

基于信任扩展和列表级排序学习的服务推荐方法

方晨^{1,2}, 张恒巍^{1,2}, 张铭^{1,2}, 王晋东^{1,2}

(1. 信息工程大学三院, 河南 郑州 450001; 2. 数字工程与先进计算国家重点实验室, 河南 郑州 450001)

摘 要: 针对传统基于信任网络的服务推荐算法中信任关系稀疏以及通过 QoS 预测值排序得到的服务推荐列表不一定最符合用户偏好等问题, 提出基于信任扩展和列表级排序学习的服务推荐方法 (TELSR)。在分析服务排序位置信息的重要性后给出概率型用户相似度计算方法, 进一步提高相似度计算的准确性; 利用信任扩展模型解决用户信任关系稀疏性问题, 并结合用户相似度给出可信邻居集合构建方法; 基于可信邻居集合, 利用列表级排序学习方法训练出最优排序模型。仿真实验表明, 与已有算法相比, TELS R 在具有较高推荐精度的同时, 还可有效抵抗恶意用户的攻击。

关键词: 服务推荐; 排序学习; 概率型用户相似度; 信任关系

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018007

Trust expansion and listwise learning-to-rank based service recommendation method

FANG Chen^{1,2}, ZHANG Hengwei^{1,2}, ZHANG Ming^{1,2}, WANG Jindong^{1,2}

1. The Third College, Information Engineering University, Zhengzhou 450001, China

2. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Abstract: In view of the problem of trust relationship in traditional trust-based service recommendation algorithm, and the inaccuracy of service recommendation list obtained by sorting the predicted QoS, a trust expansion and listwise learning-to-rank based service recommendation method (TELSR) was proposed. The probabilistic user similarity computation method was proposed after analyzing the importance of service sorting information, in order to further improve the accuracy of similarity computation. The trust expansion model was presented to solve the sparseness of trust relationship, and then the trusted neighbor set construction algorithm was proposed by combining with the user similarity. Based on the trusted neighbor set, the listwise learning-to-rank algorithm was proposed to train an optimal ranking model. Simulation experiments show that TELS R not only has high recommendation accuracy, but also can resist attacks from malicious users.

Key words: service recommendation, learning-to-rank, probabilistic user similarity, trust relationship

1 引言

随着互联网的普及和云计算技术的迅猛发展, 网络上提供的 Web 服务呈指数级增长。用户迫切地需要一种有效的服务推荐方法来解决其面临的选

择困境。因此, 服务推荐技术在服务计算领域获得了广泛的关注。Web 服务的服务质量 (QoS, quality of service) 包括服务失效率、响应时间、成本、吞吐量等^[1], 是用户进行服务选取时需要考虑的重要属性之一。而由于 Web 服务广泛地分布在网络中,

收稿日期: 2017-04-05; 修回日期: 2017-12-26

通信作者: 张恒巍, 13083710760@163.com

基金项目: 国家自然科学基金资助项目 (No.61303074, No.61309013); 河南省科技攻关计划基金资助项目 (No.12210231003)

Foundation Items: The National Natural Science Foundation of China (No.61303074, No.61309013), Henan Science and Technology Research Project (No.12210231003)

一些 QoS 属性如响应时间、吞吐量等经常受到网络环境动态变化的影响,具有很大的不确定性,这就造成了服务推荐可靠性差的问题。

为解决此问题,研究者们考虑将协同过滤算法应用到服务推荐过程中,通过预测 QoS 值并以此对服务进行排序来实现推荐^[2]。为了提高 QoS 预测的准确性,研究者们对传统协同过滤算法做出了一系列改进,包括引入用户的信任网络^[3]、服务调用模式^[4]、服务的上下文信息^[5]等。主要存在的问题为 1) 没有有效利用服务的排序位置信息; 2) 引入的信任网络中用户直接信任关系稀疏,难以提供足够的辅助信息。

同时,随着研究的深入,部分研究者发现, QoS 值预测的准确性并不代表服务推荐的准确性。假设某用户对服务 *a* 和服务 *b* 的某一 QoS 属性(如可靠性)的评价值分别为 2 和 3,使用不同的协同过滤算法来预测 QoS 值可能会得到不同的推荐结果:一种预测结果是服务 *a* 的 QoS 值为 2.5、服务 *b* 的 QoS 值为 3.6; 另一种预测结果是服务 *a* 的 QoS 值为 2.5、服务 *b* 的 QoS 值为 2.4。2 种预测结果的平方误差均为 $0.5^2 + 0.6^2$,但是,得到的推荐列表中服务 *a* 和服务 *b* 的排序却是相反的。由此说明,仅仅根据 QoS 预测值排序得到的服务推荐列表并不能非常准确地反映用户的偏好。

近几年来,有研究者考虑将排序学习技术引入推荐算法中来,通过直接优化最终的排序列表来提高推荐系统的准确性^[6]。作为一种强监督性机器学习算法,排序学习能够整合大量复杂特征并自动学习最优参数,降低了考虑单个因素进行排序的风险,且能够通过多种方法来规避过拟合问题,获得了学术界越来越多的关注^[7]。然而,目前很少有研究将传统协同过滤算法与排序学习技术结合起来,并应用到服务推荐领域。

针对上述问题,本文提出基于信任扩展和列表级排序学习的服务推荐方法(TELSR, trust expansion and listwise learning-to-rank based service recommendation method)。该方法首先在分析服务排序位置信息重要性的基础上,给出概率型用户相似度计算方法(PUSC, probabilistic user similarity computation method),提高相似度计算准确性;然后,提出信任扩展模型充分挖掘用户信任网络信息,并结合用户相似度构建可信邻居集合;最后,利用可信邻居集合改进列表级排序学习算法,通过训练得

到最符合用户偏好的服务推荐列表。本文的主要贡献有以下 3 点。

1) 给出概率型用户相似度计算方法,有效利用了服务的排序位置信息,提高了相似度计算准确性。

2) 提出信任扩展模型,充分挖掘了用户信任关系,构建出可信邻居集合,能够抵抗恶意用户的攻击。

3) 改进列表级排序学习算法,可输出最符合用户兴趣偏好的服务推荐列表。

2 相关工作

协同过滤最早是由 Goldberg 等^[8]在 1992 年提出的,后来被广泛应用于电子商务领域,并且取得了极大的成功。其核心思想是:在用户群中寻找与目标用户评分行为相似的邻居用户,然后基于这些邻居用户对服务的评分向目标用户做出推荐^[9]。目前,已有很多学者将协同过滤算法应用到服务推荐过程中,并对其做出了一系列的改进。王海艳等^[10]引入服务的推荐个性属性特征来改进传统的相似度计算式,并结合用户之间的信任关系对服务的评分值进行预测,进而对用户做出推荐。Liu 等^[11]利用服务的流行度特征改进相似度计算,并根据用户和服务的地理位置来缩小相似用户的寻找范围,相比传统推荐算法更加高效;Hu 等^[12]在寻找相似邻居时融入了服务调用的时间信息,并通过线性加权的方式综合了基于相似用户和相似服务的 QoS 预测结果。文献[10~12]均是通过改进相似度计算来提高算法准确性,属于基于近邻的协同过滤算法。此外,还有部分研究者利用数学模型来预测服务的 QoS,并取得了较好的成果。Wei 等^[13]利用矩阵分解模型将高维的用户—服务矩阵分解为低维的用户矩阵和服务矩阵,并将位置属性融入矩阵分解的正则项中,有效提高了 QoS 预测精度;胡堰等^[14]借助隐含类别表示用户指标偏好、用户及服务情境三者之间的依赖关系,并建立隐语义概率模型用于预测用户在特定服务情境下的个性化指标偏好,然后计算出每个候选服务的效用值进行推荐;Wang 等^[15]考虑到了 QoS 值在不同时间段的动态变化特性,对 QoS 预测值的残差进行零均值拉普拉斯先验分布假设,将 QoS 预测问题转化为 Lasso 回归问题进行求解。文献[13~15]有效利用了数学模型的精确性,属于基于模型的协同过滤算法。可见,上述工作的研究重点均集中在提高 QoS 值预测的准确性方面,而近年来有研究者发现 QoS 值预测的准确性并不能

