基于SVD++隐语义模型的信任网络推荐算法

陈佩武[1]，束方兴[2]
1. 平安科技（深圳）有限公司，广东 深圳 518031；
2. 北京大学互联网研究院（深圳），广东 深圳 518055

摘要
推荐算法通常基于用户的行为数据进行建模，然而显式行为数据的稀疏性可能会引起推荐算法的冷启动问题。为了降低数据稀疏和冷启动问题对推荐算法效果的影响，在已有显式信任关系的基础上，基于用户相似度引入隐式信任关系，通过SVD++隐语义模型设计了新的推荐算法。为了提升算法效果，进一步融合邻域模型，推导出算法评分预测式及损失函数。在Epinions开源数据集中将RMSE和MAE作为测试指标，在全体用户集和冷启动用户集上进行对比实验。实验结果显示，设计的推荐算法可以在一定程度上改善原推荐算法的冷启动问题，并取得更好的评分预测效果。

关键词
推荐算法；隐语义模型；信任网络；评分预测

中图分类号：TP315 文献标识码：A doi: 10.11959/j.issn.2096-0271.2021041

A recommender algorithm based on SVD ++ model under trust network

CHEN Peiwu[1], SHU Fangxing[2]
1. Ping An Technology (Shenzhen) Co., Ltd., Shenzhen 518031, China
2. Internet Research Institute, Peking University, Shenzhen 518055, China

Abstract
Recommender algorithms are usually modeled based on user behavior data. However, the sparseness of explicit behavior data may cause the cold start problem of recommender algorithms. In order to solve the impact of data sparseness and cold-start problems on the effect of recommender algorithms, implicit trust relationship based on user similarity was introduced based on the existing revealed trust relationship, and a new recommender algorithm was designed through the SVD++ implicit semantic model. In order to improve the effect of the algorithm, the neighborhood model was integrated further, and the algorithm score prediction formula and loss function were derived. In the Epinions open source data set, RMSE and MAE were used as test indicators, and comparative experiments were conducted on

the entire user set and the cold start user set. The experimental results show that the recommender algorithm can optimize the cold start problem of the original recommender algorithm to a certain extent, and achieve a better rating prediction accuracy.

Key words
recommender algorithm, latent factor model, trust network, rating prediction

1 引言

随着互联网技术的全面发展，海量数据得以产生，推动人类社会从信息匮乏的时代走入了信息过载的时代。作为解决信息过载问题的利器，推荐系统受到各大互联网公司以及研究机构的青睐。作为推荐系统的核心，推荐算法成为研究的关键和热点。然而传统的推荐算法（如协同过滤（collaborative filtering，CF）算法）存在评分数据库的数据稀疏性和用户偏好信息数据有限等问题，即推荐系统的数据稀疏和冷启动问题。以上问题会制约推荐系统的推荐效果，破坏用户体验。为了缓解数据稀疏和冷启动问题，通常会在推荐系统中引入新维度的数据。然而近年来，社交网络的快速发展给推荐算法的研究带来了新的推动力[1]。在社交网络中，用户不但会展示自己的个性和偏好，还会与偏好类似、相互信任的其他用户构建联系。因此如何进一步在社交网络的研究中利用社会化信息进行信任构建，提升推荐算法的有效性，成为一个重要的研究课题。

隐语义模型（latent factor model，LFM）最早是在文本挖掘邻域被提出的，其主要被用来寻找文本中的隐含语义。虽然学术界已经提出了多个基于隐语义模型改进的推荐算法，但是信任网络中还少有以隐语义模型为基础进行改进的推荐算法[2]，且效果并不理想。为了优化推荐算法的效果，本文采用以隐语义模型为基础，与其他模型融合的思路进行推荐算法设计。另外，评分预测（rating prediction）也是推荐系统研究的关键问题，即通过已知的用户历史评分记录来预测未知的评分。因此，基于隐语义模型与其他模型融合设计推荐算法，并从评分预测的角度进行实验验证，具有重要的研究意义和实用价值。本文结合信任网络的特点与用户关系数据进行建模，基于用户之间的相似度设定用户之间的隐式信任关系，结合显式信任关系与评分等数据一起进行预处理，生成矩阵。在隐语义模型选择方面，本文借鉴奇异值分解（single value decomposition，SVD）++模型[3]，结合信任网络的特点进行改进，并融合邻域模型，设计了推荐算法的评分预测函数和损失函数，利用随机梯度下降算法进行迭代求导，得到优化后的推荐算法评分预测值。之后本文在Epinions开源数据集上进行离线对比实验，验证了本文设计的推荐算法在信任网络中的推荐效果优于其他对比算法，对于解决推荐系统的冷启动问题有良好效果。

本文主要的贡献如下。

- 与传统仅基于用户的物品评分记录的推荐算法以及相关模型相比，本文加入了信任网络背景，通过对信任网络特点和用户行为进行分析，将用户反馈数据分为隐式行为数据和显式行为数据，并进行整合，利用信任信息对推荐算法的评分效果进行改善。
- 在信任网络的研究背景下，本文借

