基于覆盖随机游走算法的服务质量预测

张以文[1),2)] 汪开斌[2)] 严远亭[2)] 陈 洁[1),2)] 何 强[3)] 李 炜[1),2)]

[1)](安徽大学计算智能与信号处理教育部重点实验室 合肥 230031)

[2)](安徽大学计算机科学与技术学院 合肥 230601)

[3)](澳大利亚斯威本科技大学电子信息及软件工程学院 墨尔本 3122)

摘 要 随着互联网上 Web 服务的日益增多,面对大量功能相同的候选服务,用户希望能够选择质量最优的候选服务.然而,用户通常并不知道所有候选服务的服务质量(Quality of Service,QoS).因此,基于 Web 服务的历史记录预测 QoS 值得到了广泛关注.传统的基于协同过滤(CF)的预测方法可能会遭遇数据稀疏、用户信任等问题,导致该方法在预测精度方面表现一般.为解决上述问题,该文提出一种基于覆盖随机游走算法的服务质量预测方法.该方法首先基于用户服务历史 QoS 记录,使用改进的覆盖算法对用户进行聚类,选取与每个用户聚类次数的 Top-k 个用户为该用户的信任用户,连接所有用户与其信任用户构建用户信任网;其次,基于用户信任网提出一种随机游走预测方法,在随机游走的过程中,不仅考虑目标服务的 QoS 信息,同时考虑相似服务的 QoS 信息,以确保 QoS 预测的准确性;最后,每次随机游走获得一个 QoS 值,为使预测更加准确,作者进行多次随机游走,汇总所有 QoS 值进行预测.为验证文中方法的有效性,作者在真实的 Web 服务数据集进行了大量实验,其中包括来自 339 个用户的 5825 个真实世界 Web 服务的 1974675 个 Web 服务调用.实验结果表明文中方法在预测精度上明显优于现有方法,同时可以很好地解决推荐系统的数据稀疏和用户信任问题.

关键词 服务质量;质量预测;随机游走;覆盖算法;协同过滤

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2018.02756

Service Quality Prediction Based on Covering Random Walk Algorithm

ZHANG Yi-Wen[1),2)] WANG Kai-Bin[2)] YAN Yuan-Ting[2)] CHEN Jie[1),2)] HE Qiang[3)] LI Wei[1),2)]

[1)](*Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei* 230031)

[2)](*School of Computer Science and Technology, Anhui University, Hefei* 230601)

[3)](*School of Information Technology, Swinburne University of Technology, Melbourne* 3122)

Abstract As the software components, Web services are designed to support interoperable machine-to-machine interaction over a network. With the increasing number of the Web services on the Internet, users always want to choose the best candidate services when facing a large number of similar candidate services with the same function. However, users usually do not know the Quality of Service (QoS) of all candidate services. Therefore, the prediction of QoS based on the historical records of the Web services has attracted extensive attention from academia and industry. The traditional prediction methods based on Collaborative Filtering (CF) have been successfully employed by some studies to address the Web service prediction problem. However, these methods usually suffer from the problems such as data sparseness and user trust, which may cause poor performance even failures of Web service QoS prediction. In the recent literature, the random walk algorithm is successfully applied to the recommendation models, which can effectively solve the data sparse problem. However, the performance of recommendation accuracy is not

收稿日期:2017-12-26;在线出版日期:2018-07-09.本课题得到国家自然科学基金(61602003)、国家科技支撑计划(2015BAK24B01)、安徽省自然科学基金(1808085MF197)资助.**张以文**,男,1976 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为服务计算、推荐系统、群体智能.E-mail:zywahu@qq.com.**汪开斌**,男,1993 年生,硕士研究生,主要研究方向为服务计算.E-mail:wkbahu@qq.com.**严远亭**,男,1986 年生,博士,讲师,主要研究方向为机器学习、粒度计算.**陈 洁**,女,1982 年生,博士,副教授,主要研究方向为机器学习、粒度计算.**何 强**,男,1982 年生,博士,高级讲师,主要研究方向为服务计算、软件工程.**李 炜**,女,1969 年生,博士,教授,主要研究领域为软件工程、数据挖掘.

ideal, especially when it applied to the user-service classic recommendation model. To solve the above problems, this paper proposes a service quality prediction method based on the covering random walk algorithm, which is a random walk model that takes into account the influence of trust between users and users on Web service prediction, and the method employs a covering algorithm to find trust users. The clustering method does not require the number of clusters to be pre-specified or the initial centers to be manually selected, thus it can ensure the stability of the prediction. Firstly, the proposed method, based on the QoS records of user service history, employs an improved covering algorithm to cluster the users, and the Top-k users who are clustered with each user are selected as trusted users of the user, then all users are connected to their trusted users to build a user trust network. Secondly, based on the user trust network, a random walk prediction method is proposed. In the process of the random walk, not only does the QoS information of the target service is considered, but also the QoS information of the similar service is considered. The reason is that the relationship between these users and the source user will become weak and the QoS values obtained will become unreliable when the walks go too far from the source user in the trust network. Finally, a single random walk can only return a QoS value. To acquire more QoS values and obtain a more reliable prediction, several random walks are desirable and integrate all QoS values returned by different random walks for prediction. To verify the performance of the proposed approach, we conducted a large number of experiments in the real Web service datasets, including 1 974 675 Web service records of 5825 real-world Web services from 339 users. The experimental results show that the proposed method is obviously superior to the existing methods in the prediction accuracy, and it can solve the problems of data sparseness and user trust of the recommended system at the same time.

