	Comedy	Romance	Action
i_1	0	1	1	0	1
i_2	1	0	0	0	1
i_3	0	1	1	0	0
i_4	1	1	0	0	0
i_5	0	1	0	0	1

2) TF-IDF 加权处理. 利用公式(5) 计算项目之间的相似度，如表 1 中 i_1 与 i_3 和与 i_5 的相似度为 $\mathrm{sim}(i_1,i_3)=\mathrm{sim}(i_1,i_5)=0.816$，并由此可知 i_3 和 i_5 在推荐集合中同等重要. 但实际上 i_3 和 i_5 的相似度并不高，因此只利用布尔矩阵计算相似度会掩盖强特征并导致推荐精度下降；故本文使用 TF-IDF 对属性加权，首先利用式(1) 得到每个项目的向量(表 2)，然后利用式(5) 再计算 i_3 和 i_5 与 i_1 的相似度，分别得 0.938 和 0.036. 因为“Drama”特征是项目共有的，区分能力弱，加权后的值为 0.024，而强特征“Comedy” 加权后的值为 0.199，强特征更加明显，说明其区分能力增强.

表 2 加权后的项目属性矩阵

ID	Thriller	Drama	Comedy	Romance	Action
i_1	0	0.024	0.199	0	0.074
i_2	0.199	0	0	0	0.074
i_3	0	0.024	0.199	0	0
i_4	0.199	0.024	0	0	0
i_5	0	0.024	0	0	0.074

2.1.2 评分矩阵缺失值的混合填充规则 融合属性的填充方法是利用加权后的项目属性矩阵来计算每个项目之间的相似度，从中选取与评分矩阵中待填充项目相似度较高的项目集合，然后再通过加权求和方式进行填充. 由于项目采用属性特征来表示，所以相似度越高的项目，越能突显他们的强特征；因此，本文填充过程选取一个经验阈值 0.9[5]，即相似度大于 0.9 的项目作为目标项目的相似项目. 对用户 u 的未打分项 j 进行填充，采用以下公式：

$$\mathrm{fill}_{uj}=\frac{\sum\limits_{i\in \mathrm{sim}(j)} r_{ui}\cdot \mathrm{sim}(i,j)}{\sum\limits_{i\in \mathrm{sim}(j)} \mathrm{sim}(i,j)}, \tag{6}$$

其中 $\mathrm{sim}(i)$ 表示“相似度较高”的项目集合，$N(u)$ 表示“用户 u 已评分”的项目集合. 由于评分矩阵的稀疏问题，$\mathrm{sim}(i)$ 与 $N(u)$ 的交集存在空集情况，故无法全部采用项目特征属性的填充方法；因此，可采用评分均值 $\mathrm{mean}_u=\left(\sum\limits_{i=1}^{n} r_{ui}\right)/n$ 进行填充[12]. 最终得到的融合项目特征属性的填充方法所示如下：

$$\mathrm{Rating}=\begin{cases} r_{ui}, & i\in N(u),\\ \mathrm{fill}_{uj}, & N(u)\cap \mathrm{sim}(i)\neq\varnothing,\\ \mathrm{mean}_u, & N(u)\cap \mathrm{sim}(i)=\varnothing. \end{cases} \tag{7}$$

2.2 基于 SVD 的隐语义模型训练

隐语义空间中，向量的每个因子用来解释评分值的特征，这些因子是从用户的反馈推断出来的. 因子可以解释成喜剧、悲剧或情节等这些明显的维度，也可以解释性格发展的深度或者“突变”等隐性特征. 在训练隐语义模型时，重要的参数有 4 个：因子个数 F，学习速率 γ，正则化参数 λ 和迭代次数 N. 实验发现，精心选择学习率 γ 和正则化参数 λ 可以提高准确度. F 的值越大越能体现现实复杂情况，但是相应的训练时间会大大增加. 本文依据文献[13] 选择参数经验值，$F=10$、$\gamma=0.02$、$\lambda=0.1$. SVD 训练算法如下：

