doi:10.3772/j.issn.1002-0470.2021.01.011

基于综合信任的奇异值分解推荐算法研究①

张大鹏②* 张 伟③* 张丽君* 刘雅军**

(*燕山大学信息科学与工程学院 秦皇岛 066004)

(**河北建筑工程学院信息工程学院 张家口 075000)

摘 要 针对传统矩阵分解推荐算法中数据稀疏、冷启动和用户信任矩阵数据稀疏等问题,本文提出了一种改进的基于路径的信任计算模型,利用用户-用户之间的直接信任关系和受信任者对用户信任关系的影响,计算用户-用户之间存在的间接信任关系,从而填充用户-用户信任矩阵。在此基础上,将用户之间的信任关系与奇异值分解(SVD)模型相结合,提出了一种融合综合信任的奇异值分解算法,即 CT-SVD 算法。该推荐算法结合用户评分矩阵和信任关系矩阵,对传统的奇异值分解推荐算法进行优化,提高了推荐系统评分预测的准确性。在 FilmTrust 和 Ciao 数据集上的实验结果表明,该算法能够有效地缓解推荐系统的数据稀疏性和冷启动问题。

关键词 推荐系统;信任网络;综合信任;奇异值分解(SVD)

0 引 言

推荐系统随着互联网的发展和海量数据中找到感兴趣的信息条件下而产生,并快速发展。经过多年的累积和沉淀,传统推荐算法已经成功应用于诸多领域,如电子商务[1]、音乐[2]等。其中,协同过滤算法已成为目前应用最为广泛的算法。

在众多协同过滤推荐算法中,矩阵分解推荐算法凭借其能够将稀疏且高维的用户-项目评分矩阵分解为两个低维矩阵的优势获得了关注,它能减少计算复杂度并提高推荐精度。

在普通的数据集中往往只有用户、物品、评分者3类数据,这些数据普遍非常稀疏,例如本文实验所使用的 FlimTrust 数据集和 Ciao 数据集,评分密度只有 1.14% 和 0.03%,通常数据集越大,评分密度越低。所以无法从这些数据中挖掘出过多信息,并且对于新用户或者新物品来说,关于它们的评分数据集中或被评分数量过少也无法产生精确的推荐。

越来越多的学者开始研究如何利用社交网络中的信息来对推荐系统进行改进,如在推荐算法中加入信任关系或时间信息。信任信息凭借其主观性、动态性和可传播性等特点逐渐成为推荐算法中的热点。实际上,用户的信任网络存在传递关系,用户对某人的信任程度会受到社交网络的影响。而现有的利用信任信息的矩阵分解推荐算法大都只是利用用户与用户之间存在的明显信任关系,而没有进一步挖掘信任网络中存在的隐含信任关系。

本文针对现有矩阵分解算法中存在的问题和挑战,融合信任网络中的隐含信任和奇异值分解(singular value decomposition,SVD)模型,提出一种推荐算法 CT-SVD。首先,在信任网络中利用用户-用户之间的直接信任关系,通过引入路径长度与可信度的概念对现有的基于路径的信任计算模型进行改进,挖掘出没有直接信任关系的用户之间的潜在信任关系,减少信任关系稀疏性,利用用户之间的信任关系进行推荐。之后利用奇异值分解模型将用户-

① 国家自然科学基金面上(61973261)资助项目。

② 男,1979 年生,博士,副教授;研究方向:人工智能;E-mail: daniao@ysu.edu.cn

③ 通信作者,E-mail: 821819013@qq.com

(收稿日期:2019-12-16)

物品数据进行降维处理,计算低维空间中用户的相似度,对用户未评分的物品进行评分预测,结合信任关系,将评分预测高的物品推荐给用户而提高推荐质量。

1 相关工作

协同过滤就是从海量用户中发掘出小部分与目标用户品位比较类似的用户集,然后根据用户集中用户喜欢的其他东西组成排序的目录,推荐给目标用户。现有的协同过滤算法大都是基于矩阵分解模型。文献[3]提出了 PMF(probabilistic matrix factorization)模型,该模型是在矩阵分解的模型中引入概率模型进一步优化推荐结果。文献[4]提出了 SVD 模型,该模型使用 SVD 对用户-项目的评分矩阵降维,得到降维后的用户特征和项目特征,再进行协同过滤推荐,较好地缓解了数据稀疏问题。文献[5]提出了 SVD++ 模型,是在 SVD 模型基础上引入隐式反馈,将用户的历史评分数据、用户的历史浏览记录、项目的历史评分数据和项目的历史浏览记录等作为新的参数,得到一个包含多个推荐参数的模型,从而缓解推荐系统中的冷启动问题。然而,上述算法模型仅考虑了用户与项目之间的关系,忽略了用户与用户之间存在的联系,与现实社交关系的关联不大,这是不实用的。

研究发现,由于稳定和持久的社会关系,相较于陌生人和供应商,人们更愿意相信来自他们朋友的建议[6]。学者们在将信任关系引入推荐模型后,发现信任关系能更好地缓解协同过滤算法中的数据稀疏和冷启动问题,更好地提高推荐效果[7-9]。文献[10]提出了一种引入信任关系的随机游走模型。文献[11]提出了 SoRec 算法,其采用概率矩阵分解的方法,将信任关系矩阵与用户项目评分矩阵相结合。该方法将使用两个向量的内积来确定观测数据,使用 logistic 函数约束内积,将两个向量的关系映射到一个非线性空间,自动地将用户的喜好和他们信任朋友的喜好融合在一起。但是该算法只使用了用户之间的信任信息,忽略了用户之间的信任传播。文献[12]提出了矩阵因子分解技术,将信任传播机制引入到 SocialMF 模型。为了融合信任关系,SocialMF 是目标用户的直接邻居的特征向量,并且特征向量的位置相对独立,不涉及非邻居。如果不将其他用户的缺陷插入目标用户,则该用户就是目标用户的直接邻居,由于该用户的信任关系在社交互联网上非常少见,因此直接邻居方法可以有效地减少数据稀缺引起的计算复杂性。文献[13]提出的 TrustSVD 算法是在 SVD++ 算法的基础上加入用户与用户之间的显性信任关系的推荐运算。具体来说,信任的隐含影响(信任谁)可以通过扩展用户建模自动地添加到 SVD++ 模型中,信任的显式影响(信任值)被用来约束特定于用户的向量应该符合他们的社会信任关系。算法在预测未知项目的评级时,考虑了评分的显性和隐性影响,还融合了用户之间的信任信息。采用加权正则化技术,避免模型学习过程中产生过拟合问题,进一步对用户和项目特定的潜在特征向量进行正则化。