Keywords　quality of service; quality prediction; random walk; covering algorithm; collaborative filtering

1　引　言

Web 服务是用来支持机器与机器间跨网络互操作而设计的软件组件[1]. 随着互联网上 Web 服务的日益增多,向用户推荐理想的 Web 服务变得更具挑战性. 为从一组具有相同功能的 Web 服务中找到高质量的服务,服务质量(Quality of Service,QoS)被广泛用于描述和评估 Web 服务的非功能属性. 服务质量通常被定义为一组 Web 服务的用户感知属性,如响应时间、吞吐量、可用性和可靠性等. 基于 QoS 的 Web 服务发现和选择得到了学术界和工业界的广泛关注[2-4]. 然而,现实中多数 Web 服务的 QoS 信息对用户来说是未知的,并且 Web 服务的 QoS 属性容易受到环境的影响,不同位置的用户在同一个服务上可能观察到不同的 QoS 值. 而且一些 QoS 属性(如信任度、可靠性)很难被评估,需要长时间观察和大量的调用. 因此这些挑战需要更有效的方法来获取 Web 服务的 QoS 信息.

近年来对 Web 服务 QoS 预测的方法主要是基于协同过滤推荐技术[2-3],一般包括基于记忆的协同过滤和基于模型的协同过滤两种. 基于记忆的协同过滤主要过程是首先通过皮尔逊相关系数(Pearson Correlation Coefficient,PCC)计算相似度来寻找相似用户或相似服务[5-6],然后使用相似用户或相似服务的 QoS 值对缺失值进行预测. 基于模型的协同过滤方法需要根据训练集数据和机器学习方法得到一个复杂模型[7],并结合相似用户的历史数据来预测目标服务的 QoS 值. 当用户服务的 QoS 矩阵相对密集时协同过滤方法是非常有效的. 然而在现实中,一个用户通常只调用过很少的服务,所以用户服务的 QoS 矩阵往往比较稀疏. 这种情况下,传统的协同过滤方法很难通过计算相似度寻找相似用户或相似服务. 因此,QoS 预测精度会受到限制. 此外,在对服务评价过程中,可能会出现一些用户对一些服务进行恶意的评价,即可能对不好的服务给出很高的评分或对好的服务给出很低的评分,进一步降低了预测精度.

基于信任的预测方法可以一定程度上缓解恶意评价问题,提高预测精度. 然而当数据十分稀疏时,

该类方法的预测性能则有待进一步提高. 为说明本文的研究动机,图 1 给出一个 Web 服务应用场景,其中 s_1,s_2,s_3,…,s_8 代表 Web 服务,每个用户的箭头指向其信任用户,服务图标下方表示用户对服务的评分. 假设推荐系统希望预测用户 u_1 对目标服务 s_6 的 QoS 值,当数据十分稀疏时,系统发现用户 u_1 的信任用户中没有用户评价过目标服务 s_6,这将导致预测失败. 而实际上用户 u_1 信任用户的信任用户有对服务 s_6 做过评价. 如何在 Web 服务 QoS 信息十分稀疏的情况下同时考虑用户信任关系,从而提高 QoS 预测精度是本文的研究目的. 为此,本文提出了基于覆盖随机游走算法(Covering Random Walk Algorithm),简称 CRWA. 实验结果表明 CRWA 方法较大程度上提高了预测精度. 本文的主要贡献如下:

(1)本文将覆盖算法应用到 Web 服务 QoS 预测中,利用该算法计算用户信任度和服务关联度. 与经典的聚类算法相比,改进的用于聚类的覆盖算法不需要预先指定类的数量和初始质心,从而保证了预测的稳定性;

(2)本文在聚类结果的基础上,选取每个用户的 Top-k 个信任用户,构建用户信任网,进而结合随机游走提出了一种覆盖随机游走算法 CRWA. 该算法不仅考虑了用户之间的信任关系,对数据高度稀疏的应用场景同样具有很高的预测精度;

(3)本文使用真实的 Web 服务 QoS 数据集对所提出的方法进行了实验评估. 结果表明,CRWA 兼顾用户信任问题的同时能够很好地处理数据高度稀疏的问题.