输入：用户评分矩阵 $\boldsymbol{r}_{ui}$

输出：用户隐特征矩阵 $\boldsymbol{P}$，项目隐特征模型 Q

Begin

initialize：$P,Q,F,\gamma,\lambda,N,step=1$

while $step<N$

do

对 Γ 中的每个用户 u：

对 $N(u)$ 中的每个项目 i：

执行公式(2)；

计算 e_{ui}；

for $f\leftarrow 1,2,\cdots,F$：

执行公式(4)；

$step\leftarrow step+1$；

End

2.3 模型评估

本文的评估指标采用的是推荐算法中最常用的指标，即评估预测评分和实际评分的均方差(MAE)和均方根误差(RMSE)[14]，这 2 个指标的

优点是计算精度时对推荐系统的目标不需任何假设. 文献[15]表明，RMSE 值细微的变化都会对推荐系统的精确度产生很大影响. RMSE 的计算公式如下：

$$RMSE=\sqrt{\frac{1}{|\Gamma_{\text{test}}|}\sum_{u,i\in\Gamma_{\text{test}}}(\hat{r}_{ui}-r_{ui})^2},\quad(8)$$

式中 r_{ui} 通常是用户调查或在线获得的. 本文测试集 Γ_{test} 的 r_{ui} 是已知的. 为了评估模型，在实验中隐藏测试集中用户的部分评分数据. MAE 的计算公式如下：

$$MAE=\sqrt{\frac{1}{|\Gamma_{\text{test}}|}\sum_{u,i\in\Gamma_{\text{test}}}|\hat{r}_{ui}-r_{ui}|},\quad(9)$$

从式(8) 和式(9) 可看出，较大的 RMSE 或 MAE 都意味着较大的预测误差. 与 MAE 相比，RMSE 会惩罚大的误差，例如误差为{2,2,2,0} 和{3,0,0,0} 这两种情况时，RMSE 适用于分辨出第 1 种情况，而 MAE 适用于分辨出第 2 种情况.

3　实验结果与分析

实验数据采用 GroupLens 研究组的数据集 MovieLens 1M. 评分数据约有一百万条记录，由 6 040 名用户对 3 952 部电影进行评分，稀疏度达到 96.05%. 实验过程中将评分矩阵按比例 8∶2 分为训练集 Γ_{train} 和测试集 Γ_{test}. 首先，对训练集 Γ_{train} 进行融合项目属性的填充处理，然后通过 SVD 算法构建隐语义模型，结果如图 1 所示.

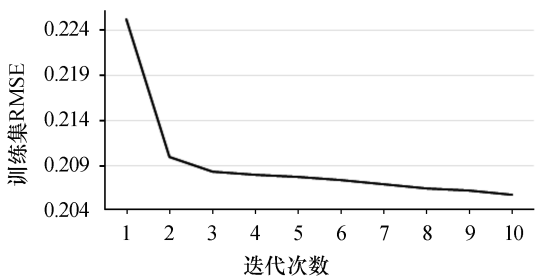

图 1　融合项目属性特征的模型建立过程

由图 1 可以看出，随着迭代次数的增加，模型对训练集的拟合程度增高. 模型评估过程是通过式(5)从隐语义模型中找到最相似的用户，用该用户隐空间的特征向量来表示当前用户 u，然后利用公式(8)和(9)计算 RMSE 和 MAE.

为验证算法的有效性，选择本文提出的 mix-SVD 方法与传统的基于项目的余弦相似度方法(IB-CF)、零填充隐语义模型(zero-SVD)和众数填充隐语义模型(mode-SVD)进行对比实验. 模型评估标准选择训练集拟合程度、测试集的 RMSE 和 MAE 3 个指标，对比结果如图 2 所示. 由图 2 可知：在测试集上，mix-SVD 与 zero-SVD 相比，其 RMSE 降低了 6.7%，MAE 降低了13.8%；mix-SVD 与 mode-SVD 相比，其 RMSE 降低了 4.8%，MAE 降低了 17.7%. mix-SVD 的拟合程度也好于其他 3 种方法，由此表明，混合填充的隐语义模型相比其他 3 个方法更好.