鉴SVD++隐语义模型进行推荐算法的设计和实现，融合显式信任因子及隐式信任因子优化算法的推荐评分预测效果。

● 本文以隐语义模型为算法基础，借鉴了基于邻域模型的推荐算法思想，将两种模型从全局优化的角度进行融合，并从评分预测的角度对推荐算法设计了对比实验。本文使用评测指标平均绝对误差（mean absolute error，MAE）和均方根误差（root mean square error，RMSE）分别在冷启动数据集和全体数据集中对比加入信任关系前后的SVD++算法的效果，从而证明信任网络的加入对原有推荐算法的提升效果。然后将本文设计的推荐算法与其他信任网络下的典型算法进行对比测试，并证明了本文设计的算法在评分预测问题上的优势。

2 研究背景

2.1 推荐系统的发展

针对推荐系统的研究开始于20世纪90年代，随着网络的普及，该技术被逐步应用到各个行业。推荐系统主要由用户建模部分、推荐对象建模部分以及推荐算法部分共同组成，其中推荐算法部分是整个推荐系统的核心，也是研究的关键和热点。目前被广泛使用的协同过滤推荐算法[4]来源于Tapestry和GroupLens系统的论文[5-6]，该算法被用于邮件和新闻的过滤。协同过滤算法也被称为基于邻域模型的推荐算法，即邻域实质相似项的集合。用户间存在相似的兴趣，或者某些物品间存在相似的特征，因此基于邻域的推荐算法又被分为两类：基于用户的协同过滤（UserCF）推荐算法和基于物品的协同过滤（ItemCF）推荐算法[7]。UserCF算法主要包括两步：找出与目标用户有相似兴趣的用户集；把用户集中其他用户喜欢且目标用户没标注过的物品推荐给目标用户。余弦相似度可用下式计算：

$$\mathrm{sim}\cos(a,b)=\frac{|S(a)\cap S(b)|}{\sqrt{|S(a)||S(b)|}} \quad (1)$$

其中，$S(a)$和$S(b)$分别表示用户a、b有反馈信息的物品集合。在推荐系统中，用户数量往往较大，用户之间两两计算相似度会带来较高的算法复杂度[8]，因此如何定义用户间的相似度成为优化研究的重点，如Breese J S等人[9]提出利用用户行为进行相似度计算。

由于UserCF算法复杂度较高，目前Amazon、Netflix等平台都将ItemCF算法作为推荐算法的基础[10]。可使用ItemCF算法向用户推荐他们喜欢的物品的相似产品，该算法并不使用物品的内容属性进行相似度计算，而是通过分析用户的行为数据来计算不同物品之间的相似度。ItemCF算法的步骤包括两步：进行物品间的相似度计算；基于用户的行为数据特点推荐产品。ItemCF算法的评分预测式如下：

$$\hat{r}_{ai}=\overline{r}_i+\frac{\sum_{j\in R(j,k)\cap I(u)}\mathrm{Si}(i,j)(r_{a,j}-\overline{r}_J)}{\sum_{b\in R(i,k)\cap I(u)}\mathrm{Si}(i,j)} \quad (2)$$

其中，Si(i,j)表示物品i与j之间的相似度，$R(i,k)$表示前k个与物品i相似的物品，$I(u)$表示被用户u评论的物品集合，$r_{a,j}$表示用户a对物品j的兴趣，$\overline{r}_j$表示感兴趣的均值。在有用户m个、产品n个、用户行为记录k个、隐类f个、算法迭代d次的情况下，基于邻域模型的推荐算法和基于隐语义模型的推荐算法各有优势，具体见**表1**。如果在推荐算法的设计中将两类模型进行融合，可能会起到优势互补的作用，从而获得更好的推荐效果。

隐语义模型的思想最初于文本挖掘领域被提出，后来扩展改进的方法包

表1 隐语义模型和邻域模型特点

对比项	隐语义模型	邻域模型
理论基础	有监督的机器学习	统计、协同过滤
时间复杂度	$O(k \cdot f \cdot d)$	UserCF：$O(n \cdot (k/n)^2)$ ItemCF：$O(m \cdot (k/m)^2)$
空间复杂度	$O((m+n) \cdot f)$	UserCF: $O(m \cdot m)$ ItemCF: $O(n \cdot n)$
推荐解释	无法提供推荐原因解释	ItemCF可以解释推荐结果
实时推荐	每次生成用户推荐列表时需要重新计算，不适合实时推荐计算	UserCF和ItemCF可将用户和物品表缓存，从而进行实时预测

括隐含狄利克雷分布（latent Dirichlet allocation，LDA）主题模型[11]、概率潜在语义分析（probabilistic latent semantic analysis，PLSA）、主题建模（topic model）等。通过矩阵降维的方式进行评分矩阵补全、应用隐语义模型应对推荐系统中冷启动问题成为一个有效方法。奇异值分解是一种用于发现文本中潜在因子的方法，它将高度相关并共同出现的相似词语作为因子，从而把大规模的文本向量矩阵拆解为低阶的相似矩阵。除此之外，相关的方法还有主成分分析（principal component analysis，PCA）[12]和隐含狄利克雷分布主题模型等。早期，评分矩阵稀疏且SVD算法计算复杂度较高，导致基于SVD的推荐算法没有引起重视。直到Koren Y等人[3]在Funk-SVD的基础上加入偏置项，并且把用户历史行为考虑在内提出SVD++模型之后，才大大提升了算法的效果。