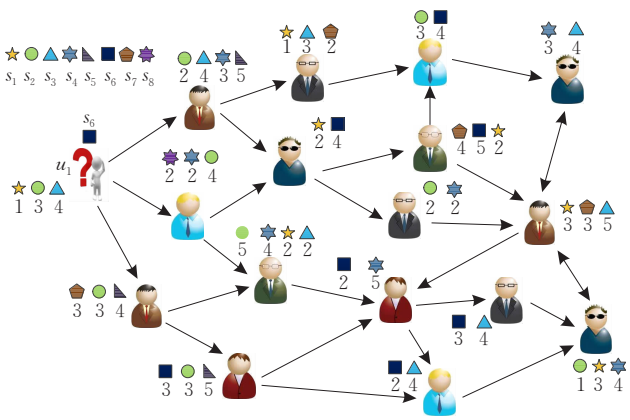


图 1 Web 服务应用场景

本文第 2 节介绍 QoS 预测的相关工作;第 3 节详细解释 CRWA 算法的实现过程;第 4 节是实验结果与分析;最后对本文工作进行总结.

2 相关工作

协同过滤已经广泛应用于各种商业推荐系统执行个性化推荐[8-9]. Breese 等人[10]提出基于用户的协同过滤方法. Sarwar 等人[11]提出基于项目的协同过滤方法. McLaughlin 和 Herlocker[12]通过修改传统的 Pearson 相关系数(PCC)来改进用户相似度计算. Liu 等人[13]研究了项目的个性化影响,通过用户评价过的公共的项目来计算用户之间的相似度. 如今,协同过滤已成功应用于 QoS 感知的 Web 服务预测和推荐. Shao 等人[5]将协同过滤技术用于 Web 服务 QoS 预测,提出了一种基于用户的协同过滤方法. Zheng 等人[14]提出了 WSRec,一种混合的协同过滤方法,为预测用户和目标服务缺失的 QoS 值,结合了基于用户和基于服务的协同过滤技术,并引入信心权重来平衡基于用户和基于服务的协同过滤所产生的影响. 在这些工作的基础上,一些研究提出了改进的协同过滤方法,以进一步提高预测精度. 考虑到用户和服务的个性化特征,Jiang 等人[15]通过发现用户和服务的 QoS 个性化特征以改进相似度计算方法,从而提出了一种个性化的协同过滤 Web 服务 QoS 预测方法. Wu 等人[16]通过使用用户服务 QoS 矩阵的数据平滑技术提出了一种改进的协同过滤方法,可以在一定程度上缓解数据稀疏性问题,从而提高 QoS 预测精度. Ma 等人[17]客观的分析了 Web 服务 QoS 值,对传统协同过滤方法进行精细化,提升了 QoS 预测性能. Qiu 等人[18]将用户的声誉纳入到 Web 服务 QoS 预测和推荐的协同过滤中.

基于协同过滤方法进行 QoS 预测的关键是计算相似用户或相似服务. 因此,一些工作尝试使用聚类算法来计算相似用户或相似服务. Zhang 等人[19]提出了一种模糊聚类方法来聚类用户. 该方法具有模糊聚类和 PCC 的优点,但该方法的主要限制是初始选择的质心对聚类结果有显著影响. Yu 等人提出了一种名为 CluCF 的方法[20],采用 K-means 算法对用户和 Web 服务进行聚类,该方法的重点是降低更新新用户和新服务聚类的复杂性. 在对服务评价过程中,可能会出现部分用户对服务进行恶意评价的情况,导致计算相似用户或相似服务效果不佳. 因此,Wu 等人[21]采用 K-means 算法对用户进行聚类,该方法的目的是识别不可信任用户,文献[21]提出的预测方法作为 UCAPK 在本文实验部分得到了实现.

虽然上述聚类方法在一定程度上提高了预测精度,但基于 K-means 预测方法的聚类结果严重依赖

于预先指定的聚类数和最初选择的质心，并且当数据十分稀疏时，这些方法在预测精度上仍然表现不佳. 在最近的文献中，随机游走算法被应用于推荐模型中[22-23]，可以有效地解决数据稀疏问题. 然而，在推荐准确性方面，表现不太理想，特别是应用于用户服务经典推荐模型时. 一些研究提出了改进的随机游走算法，以提高预测精度. Jamali 和 Ester[24] 提出了随机游走模型 TrustWalker，结合基于信任和基于项目的协同过滤推荐算法. 然而该方法假设提供了用户信任网，并给出了用户之间的信任值，忽略了信任用户之间的差异. Tang 和 Dai 等人[25] 提出了随机游走模型 WSWalker，结合基于地理位置和协同过滤的方法进行 QoS 预测.

针对上述问题，以及受现有研究的启发，本文提出一种基于覆盖随机游走算法的服务质量预测方法. 该方法将覆盖算法与随机游走算法相结合，改进的覆盖算法不需要预先指定类的数量和初始质心，并且在随机游走算法的基础上不仅考虑了目标服务，同时考虑了目标服务的相似服务，可有效提高服务质量的预测精度.