确保服务推荐的准确性，引言已给出相关示例。

排序学习作为一种强监督性机器学习算法，能够直接针对最终的推荐列表进行优化，这一特性可以避免根据 QoS 值排序来间接得到推荐列表带来的缺陷。根据优化目标的不同，排序学习主要分为 3 类：点级 (pointwise)、对级 (pairwise)、列表级 (listwise)^[7]。点级排序的处理对象是单独的一个项目，通过预测评分实现推荐，其相当于传统的预测 QoS 值的服务推荐方法；对级排序是根据评分来定义项目对之间的偏序关系，最终通过整合所有项目对的偏序关系得到整个排序列表，而其时间复杂度高，且在整合推荐列表时会损失一定的准确性；列表级排序的处理对象是所有的项目，直接对整个排序列表进行优化，在运行效率和推荐准确性方面具有更明显的优势，因此，成为被研究最多的方法。

Huang 等^[6]利用基于排列概率的相似度来寻找更准确的邻居用户，然后通过最小化目标用户和邻居用户在未评分项目集合上的交叉熵损失函数，来得到最优的排序列表。Weimer 等^[16]提出了一种最大化边界矩阵因式分解算法 CoFiRank，通过直接优化排序评价标准 NDCG 来进行推荐。Shi 等^[17]提出一种基于上下文感知的推荐方法，利用张量分解优化 MAP 评测准则，是首个能够挖掘用户隐式反馈和上下文信息，并将列表级排序学习和协同过滤算法相结合的方法。但是列表级排序学习算法依然面临着用户恶意评分、数据稀疏性等传统难题，且目前还缺乏将该算法改进并应用到服务推荐领域的研究。

3 基于信任扩展和列表级排序学习的服务推荐方法

本节给出基于信任扩展和列表级排序学习的服务推荐方法，该方法首先将用户表示为已调用服务集合的概率分布，基于 Kullback-Leibler (KL) 距离进行概率型用户相似度的计算，以此提高用户相似度计算的准确性；然后，利用信任扩展模型充分挖掘用户信任网络中的信任关系，并结合用户相似度构建为目标用户构建可信邻居集合，以此抵抗某些恶意用户的攻击；最后，利用可信邻居集合改进列表级排序学习算法，训练出最优的服务排序列表推荐给用户。其中，概率型用户相似度计算方法、可信邻居集合构建算法 (TNSC, trusted neighbor set construction algorithm) 以及列表级排序学习预测算法 (PABL, prediction algorithm based on listwise learning-to-rank) 为 TELSR 的

核心，下面重点对它们进行介绍。

3.1 概率型用户相似度计算方法

协同过滤算法的核心步骤是寻找相似用户，可采用的方法主要有 Pearson 相关系数、余弦相似性、修正的余弦相似性等^[10]。目前大多数服务推荐算法都是基于 Pearson 相关系数改进得来的，其基本定义如下。

定义 1 Pearson 相关系数。若 $r(u, i)$ 和 $r(v, i)$ 分别为用户 u 和 v 对服务 i 的 QoS 评价值， $\bar{r}(u)$ 和 $\bar{r}(v)$ 分别为用户 u 和 v 的平均 QoS 评价值， $I_{u,v}$ 为用户 u 和 v 共同调用的服务集合，则利用 Pearson 相关系数，计算用户 u 和 v 的相似度为

$$PCC(u, v) = \frac{\sum_{i \in I_{u,v}} (r(u, i) - \bar{r}(u))(r(v, i) - \bar{r}(v))}{\sqrt{\sum_{i \in I_{u,v}} (r(u, i) - \bar{r}(u))^2} \sqrt{\sum_{i \in I_{u,v}} (r(v, i) - \bar{r}(v))^2}} \quad (1)$$

用户调用服务示例如图 1 所示，假设用户 A 、 B 、 C 共同调用过的服务集合为 $I = \{a, b, c, d\}$ ，这 4 个服务的 QoS (如可用性，用百分制表示) 分别为 $A=(0, 20\%, 80\%, 100\%)$ ， $B=(10\%, 0, 80\%, 100\%)$ ， $C=(0, 22\%, 100\%, 89\%)$ ，采用 Pearson 相关系数计算得 $PCC(A, B) = PCC(A, C) = 0.97$ ，即用户 B 、用户 C 和用户 A 是同等相似的。从图 1 可以看出，如果根据服务可用性大小对服务进行排序，用户 B 对于服务 a 、 b 的排序与用户 A 是相反的，而用户 C 对于服务 c 、 d 的排序与用户 A 是相反的。此时，若利用用户 B 做推荐，则其向用户 A 推荐的最好服务是 d ，正好符合用户 A 的需求；若利用用户 C 做推荐，则其向用户 A 推荐的最好服务是 c ，违背用户 A 的需求。相比之下，用户 B 的推荐结果更加可信，所以理论上用户 B 与用户 A 的相似度应该更大。原因在于，在实际推荐系统中，用户主要关注排在推荐列表中前面质量较优的服务，对于排在后面质量较差 (如可用性低于 50%) 的服务给予的关注较少。因此，服务的排序位置是除了 QoS 数据之外另一个能够反映用户兴趣偏好的重要信息。

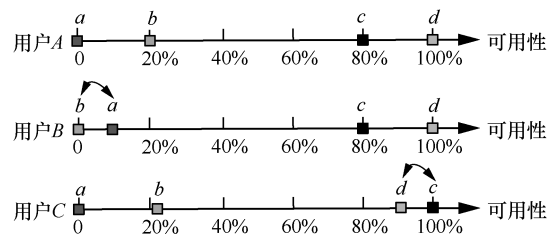


图 1 用户调用服务示例

基于此, 本文充分挖掘服务的排序位置信息, 借鉴 Mollica 等^[18]提出的 Plackett-Luce 模型, 将每个用户表示为已调用服务集合排列的概率分布, 然后进行用户相似度的计算, 从而找到更加准确的相似用户。为方便下文讨论, 定义如下。

定义 2 服务排列。若存在一个服务集合 $I = \{i_1, i_2, i_3, \dots, i_n\}$, 按照用户偏好度由大到小的顺序对服务进行排序, 得到服务排列 $\pi = \langle \pi_1, \pi_2, \dots, \pi_n \rangle$, 其中, $\pi_i \in I$ 且 $\pi_i \neq \pi_j (i \neq j)$ 。记所有可能的排列的集合为 Ω^I 。

对于有 n 个服务的集合, 共有 $n!$ 种不同的服务排列, 但每一种排列的概率是不同的, QoS 值更优的服务排在前面的概率应该更大。本文中只考虑效益型 QoS 属性 (即 QoS 值越大越好, 如吞吐量、可用性等), 对于成本型 QoS 属性 (即 QoS 值越小越好, 如响应时间、成本等), 可将其通过负值计算转化为效益型 QoS 属性。

定义 3 排列概率。给定一个服务排列 $\pi = \langle \pi_1, \pi_2, \dots, \pi_n \rangle$ 和对应服务的 QoS 值 $\{r(\pi_1), r(\pi_2), \dots, r(\pi_n)\}$, 则排列 π 的概率为

$$P(\pi) = \prod_{i=1}^n \frac{\varphi(r(\pi_i))}{\sum_{k=i}^n \varphi(r(\pi_k))} \quad (2)$$