2.2 评分预测与Top*N*推荐问题

利用已有数据信息来预测用户对未评分的物品集合的评分的研究被称为推荐系统研究中的评分预测[12]。在评分预测研究中，基本数据集是用户的历史评分数据，在该数据集中，通常每条记录数据用三元组(u,i,r)来表示。其中，u表示用户，i表示物品，r表示对该物品的评分。评分预测的准确性可以通过RMSE和MAE来衡量。在推荐系统中，真实值r_{ui}是用户u对物品i的评分，预测值$\hat{r}_{ui}$是推荐系统给出的预测结果，又因为在离线实验中数据集被分为测试集和训练集，所以观测次数在这里指的是测试集的大小。RMSE和MAE的定义如下：

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{(u,i)\in T}(r_{ui}-\hat{r}_{ui})^2}{|\mathrm{Test}|}} \quad (3)$$

$$\mathrm{MAE}=\frac{\sum_{(u,i)\in T}|r_{ui}-\hat{r}_{ui}|}{|\mathrm{Test}|} \quad (4)$$

其中，T表示整个测试集，Test表示实际测试时使用的测试集。

从指标的定义可以看出，两个测试指标的值与评分预测的准确性呈现负相关的趋势，即测评指标越小，评分预测准确性越高。相较于MAE指标，RMSE指标因为平方和的形式更加放大了误差的影响，对于推荐算法的评测来说，RMSE更加严格。

Top*N*推荐问题指在给用户的个性化推荐列表中，设置哪些产品进行优先呈现[13]。Top*N*推荐预测的效果一般使用准确率（precision）和召回率（recall）来衡量，两者分别表示有多少比例的推荐选项被用户选中以及有多少比例的推荐选项进入了用

户的最终列表，其定义如下：

$$\text{precision}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \quad (5)$$

$$\text{recall}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \quad (6)$$

其中，$R(u)$表示训练集上生成的给用户的推荐列表，$T(u)$表示用户在测试集上生成的行为列表，U表示用户集。目前大多数推荐算法的相关研究是基于用户的评分数据的[14]，因此有许多研究人员探索如何优化RMSE和MAE两个评分指标。同时Top$N$推荐问题也受到产业界各大互联网公司的关注[15]。本文结合研究背景进行实验设计，主要从评分预测的角度来定义推荐算法，并进行评测。

3 算法研究与模型训练

3.1 基于SVD++的信任网络推荐算法设计

本文在信任网络的研究背景下，从推荐算法研究中的评分预测问题出发设计推荐算法，即在确定用户的相关历史数据和相关信任关系的情况下，对用户未评分的物品进行评分预测。传统的隐语义模型主要利用矩阵因子分解的方式，从评分预测的角度定义推荐算法的损失函数，没有完全考虑用户的历史行为数据[16]。Koren Y提出的Simon模型加入了用户的历史行为数据，其设计的SVD++算法取得了良好的评分预测效果。Simon模型的评分预测式如下：

$$\hat{r}_{ui}=b_{ui}+\boldsymbol{v}_i^{\mathrm{T}}(\boldsymbol{u}_u+|N(u)|^{-\beta}\sum_{j\in N(u)}y_j) \quad (7)$$

其中，b_{ui}表示预测的基准估计值，$\boldsymbol{u}_u$表示从评分矩阵中分解的用户向量，$\boldsymbol{u}_u$与$\boldsymbol{v}_i$的向量乘积表示用户的显式评分对预测结果的影响。$N(u)$表示与用户u的潜在偏好有联系的群体，y_j表示$N(u)$群体的隐式反馈。因为数据集具有稀疏性特点，所以隐式数据的体量多于显式数据，因而本文设置正则化处理系数$-\beta$，$\beta\in[0,1]$。$\beta=0$表示隐式反馈影响较大，$\beta=1$则表示隐式反馈的影响几乎被抵消，本文选取$\beta=0.5$。结合信任网络中的奇异值分解，首先将用户向量与信任者向量进行统一，之后逐步把信任关系代入SVD++模型：

$$\hat{r}_{ui}=b_{ui}+\boldsymbol{v}_i^{\mathrm{T}}(\boldsymbol{u}_u+|N(u)|^{-0.5}\sum_{j\in N(u)}y_j+\sum_{v\in T(u)}p_v) \quad (8)$$

其中，p_v表示可以反映显式信任的被信任者，$\boldsymbol{v}_i^{\mathrm{T}}p_v$表示用户$u$信任的用户$v$对用户$u$进行未知物品评分预测的影响，$\boldsymbol{v}_i^{\mathrm{T}}u_u$表示用户$u$作为显式信任者对评分预测的影响。然后需要考虑隐式信任关系的影响，同理可用SVD将隐式信任关系的矩阵进行分解：

$$\hat{t}_{uv}^{i}=\boldsymbol{w}_v^{\mathrm{T}}\boldsymbol{u}_u \quad (9)$$

其中，$\boldsymbol{w}_v^{\mathrm{T}}$表示被隐式信任的用户的因子向量，$\boldsymbol{u}_u$表示隐式信任用户的因子向量。因此可将$\boldsymbol{w}_v^{\mathrm{T}}$放入评分预测式中，得到：

$$\hat{r}_{ui}=b_{ui}+\boldsymbol{v}_i^{\mathrm{T}}(\boldsymbol{u}_u+|N(u)|^{-0.5}\sum_{j\in N(u)}y_j+\sum_{v\in T(u)}p_v+|T_i(u)|^{-0.5}\sum_{v\in T_i(u)}\boldsymbol{w}_v) \quad (10)$$