3　CRWA 预测方法

本文提出的 CRWA 服务质量预测方法总体框架描述如图 2 所示. CRWA 主要包括以下几个功能模块：

(1) 用户信任网的构建. 根据每个服务的历史 QoS 值使用覆盖算法对用户进行聚类，求出每个用户与其他用户覆盖在同一个类的次数，对该次数从大到小进行排序，接着使用 Top-k 机制选取前 k 个用户为信任用户，连接每个用户与其 k 个信任用户建立用户信任网；

(2) 随机游走. 根据每个用户的历史 QoS 值使用覆盖算法对服务进行聚类. 基于用户与用户、服务与服务的聚类次数，在用户信任网进行随机游走. 该算法不仅考虑了目标服务的 QoS 值，同时也考虑了相似服务的 QoS 值. 一次随机游走返回用户 u 对目标服务 i 的一个 QoS 预测值；

(3) QoS 预测. 进行多次随机游走，将多次随机游走返回的所有评分汇总，预测用户 u 对目标服务 i 的 QoS 值 $\hat{r}_{u,i}$.

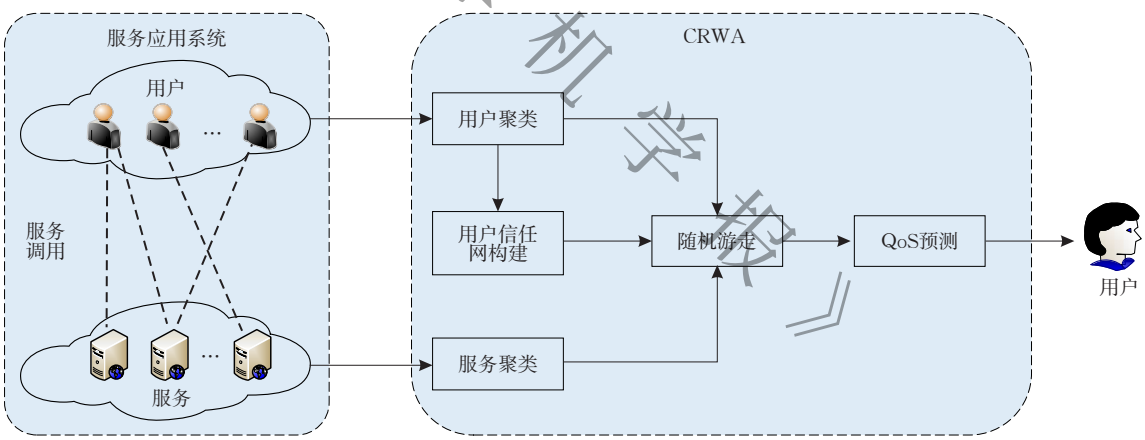


图 2　CRWA 方法的总体描述

在用户信任网进行随机游走搜索，并且在服务选择部分考虑相似服务的 QoS 值以避免在信任网中过度深入. 我们的方法优先选择信任用户与原用户更相似来提高准确率. 为了预测原用户 u_1 对目标服务 i 的 QoS 值，首先咨询用户 u_1 的信任用户的 QoS 信息，如果该信任用户对目标服务 i 有过评价，则该 QoS 值将作为用户 u_1 对目标服务 i 的一个 QoS 预测. 否则，选择该信任用户已评价的与目标服务相似的服务的 QoS 值作为对服务 i 的 QoS 预测，或继续咨询该信任用户的信任用户. 随机游走算法的细节将会在本节的后面讨论，每次随机游走返回一个评分. 进行多次随机游走，将不同随机游走返回的所有评分汇总视为预测评分 $\hat{r}_{u_1,i}$.

在下面的小节中，我们将讨论随机游走算法的细节. 为了便于后面的阐述，表 1 列出了本文方法所

表 1　CRWA 中使用的符号

符号	描述
$\phi_{u,i,k}$	随机游走第 k 步到达用户 u 并停止的概率
$X_{u,i,k}$	原用户 u 第 k 步游走到达某一用户的随机变量
$X_{u,i}$	原用户 u 游走一些步数到达某一用户的随机变量
S_u	在集合 NU_u 中选择某一用户的随机变量
$Y_{u,i}$	选择用户 u 评价过的某一服务的随机变量
$XY_{u,i}$	原用户 u 开始随机游走在某一用户停止并选择该用户评价过的某一服务的随机变量
$r_{u,i}$	用户 u 对服务 i 的真实 QoS 值
$\hat{r}_{u,i}$	用户 u 对服务 i 的预测 QoS 值
$t_{u,i}$	用户 u 和用户 v 之间的信任值

用的所有符号，用 $u,v,w,\cdots$ 表示用户，$i,j,\cdots$ 表示服务，k 表示每次随机游走的步数，NU_u 表示用户 u 的信任用户集合.

3.1 构建用户信任网

Tang 等人[25]认为用户地理位置越相近则越相似. 他们将每个用户与其地理位置最近的 k 个用户连接构建用户网，通过用户的经度和纬度计算距离. 但是我们知道尽管两个用户在物理距离接近，但他们的网络距离可能很远，从而会影响预测精度，且该方法没有考虑用户信任的问题. 在本文中，我们考虑将每个用户和他们最信任的 k 个用户连接起来构建用户信任网，这些用户将会对预测精度提供有意义的信息. 如何确定用户之间的信任值是关键问题.