近几年出现了一些在信任关系上加入隐性信任关系的推荐算法[14-17],而这些算法都只是简单地使用用户-用户之间的直接信任关系来聚合或直接计算信任者的主观意愿来进行间接信任值计算,而没有考虑到被信任者在信任网络中所起到的作用。在此基础上,本文在间接信任计算过程中,充分考虑了被信任者在信任预测过程中的影响,提出了加入可信度的路径信任计算模型。

2 信任模型

2.1 相关定义

为了更加直观地了解社交网络,假设在推荐系统中,存在局部的连续性信任网络,图 1 中共有 6 个用户($u1 \rightarrow u6$),图中存在的有向实线表示用户之间存在信任关系,实线上的数字所代表的是直接信任值。

本文挖掘信任网络中隐藏的信任关系是指对某一特定的用户 u,以信任网络作为基础,推导出与其没有直接关联的用户的信任度,换句话说,就是想通过信任网络中存在的直接信任关系,找到用户之间潜在的信任关系,从而填充信任网络。例如图 1 中

用户 $u1$ 对 $u4$、$u5$、$u6$ 也应该存在着一定的信任值，本文通过用户之间的信任网络，采用适当的方式将这些信任值推导出来，而推导出来的信任值被称为间接信任值。

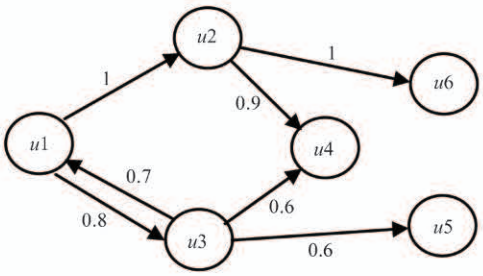


图 1 局部信任网络图

不同的领域对信任的定义不同，例如在社会网络中，信任被定义为基于被信任者将来行为的正面期望的信念[18]。针对本文的信任模型的建立与计算，现给出如下定义。

定义 1 信任。用户在相互交互历史的基础上，建立起来的相互依赖的关系，是目标用户对其他用户是否能为其提供有价值的评分能力的期望[19]。

定义 2 信任值。表示节点与节点之间信任关系的重要程度。信任值的范围为[0,1]，信任值越大表示用户之间的信任程度越高，0 表示不信任，1 表示完全信任。

2.2 直接信任

直接信任值的大小通常由数据集给出，不需要再进行计算。但是这样的数据的信任值往往只有 0 值和 1 值，无法很好地表达现实社会中的信任关系，并且信任数据集中的数据往往都比较稀疏，如本文所使用的 FilmTrust 和 Ciao 数据集，所以需要对数据集中的直接信任关系进行挖掘，让信任关系更加合理、密集。

2.3 间接信任及计算

间接信任关系一般是通过数据集中存在的用户-用户之间的直接信任关系推导出来的，例如 $u1$ 对 $u6$ 的信任度可由 $u1 \rightarrow u2 \rightarrow u6$ 推导出来。本文采用了改进的基于路径的信任预测算法对潜在的信任关系进行预测 t_{uk}，即改进的 MoleTrust 算法。最基本的 MoleTrust 算法[20]信任计算公式如下：

$$MT_{uv} = \frac{\sum_{k \in P(u)} t_{uk} t_{kv}}{\sum_{k \in P(u)} t_{uk}} \tag{1}$$

其中，$P(u)$ 表示用户 u 与用户 v 之间存在信任关系的用户集合，t_{uk} 表示用户 u 与用户 k 之间存在的信任值，t_{kv} 表示用户 k 与用户 v 之间存在的信任值。

由式(1)可知，MoleTrust 算法就是利用信任传递来衡量信任关系，但是没有考虑信任值随着路径的增加而变小，例如在图 1 中 $MT_{u1u6} = 1$ 显然是不合理的。本文在 MoleTrust 算法的基础上引入了信任递减路径长度对算法进行修改。路径长度是指源节点距离目标节点所需要经过弧的数目。例如在图 1 信任网络中，用户 $u1$ 与 $u2$、$u3$ 的路径长度为 1，而与 $u4$、$u5$、$u6$ 的路径长度为 2。

加入路径长度之后的间接信任计算如式(2)所示：

$$MT_{nuv} = \frac{MT_{uv}}{d} \tag{2}$$

其中，d 表示路径长度，由公式可知，用户 u 与目标用户 v 的路径长度越长，获得的信任值越小。为了避免多步信任产生的噪声与过长时间的搜索，本文限定 $d \leqslant 2$。

此时改进的 MoleTrust 算法充分考虑了信任传递性以及信任递减性，但是在计算过程中，该算法只考虑信任者，而没有考虑被信任者的影响，由此，本文引入了可信度对算法进行进一步的改进与完善。

可信度是指节点在信任网络中的可信程度。通俗地讲，一个用户在信任网络中可以被别人相信的程度就是可信度，这是该用户在信任网络中自身的一个属性，区别于信任值。可信度的取值范围为[0,1]，可信度越大，说明该用户越值得信赖。本文采用信任网络中入度的概念来计算可信度。可信度的具体计算公式如下：

$$Rel_{uv} = \frac{ind(v)}{\sum_{k \in t(u)} ind(k)} \tag{3}$$

其中，$ind(v)$ 表示用户 v 在信任网络中的入度，此时的入度值为信任某一用户的用户数量，$t(u)$ 表示用户 u 信任的用户集合。通过上式分析可知，信任用户 v 的用户越多，用户 v 可信度越高；用户 u 所表示的信任用户越少，可信度越高。