其中, $\varphi(\bullet)$ 为任意单调递增且取值为严格正值的函数, 这里, 令 $\varphi(x) = e^x$ 。

在海量的 Web 服务环境中, n 的值往往很大, 如果要计算出 $n!$ 种不同排列的概率, 那将会消耗大量计算资源和时间。Cao 等^[19]提出了一种排列模型, 只考虑排列中位于前 k 个位置的服务。

定义 4 top- k 排列集合。对于一个服务集合 I , 属于 I 的一个 top- k 排列集合 $T_k(i_1, i_2, \dots, i_k)$ 包括所有前 k 个服务为 $i_1, i_2, \dots, i_k \in I$ 的排列, 即

$$T_k(i_1, i_2, \dots, i_k) = \{\pi = \langle \pi_1, \pi_2, \dots, \pi_k, \dots, \pi_n \rangle \mid \pi \in \Omega^I, \pi_j = i_j, j = 1, 2, \dots, k\} \quad (3)$$

根据定义 4, 对于服务集合 I 来说, 共有 $\frac{n!}{(n-k)!}$ 种不同的 top- k 排列集合, 将其记为集合 T_k^S 。Cao 等^[19]给出了一种计算 $T_k(i_1, i_2, \dots, i_k)$ 概率的简便方法, 介绍如下。

定义 5 top- k 概率。记 top- k 排列集合 $T_k(i_1, i_2, \dots, i_k)$ 的概率为 $P(T_k(i_1, i_2, \dots, i_k))$, 其计算式为

$$P(T_k(i_1, i_2, \dots, i_k)) = \prod_{j=1}^k \frac{\varphi(r(\pi_j))}{\sum_{l=j}^n \varphi(r(\pi_l))}, \forall j = 1, \dots, k, \pi_j = i_j \quad (4)$$

其中, $r(\pi_j)$ 是排在第 j 个位置的服务的 QoS 值。

由此可见, 计算 top- k 排列集合的概率比直接计算 I 中全排列的概率要更有效率, 且更能直观地反映出服务集合 I 中的概率分布。

设用户 u 和 v 共同调用过的服务集合记为 $I_{u,v}$, $T_k^{I_{u,v}}$ 表示 $I_{u,v}$ 上所有的 top- k 排列集合, 则用户 u 和 v 在 $T_k^{I_{u,v}}$ 上的概率分布可由式(4)计算而得, 分别记为 P_u 和 P_v 。用户 u 和 v 的相似度可以通过衡量 P_u 和 P_v 之间的距离而得, 2 个概率分布越相近, 表明用户 u 和 v 的评价行为越相似。本文采用 KL 距离来衡量 2 个排列分布的距离^[6]。

定义 6 KL 距离。概率分布 P_u 和 P_v 之间的 KL 距离可定义为

$$D_{KL}(P_u \| P_v) = \sum_{g \in T_k^{I_{u,v}}} P_u(g) \text{lb} \left(\frac{P_u(g)}{P_v(g)} \right) \quad (5)$$

其中, $P_u(g)$ 和 $P_v(g)$ 分别为用户 u 和 v 的 top- k 排列集合 $g \in T_k^{I_{u,v}}$ 的概率。由式(5)可知, $D_{KL}(P_u \| P_v) \neq D_{KL}(P_v \| P_u)$, 即 KL 距离是非对称的。因此, 本文基于 KL 距离提出一种对称的概率型用户相似度计算方法。

定义 7 概率型用户相似度。用户 u 和 v 之间的概率型相似度可定义为

$$\text{sim}(u, v) = \begin{cases} 1 - \frac{1}{2}(D_{KL}(P_u \| P_v) + D_{KL}(P_v \| P_u)), & |I_{u,v}| \geq c \\ \left[1 - \frac{1}{2}(D_{KL}(P_u \| P_v) + D_{KL}(P_v \| P_u)) \right] \frac{|I_{u,v}|}{c}, & |I_{u,v}| < c \end{cases} \quad (6)$$

其中, c 为 $|I_{u,v}|$ 的阈值, 它可以避免由于用户 u 和 v 共同调用服务过少而导致相似度计算偏大的问题。概率型用户相似度计算方法如算法 1 所示。

算法 1 概率型用户相似度计算

输入 用户 u 和 v 共同调用服务集合 $I_{u,v}$, 参数 k , 阈值 c

输出 u 和 v 的概率型用户相似度 $\text{sim}(u, v)$

begin

1) $T_k^{I_{u,v}} \leftarrow I_{u,v}$ 上所有的 top- k 排列集合

- 2) $D_{KL}(P_u \| P_v) \leftarrow 0; D_{KL}(P_v \| P_u) \leftarrow 0$
 - 3) for each $g \in T_k^{u,v}$
 - 4) 利用式(4)计算用户 u 出现排列 g 的概率 $P_u(g)$ 以及用户 v 出现排列 g 的概率 $P_v(g)$
 - 5) $D_{KL}(P_u \| P_v) \leftarrow D_{KL}(P_u \| P_v) + P_u(g) \text{lb} \left(\frac{P_u(g)}{P_v(g)} \right)$
- $$D_{KL}(P_v \| P_u) \leftarrow D_{KL}(P_v \| P_u) + P_v(g) \text{lb} \left(\frac{P_v(g)}{P_u(g)} \right)$$
- 6) end for
 - 7) 利用式(6)计算用户 u 和 v 的概率型用户相似度 $sim(u,v)$

end

利用算法 1，计算图 1 中用户 A 和 B 的相似度 $sim(A,B)=0.95$ ，用户 A 和 C 之间的相似度 $sim(A,C)=0.82$ ，因此用户 A 和 B 的相似度更大，符合“推荐系统中用户更在意排在前面质量较优的服务”这一结论。

3.2 可信邻居集合构建算法

利用 PUSC 可计算其他用户与目标用户的相似度，然后选取出相似度较大的用户作为邻居进行推荐。但是当推荐系统中存在恶意用户对服务 QoS 值进行虚假评价时，此时，若把这类用户当作邻居，会极大影响推荐的精度。基于此，本文利用用户间的信任关系建立可信邻居集合来进行推荐，从而避免恶意用户的攻击。为了解决传统基于信任的服务推荐算法中信任关系稀疏性问题，本文提出信任扩展模型，同时考虑直接信任关系和间接信任关系。

定义 8 直接信任关系。推荐用户与目标用户有共同的服务调用记录，通过交互彼此的服务调用信息而建立起的一种信任关系，记为 RT 。

假设用户 u 和 v 都调用过服务 i ，其 QoS 评价价值分别为 $r(u,i)$ 和 $r(v,i)$ ，则用户 v 可视为用户 u 的一个推荐用户。若 $r(u,i)$ 和 $r(v,i)$ 的误差绝对值不大于阈值 ε ，则认为用户 v 的此次推荐行为是有效的，否则是无效的。

Beta 信任模型^[20]基于概率论而进行信任度量，其概率密度函数用于计算二进制事件 (x, \bar{x}) 的后验概率，计算式为

$$f(x) = \frac{\Gamma(p+q+2)}{\Gamma(p+1)\Gamma(q+1)} x^p (1-x)^q, 0 < x < 1 \quad (7)$$

其中， p 表示事件 x 出现的次数， q 表示事件 \bar{x} 出现的

次数， Γ 为 Gamma 函数，其值为 $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du$ 。

由于其信任模型简单易懂，计算复杂度小，因此，比较适用于存在大量服务推荐信息的场合。本文采用该信任模型对用户之间的直接信任关系进行度量。

定义 9 直接信任度。用户 u 和 v 之间的直接信任度 $RT(u,v)$ 可以表示为 Beta 信任模型的数学期望，即

$$RT(u,v) = E(x) = \frac{p_{u,v} + 1}{p_{u,v} + q_{u,v} + 2} \quad (8)$$

其中， x 代表在用户 u 和 v 所有服务信息交互过程中，用户 v 为用户 u 做出的有效推荐行为； $p_{u,v}$ 代表有效推荐行为次数， $q_{u,v}$ 代表无效推荐行为次数。