在本文的工作中，我们使用改进的覆盖算法计算用户间的信任值. 传统的覆盖算法是由张铃教授和张钹院士根据神经网络的几何意义提出的[26]，利用 M-P 神经元模型，得出了一个领域覆盖的规则，具有识别率高、计算速度快等优点.

然而传统的覆盖算法解决的是分类问题，分类算法属于有监督学习，必须事先明确知道各个类别的信息. 在多数情况下该条件无法得到满足，尤其是在处理海量数据的时候，如果通过预处理使得数据满足分类算法的要求，则代价非常大. 本文方法在构建用户信任网时，计算用户间的信任值是最关键的问题，分类算法无法满足本文研究的需要，但聚类方法可以很好地解决. 因此，我们改进了传统的覆盖算法. 改进的覆盖聚类算法与经典的聚类算法相比，不需要预先指定类的数量和初始质心，保证了预测的稳定性.

根据传统覆盖算法的分析，我们改进了该算法，可以得出使用覆盖算法进行聚类，该算法将数据点 $D=\{d_1,d_2,\cdots,d_g\}$ 划分成多个覆盖，其中 $g\leq m$，m 表示样本点的个数. 在对这种迭代算法的描述中，每次迭代时新产生的覆盖被称为当前覆盖，用 C_{cr} 表示，覆盖中心用 c_{cr} 表示，覆盖半径用 r_{cr} 表示. $D_{uc}(D_{uc}\in D)$ 表示未被覆盖的样本点集合. 算法的主要步骤如下：

1. 求出 D_{uc} 的重心 $CD=(\bar{x}_1,\cdots,\bar{x}_p)$，其中 $\bar{x}_p=\dfrac{\sum_{i=1}^{g}x_{i,p}}{g}$，$x_{i,p}$ 表示集合 d_i 的第 p 维坐标；

2. 确定 D_{uc} 中离重心 CD 最近的样本点 $c_{cr}(cr=1)$，并将其作为第一个覆盖 C_1 的中心，如图 3(a)所示：

$$\min\sqrt{\sum_{j=1}^{p}(x_{i,j}-x_{D,j})^2},\ \forall d_i\in D_{uc} \tag{1}$$

其中 $x_{D,j}$ 表示 CD 的第 j 维坐标；

3. 计算 D_{uc} 中所有样本点到 c_{cr} 的平均距离 r_{cr}，并把 r_{cr} 作为当前覆盖的半径：

$$r_{cr}=\frac{\sum_{d\in D_{uc}}\sqrt{\sum_{j=1}^{p}(x_{d,j}-x_{c,j})^2}}{|D_{uc}|} \tag{2}$$