综上所述,最终得到的间接信任度的计算公式为

$$T_{uv} = \alpha \times MT_{nuv} + (1 - \alpha) \times Rel_{uv} \tag{4}$$

其中,α 表示信任倾向权重,$\alpha \in [0, 1]$。$\alpha = 0$ 时,间接信任度的计算仅考虑用户可信度的影响;$\alpha = 1$ 时,间接信任度的计算仅考虑加入路径长度的 MoleTrust 算法的影响。而 α 的值则采用信任网络中入度与出度进行计算,计算方式为

$$\alpha = \frac{outd(u)}{outd(u) + ind(v)} \tag{5}$$

其中,$outd(u)$ 表示用户 u 在信任网络中的出度,此时的出度值为某一用户信任的用户的数量。通过上式分析可知,当用户 u 在信任网络中的出度越高,即用户 u 的信任关系越多时,该用户在计算间接信任时所占的权重就越大。同理,被信任者 v 的入度越高时,即信任网络中,信任用户 v 的人越多,那么用户 v 的权重也就越大。

3 融合综合信任的奇异值分解

3.1 SVD 推荐模型

传统的 SVD 模型首先将用户的历史行为数据转化为评分矩阵,然后通过分解评分矩阵得到用户特征和项目特征,进行协同过滤推荐时,再根据分析结果预测评分,这样较好地缓解了数据稀疏问题。评分预测公式如式(6)所示。

$$r^*_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}} p_u \tag{6}$$

其中,u 为所有评分的平均分,p_u 和 q_i 分别对应用户和物品的特征向量,b_u 和 b_i 分别代表用户 u 和物品 i 相对 μ 偏移,具体计算方式如下:

$$b_u = \frac{1}{|I_u|} \sum_{i \in I_u} (r_{u,i} - \mu) \tag{7}$$

$$b_i = \frac{1}{|U_i|} \sum_{u \in U_i} (r_{u,i} - b_u - \mu) \tag{8}$$

其中,I_u 表示用户 u 评价过的物品的集合,U_i 表示对物品进行过评价的用户的集合。

SVD ++[5] 模型是在 SVD 模型基础上引入隐式反馈,将用户的历史评分数据、用户的历史浏览记录、项目的历史评分数据、项目的历史浏览记录等作为新的参数,从而得到一个包含多个推荐参数的模型,该模型的评分预测公式为

$$r^*_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}} (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i) \tag{9}$$

其中,参数 y_i 表示用户 u 对项目 i 的评分。

TrustSVD 是在 SVD ++算法的基础上加入用户与用户之间的显性信任关系进行推荐运算,该算法的评分预测公式为

$$r^*_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}} (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} w_v) \tag{10}$$

其中,参数 T_u 表示用户 u 信任的用户的集合,w_v 表示用户 u 对用户 v 的信任度。

3.2 CT-SVD 模型

假设,本文所使用的评分矩阵 $\boldsymbol{R} = [r_{u,i}]_{m \times n}$,$m$ 表示用户数,n 表示物品数,直接信任网络为 $G = (n, e)$,其中 n 为信任网络中所有节点的集合,e 为信任网络中所有边的集合,表示用户之间的信任关系。由信任网络 G 得到信任矩阵 $\boldsymbol{M} = [t_{uv}]_{m \times m}$,其中 $t_{uv} \in [0, 1]$ 表示用户 t 与用户 v 之间的信任值。用户之间的直接信任值为 1,间接信任值由式(4)计算得出。

在 TrustSVD 模型中,计算预测评分时只考虑了用户之间的直接信任,得到的预测评分计算公式如式(11)所示。

$$r^*_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}} (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^D|^{-\frac{1}{2}} \sum_{v \in T_u^D} w_v) \tag{11}$$

其中,T_u^D 为用户 u 的直接信任用户集合,w_v 为直接信任中用户 u 是否信任用户(受托人)的用户特定潜在特征向量。由该公式可知该模型计算的是评分信息的显隐性反馈和直接信任信息对预测评分的影响。

本文提出的 CT-SVD 算法还需要考虑间接信任以及被信任者对推荐效果的影响,若只考虑间接信任关系的影响,可以得到的信任预测公式如下:

$$r^*_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}} (p_u + |I_u|^{-\frac{1}{2}} \sum_{i \in I_u} y_i + |T_u^I|^{-\frac{1}{2}} \sum_{v \in T_u^I} f_v) \tag{12}$$

其中，T_u^I 为用户 u 的间接信任用户集合，f_v 为间接信任中用户 u 是否信任用户（受托人）的用户特定潜在特征向量。由该公式可知该模型计算的是评分信息的显隐性反馈和间接信任信息对预测评分的影响。

上面的两个模型分别考虑了信任网络中的直接信任和间接信任对评分预测的影响，本文提出了将直接信任与间接信任通过线性组合的方式相结合，同时在间接信任计算时考虑受信者影响的算法，并命名为 CT-SVD。CT-SVD 算法的用户对项目的预测评分通过式（13）计算。

$$r_{u,i}^{*} = \mu + b_u + b_i + q_i^{\mathrm{T}}(p_u + |I_u|^{-\frac{1}{2}}\sum_{i\in I_u} y_i + \beta|T_u^D|^{-\frac{1}{2}}\sum_{v\in T_u^D} w_v + (1-\beta)|T_u^I|^{-\frac{1}{2}}\sum_{v\in T_u^I} f_v) \tag{13}$$

其中，β 为权重值，当 $\beta = 0$ 时，评分预测过程中只考虑间接信任的影响，当 $\beta = 1$ 时，评分预测过程中只考虑直接信任的影响。结合实际情况，一般认为在评分预测过程中，直接信任比间接信任的影响要大，所以取 $\beta \geqslant 0.5$。

式（13）将用户之间的直接信任特征以及通过信任预测得到的间接信任特征融合到预测评分 $r_{u,j}^{*}$ 中，进一步优化得到新的用户模型 $p_u + |I_u|^{-\frac{1}{2}}\sum_{i\in I_u} y_i + \beta TuD - 12v \in TuDwv + 1 - \beta Tu1 - 12v \in TuIfv$，在评分过程中，本文不仅考虑了用户-项目评分矩阵，还考虑了用户-用户直接信任矩阵以及潜在的间接信任矩阵，损失函数计算如下：

$$\begin{aligned} L = {} & \frac{1}{2}\sum_u\sum_{j\in I_u}(r_{u,j}^{*} - r_{u,j})^2 + \frac{\lambda_t^D}{2} + (1-\beta)\sum_{v\in T_u^I}(t_{u,v}^{I*} - t_{u,v}^{I})^2 + \frac{\lambda}{2}\Big(\sum_u b_u^2 + \sum_j b_j^2 \\ & + \sum_u \|p_u\|_F^2 + \frac{\lambda}{2}\sum_j \|q_j\|_F^2 + \frac{\lambda}{2}\sum_i \|y_i\|_F^2 + \sum_v \|w_v\|_F^2 + \sum_v \|f_v\|_F^2\Big) \end{aligned} \tag{14}$$