由式(8)可知，用户 u 和 v 之间的直接信任度与有效推荐行为次数成正比，因此，其可以甄别某些用户反常的恶意评价行为。然而，在实际推荐系统中，用户之间的相互交互记录往往较少，导致直接信任关系稀疏性问题。为此，本文利用信任关系的传递特性来扩大用户的信任范围，并给出如下定义。

定义 10 间接信任关系。推荐用户与目标用户没有共同的服务调用记录，其信任关系是通过中间若干个用户之间的直接信任关系传递而形成的，记为 IRT 。

间接信任传递关系如图 2 所示。由图 2 可知，用户 u 和 v 均与用户 m 和 n 之间存在直接信任关系，根据信任关系的传递性，可以通过用户 m 和 n 建立起用户 u 和 v 之间的间接信任关系。

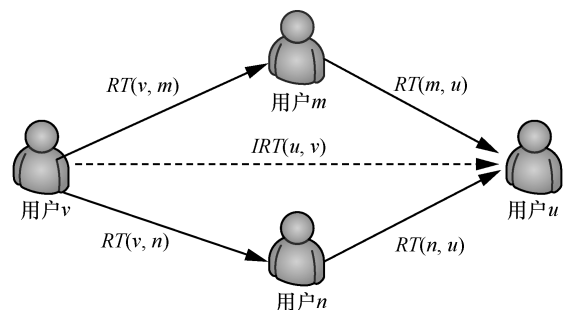


图 2 间接信任传递关系

定义 11 间接信任度。若用户 u 的直接信任集合为 D ，利用 D 中所有与用户 v 有直接信任关系的用户来进行信任传递，则用户 u 和 v 之间的间接信任度为

$$IRT(u, v) = \frac{\sum_{k \in D} w_k RT(v, k)}{\sum_{k \in D} w_k} \quad (9)$$

其中, w_k 为集合 D 中的用户 k 向 u 推荐 v 的权重, 其大小为用户 u 与 k 之间的直接信任度; $RT(v, k)$ 为用户 v 与 k 之间的直接信任度。

定义 12 综合信任度。通过综合直接信任度和间接信任度, 得到用户之间的综合信任度 CT 为

$$CT(u, v) = \alpha RT(u, v) + (1 - \alpha) IRT(u, v) \quad (10)$$

其中, 参数 α 用于控制直接信任度 $RT(u, v)$ 在综合信任度计算中所占的权重。 $CT(u, v)$ 越大, 代表推荐用户 v 与目标用户 u 之间的信任关系越强, 其推荐行为的可信性越高。

定义 13 可信相似度。综合考虑用户 u 和 v 之间的概率相似度和综合信任度, 得到用户 u 和 v 的可信相似度为

$$T(u, v) = \frac{2sim(u, v) \times CT(u, v)}{sim(u, v) + CT(u, v)} \quad (11)$$

基于可信相似度的定义, 本文提出可信邻居构建算法, 如算法 2 所示。

算法 2 可信邻居构建

输入 目标用户 u , 其他用户集合 U , 参数 l

输出 目标用户 u 的可信邻居集合 N_u

begin

1) for 每个用户 $u_k \in U$

2) 利用 PUSC 计算概率型用户相似度 $sim(u, u_k)$

3) 利用式(8)计算直接信任度 $RT(u, u_k)$

4) 利用式(9)计算间接信任度 $IRT(u, u_k)$

5) 利用式(10)计算综合信任度 $CT(u, u_k)$

6) 利用式(11)计算可信相似度 $T(u, u_k)$

7) end for

8) $N_u \leftarrow$ 按照可信相似度由大到小对用户进行排序, 选取前 l 个用户作为目标用户 u 的可信邻居集合

end

3.3 列表级排序学习预测算法

为了利用可信邻居集合来提高服务推荐的准确性, 本文首先利用矩阵分解模型来预测服务的 QoS 值, 然后利用列表级排序学习算法训练出最优的服务排序模型。为了方便描述, 定义参数如下。

1) 参数定义

$R = (r(u_j, i_k))_{m \times n}$: 用户服务评分矩阵, 其中, m 为用户的个数, n 为服务的个数。

$r(u_j, i_k)$: 用户 u_j 对于服务 i_k 的 QoS 评价价值。

$U_{d \times m}$: $d \times m$ 维的用户隐含特征矩阵。

$V_{d \times n}$: $d \times n$ 维的服务隐含特征矩阵。

d : 隐含特征数。

U_j : 矩阵 U 的第 j 列向量, 代表用户 u_j 的隐含特征向量。

V_k : 矩阵 V 的第 k 列向量, 代表服务 i_k 的隐含特征向量。

$T(u_j, u_m)$: 用户 u_j 和用户 u_m 之间的可信相似度。

N_{u_j} : 用户 u_j 的可信邻居集合。

2) 矩阵分解模型

矩阵分解模型是在协同过滤推荐算法中应用最为广泛的模型之一。其主要思想是将用户服务评分矩阵 R 近似分解为低维的用户隐含特征矩阵 U 和服务隐含特征矩阵 V , 计算式为

$$R \approx U^T V \quad (12)$$

则用户 u_j 对服务 i_k 的 QoS 预测值为

$$\hat{r}(u_j, i_k) = g(U_j^T V_k) \quad (13)$$

算法通过最小化预测评分矩阵和原评分矩阵的误差来实现 QoS 值的精确预测^[21]。在现实生活中, 人们对于一个服务的评价往往会受到所信任好友的影响。因此, 为了提高服务推荐的准确性, 本文在预测 QoS 值时加入可信邻居用户的影响, 将式(13)改进为

$$\hat{r}(u_j, i_k) = g[\beta U_j^T V_k + (1 - \beta) \sum_{m \in N_{u_j}} T(u_j, u_m) U_m^T V_k] \quad (14)$$

其中, 参数 β 用于控制目标用户的 QoS 预测值受其可信邻居影响的权重, $g(x)$ 是一种逻辑函数, 本文

$$\text{令 } g(x) = \frac{1}{1 + e^{-x}}。$$

3) 列表级排序学习模型

列表级排序学习直接针对最终的排序列表进行优化, 可以避免仅仅根据 QoS 值排序带来的不准确性。其核心思想是: 将预测排序列表和正确排序列表之间的交叉熵作为损失函数, 通过训练过程最小化其交叉熵, 从而使最终得到的预测排序模型最接近正确排序模型^[7]。本文基于 top-1 概率, 将交叉熵损失函数定义如下。

定义 14 top-1 概率。服务 i 在用户 u 的推荐列表中排在第一位置的概率, 定义为

$$P_u(r(u, i)) = \frac{\varphi(r(u, i))}{\sum_{j=1}^n \varphi(r(u, j))} \quad (15)$$

定义 15 交叉熵损失函数。根据训练数据集提供的已知 QoS 值 $r(u_j, i_k)$, 计算得到正确排序列表的概率分布, 根据式(14)中预测 QoS 值 $\hat{r}(u_j, i_k)$, 计算得到预测排序列表的概率分布, 则两者的交叉熵损失函数定义为

$$L = -\sum_{j=1}^m \sum_{k=1}^n I_{jk} P_{u_j}(r(u_j, i_k)) \text{lb}(\hat{P}_{u_j}(\hat{r}(u_j, i_k))) + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) \quad (16)$$