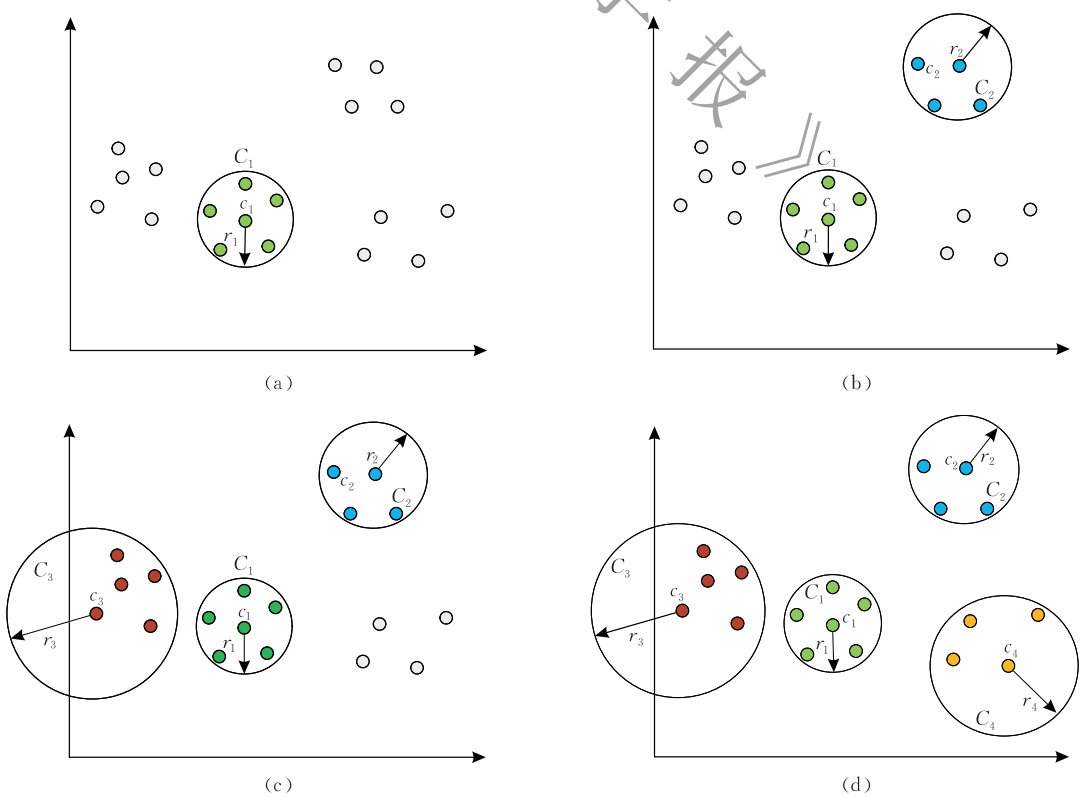


图 3 覆盖聚类过程示例

4. 计算出 D_{uc} 中离当前覆盖中心 c_{cr} 最远的样本点作为下一次覆盖的中心 c_{cr}，如图 3(b)所示：

$$\max\sqrt{\sum_{j=1}^{p}(x_{i,j}-x_{c,j})^2},\ \forall d_i\in D_{uc} \tag{3}$$

从 D_{uc} 中删除当前覆盖内所有的样本点；

5. 重复步骤 3 和步骤 4，直到 D_{uc} 为空为止，如图 3(c)和(d).

对于调用同一个服务的不同用户，如果他们对服务的 QoS 值越相近，当使用覆盖算法聚类时，他们被分在同一个覆盖中的可能性越大. 也就是说，根据每个服务的历史 QoS 值使用覆盖算法对用户进行聚类，在同一个覆盖内用户间的相似度大于不同覆盖的用户间的相似度. 因此我们对每个服务的不同用户进行聚类，求出每个用户与其他用户覆盖在同一个类的次数，并对该次数排序. 如果两个用户被聚类到同一覆盖中的次数越多，说明这两个用户之间的信任度越高，用户 u 和用户 v 聚类在同一覆盖的次数记为 $t_{u,v}$. 选取与每个用户覆盖次数的 Top-k 个用户为该用户的信任用户，通过连接每个用户与其信任用户构建用户信任网. 用户信任网可以被定义为 $G=\langle U,TU\rangle$，其中 $TU=\{(u,v)\mid u\in U,v\in NU_u\}$.

3.2　随机游走

假设从原用户 u_0 开始随机游走，在随机游走的第 k 步，到达某一用户 u，如果用户 u 已经对目标服务 i 做过评价，则停止随机游走并返回 $r_{u,i}$ 作为本次随机游走的结果. 如果用户 u 未对目标服务 i 做过评价，则有两种选择：

(1) 以概率 $\phi_{u,i,k}$ 停止随机游走，随机选择用户 u 评价过的与目标服务 i 相似的服务 j，并返回 $r_{u,j}$ 作为本次随机游走的结果；

(2) 以概率 $1-\phi_{u,i,k}$ 继续随机游走到用户 u 的某一信任用户 v.

如果决定在用户 u 继续进行随机游走，则不得不选择该用户的某一信任用户并走向他. Jamali 等人[24]认为用户间的信任度没有差异，选择走向每个信任用户的概率都相等，然而用户与用户间的信任值是有高低之分的. 在本文中，我们认为信任用户间是有差异的，现设 S_u 为从 NU_u 中选择某一用户 v 的随机变量，在用户 u 的信任用户集合中选择用户 v 的概率定义为

$$P(S_u=v)=\frac{t_{u,v}}{\sum_{w\in NU_u}t_{u,w}} \tag{4}$$

则有

$$P(X_{u_0,i,k+1}=v\mid X_{u_0,i,k}=u,\hat{R}_{u,i})=$$
$$(1-\phi_{u,i,k})\times P(S_u=v)=(1-\phi_{u,i,k})\times\frac{t_{u,v}}{\sum_{w\in NU_u}t_{u,w}} \tag{5}$$

其中，$X_{u_0,i,k+1}=v$ 表示原用户 u_0 在第 $k+1$ 步游走到达用户 v. $\hat{R}_{u,i}$ 的条件是在第 k 步用户 u 未评价过目标服务 i. 从用户 u 到用户 v 的游走概率独立于之前的行走步数，但是 $\phi_{u,i,k}$ 依赖于步数 k，所以并不独立于之前的随机游走步数.

如果决定停止在用户 u，则不得不选择用户 u 已评价的某一服务 j. Jamali 等人[24]和 Tang 等人[25]使用皮尔逊相关系数(PCC)计算服务间的相似度，利用服务间的相似度来作为选择概率. 但是我们知道在现实生活中，用户服务矩阵非常稀疏. 此外在服务评价的过程中可能会出现一些恶意用户给出一些恶意的评价，导致一些 QoS 值可能不准确，因此使用 PCC 计算相似度可能不够准确，导致这些方法预测精度不高. 在本文中，和基于每个服务给用户聚类类似，我们基于每个用户给服务进行覆盖聚类. 根据目标服务 i 和服务 j 在同一个覆盖的次数与服务 i 和该用户已评价的所有服务在同一个覆盖的总次数的比值作为选择概率. 选择服务 j 的概率计算如下：

$$P(Y_{u,i}=j)=\frac{x_{i,j}}{\sum_{l\in I_u}x_{i,l}} \tag{6}$$

其中，I_u 表示用户 u 已评价的服务集合，$Y_{u,i}$ 表示选择用户 u 评价过的与目标服务 i 相似的服务 j 的随机变量. $x_{i,j}$ 表示服务 i 和服务 j 在同一个覆盖的次数. 用 $r_{u,j}$ 作为本次随机游走的结果.