为了避免模型在学习过程中产生过拟合，需要对函数加入正则化项，具体的添加方式为加大对存在冷启动行为的用户或物品的惩罚力度，因为其出现过拟合的可能性较小。加入正则项后，经过整理得到的目标损失函数如式（15）所示。

$$\begin{aligned} L = {} & \frac{1}{2}\sum_u\sum_{j\in I_u}(r_{u,j}^{*} - r_{u,j})^2 + \frac{\lambda_t^D}{2}\sum_u\sum_{v\in T_u^D}(t_{u,v}^{D*} - t_{u,v}^{D})^2 + \frac{\lambda_t^I}{2}\sum_u\sum_{v\in T_u^I}(t_{u,v}^{I*} - t_{u,v}^{I})^2 + \frac{\lambda}{2}\sum_u |I_u|^{-\frac{1}{2}} b_u^2 \\ & + \frac{\lambda}{2}\sum_j |U_j|^{-\frac{1}{2}} b_j^2 + \sum_u \Big(\frac{\lambda}{2}|I_u|^{-\frac{1}{2}} + \frac{\lambda_t^D}{2}|T_u^D|^{-\frac{1}{2}} + \frac{\lambda_t^I}{2}|T_u^I|^{-\frac{1}{2}}\Big)\|p_u\|_F^2 + \frac{\lambda}{2}\sum_j |U_j|^{-\frac{1}{2}}\|q_j\|_F^2 \\ & + \frac{\lambda}{2}\sum_j |U_i|^{-\frac{1}{2}}\|y_i\|_F^2 + \frac{\lambda}{2}|T_v^{D+}|^{-\frac{1}{2}}\sum_v \|w_v\|_F^2 + \frac{\lambda}{2}|T_v^{I+}|^{-\frac{1}{2}}\sum_v \|f_v\|_F^2 \end{aligned} \tag{15}$$

其中，$\|\cdot\|_F$ 表示 frobenius 范数，λ 为模型参数，U_i、U_j 为对项目 i、j 进行评分的用户集合，T_v^{D+}、T_v^{I+} 分别表示信任用户 v 的直接信任用户集合与间接信任用户集合。

为获得推荐模型，本文通过随机梯度下降法来求解上述损失函数的优化解。各参数的梯度计算公式如式（16）所示。

$$\begin{aligned} \nabla b_u &= \sum_{j\in I_u} e_{u,j} + \lambda|I_u|^{-\frac{1}{2}} b_u \\ \nabla b_j &= \sum_{u\in U_j} e_{u,j} + \lambda|U_j|^{-\frac{1}{2}} b_j \\ \nabla p_u &= \sum_{j\in I_u} e_{u,j} q_j + \lambda_t^D\sum_{v1\in T_u^D} e_{u,v1} w_{v1} + \lambda_t^I\sum_{v2\in T_u^I} e_{u,v2} f_{v2} + (\lambda|I_u|^{-\frac{1}{2}} + \lambda_t^D|T_u^D|^{-\frac{1}{2}} + \lambda_t^I|T_u^I|^{-\frac{1}{2}})p_u \\ \nabla q_j &= \sum_{u\in U_j} e_{u,j}(p_u + |I_u|^{-\frac{1}{2}}\sum_{i\in I_u} y_i + \beta|T_u^D|^{-\frac{1}{2}}\sum_{v\in T_u^D} w_v + (1-\beta)|T_u^I|^{-\frac{1}{2}}\sum_{v\in T_u^I} f_v) + \lambda|U_j|^{-\frac{1}{2}} q_j \\ \nabla y_i &= \sum_{j\in I_u} e_{u,j}|I_u|^{-\frac{1}{2}} q_j + \lambda|U_i|^{-\frac{1}{2}} y_i \\ \nabla w_v &= \sum_{j\in I_u} e_{u,j}\beta|T_u^D|^{-\frac{1}{2}} q_j + \lambda_t^D e_{u,v}^D p_u + \lambda|T_u^{D+}|^{-\frac{1}{2}} w_v \\ \nabla f_v &= \sum_{j\in I_u} e_{u,j}(1-\beta)|T_u^I|^{-\frac{1}{2}} q_j + \lambda_t^I e_{u,v}^I p_u + \lambda|T_u^{I+}|^{-\frac{1}{2}} f_v \end{aligned} \tag{16}$$

其中，$e_{u,j} = r_{u,j}^{*} - r_{u,j}$、$e_{u,v}^D = t_{u,v}^{D*} - t_{u,v}^D$ 和 $e_{u,v}^I = t_{u,v}^{I*} - t_{u,v}^I$ 分别表示用户 u 对物品 j 的评分预测误差、用户 u 对用户 v 的直接信任预测误差以及间接信任预测误差。通过随机梯度的不断迭代，参数也会在每

次迭代后更新,最后得到最佳答案。

本文的主要工作有 3 个方面。

(1) 利用用户-用户之间的直接信任关系,通过引入路径长度与可信度的概念对现有的一个基于路径的信任计算模型进行改进,挖掘出没有直接信任关系的用户之间的潜在信任关系,减少信任关系稀疏性,并且利用用户之间的信任关系进行推荐。

(2) 结合用户之间直接存在的信任关系与挖掘出的信任关系对评分预测进行规格化约束,缓解协同过滤算法的冷启动与数据稀疏性问题。

(3) 利用奇异值分解模型将数据进行降维处理,计算出多维空间中用户的相似度,对用户未评分的物品进行评分预测,结合信任关系,将评分预测高的物品推荐给用户,提高系统推荐质量。