其中, $\|\cdot\|_F^2$ 表示 Frobenius 范数, λ 为正则化参数, 用于防止损失函数 L 在学习过程中的过拟合; I_{jk} 为指示参数, 当用户 u_j 调用过服务 i_k 时, $I_{jk} = 1$, 否则, $I_{jk} = 0$ 。

基于交叉熵损失函数的定义, 给出列表级排序学习预测算法, 具体如算法 3 所示。

算法 3 列表级排序学习预测

输入 训练数据集, 学习速率 η , 最大迭代次数 t_{\max} , 阈值 θ

输出 每一个用户的最佳服务推荐列表

begin

1) 根据训练数据集, 利用式(15)计算得到正确排序列表的概率分布

2) 利用式(14)算出初始所有服务的 QoS 预测值

3) 利用式(15)算出初始预测排序列表的概率分布

4) 利用式(16)计算初始交叉熵损失函数 L

5) **for** $t \leq t_{\max}$

6) 更新用户隐含特征矩阵 U 和服务隐含特征矩阵 V

$$U_j \leftarrow U_j - \eta \frac{\partial L}{\partial U_j}$$

$$V_k \leftarrow V_k - \eta \frac{\partial L}{\partial V_k}$$

7) 记录上次交叉熵损失函数 L

8) 利用式(14)算出新的所有服务的 QoS 预测值

9) 利用式(15)算出新的预测排序列表的概率分布

10) 利用式(16)计算新的交叉熵损失函数 L'

11) **if** $(|L' - L| < \theta)$ **break**;

12) **else** $t \leftarrow t + 1$

13) **end for**

14) **for** 每个用户 $u_j \in U$

15) 将所有服务按照 top-1 概率由大到小进行排序, 得到最佳服务排序列表, 推荐给用户

16) **end for**

end

由算法 3 可知, PABL 首先利用训练数据集计算得到正确排序列表的概率分布, 如步骤 1)所示; 再根据 QoS 预测值计算得到初始预测排序列表的概率分布, 如步骤 2)~步骤 3)所示; 然后采用随机梯度下降算法, 通过不断调整用户隐含特征矩阵 U 和服务隐含特征矩阵 V , 来最小化交叉熵损失函数 L , 如步骤 5)~步骤 10)所示; 算法的终止条件是交叉熵损失函数 L 的平均绝对误差小于阈值 θ 或达到设定的最大迭代次数 t_{\max} , 如步骤 11)~步骤 12)所示; 最终根据每个服务的 top-1 概率由大到小进行排序, 得到每一个用户的最佳服务推荐列表。

3.4 TELSR 描述

经过上述分析, TELSR 的具体过程如下。

1) 运用 PUSC 计算每一个用户与其他用户的概率型相似度。

2) 运用 TNSC 为每一个用户建立可信邻居集合。

3) 根据训练数据集得到用户服务评分矩阵, 初始化用户隐含特征矩阵 U 和服务隐含特征矩阵 V 。

4) 运用 PABL 得到每一个用户的最佳服务推荐列表。

TELSR 首先通过 Plackett-Luce 模型将用户表示为已调用服务集合的概率分布, 并利用 PUSC 计算用户的概率型相似度, 其优点在于利用了服务的排序位置信息, 使相似度计算更加准确; 为了消除推荐系统中恶意用户随意打分的影响, 利用 TNSC 为用户建立可信邻居集合, 其优点在于充分挖掘用户间的直接信任关系和间接信任关系, 缓解了用户信任网络稀疏性问题; 最终利用可信邻居集合改进 QoS 预测值的准确性, 并利用列表级排序学习的强大数据处理能力, 训练出最优的排序模型, 为用户提供最符合其偏好的服务推荐列表。

3.5 算法时间复杂度

设推荐系统中共有 m 个用户, n 个候选服务, d 个特征数, 每个用户平均调用过 n_1 个服务, 每 2 个用户共同调用过的服务数目平均为 n_2 , 每个用户的直接信任用户数目平均为 m_1 , 每个用户的可信邻居数目平均为 m_2 , 矩阵 R 中非零元素个数为 $|R|$, k 为相似度计算时取前 k 个服务。当给一个特定用户 u 提供服务推荐列表时, 在 PUSC 中, 用户共同调用服务的 top- k 排列集合共有 $\frac{n_2!}{(n_2-k)!}$ 个, 所以算

法需要循环 $\frac{n_2!}{(n_2-k)!}$ 次, 需要计算出用户 u 与所有

用户的相似度, 所以总时间复杂度为 $O\left(\frac{mn_2!}{(n_2-k)!}\right)$;

在 TNSC 中, 计算直接信任度的复杂度为 $O(n_2)$, 间接信任度的复杂度为 $O(m_1)$, 需要计算用户 u 与所有用户的信任度, 所以总时间复杂度为 $O(m(m_1+n_2))$; 在 PABL 中, 计算交叉熵损失函数 L 的时间复杂度为 $O(|R|d+|R|m_2d)$, 计算偏导函数

$\frac{\partial L}{\partial U_j}$ 、 $\frac{\partial L}{\partial V_k}$ 的时间复杂度分别为 $O(|R|d+|R|m_2d)$ 、

$O(|R|d+|R|n_1m_2d)$, 计算所有候选服务 top-1 概率的时间复杂度为 $O(nd+nm_2d)$ 。在算法实际运行过程中, 一般 k 的取值均较小 (说明见第 4.3 节), 所以可以认为算法 TELSR 的时间复杂度与用户数目、候选服务数目以及矩阵 R 中非零元素个数是线性相关的。因此, 算法 TELSR 可以应用到大型的 Web 服务数据集上。

4 实验结果与分析

4.1 实验设置

本实验使用由 Zheng 等^[22]收集并公开发布的 WS-DREAM 数据集, 它是由分布在全球 20 多个国家的 150 个电脑节点收集的 QoS 信息, 构成了约 150 万条 QoS 调用记录, 其内容主要包括 Web 服务的往返响应时间 RTT、数据块大小、响应结果等属

性, 表 1 展示了该数据集的部分服务实例信息。

本文选用往返响应时间 RTT 作为评价 QoS 的标准, 当用户调用某服务超过 100 次时, 计算出 RTT 的均值, 最终得到一个 150×100 的用户服务矩阵。为了研究算法的推荐准确性, 本文从原始的用户服务矩阵中随机地剔除部分 QoS 值, 形成了 5 个不同的稀疏矩阵, 其密度分别为 0.04、0.08、0.12、0.16、0.20。之所以选择小密度矩阵, 是因为在海量的 Web 服务环境中, 用户只调用过很少的服务, 因此, 其真实的用户服务矩阵就是很稀疏的。

本文使用 5 重交叉验证作为实验方法, 将稀疏矩阵随机分为 5 份, 每次选择其中的 4 份即矩阵的 80% 作为训练集, 选择余下的 1 份即矩阵的 20% 作为测试集。每次实验重复 5 次, 取平均值得到最终的评估结果。由于本文认为用户更在意最终获得的服务推荐列表中服务排序的准确性, 因此, 采用 NDCG (normalized discounted cumulative gain) 作为衡量算法推荐性能的标准。NDCG 值越大, 表示算法的推荐性能越好^[7]。