其中，$T_i(u)$表示与用户u有相关隐式信任关系的群体。为了使推荐算法在评分预测过程中取得良好效果，需要解出评分预测式中的未知系数，本文通过确定损失函数将求解未知系数转换为对损失函数最小值的求解问题。将待优化的损失函数设定如下：

$$L=\sum_{(u,i)\in\Omega}(r_{ui}-\hat{r}_{ui})^2+\lambda_{t1}\sum_{(u,v)\in\Phi}(t_{uv}-\hat{t}_{uv})^2+\lambda_{t2}\sum_{(u,v)\in\Lambda}(t^{i}_{uv}-\hat{t}^{i}_{uv})^2 \quad (11)$$

其中，r_{ui}表示获得的评分，$\hat{r}_{ui}$表示预测得分，t_{uv}表示用户v对用户u的显式信任值，t^{i}_{uv}代表用户v对用户u的隐式信任值，Ω表示用户与物品集，Φ表示用户显式信任关系

集，Λ表示用户间的隐式信任关系集，λ_{t1}与λ_{t2}分别表示控制显式及隐式信任的正则化影响因子。在机器学习的拟合过程中，常会遇到过度拟合的情况，在优化损失函数的最小值时因为过度拟合样本误差，导致样本点外的函数计算结果偏移较大。本文使用一个过拟合参数λ来避免过拟合，加入λ后得到的损失函数如下：

$$L=\sum_{(u,i)\in\Omega}(r_{ui}-\hat{r}_{ui})^2+\lambda_{t1}\sum_{(u,v)\in\Phi}(t_{uv}-\hat{t}_{uv})^2+\lambda_{t2}\sum_{(u,v)\in\Lambda}(t^i_{uv}-\hat{t}^i_{uv})^2+\lambda(b_u^2+b_i^2+\sum_i\|v_i\|_F^2+\sum_u\|u_u\|_F^2+\sum_j\|y_j\|_F^2+\|p_v\|_F^2+\|w_v\|_F^2) \tag{12}$$

对损失函数中加入的避免过拟合项进行正则化处理，并进行系数处理，得到基于隐语义模型的信任网络推荐算法的损失函数：

$$2L=\sum_{(u,i)\in\Omega}(r_{ui}-\hat{r}_{ui})^2+\lambda_{t1}\sum_{(u,v)\in\Phi}(t_{uv}-\hat{t}_{uv})^2+\lambda_{t2}\sum_{(u,v)\in\Lambda}(t^i_{uv}-\hat{t}^i_{uv})^2+\lambda(b_u^2+b_i^2)+\sum_u(\lambda|N(u)|^{-0.5}+\lambda_{t2}|T_i(u)|^{-0.5})\|u_u\|_F^2+\lambda(\sum_j|U(j)|^{-0.5}\|y_j\|_F^2+\sum_i|U(i)|^{-0.5}\|v_i\|_F^2)+\lambda(|X(v)|^{-0.5}\|p_v\|_F^2+|\mathrm{Xi}(v)|^{-0.5}\|w_v\|_F^2) \tag{13}$$

其中，U_i和U_j分别表示对物品i和物品j进行过评分的用户集，$X(v)$和$\mathrm{Xi}(v)$分别表示显式信任用户集合以及隐式信任用户集合，正则化的惩罚规则主要用于可能造成评分过拟合的冷门对象，完成基于SVD++隐语义模型针对信任网络进行的推荐算法的评分预测式以及损失函数的设计，下一步进行隐语义模型与邻域模型的融合。

3.2 信任网络中邻域模型和隐语义模型的推荐算法融合

基于邻域模型的推荐算法分为ItemCF算法和UserCF算法两类，本文将从ItemCF算法的角度设计适合与隐语义模型进行融合的邻域模型。两类模型基于同一数据集，只是从不同的角度来推测用户的兴趣。本文把两类模型的不确定参数放入同一损失函数中进行求解，然后对评分预测的结果进行拟合，从而将两类模型进行融合。整合邻域模型的评分预测式，将评分预测式进行正则化后，将其中的基准评分估计值设置为隐语义模型中的评分预测式（式（10）），则得到式（14）：

$$\hat{r}_{ui}=u+b_i+b_u+\boldsymbol{v}_i^{\mathrm{T}}(u_u+|N(u)|^{-0.5}\sum_{j\in N(u)}y_j+\sum_{v\in T(u)}p_v+|T_i(u)|^{-0.5}\sum_{v\in T_i(u)}w_v)+|R(u)|^{-0.5}\sum_{j\in R(u)}(r_{uj}-b_{uj})s_{ij}+|N(u)|^{-0.5}\sum_{j\in N(u)}c_{ij} \tag{14}$$

其中，w_v表示用户显式反馈的未知参数，s_{ij}表示用户i与j之间的相关性对基准估值的偏移度，c_{ij}表示用户i对j的隐式反馈对评分预测带来影响的权重，以优化算法的空间复杂度和时间复杂度。本文使用式（14）作为评分预测式，参考式（13）得到融合后的预测损失函数：

$$2L=\sum_{(u,i)\in\Omega}(r_{ui}-\hat{r}_{ui})^2+\lambda_{t1}\sum_{(u,v)\in\Phi}(t_{uv}-\hat{t}_{uv})^2+\lambda_{t2}\sum_{(u,v)\in\Lambda}(t^i_{uv}-\hat{t}^i_{uv})^2+\lambda(b_u^2+b_i^2)+\sum_u(\lambda|N(u)|^{-0.5}+\lambda_{t2}|T_i(u)|^{-0.5})\|u_u\|_F^2+\lambda(\sum_j|U(j)|^{-0.5}\|y_j\|_F^2+\sum_i|U(i)|^{-0.5}\|v_i\|_F^2)+\lambda(|X(v)|^{-0.5}\|p_v\|_F^2+|\mathrm{Xi}(v)|^{-0.5}\|w_v\|_F^2)+\lambda(\sum_{j\in R(u)}s_{ij}^2+\sum_{j\in N(u)}c_{ij}^2) \tag{15}$$