3.3 时间复杂度分析

本文算法的时间开销来自两个部分,即改进的 MoleTrust 信任计算和 CT-SVD 推荐算法。改进的 MoleTrust 用于计算潜在的信任关系中的信任值,即对用户信任关系网络进行宽度遍历的过程。但是由于信任数据稀疏性和路径长度的问题,时间复杂度要远小于 $O(n^2)$。CT-SVD 推荐算法的主要计算开销来自于目标损失函数 L 和变量梯度的计算,计算目标损失函数 L 的时间复杂度是 $O(d|\boldsymbol{R}|+d|\boldsymbol{T}^D\boldsymbol{R}|+DTIR)$,其中 d 表示 d 维特征向量;$\boldsymbol{R}$ 表示用户-项目矩阵;$\boldsymbol{T}^D$ 表示用户直接信任矩阵;$\boldsymbol{T}^I$ 表示用户间接信任矩阵。变量梯度的时间复杂度为 $O(td|\boldsymbol{R}|+td|\boldsymbol{T}^DR|+td|\boldsymbol{T}^I\boldsymbol{R}|)$,其中 t 表示训练迭代次数。以上分析可知,算法的时间复杂度与用户-评分数和信任关系数量呈线性关系。

4 实验及结果分析

本文的实验过程为对比实验,实验系统为 Window8.1 专业版,内存为 8 GB、64 位,实验代码均由 Python 编写。将本文算法与传统的信任推荐算法和矩阵分解算法进行性能对比。研究的主要问题如下。

(1) 对于数据集中所有的用户,本文算法与传统的信任推荐算法和矩阵分解算法的推荐精度对比。

(2) 针对冷启动用户,本文算法与传统的信任推荐算法和矩阵分解算法的推荐性能对比。

(3) 当实验的训练集大小不同时,尤其是训练集数据极端稀疏的情况下,本文算法与传统的信任推荐算法和矩阵分解算法的推荐效果对比。

4.1 数据集及评价标准

根据模型以及实验的要求,实验所选用的数据集中的数据不仅要包含用户-项目评分信息,还包括用户-用户信任数据,所以本文选用了 FilmTrust 数据集以及 Ciao 数据集作为实验数据集,两个数据集的具体信息如表 1 所示。

表 1 实验数据集信息

	用户数	项目数	评分数	稀疏度	信任者	被信任者	信任数	稀疏度
FilmTrust	1508	2071	35 497	1.14	609	732	1853	0.42
Ciao	7375	99 746	280 391	0.03	6792	7297	111 781	0.23

本文采用平均绝对值误差(mean absolute error, MAE)及均方根误差(root mean squared error, RMSE)进行推荐性能对比。MAE 能很好地反映预测值误差的实际情况;RMSE 则可以用来衡量预测值与真实值之间的偏差,更好地反映均方根误差情况。平均绝对值误差(MAE)及均方根误差(RMSE)的计算公式如式(17)和(18)所示。

$$MAE=\frac{\sum_{u,j}|r^*_{u,j}-r_{u,j}|}{N} \tag{17}$$

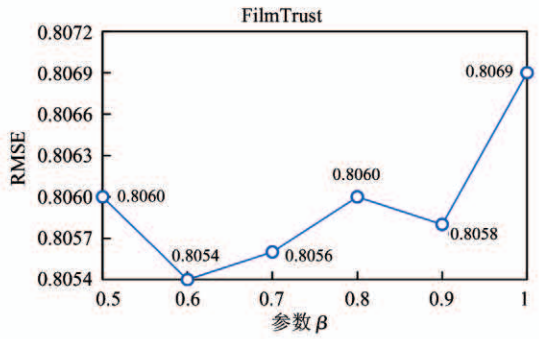
$$RMSE=\sqrt{\frac{\sum_{u,j}(r^*_{u,j}-r_{u,j})^2}{N}} \tag{18}$$

其中,$r_{u,j}$ 表示用户 u 对项目 j 的实际评分,N 表示测试集中的评分数量。评判的标准则是 MAE 与 RMSE 的值越小,实验误差越小,预测结果越精确,

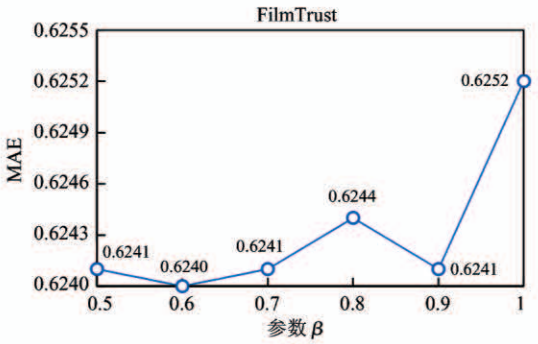
推荐性能越好。

4.2　参数对实验结果的影响

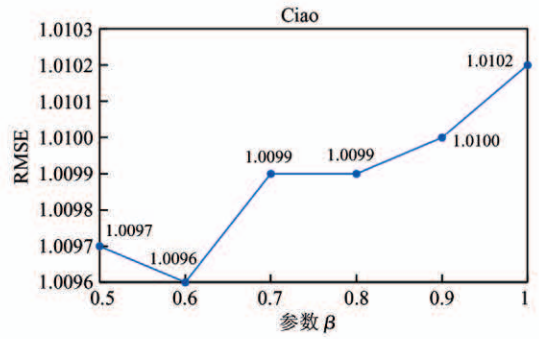
本文中，参数 β 表示在评分预测过程中直接信任关系与间接信任关系的影响程度。实验时，将潜在特征向量固定为 5，$\lambda_t^D = \lambda_t^I = 0.001$，迭代次数 1000 次，在评分预测过程中，一般默认直接信任比间接信任的影响要大，所以 $\beta \in [0.5, 1]$，并且以步长 0.1 逐渐增大。实验结果如图 2 所示。



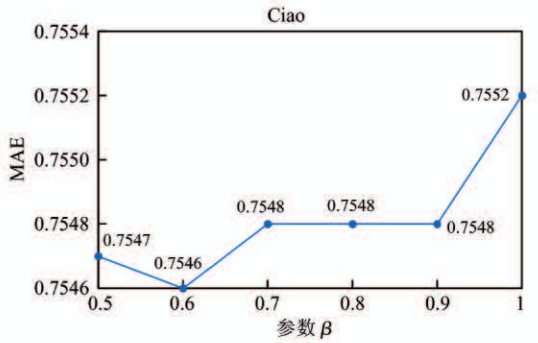
（a）在 FilmTrust 上的 RMSE 结果



（b）在 FilmTrust 上的 MAE 结果



（c）在 Ciao 上的 RMSE 结果



（d）在 Ciao 上的 MAE 结果

图 2　不同值在两个数据集上的实验结果

由图 2（a）、（b）分析可知，对于数据集 FilmTrust 的 RMSE 与 MAE 值，总体变化趋势遵循先递减后递增。当参数 $\beta = 0.6$ 时，数据集 FilmTrust 的 RMSE 与 MAE 值均为最小，当参数 $\beta > 0.6$ 时 RMSE 与 MAE 的值开始变大，虽然在 $\beta = 0.9$ 时，产生了波动，但总体趋势还是在变大。同理，由图 2（c）、（d）分析可知，对于数据集 Ciao 的 RMSE 与 MAE 值来说，整体趋势也是先降低后升高。当参数 $\beta = 0.6$ 时，数据集 Ciao 的 RMSE 与 MAE 值均为最小，当 $0.7 < \beta < 0.8$ 时，RMSE 与 MAE 的值保持不变，随后增大。所以，本文在两个数据集的实验中的影响参数 β 均取 0.6。