在服务推荐系统中, 用户对于 QoS 的评价可以作为相关度。假设共有 m 个用户, $r(u, j)$ 为用户 u 的服务排序列表中第 j 个服务的 QoS 值, $t \in R$ 为截断阈值。对于用户 u , 其 $NDCG_u$ 值定义为

$$NDCG_u @t = \sum_{j=1}^t \frac{2^{r(u,j)} - 1}{1b(j+2)} \quad (17)$$

其中, $NDCG @t$ 为所有用户的 $NDCG_u @t$ 的平均值, 定义为

$$NDCG @t = \frac{1}{m} \sum_{u=1}^m NDCG_u @t \quad (18)$$

4.2 参数对算法性能的影响

由式(10)可知, 参数 α 控制着直接信任度和间接信任度在综合信任度计算中所占的权重。实验中以 0.1 的步长增加 α 的值, 记录推荐性能 $NDCG @10$ 的变化情况。为了更加清楚地了解参数 α 对于推荐性能的影响, 根据不同的参数 β , 分 5 组

表 1 部分数据集信息

IP 地址	服务 ID	响应时间/ms	数据块大小/KB	HTTP code	HTTP message
35.9.27.26	8 451	2 736	582	200	OK
35.9.27.26	8 460	804	14 419	200	OK
35.9.27.26	8 953	20 176	2 624	-1	TimeoutException
192.33.90.66	8 712	719	631	200	OK

进行实验，结果如图 3 所示。从图 3 中可以看出，对于不同的参数 β 值，推荐性能随不同参数 α 值的变化情况基本相同， α 值过大或过小都会降低算法的推荐性能，当 $\alpha = 0.7$ 时，算法的推荐性能最好，说明综合用户之间的直接信任度和间接信任度能够有效提高算法的推荐性能。因此，对比实验均设置 $\alpha = 0.7$ 。

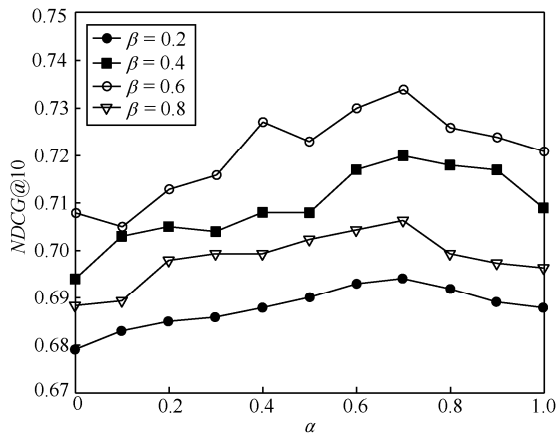


图 3 参数 α 对于推荐性能的影响

由式(14)可知，参数 β 用于控制可信邻居用户在 QoS 预测值中的影响权重。利用与参数 α 类似的实验方法来研究参数 β 对于推荐性能的影响，结果如图 4 所示。从图 4 可以看出，对于不同的参数 α ，推荐性能随不同参数 β 取值的变化情况基本相同。当 $\beta = 0.6$ 时，算法的推荐性能最好，说明可信邻居可以有效提高算法的推荐性能。因此，对比实验均设置 $\beta = 0.6$ 。

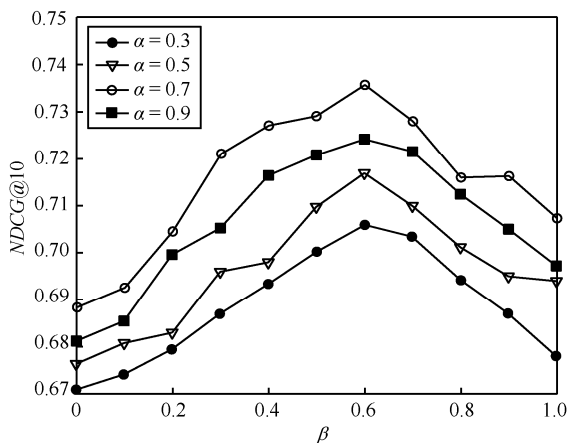


图 4 参数 β 对于推荐性能的影响

参数 d 为矩阵分解过程中用户和服务的特征数。将 d 分别取值为 10、15、20、25、30、35，记

录 TELSR 运行时间和推荐性能的变化，结果如图 5 所示。从图 5 中可以看出，随着隐含特征数 d 的增加，算法的推荐性能得到了一定的提升，但是算法的运行时间也随之快速增长。当 d 从 10 增加到 35 时，算法的推荐准确率提高了 6.6%，运行时间增加了近 200%，说明在实际情况中，仅需要较小的隐含特征维数，就能够取得较好的推荐效果，且算法的运行时间也较短。因此，对比实验均设置 $d = 10$ 。

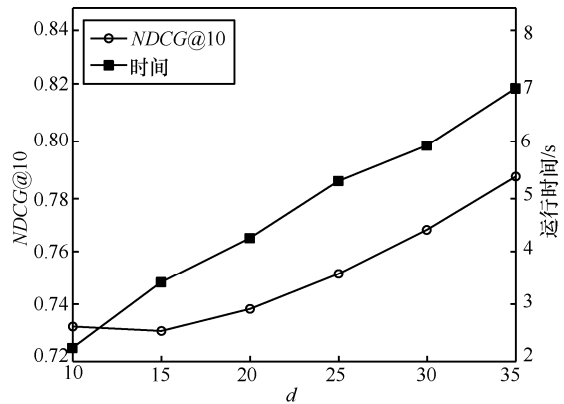


图 5 隐含特征数 d 对于推荐性能的影响

4.3 算法运行时间的比较

PUSC 基于 top- k 概率计算用户的概率型相似度。 k 值越大，用户的相似度计算越准确，算法的推荐性能可能越好。但是由第 3.5 节时间复杂度分析可知， k 值越大，会导致算法的时间复杂度变大。为了得到 k 的合理取值，本文将 k 依次取值为 1、2、3，矩阵密度为 0.08，记录算法的运行时间和推荐性能 $NDCG@10$ ，如表 2 所示。

k 取值	NDCG@10	运行时间/s
1	0.732 4	2.2
2	0.745 6	5.5
3	0.753 7	9.7

由表 2 可以看出，随着 k 的取值不断增大，算法的推荐性能提高非常小，但是算法的运行时间却大幅增加。当 $k=3$ 时，算法的推荐性能 $NDCG@10$ 比 $k=1$ 时提高了 2.9%，但是运行时间增加了 341%。而 $k=1$ 时，算法就可以在较短的运行时间内实现较好的推荐性能。因此，本文实验中均设置 $k=1$ 。

为了衡量算法的运行时间性能，将 TELSR 与以下 4 种经典的推荐算法做比较。

1) CF-DNC^[23]

该算法首先利用“兴趣相似用户集选取算法”动态选取目标用户的相似邻居，然后提出“用户信任计算模型”，筛选出目标用户的可信邻居用户集，最后提出了一种新的协同过滤算法，综合利用可信邻居的评分信息，对服务的评分值进行预测。

2) TACF^[12]

该算法有效融合了服务调用时间信息，提出“时间感知的相似度算法”，寻找更加准确的相似用户和相似服务，然后设计“个性化随机游走算法”来克服数据的稀疏性，最后利用混合协同过滤算法预测服务的 QoS 值。

3) listPMF^[24]

该算法改进了概率矩阵分解模型，根据用户评分得到用户的偏好序列，并通过最大化预测的偏好序列和已知的偏好序列的后验概率来实现项目的推荐，属于基于列表级排序的协同过滤算法。

4) listCF^[6]

该算法通过计算用户共同打分项目集合的 Jansen-Shannon 散度，来度量用户的相似度，并通过最小化目标用户和邻居用户的加权交叉熵损失函数来做预测，属于基于列表级排序的协同过滤算法。

本文的主要参数设置为 $\alpha = 0.7$ 、 $\beta = 0.6$ 、 $d = 10$ 、 $k = 1$ 、 $l = 20$ ，其余对比算法均为原文献中说明的最优参数。所有算法均使用密度为 0.04、0.08、0.12、0.16、0.20 的 5 个稀疏矩阵进行实验，并记录下算法的运行时间，结果如图 6 所示。