综上，式（14）与式（15）即基于隐语义模型和邻域模型融合后的适用于信任网络的推荐算法评分预测式与预测损失函数。

3.3 模型的迭代训练

对目标函数的最优值进行求解通常会使用梯度下降（gradient desent）法，其中梯度为多元函数的偏导向量。逐步求出损失函数的最小值，需要确定每个步骤的步

长以及方向，通过梯度计算可以得到函数下降最快的方向。对于未确定参数，采用梯度下降算法，可表示为：

$$\theta \leftarrow \theta - \alpha \cdot \frac{\partial L}{\partial \theta} \tag{16}$$

如**图1**所示，对于参数θ而言，损失函数L对其求偏导即表示梯度，其中α表示该方向上的步长，也被称为学习速率。梯度算法可以分为两类：随机梯度下降（stochastic gradient descent，SGD）算法、批梯度下降（batch gradient descent，BGD）算法。其中，BGD算法为通过最小化所有训练样本的损失函数而得到的全局最优解，而SGD算法则大体向全局最优的方向求解，但是SGD算法可以不用每次迭代过程中都使用全体训练样本，因此SGD的复杂度相对较小，本文使用复杂度优先的SGD算法。

首先，定义评分值、显式信任数值以及隐式信任数值的表达误差：

$$e_{ui} = r_{ui} - \hat{r}_{ui} \tag{17}$$

$$e_{uv} = t_{uv} - \hat{t}_{uv} \tag{18}$$

$$e_{uv}^{i} = t_{uv}^{i} - \hat{t}_{uv}^{i} \tag{19}$$

损失函数对参数b_u、b_i、u_u、v_i、y_j、p_v、w_v、s_{ij}、c_{ij}求偏导：

$$\frac{\partial L}{\partial b_u} = \sum\nolimits_{i \in Iu} e_{ui} + \lambda b_u \tag{20}$$

$$\frac{\partial L}{\partial b_i} = \sum\nolimits_{u \in U(i)} e_{ui} + \lambda b_i \tag{21}$$

$$\frac{\partial L}{\partial u_u} = \sum\nolimits_{i \in Iu} e_{ui} v_i + \lambda_{t1} \sum\nolimits_{v \in Tu} e_{uv} p_v + \lambda_{t2} \sum\nolimits_{v \in T^i u} e_{uv}^{i} w_v + (\lambda |N(u)|^{-0.5} + \lambda_{t2} |Ti(u)|^{-0.5}) u_u \tag{22}$$

$$\frac{\partial L}{\partial v_i} = \sum\nolimits_{u \in U(i)} e_{ui} (u_u + |N(u)|^{-0.5} \sum\nolimits_{j \in N(u)} y_j + \sum\nolimits_{v \in Tu} p_v + |Ti(u)|^{-0.5} \sum\nolimits_{v \in T^i u} w_v) + \lambda |U(i)|^{-0.5} v_i \tag{23}$$

$$\forall j \in N(u): \frac{\partial L}{\partial y_j} = \sum\nolimits_{i \in Iu} e_{ui} |N(u)|^{-0.5} v_i + \lambda |U(i)|^{-0.5} y_j \tag{24}$$

$$\frac{\partial L}{\partial p_v} = \sum\nolimits_{i \in Iu} e_{ui} v_i + \lambda_{t1} e_{uv} u_u + \lambda |X(v)|^{-0.5} p_v \tag{25}$$

$$\forall v \in Ti(u): \frac{\partial L}{\partial w_v} = \sum\nolimits_{i \in Iu} e_{ui} |Ti(u)|^{-0.5} v_i + \lambda_{t1} e_{uv} u_u + \lambda |X(v)|^{-0.5} p_v \tag{26}$$