4.3　实验结果及分析

为了评估对比本文提出的 CT-SVD 算法的性能，现采用以下算法进行对比实验。

（1）SoRec 是 Ma 等人[11]提出的一种将信任关系矩阵与用户-项目矩阵进行整合的推荐模型。

（2）SocialMF 是 Jamali 等人[12]提出的一种使用矩阵因子分解技术，将信任传播机制引入的推荐模型，该算法只是简单地考虑了直接信任关系。

（3）SVD++ 是 Koren[5]提出的在 SVD 模型上通过引入隐式参数的形式，将用户的评分数据和项目的评分数据作为新的参数，从而得到一个更为精细的推荐模型。

（4）TrustSVD 是 Guo 等人[13]提出的在 SVD++ 算法的基础上加入用户与用户之间的显性信任关系进行推荐运算。

实验 1　针对全体用户的不同推荐算法的性能对比实验中，将原始数据集随机分割为 5 份，每份为原始数据集的 20%，采用 5-cross 交叉验证法，得出的原始数据在潜在因子数量分别为 5、10 时的算法性能，实验结果如表 3 所示。

表 2　对比算法实验参数设置

模型	FilmTrust	Ciao
SoRec	$\lambda = 0.001$, $\lambda_c = 0.1$	$\lambda = 0.001$, $\lambda_c = 0.01$
SocialMF	$\lambda = 0.001$, $\lambda_t = 1$	$\lambda = 0.001$, $\lambda_t = 1$
SVD++	$\lambda = 0.1$	$\lambda = 0.1$
TrustSVD	$\lambda = 1.2$, $\lambda_t = 0.9$	$\lambda = 0.5$, $\lambda_t = 1$
CT-SVD	$\lambda = 0.8$, $\lambda_t^D = \lambda_t^I = 0.001$, $\beta = 0.6$	$\lambda = 0.8$, $\lambda_t^D = \lambda_t^I = 0.0001$, $\beta = 0.6$

表 3　各算法对全体用户的推荐性能对比

数据集	潜在因子	度量方法	SoRec	SocialMF	SVD++	TrustSVD	CT-SVD	提升
FilmTrust	$d = 5$	MAE	0.6522	0.6470	0.6330	0.6351	0.6240	1.42%
		RMSE	0.8504	0.8423	0.8319	0.8221	0.8054	2.03%
	$d = 10$	MAE	0.6560	0.6534	0.6301	0.6354	0.6238	1.00%
		RMSE	0.8520	0.8470	0.8316	0.8224	0.8052	2.09%
Ciao	$d = 5$	MAE	0.9318	0.8203	0.8004	0.7798	0.7547	3.22%
		RMSE	1.2399	1.0824	1.1058	1.0607	1.0096	4.82%
	$d = 10$	MAE	0.9097	0.8194	0.7974	0.7789	0.7512	3.56%
		RMSE	1.2081	1.0977	1.1028	1.0585	1.0084	4.73%

由表 3 可知,在不同的潜在因子数下,SVD++算法的表现比加入信任关系的 SoRec 算法和 SocialMF 算法的表现要好,而 SocialMF 算法的表现也要优于 SoRec 算法,这说明单纯地将稀疏性极大评分矩阵和信任矩阵用于推荐系统效果较差,也说明了精细的模型有助于提高推荐精度。此外,TrustSVD 算法的优越性总体上高于 SVD++算法,说明信任矩阵的引入可以在 SVD++模型上产生好的推荐效果。

最后,相对于 SoRec 算法、SocialMF 算法和SVD++算法、TrustSVD 算法,CT-SVD 算法在 MAE 及 RMSE 上都有一定的提高。对于数据集 FilmTrust 来说,CT-SVD 算法的提升度为 1% ~2%,而 Ciao 数据集中 CT-SVD 算法的提升度为 3% ~5%,明显高于数据集 FilmTrust 的提升度,可能是由于数据集 FilmTrust 的数据稀疏度大于 Ciao 数据集。总体分析可知,信任关系中隐含的间接信任关系能够提高推荐精度。

实验 2　针对冷启动用户的不同推荐算法的性能对比实验中,冷启动用户的定义为在数据集中对物品评分数小于等于 5 的用户。实验结果如表 4 所示。

表 4　各算法对冷启动用户的推荐性能对比

数据集	潜在因子	度量方法	SoRec	SocialMF	SVD++	TrustSVD	CT-SVD	提升
FilmTrust	$d=5$	MAE	0.7464	0.7208	0.6700	0.6737	0.6228	7.04%
		RMSE	0.9473	0.9421	0.9120	0.8782	0.8014	8.75%
	$d=10$	MAE	0.7794	0.7384	0.6720	0.6739	0.6227	7.35%
		RMSE	0.9746	0.9572	0.9144	0.8806	0.8033	8.78%
Ciao	$d=5$	MAE	0.9353	0.8719	0.7789	0.7379	0.7424	-0.61%
		RMSE	1.2333	1.1678	1.0737	1.0488	1.0281	1.97%
	$d=10$	MAE	0.9208	0.8650	0.7819	0.7382	0.7425	-0.58%
		RMSE	1.2110	1.1542	1.0747	1.0490	1.0281	1.99%