为了定义整个概率分布，我们定义当条件 $r_{u,i}$ 为真或用户 u 未评价过的服务的概率如下：

$$\forall v\neq u\ P(X_{u_0,i,k+1}=v\mid X_{u_0,i,k}=u,R_{u,i})=0 \tag{7}$$
$$\forall j\notin I_u\ P(Y_{u,i}=j)=0 \tag{8}$$

当随机游走 k 步到达用户 u，以 $\phi_{u,i,k}$ 概率停止游走，并选择用户 u 已评价服务的 QoS 值作为本次游走的结果. 该概率和式(6)有关. 我们考虑式(6)的最大值作为停止游走的概率.

一般而言，离原用户较远的用户对目标服务 i 的评分可靠性越低. 因此，在用户信任网中随机游走的越深，继续游走的概率就会越小，$\phi_{u,i,k}$ 的值则会越大. 结合 $\phi_{u,i,k}$ 中随机游走的步数 k，我们设计了 $\phi_{u,i,k}$ 的计算方式，k 值越大停止概率越大.

$$\phi_{u,i,k}=\max\frac{x_{i,j}}{\sum_{l\in I_u}x_{i,l}}\times\frac{1}{1+e^{-\frac{k}{2}}} \tag{9}$$

对于随机游走的每一步,有如下 3 种停止条件:

(1)到达某一用户,该用户对目标服务已评价,则返回该用户对目标服务的 QoS 值作为本次随机游走的结果;

(2)到达用户 u 并决定在该用户停止,选择该用户已评价的某一服务,返回用户 u 对该服务的 QoS 值作为本次随机游走的结果;

(3)一次随机游走可能一直走下去.为了防止这种情况,我们需要限制随机游走的深度.根据社交网络中"六度分离"的思想,我们设置最大深度为 6.

3.3 终止条件

为获得更多的 QoS 值并且得到更可靠的预测,需要多次随机游走.我们需要决定何时完成足够的随机游走以获得准确的预测值 $\hat{r}_{u,i}$.

设 r_i 表示第 i 次随机游走返回的 QoS 值,$\bar{r}$ 表示 T 次随机游走返回的平均 QoS 值.我们计算 T 次随机游走结果的方差如下:

$$\sigma^2=\frac{\sum_{i=1}^{T}(r_i-\bar{r})^2}{T} \tag{10}$$

同时假设变量 σ_i^2 为前 i 次随机游走结果的方差.由于 Web 服务的 QoS 值通常在有限的范围内,因此可以证明当 T 不断增加时,方差将收敛到一个常数.如果 $|\sigma_{i+1}^2-\sigma_i^2|<\varepsilon$,则随机游走终止,其中 ε 是一个小的可调参数.

3.4 QoS 预测

我们使用多次随机游走返回的 QoS 值来预测原用户 u 对目标服务 i 的 QoS 值,如式(11)所示.为了进一步在 QoS 预测中考虑用户、服务以及用户服务的个性特征,我们把式(11)演变成三个公式,即式(12)、式(13)和式(14),其中 $\bar{u}$ 表示用户 u 已评价服务的 QoS 平均值,$\bar{i}$ 表示服务 i 的 QoS 平均值,参数 $\lambda(0\leq\lambda\leq1)$ 用于决定预测方法依赖基于用户和基于服务的程度.

$$\hat{r}_{u,i}=\frac{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))r_{v,j}}{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))} \tag{11}$$

$$\hat{r}_{u,i}^{U}=\bar{u}+\frac{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))(r_{v,j}-\bar{v})}{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))} \tag{12}$$

$$\hat{r}_{u,i}^{I}=\bar{i}+\frac{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))(r_{v,j}-\bar{j})}{\sum_{\{(v,j|R_{v,j})\}}P(XY_{u,i}=(v,j))} \tag{13}$$

$$\hat{r}_{u,i}^{UI}=\lambda\times\hat{r}_{u,i}^{U}+(1-\lambda)\times\hat{r}_{u,i}^{I} \tag{14}$$

当随机游走从原用户 u 开始出发寻找目标服务 i,$XY_{u,i}=(v,j)$ 表示停止在用户 v 并选择用户 v 评价过的某一服务 j 的随机变量.如前所述,$R_{v,j}$ 是布尔变量表示用户 v 是否评价过服务 j.其概率有如下三种情况:

$$P(XY_{u,i}=(v,j))=\begin{cases}P(X_{u,i}=v)\phi_{u,i}P(Y_{v,i}=j), & v\neq u;i\neq j\\ P(X_{u,i}=v), & v\neq u;i=j\\ \phi_{v,i,1}P(Y_{v,i}=j), & v=u;i\neq j\end{cases} \tag{15}$$

在第一种情况中,用 $\phi_{u,i}$ 代替 $\phi_{u,i,k}$ 是因为未考虑到因素 k,不知道到达用户 v 需要多少步.情况 $v=u$ 和 $i=j$ 表示用户自己评价过目标服务,无须预测.