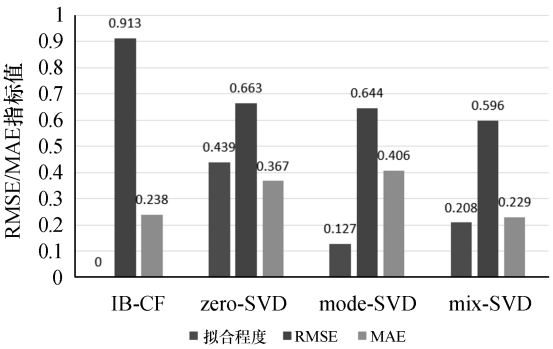

图 2　不同模型实验对比图

实验中发现，mode-SVD 对训练集拟合得非常好，而对于测试集的效果却并不理想，说明 mode-SVD 是强制按照用户习惯来填充的，由此造成了过拟合现象. 实验中还发现，在最后模型评估过程中，如果找到部分的相似用户集合，然后采用对该集合加权求和的方式来表示被推荐用户的特征向量，那么被推荐用户的精度会在模型训练的基础上进一步得到改善，如图 3 所示. 但在选择的相似用户超过 3 个以上时，精度提升得并不明显，甚至会出现不稳定情况，说明相似用户的数量选取不宜过大.

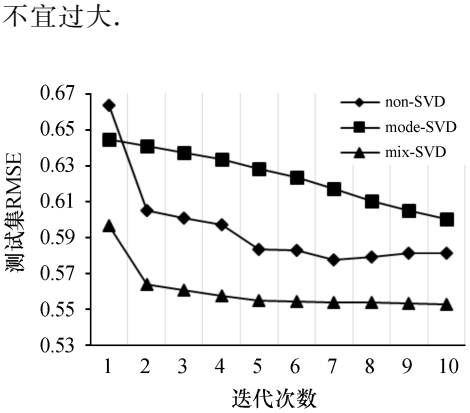

图 3　3 种模型对比实验的结果

4 结束语

本文提出一种融合项目属性特征的SVD协同过滤算法,该方法与相似度推荐方法、零填充方法以及众数填充方法相比,缓解了稀疏性带来的推荐精度下降问题,从而提高了推荐的精确度.本文的填充方法仅利用了项目的种类属性,因此融合的内容比较单一,而现实中的项目和用户的属性纷繁复杂,所以今后将在后续的工作中研究如何挖掘多模态的项目属性,以获得更高精度的推荐策略.

参考文献:

[1] 刘红霞.基于协同过滤技术的推荐系统[J].信息安全与技术,2016,7(3):24-26.

[2] Ponnam L T, Punyasamudram S D, Nallagulla S N, et al. Movie recommender system using item based collaborative filtering technique[C]// International Conference on Emerging Trends in Engineering, Technology and Science. Pudukkottai, India: IEEE, 2016.

[3] 张玉连,袁伟.隐语义模型下的科技论文推荐[J].计算机应用与软件,2015(2):37-40.

[4] 李博,陈志刚,黄瑞,等.基于LDA模型的音乐推荐算法[J].计算机工程,2016,42(6):175-179.

[5] Kumar R, Verma B K, Sunder S. Social popularity based SVD++ recommender system[J]. International Journal of Computer Applications, 2014, 87(14):33-37.

[6] 陈宗言,颜俊.基于稀疏数据预处理的协同过滤推荐算法[J].计算机技术与发展,2016,26(7):59-64.

[7] 谢海江,侯梦薇,赵季忠,等.一种多层混合的推荐模型研究[J].中国科技论文,2015,10(14):1660-1664.

[8] 姜维,庞秀丽.面向数据稀疏问题的个性化组合推荐研究[J].计算机工程与应用,2012,48(21):21-25.

[9] 孙明.基于语义的信息检索与关联推荐关键技术研究[D].成都:电子科技大学,2015.

[10] Hastie T, Mazumder R, Lee J D, et al. Matrix completion and low-rank SVD via fast alternating least squares[J]. Journal of Machine Learning Research, 2015,16(1):3367-3402.

[11] 周海平,黄凌英,刘妮,等.基于评分预测的协同过滤推荐算法[J].数据采集与处理,2016,31(6):1234-1241.

[12] 孙龙菲,黄梦醒.综合用户特征和项目属性的协作过滤推荐算法[J].计算机应用研究,2014,31(2):384-387.

[13] 项亮.动态推荐系统关键技术研究[D].北京:中国科学院研究生院,2011.

[14] McNee S M, Riedl J, Konstan J A. Being accurate is not enough: how accuracy metrics have hurt recommender systems [C]//CHI'06 Extended Abstracts on Human Factors in Computing Systems. Montréal, Québec, Canada: ACM, 2006: 1097-1101.

[15] Said A, Bellogin A, Alejandro N. Comparative recommender system evaluation benchmarking recommendation frameworks[C]//ACM Conference on Recommender Systems, 2014. CA, USA: RecSys, 2014: 129-136.