由表 4 可知,在对数据集中的冷启动用户进行评分预测时,相对于全体用户,SoRec 算法、SocialMF 算法、SVD ++ 算法和 TrustSVD 算法的 RMSE 与 MAE 的值在 FilmTrust 数据集上普遍变大,在 Ciao 数据集上 SVD++ 的 RMSE 与 MAE 的值反而变小,这可能是实验数据集的数据密度不同,导致 SVD ++ 所构造算法的精确度不同造成的。此外,在冷启动用户表中,对于两个数据集来说,各算法的 RMSE 与 MAE 值总体是从左到右逐渐减小,这也说明了,对于冷启动用户,只考虑现有的评分矩阵和信任矩阵而不去挖掘数据之间的联系是不行的,随着模型的不断精细,推荐效果也会越来越好。

最后,观察本文的 CT-SVD 模型数据可以得出,CT-SVD 模型的 MAE 及 RMSE 相对于其他算法都有一定的提高。总体来看,CT-SVD 模型依然是其他对比算法中实验效果最好的,可以有效地缓解用户冷启动问题。

实验 3 针对不同规模训练集的算法性能对比。为了进一步全面地对比本文算法与其他算法的性能,拟采用不同规模训练集进行模型训练。本实验将原始数据集按照不同的比例随机分为训练集和测试集,然后进行测试。具体地,分别采用规模为 50%、60%、70%、80% 和 90% 的训练集,在潜在因子数量为 5 的条件下,实验参数均为各算法的最优参数,实验结果如表 5、表 6 所示。

将表 5、表 6 中各个算法的 MAE 值转化为图的形式,可以更加直观地分析数据中所隐藏的实验结果。具体结果如图 3 所示。

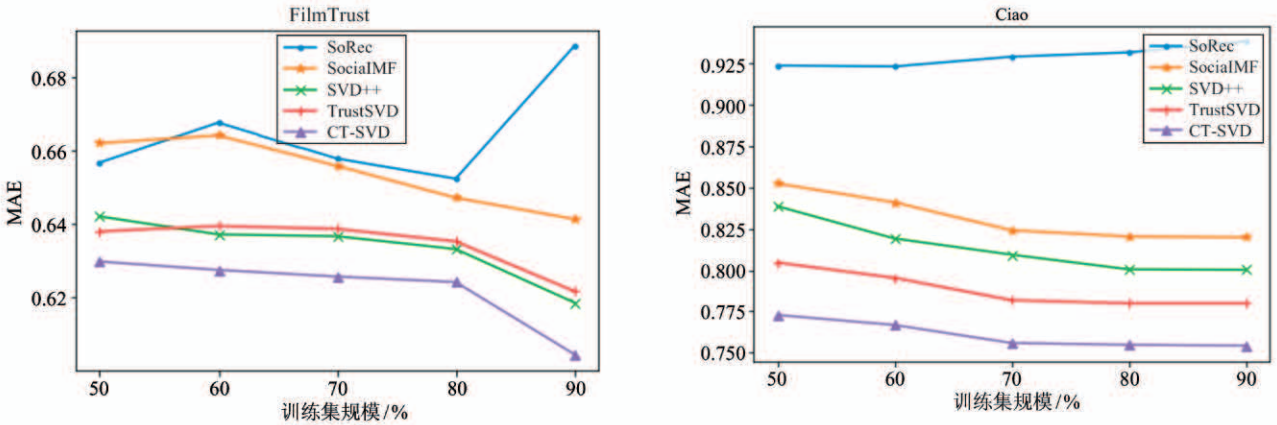
由表中数据可知,随着训练集规模的增加,FilmTrust数据集的所有推荐算法在两个数据集上

表 5 各算法在 FilmTrust 数据集上的不同规模训练集的推荐性能对比

训练集规模	度量方法	SoRec	SocialMF	SVD++	TrustSVD	CT-SVD	提升效果
50%	MAE	0.6566	0.6619	0.6419	0.6378	0.6296	1.29%
	RMSE	0.8569	0.8586	0.8423	0.8246	0.8150	1.16%
60%	MAE	0.6675	0.6640	0.6370	0.6393	0.6273	1.88%
	RMSE	0.8596	0.8698	0.8360	0.8262	0.8107	1.88%
70%	MAE	0.6577	0.6557	0.6365	0.6385	0.6255	1.73%
	RMSE	0.8570	0.8527	0.8368	0.8283	0.8100	2.21%
80%	MAE	0.6522	0.6470	0.6330	0.6351	0.6240	1.42%
	RMSE	0.8504	0.8426	0.8319	0.8221	0.8054	2.03%
90%	MAE	0.6884	0.6412	0.6184	0.6215	0.6043	2.76%
	RMSE	0.8456	0.8315	0.8068	0.8023	0.7813	2.62%

表 6 各算法在 Ciao 数据集上的不同规模训练集的推荐性能对比

训练集规模	度量方法	SoRec	SocialMF	SVD++	TrustSVD	CT-SVD	提升效果
50%	MAE	0.9239	0.8524	0.8387	0.8045	0.7728	3.94%
	RMSE	1.2312	1.1323	1.1530	1.0881	1.0284	5.49%
60%	MAE	0.9233	0.8410	0.8191	0.7953	0.7667	3.60%
	RMSE	1.2300	1.1268	1.1389	1.0875	1.0315	5.15%
70%	MAE	0.9292	0.8241	0.8093	0.7817	0.7558	3.31%
	RMSE	1.2370	1.1067	1.1174	1.0597	1.0111	4.59%
80%	MAE	0.9318	0.8203	0.8004	0.7798	0.7547	3.22%
	RMSE	1.2399	1.0824	1.1058	1.0607	1.0096	4.82%
90%	MAE	0.9384	0.8198	0.8001	0.7798	0.7542	3.28%
	RMSE	1.2427	1.1047	1.1012	1.0567	1.0051	4.88%



(a) 在 FilmTrust 上的实验对比　　(b) 在 Ciao 上的实验对比

图 3　各算法在两个数据集上的不同规模训练集的 MAE 值对比

的推荐性能都有所提高,而在 Ciao 数据集上,除了 SoRec 算法外的推荐算法的推荐性能也是提高的。并且在两个数据集上,TrustSVD 算法和 CT-SVD 算法的推荐性能都要优于 SoRec 算法、SocialMF 算法和 SVD++算法。SoRec 算法的推荐效果不好,可能是由于数据集过度稀疏,并且随着训练集规模的变大,用于预测实验结果的测试集评分矩阵更加稀疏,导致预测难度更大,结果也难精确。另外,相比于 FilmTrust 数据集,Ciao 数据集在推荐精度提升效果上有明显提高,这是因为相对于 FilmTrust 数据集,Ciao 数据集的数据量更加多,数据更加稀疏,所以从中能够获得更多的隐藏信息用于推荐预测。