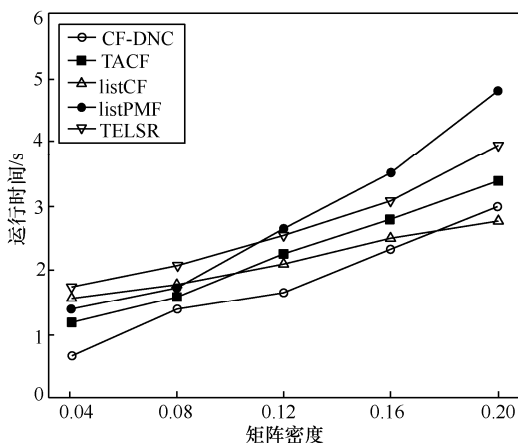


图 6 不同算法运行时间的比较

由图 6 可以看出，随着矩阵密度的增加，本文提出的 TELSR 的运行时间比 listPMF 短，但是比 CF-DNC、TACF 和 listCF 稍长，具体原因如下。

1) listPMF 基于用户和项目的隐含特征矩阵来预测用户的偏好序列。当矩阵密度增加时，需要预测的用户偏好序列数量增多，且随着隐含特征矩阵的不断更新，算法计算量成倍增长，导致 listPMF 的运行时间大幅度增加，甚至超过了 TELSR。

2) CF-DNC 和 TACF 均是在传统的 Pearson 相关系数的基础上改进的相似度计算方法，可以直接利用数据集中的 QoS 值。而 TELSR 采用 PUSC，首先需要根据已有的 QoS 数据计算出用户调用服务的概率分布，才能进行下一步的相似度计算。

3) listCF 选取出邻居用户之后，直接利用列表级排序算法进行 QoS 值预测。而 TELSR 还增加了用户信任度的计算，并结合用户相似度提出了可信邻居构建算法 TNSC。

由于 TELSR 属于混合型算法，内容同时涉及用户相似度、信任度和 QoS 预测，因此，其计算量更大，但根据第 3.5 节和图 6 可知，随着矩阵密度的增加，其运行时间仍保持了线性增长的趋势，说明 TELSR 算法是可以应用在大型 Web 服务数据集上的。

4.4 推荐准确性的比较

为了衡量算法的推荐性能，依然将 TELSR 与上面 4 种经典的推荐算法做对比。本文算法的主要参数设置为 $\alpha = 0.7$ 、 $\beta = 0.6$ 、 $d = 10$ 、 $k = 1$ 、 $l = 20$ ，其余对比算法均为原文献中说明的最优参数。所有算法均使用密度为 0.04、0.08、0.12、0.16、0.20 的 5 个稀疏矩阵进行实验，并记录下每种矩阵下的 $NDCG@10$ 、 $NDCG@20$ 。具体情况如表 3 所示，其中括号内的百分数表示 TELSR 比相应算法提高的百分比。由表 3 可以看出以下 2 点。

1) TELSR 的推荐性能在密度更大的矩阵上提升得更为明显。例如，与 CF-DNC 相比，在密度为 0.20 的矩阵上，TELSR 在 $NDCG@10$ 、 $NDCG@20$ 提升的百分比分别为 10.25%和 8.57%，而在密度为 0.04 的矩阵上，其提升百分比仅为 5.57%和 4.48%。原因在于，矩阵密度的增加使用户能够找到更多共同调用的服务，从而提高了相似度计算的准确率，增强了算法的推荐性能。

2) 在 5 种算法中，listCF 和 TELSR 表现最为优异，它们均属于 listwise CF，但是 TELSR 在 listCF 的相似度的基础上加入了信任度的计算，因此其推荐性能相对于 listCF 有所提升，且提升幅度随着矩阵密度的增加而增加。因为随着矩阵中已知 QoS 值

表 3 推荐准确性比较

矩阵密度	NDCG	CF-DNC	TACF	listPMF	listCF	TELSR
0.04	<i>NDCG@10</i>	0.663 4(5.57%)	0.672 7(4.12%)	0.674 7(3.81%)	0.692 9(1.08%)	0.700 4
	<i>NDCG@20</i>	0.684 5(4.78%)	0.697 5(2.83%)	0.705 1(1.72%)	0.721 8(-0.64%)	0.717 2
0.08	<i>NDCG@10</i>	0.685 9(6.78%)	0.698 3(4.89%)	0.705 0(3.89%)	0.727 3(0.70%)	0.732 4
	<i>NDCG@20</i>	0.711 1(5.43%)	0.725 5(3.34%)	0.731 0(2.56%)	0.746 1(0.48%)	0.749 7
0.12	<i>NDCG@10</i>	0.692 4(8.11%)	0.708 0(5.73%)	0.717 3(4.37%)	0.738 7(1.34%)	0.748 6
	<i>NDCG@20</i>	0.719 3(6.06%)	0.733 5(4.01%)	0.737 6(3.43%)	0.755 2(1.02%)	0.762 9
0.16	<i>NDCG@10</i>	0.698 8(9.23%)	0.717 1(6.44%)	0.725 2(5.25%)	0.751 1(1.62%)	0.763 3
	<i>NDCG@20</i>	0.723 4(7.66%)	0.739 8(5.27%)	0.745 9(4.41%)	0.768 3(1.37%)	0.778 8
0.20	<i>NDCG@10</i>	0.708 8(10.25%)	0.729 6(7.10%)	0.735 6(6.22%)	0.762 6(2.47%)	0.781 4
	<i>NDCG@20</i>	0.732 9(8.57%)	0.746 4(6.61%)	0.758 7(4.88%)	0.779 4(2.01%)	0.795 7

的服务的增多,TELSR 能够在用户之间发现更多的有效推荐行为,从而使其信任度计算更加准确,进一步提升了算法的推荐性能。

4.5 抵抗恶意用户能力的比较

CF-DNC、TACF、listCF 和 TELS R 均属于基于近邻的协同过滤推荐算法。在这类算法中,可能会存在着某些邻居用户由于利益驱动而演变为恶意用户,从而影响算法的推荐精度。因此,为了衡量其抵抗恶意用户的能力,本文通过逐渐增加恶意用户所占的比例,来观察这 4 种算法推荐性能的变化,如图 7 所示。TELSR 的主要参数设置为 $\alpha = 0.7$ 、 $\beta = 0.6$ 、 $d = 10$ 、 $k = 1$ 、 $l = 20$, 矩阵密度为 0.08, 其余对比算法均为原文献中说明的最优参数。

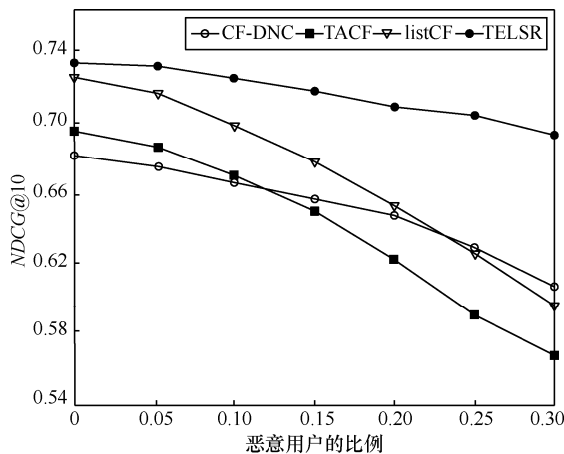


图 7 抵抗恶意用户能力比较

由图 7 可知,随着恶意用户比例的增加,TACF 和 listCF 的推荐性能下降很快,因为它们没有建立任何防御机制,一旦系统中的相似用户演变为恶意用户,算法的准确性会受到很大的影响;而 CF-DNC