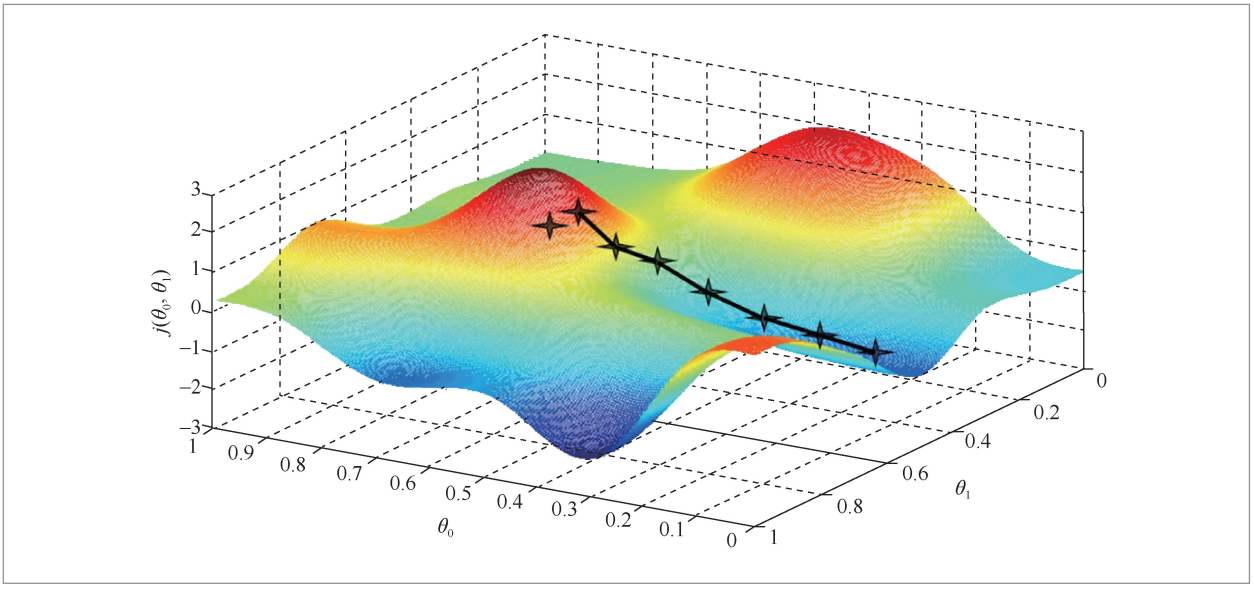

图1 采用梯度下降算法求解损失函数的最优解

$$\frac{\partial L}{\partial s_{ij}} = \sum_{i \in Iu} e_{ui} \left| R(u) \right|^{-0.5} \sum_{j \in R(u)} (r_{uj} - b_{uj}) + \lambda_{t1} s_{ij} \quad (27)$$

$$\frac{\partial L}{\partial c_{ij}} = \sum_{i \in Iu} e_{ui} \left| N(u) \right|^{-0.5} + \lambda c_{ij} \quad (28)$$

由式(17)~式(19)的计算可以得到参数的偏导结果，下面给出其关键运算步骤的伪代码，算法1的作用是通过文本进行数据输入预处理，生成3个矩阵，算法2为采用随机梯度下降算法进行推荐算法迭代训练模型的伪代码流程，并在最后给出其评分预测结果。

算法1：输入数据预处理

1. R = Matrix(m,n), T = Matrix(m,m), Ti = Matrix(m,m) //3个空矩阵声明

2. open("ratings.txt", "trust.txt") as file_r, file_t; //读取源文件数据

3. for line_r, line_t in file_t, file_t:

4. addToMatrix(line_r, R);

5. addToMatrix(line_t, T);

//逐行进行矩阵***R***、***T***的构建

6. Ti = PCC(R); //采用PCC进行相似度计算，并转换为隐式信任矩阵

算法2：利用本文的推荐算法进行评分预测

1. Input: R, T, Ti, d, λ, λ_{t1}, λ_{t2}, α, iter

2. for count = 1 to iter do:

3. $b_u \leftarrow b_u - \alpha \frac{\partial L}{\partial b_u}$, u = 1, … , m

4. $b_i \leftarrow b_i - \alpha \frac{\partial L}{\partial b_i}$, i = 1, … , n

5. $u_u \leftarrow u_u - \alpha \frac{\partial L}{\partial u_u}$, u = 1, … , m

6. $v_i \leftarrow v_i - \alpha \frac{\partial L}{\partial v_i}$, i = 1, … , n

7. $y_j \leftarrow y_j - \alpha \frac{\partial L}{\partial y_j}$, $\forall j \in N(u)$

8. $p_v \leftarrow p_v - \alpha \frac{\partial L}{\partial p_v}$,

9. $w_v \leftarrow w_v - \alpha \frac{\partial L}{\partial w_v}$, $\forall v \in Ti(u)$

10. $s_{ij} \leftarrow s_{ij} - \alpha \frac{\partial L}{\partial s_{ij}}$

11. $c_{ij} \leftarrow c_{ij} - \alpha \frac{\partial L}{\partial c_{ij}}$

12. end

13. Output: RMSE, MAE

在算法的复杂度分析方面，本文设定t为损失函数的迭代次数，d为隐语义模型降维后的向量维度，$|\Omega|$为评分矩阵中所有非空项的数目，$|\Phi|$为显式信任矩阵中所有非空项的数目，$|\Lambda|$为隐式信任矩阵中所有非空项的数目。因为在求解损失函数的过程中包含了评分预测值的计算，所以在计算时间复杂度时需要先通过式(15)计算评分预测损失函数的时间复杂度，为$O(td(|\Omega|+|\Phi|+|\Lambda|))$。然后由式(17)~式(19)求导得：$O(td|\Omega|)$、$O(td(|\Omega|+|\Phi|+|\Lambda|))$、$O(td(|\Omega|+|\Lambda|))$。假设$|K|=\max((|\Omega|,|\Phi|,|\Lambda|))$，则可以得到本文推荐算法的时间复杂度为$O(td|K|)$，其与隐语义模型的向量维度和训练迭代的次数线性相关。

4 实验

本文的实验部分主要是在开源数据集Epinions上对本文设计的推荐算法与经典的TrustMF算法进行对比，实验方法采用离线实验。将离线的实验数据集分为训练集和测试集，在训练集上进行用户的兴趣模型训练，然后在测试数据集上对训练的模型进行预测测试，并用相关指标算法进行评分。在对比对象的选择方面，选择TrustMF的原因是：相较于其他相关研究信任网络下的推荐算法(如SoRech[17]、SocialMF[18]等相关算法[19-22])，该算法可以在评分预测问题中取得较好的成绩。