相对于 SoRec 算法、SocialMF 算法、SVD++算法和 TrustSVD 算法,CT-SVD 模型在两个不同数据集上对不同规模的训练集都表现出了较好的提升效果,这也说明了本文提出的信任预测模型的合理性。信任网络中隐藏的信任关系,确实有助于填充信任网络,解决信任网络数据稀疏性问题。精确的信任计算模型能够有效提升推荐精度,改善推荐系统的性能。

5　结　论

本文以社交网络中的信任关系为理论依据,以现有的基于路径的信任计算模型为基础,受"朋友的朋友就是朋友"的思想启发,设计了一种综合考虑信任者与受信者对用户之间信任关系影响的信任预测模型,根据模型计算得出用户之间的间接信任,填充信任矩阵。同时结合矩阵分解可降维、易运算的优点,提出了一种融合综合信任关系的奇异值分解算法,利用用户-用户信任矩阵对传统的奇异值分解模型加以扩展,缓解了协同推荐中的数据稀疏性问题和新用户的冷启动问题。基于 FilmTrust 和 Ciao 数据集的实验结果表明,该算法的推荐性能优于已有的社交推荐算法和奇异值推荐算法,特别是在数据极端稀疏和冷启动的情况下,推荐性能更加优越。

另外,本文所研究的模型只考虑了信任网络中的信任关系,没有考虑不信任关系,并且本文的实验都是在单机环境中完成的,具有一定的局限性。下一步研究不仅要考虑加入不信任关系对该模型推荐性能的影响,而且还要考虑将模型移植到大数据平台,进行进一步的提升。

参考文献

[1] Huang Z, Zeng D, Chen H. A comparison of collaborative-filtering recommendation algorithms for e-commerce [J]. *IEEE Intelligent Systems*, 2007 (5): 68-78

[2] Park W I, Kang S, Choi M, et al. A music recommendation system in mobile environment [C] // International Conference on Consumer Electronics, Las Vegas, USA, 2009: 1-2

[3] Salakhutdinov R. Probabilistic matrix factorization [C] // International Conference on Neural Information Processing Systems, Kitakyushu, Japan, 2007: 1-8

[4] Funk S. Netflix update: try this at home [EB/OL]. http://sifter.org/~simon/journal/20061211.html: Net-

flix, 2011
[5] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, USA, 2008: 426-434
[6] Joachims T. Optimizing search engines using clickthrough data[C]//Proceedings of the 8th ACM SIGKDD International Conferenceon Knowledge Discovery and Data Mining, Alberta, Canada, 2002: 1-8
[7] Yang B, Lei Y, Liu J, et al. Social collaborative filtering-by trust[C]//International Joint Conference on Artificial Intelligence, Beijing, China, 2013: 2747-2753
[8] Fang H, Bao Y, Zhang J. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation[C]//The 28th AAAI Conference on Artificial Intelligence, Québec, Canada, 2014: 30-36
[9] Mi C, Peng P, Xiao L, et al. Recommendation algorithm based on user trust and interest with probability matrix factorization[C]//International Conference on Advanced Cloud and Big Data, Shanghai, China, 2017: 355-361
[10] Jamali M, Ester M. TrustWalker: a random walk model for combining trust-based and item-based recommendation [C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009: 397-406
[11] Ma H, King I, Lyu M R. Learning to recommend with social trust ensemble [C] // International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, USA, 2009: 203-210
[12] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks [C]//The 2010 ACM Conference on Recommender Systems, Barcelona, Spain, 2010: 135-142
[13] Guo G, Jie Z, Yorke-Smith N. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings[C]//The 29th AAAI Conference on Artificial Intelligence, Austin, USA, 2015: 123-129
[14] 李卫疆, 齐静, 余正涛,等. 融合信任传播和奇异值分解的社会化推荐算法[J]. 计算机工程, 2017(8): 236-242
[15] Tian Y, Qin Y B, X D Y, et al. TrustSVD algorithm based on double trust mechanism[J]. *Journal of Frontiers of Computer Science and Technology*, 2015, 9(11): 1391-1397
[16] Xi X, Zhang F Q, Li X Q. Probabilistic matrix factorization recommendation with user group and implicit trust [J]. *Computer Engineering and Applications*, 2019, 55(2): 137-141
[17] 王瑞琴, 潘俊, 冯建军. 基于信任计算和矩阵分解的推荐算法[J]. 模式识别与人工智能, 2018, 31(9): 786-796
[18] Massa P, Avesani P. Trust-aware recommender systems [C]//ACM Conference on Recommender Systems, Minneapolis, USA, 2007: 17-24
[19] Sherchan W, Nepal S, Paris C. A survey of trust in social networks[J]. *ACM Computing Surveys*, 2013, 45(4): 1-33
[20] Massa P, Avesani P. Controversial users demand local trust metrics: an experimental study on Epinions.com community[C]//National Conference on Artificial Intelligence, Pittsburgh, USA, 2005: 121-126

Research on SVD recommendation algorithm based on comprehensive trust

Zhang Dapeng*, Zhang Wei*, Zhang Lijun*, Liu Yajun**
(*School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)
(**Information Engineering College, Hebei University of Architecture, Zhangjiakou 075000)

Abstract

Aiming at the problems of data sparseness, cold start and sparse data of user trust matrix in traditional matrix decomposition recommendation algorithm, an improved path based trust computing model is proposed, which uses the direct trust relationship between users-users and the influence of the trusted on the trust relationship between users-users to calculate the indirect trust relationship between users-users and fill in the users-users trust matrix. On this basis, combining the trust relationship between users and the singular value decomposition (SVD) model, a singular value decomposition algorithm combining synthetic trust is proposed. The recommendation algorithm combines the user scoring matrix and the trust relationship matrix to the traditional singular value. The decomposition recommendation algorithm is optimized to improve the accuracy of the recommendation system score prediction. Experimental results on the FilmTrust and Ciao datasets show that the proposed algorithm can effectively alleviate the data sparsity and cold start problems of the recommended system.

Key words: recommender system, trust network, comprehensive trust, singular value decomposition (SVD)