考虑到了用户的信任关系,具备一定的抗攻击能力,但是其在计算相似度时没有考虑服务的排序位置信息,所以导致推荐精度不够高;TELSR 利用概率分布模型计算用户相似度,并结合用户的信任关系进行服务推荐,在具备较高推荐精度的同时,还能够较好地抵抗恶意用户的攻击。

5 结束语

本文针对传统服务推荐算法中仅依据 QoS 预测值排序带来的不准确性以及用户信任关系稀疏性问题,提出基于信任扩展和列表级排序学习的服务推荐方法。该方法首先在分析服务排序位置信息重要性的基础上,给出概率型用户相似度计算方法,提高了用户相似度计算的准确性;然后,利用信任扩展模型充分挖掘用户之间的信任关系,并给出可信邻居构建算法,以抵抗某些恶意用户的攻击;最后,利用可信邻居集合改进矩阵分解模型,并给出列表级排序学习预测算法,为用户训练出最优的服务排序模型。实验证明,TELSR 具有较高的推荐精度,并且可以应用到大型的 Web 服务数据集上。下一步工作将优化用户信任模型,并考虑 QoS 值的时间效应,进一步增强推荐模型在动态环境中的适用性。

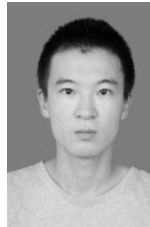
参考文献:

[1] 李玲,刘敏,成国庆.一种基于 FAHP 的多维 QoS 的局部最优服务选择模型[J].计算机学报,2015,38(10):1997-2010.
LI L, LIU M, CHENG G Q. A local optimal model of service selection of multi-QoS based on FAHP[J]. Chinese Journal of Computers, 2015, 38(10): 1997-2010.

[2] MA Y, WANG S G, YANG F C, et al. Predicting QoS values via mul-

- ti-dimensional QoS data for Web service recommendations[C]//IEEE Conference on Web Services. 2015: 249-256.
- [3] LIU Z, MA J, JIANG Z, et al. IRLT: integrating reputation and local trust for trustworthy service recommendation in service-oriented social networks[J]. Plos One, 2016, 11(3):e0151438.
- [4] 张莉, 张斌, 黄利萍, 等. 基于服务调用特征模式的个性化 Web 服务 QoS 预测方法[J]. 计算机研究与发展, 2013, 50(5):1066-1075. ZHANG L, ZHANG B, HUANG L P, et al. A personalized Web service quality prediction approach based on invoked feature model[J]. Journal of Computer Research and Development, 2013, 50(5): 1066-1075.
- [5] QI L, DOU W, ZHOU Y, et al. A context-aware service evaluation approach over big data for cloud applications[J]. IEEE Transactions on Cloud Computing, 2015, PP (99): 1.
- [6] HUANG S S, WANG S Q, LIU T Y, et al. Listwise collaborative filtering[C]//The 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). 2015: 343-352.
- [7] 黄震华, 张佳雯, 田春岐. 基于排序学习的推荐算法研究综述[J]. 软件学报, 2016, 27(3): 691-713. HUANG Z H, ZHANG J W, TIAN C Q. Survey on learning-to-rank based recommendation algorithms[J]. Journal of Software, 2016, 27(3): 691-713.
- [8] GOLDBERG D. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.
- [9] PARK C, KIM D, OH J, et al. TRecSo: enhancing top- k recommendation with social information[C]//International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee. 2016: 89-90.
- [10] 王海艳, 杨文彬, 王随昌. 基于可信联盟的服务推荐方法[J]. 计算机学报, 2014, 37(2): 301-311. WANG H Y, YANG W B, WANG S C. A service recommendation method based on trustworthy community[J]. Chinese Journal of Computers, 2014, 37(2): 301-311.
- [11] LIU J, TANG M, ZHENG Z, et al. Location-aware and personalized collaborative filtering for Web service recommendation[J]. IEEE Transactions on Services Computing, 2015, 9 (5): 686-699.
- [12] HU Y, PENG Q, HU X. A time-aware and data sparsity tolerant approach for Web service recommendation[C]//IEEE International Conference on Web Services. 2014: 33-40.
- [13] WEI L, YIN J, DENG S, et al. Collaborative Web service QoS prediction with location-based regularization[C]//IEEE International Conference on Web Services. 2012: 464-471.
- [14] 胡堰, 彭启民, 胡晓惠. 一种基于隐语义概率模型的个性化 Web 服务推荐方法[J]. 计算机研究与发展, 2014, 51(8): 1781-1793. HU Y, PENG Q M, HU X H. A personalized Web service recommendation method based on latent semantic probabilistic model[J]. Journal of Computer Research and Development, 2014, 51(8): 1781-1793.
- [15] WANG X Y, ZHU J K, ZHENG Z B, et al. A spatial-temporal QoS prediction approach for time-aware Web service recommendation[J]. ACM Transactions on the Web, 2016, 10(1): 1-25.
- [16] WEIMER M, KARATZOGLOU A, LE Q V, et al. Cofrank maximum margin matrix factorization for collaborative ranking[C]//The 21th Int'l Conference on Neural Information Processing Systems. 2007: 1-8.
- [17] SHI Y, KARATZOGLOU A, BALTRUNAS L, et al. TFMAP: optimizing MAP for top- n context-aware recommendation[C]//ACM Special Interest Group on Information Retrieval. 2012: 155-164.
- [18] MOLLICA C, TARDELLA L. Bayesian mixture of Plackett-Luce models for partially ranked data[J]. Statistics, 2015, 2(4): 208-222.
- [19] CAO Z, QIN T, LIU T Y, et al. Learning to rank: from pairwise approach to listwise approach[C]//The 2007 ACM conference on Machine learning. 2007: 129-136.
- [20] FANG W, ZHANG C, SHI Z, et al. BTRES: beta-based trust and reputation evaluation system for wireless sensor networks[J]. Journal of Network & Computer Applications, 2015(59): 88-94.
- [21] 郭弘毅, 刘功申, 苏波, 等. 融合社区结构和兴趣聚类的协同过滤推荐算法[J]. 计算机研究与发展, 2016, 53(8): 1664-1672. GUO H Y, LIU G S, SU B, et al. Collaborative filtering recommendation algorithm combining community structure and interest clusters[J]. Journal of Computer Research and Development, 2016, 53(8): 1664-1672.
- [22] ZHENG Z B, ZHANG Y L, LYU M R. Distributed QoS evaluation for real-world Web services[C]//The 8th International Conference on Web Services. 2010: 83-90.
- [23] 贾冬艳, 张付志. 基于双重邻居选取策略的协同过滤推荐算法[J]. 计算机研究与发展, 2013, 50(5): 1076-1084. JIA D Y, ZHANG F Z. A collaborative filtering recommendation algorithm based on double neighbor choosing strategy[J]. Journal of Computer Research and Development, 2013, 50(5): 1076-1084.
- [24] LIU J, WU C, XIONG Y, et al. List-wise probabilistic matrix factorization for recommendation[J]. Information Sciences, 2014(278): 434-447.

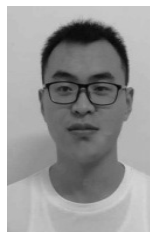
[作者简介]



方晨 (1993-), 男, 安徽宿松人, 信息工程大学硕士生, 主要研究方向为服务推荐、数据挖掘等。



张恒巍 (1978-), 男, 河南洛阳人, 博士, 信息工程大学副教授, 主要研究方向为网络安全与攻防对抗、信息安全风险评估。



张铭 (1993-), 男, 河南安阳人, 信息工程大学硕士生, 主要研究方向为云资源调度。

王晋东 (1966-), 男, 山西洪洞人, 信息工程大学教授, 主要研究方向为网络与信息安全、云资源管理。