本文实验选择的Epinions数据集为

开源数据，分为两个部分：rating_data和trust_data，其中rating_data为用户历史完成评分的数据，格式为三元组(user_id, item_id, rating_value)；trust_data为表示用户间信任关系的数据，格式为三元组形式(source_user_id, target_user_id, trust_statement_value)；该数据集的基本信息见**表2**、**表3**。

实验评测指标主要选取RMSE和MAE，这两个指标数值都反比于推荐预测的准确性，即指标数值越小，则误差越小，准确性越高。在进行实验前，将Epinions数据集随机划分为5份，1份是测试集，其余4份是训练集，算法中已知参数的设置见**表4**。

为了说明本文设计的算法对冷启动问题的优化作用，实验先从推荐系统冷启动的角度开始进行。推荐系统的冷启动是指在进行评分预测时，测试集中只选用评分物品数量小于5的用户来进行测试。实验结果见**表5**。

从实验结果可以得出，除个别情况下某一指标（d=10时本文算法的MAE略大于TrustMF的MAE）稍差，其他情况下本文算法都优于其他各个推荐算法的冷启动效果。可以证明，相比于其他参考算法，本文的推荐算法在解决冷启动问题方面更好。迭代优化实验则使用所有用户作为测试集，得到的评分预测结果见**表6**。

从**表6**可以看出，所有算法的实验评测效果均好于冷启动实验组的评测结果。这

表 2 Epinions数据集基本信息

数据集	用户数	物品数	评分数	评分数据密度	信任者数	被信任者数	信任关系数	信任数据密度
Epinions	40 163	139 738	664 824	0.011 8%	33 960	49 290	487 181	0.029%

表 3 Epinions数据集数据范围

数据集	用户ID范围	物品ID范围	评分值范围	信任者 ID范围	被信任者ID范围	信任值范围
Epinions	[1, 49 290]	[1, 139 738]	[1, 5]	[1, 49 290]	[1, 49 290]	1 / null

表 4 算法中已知参数

参数	隐语义因子数d	迭代次数iter	学习速率α	λ	λ_{t1}	λ_{t2}
值	5 / 10	100	0.01	0.9	0.8	0.5

表 5 冷启动测试评测结果

隐向量维度	测评指标	UserMean	ItemMean	SVD++	TrustMF	本文算法	提高程度
d=5	MAE	1.147	0.902	0.889	0.890	0.870	2.1%
d=5	RMSE	1.430	1.127	1.162	1.109	1.102	2.2%
d=10	MAE	1.147	0.902	0.891	0.853	0.860	−0.8%
d=10	RMSE	1.430	1.127	1.166	1.114	1.102	1.1%

表 6 使用数据集中所有用户进行测试的评测结果

隐向量维度	测评指标	UserMean	ItemMean	SVD++	TrustMF	本文算法	提高程度
d=5	MAE	0.932	0.928	0.818	0.821	0.805	1.6%
d=5	RMSE	1.216	1.101	1.057	1.058	1.042	1.4%
d=10	MAE	0.932	0.928	0.818	0.815	0.804	1.3%
d=10	RMSE	1.216	1.101	1.056	1.077	1.042	1.3%

是因为在冷启动时遇到的数据稀疏问题影响了推荐系统的效果。而采用全部用户集上的数据时，因为包括热门用户和产品，所以提升了算法的推荐效果。同时从实验结果看，相比其他算法，本文算法的评测结果有一定的优化提升。

为了进一步了解信任网络对推荐算法优化的影响，本文以用户在信任网络中的度为依据进行用户群体的划分设计测试集。将用户视为图论中的点，修改点的出度和入度之和就是修改点在信任网络图中的度。本文取隐式向量维度d=10，与TrustMF进行对比测试，如**图2**所示。

从**图2**可以看出，随着信任网络中度数的增加，推荐算法对用户兴趣的预测准确率基本呈上升趋势。而且，在多数情况下，本文的推荐算法因为结合了隐语义模型和显式信任关系，推荐结果的评分预测精度更高，效果优于对比算法TrustMF。

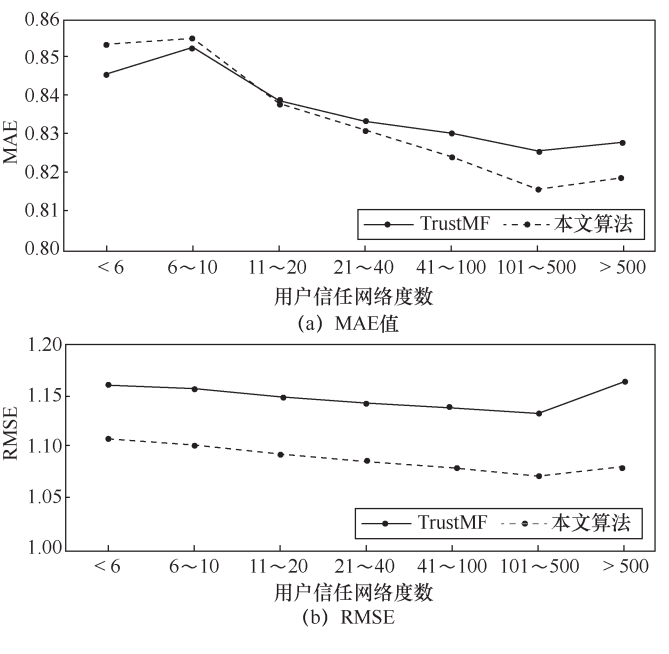

图2 当d=10时，不同信任网络度数的MAE值和RMSE值

5 结束语

作为推荐系统的核心，对推荐算法的研究受到学术界和产业界的广泛关注和重视，评分预测是推荐算法研究中的核心问题。传统推荐算法在面对显式行为数据稀疏时容易出现冷启动的问题，为了缓解该问题对推荐算法进行评分预测的准确度的影响，本文结合信任网络的特点和用户信任关系数据进行建模，设计了适用于信任网络的推荐算法，并进行了相关实验评测。本文将用户行为数据分为显式行为数据和隐式行为数据，因为输入的用户关系数据中具有二元显式信任关系，本文基于用户之间的相似度来设定用户之间的隐式信任关系，结合显式信任关系数据、评分数据以及隐式信任关系数据一起进行预处理，生成矩阵，进行建模。

在隐语义模型的选择上，本文借鉴SVD++模型，结合信任网络的特点进行改进，并融合了邻域模型进行推荐算法的评分预测函数和损失函数的设计。然后利用随机梯度下降算法进行函数中的参数迭代求导，直至损失函数收敛于最小值，从而求得优化后的相关参数，将其代入评分预测函数后得到推荐算法评分预测值。最后本文通过开源数据集，采用离线测试的方案，使用RMSE和MAE两个典型评测指标对本文设计的推荐算法和TrustMF算法进行评测实验。实验结果证明，本文设计的推荐算法在信任网络中优于其他参照算法，对于解决推荐系统的冷启动问题有良好的效果，并具有一定的实用意义。

参考文献:

[1] NIE F P, WANG X Q, HUANG H.

Clustering and projected clustering with adaptive neighbors[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2014: 977–986.

[2] WANG X B, LEI Z, GUO X J, et al. Mult-view subspace clustering with intactness-aware similarity[J]. Pattern Recognition, 2019, 88: 50–63.

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

[4] RESNICK P, IACOVOU N, SUCHAK M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]// Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. New York: ACM Press, 1994: 175–186.

[5] GOLDBERG D, NICHOLS D, OKI B M, et al. Using collaborative filtering to weave an information Tapestry[J]. Communications of the ACM, 1992, 35(12): 61–70.

[6] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. New York: ACM Press, 2001: 285–295.

[7] YANG B, LEI Y, LIU D Y, et al. Social collaborative filtering by trust[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(8): 1633–1647.

[8] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76–80.

[9] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. New York: ACM Press, 1998: 43–52.

[10] GOLDBERG K, ROEDER T, GUPTA D, et al. Eigentaste: a constant time collaborative filtering algorithm[J]. Information Retrieval, 2001, 4(2): 133–151.

[11] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. The Journal of Machine Learning Research, 2003, 3: 993–1022.

[12] KOREN Y. Factor in the neighbors: scalable and accurate collaborative filtering[J]. ACM Transactions on Knowledge Discovery from Data, 2010, 4(1).

[13] XIANG L, YUAN Q, ZHAO S W, et al. Temporal recommendation on graphs via long- and short-term preference fusion[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2010: 723–732.

[14] 贾俊, 张斌, 李志远. 基于用户行为分析的个性化推荐算法[J]. 智能科学与技术学报, 2019, 1(4): 421–426.
JIA J, ZHANG B, LI Z Y. Personalized recommendation algorithm based on user behavior analysis[J]. Chinese Journal of Intelligent Science and Technology, 2019, 1(4): 421–426.

[15] AI J, LIU Y Y, SU Z, et al. Link prediction in recommender systems based on multi-factor network modeling and community detection[J]. Europhysics Letters, 2019, 126(3): 38003.

[16] XIONG F, WANG X M, CHENG J J. Subtle role of latency for information diffusion in online social networks[J]. Chinese Physics B, 2016, 25(10): 108904.

[17] JAMALI M, ESTER M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//Proceedings of the 4th ACM Conference on Recommender Systems. New York: ACM Press, 2010: 1055–1066.

[18] MA H, YANG H X, LYU M R, et al. SoRec: social recommendation using probabilistic matrix factorization[C]//Proceedings of the 17th ACM Conference on Information and Knowledge Management. New York:

ACM Press, 2008: 931-940.
[19] XU M H, LIU S H. Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks[J]. IEEE Access, 2019, 7: 17493-17502.
[20] YANG B, ZHAO P F, PING S Q, et al. Improving the recommendation of collaborative filtering by fusing trust network[C]//Proceedings of the 8th International Conference on Computational Intelligence and Security. Piscataway: IEEE Press, 2012: 195-199.
[21] FANG H, BAO Y, ZHANG J. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation[C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2014.
[22] LIU Y, YANG C, MA J, et al. A social recommendation system for academic collaboration in undergraduate research[J]. Expert Systems, 2018, 36(1): e12365.

作者简介

陈佩武(1976-),男,平安科技(深圳)有限公司高级总监,深圳市金融智能机器人工程研究中心助理主任,主要研究方向为人工智能和大数据。

束方兴(1990-),男,北京大学互联网研究院(深圳)硕士生,主要研究方向为区块链和大数据。

收稿日期: 2020-12-05
通信作者: 束方兴, 1701213624@sz.pku.edu.cn