一次随机游走从用户 u 到达用户 v 可以有不同的步数.随机变量 $X_{u,i,k}=v$ 表示从用户 u 开始经过 k 步到达用户 v,其概率计算如下:

$$P(X_{u,i,k}=v)=\sum_{w\in U}P(X_{u,i,k-1}=w)(1-\phi_{w,i,k})P(S_w=v) \tag{16}$$

特别地,$P(X_{u,i,0}=u)=1$.因而:

$$P(X_{u,i}=v)=\frac{\sum_{k=1}^{\infty}P(X_{u,i,k}=v)}{\sum_{w\in U}\sum_{k=1}^{\infty}P(X_{u,i,k}=w)} \tag{17}$$

3.5 算法复杂度分析

假设数据集是一个 $m\times n$ 矩阵,其中包括 m 个用户和 n 个服务,矩阵中的元素表示用户调用服务的 QoS 值.为确定每个用户的信任用户,本文首先使用覆盖聚类算法对用户进行聚类.根据 3.1 节描述的算法分析可知覆盖聚类算法的时间复杂度为 $O(mn\log m)$.其次为找到每个用户的 Top-k 个信任用户,必须将每个用户和其他用户间的信任值进行排序.文中采用堆排序,因此排序的时间复杂度为 $O(\text{Top-}k\times(m-1)\log_2(m-1))=O(m\log m)$,$m$ 个用户的时间复杂度为 $O(m^2\log m)$.因此该部分总的时间复杂度为 $O(mn\log m)+O(m^2\log m)$.

在随机游走算法预测未知 QoS 值的过程中,根据社交网络中"六度分离"的思想,每次随机游走至多行走 6 步,选择用户已评价的目标服务或随机选择用户已评价的与目标服务相似的服务,时间复杂度为 $O(k)+O(1)=O(1)$.因此预测部分总的时间复杂度为 $O(Nc)$,其中 N 表示待预测 QoS 值的数量,c 表示多次随机游走达到终止条件的游走次数.综上所述 CRWA 的总体时间复杂度为 $O(mn\log m)+O(m^2\log m)+O(Nc)$.

4　实验结果与分析

4.1　数据集

为了评价本文方法的有效性，我们采用真实的 Web 服务数据集 WSDream 作为测试数据集[27]. 该数据集包括 1974675 条 QoS 记录，这些记录是通过分布在 30 个国家的 339 台计算机(用户)对分布在 73 个国家的 5825 个 Web 服务进行调用得到的. 每个用户和每个 Web 服务之间都有一条通过调用产生的 QoS 记录，每个 QoS 记录有两个 QoS 属性，响应时间(RT)和吞吐量(TP). 同时，该数据集中还收集了这些用户的 IP 地址、服务的 URL 以及它们所在的国家等信息.

在现实世界中，用户项目矩阵通常非常稀疏，因为用户通常只调用过少量的 Web 服务. 因此在本文中，为使实验更真实，我们随机删除了初始 RT 和 TP 矩阵一定数量的 QoS 值，以生成低密度矩阵. 例如，矩阵密度为 5%意味着我们随机选择 5%的 QoS 值来预测其余 95%的 QoS 值. 移除的原始 QoS 值用作预期值来研究预测准确性.

实验环境为：JDK 1.7、Scala 语言、Ubuntu 14.04 LTS、Intel i7-4790、CPU 3.60GHz 和 4GB RAM.

4.2　评估指标

为评估本文方法的预测性能，采用平均绝对误差 MAE(Mean Absolute Error)和均方根误差 $RMSE$(Root Mean Squared Error)作为评测指标. 通过计算预测的用户评分与实际用户评分之间的偏差来度量预测的准确性.

MAE 是预测的 QoS 值与实际 QoS 值之间的偏差. 定义如下：

$$MAE=\frac{\sum_{u,i}|r_{u,i}-\hat{r}_{u,i}|}{L} \tag{18}$$

$RMSE$ 定义如下：

$$RMSE=\sqrt{\frac{\sum_{u,i}(r_{u,i}-\hat{r}_{u,i})^2}{L}} \tag{19}$$

其中，$r_{u,i}$ 和 $\hat{r}_{u,i}$ 分别表示实际 QoS 值和预测 QoS 值，L 表示预测的 QoS 值数量. 从公式可以观察到 $RMSE$ 对较大的误差反应比较敏感. MAE 和 $RMSE$ 值越小表示预测方法的预测性能越好，反之越差. 本文方法设置 CRWA 的终止条件 $\varepsilon=0.0001$，多次随机游走的最大阈值为 10000 次.

4.3　性能对比

为更好地评估本文方法 CRWA 的性能，在实验中比较下面几种较经典的 QoS 预测方法. 列举如下：

(1) UPCC(基于用户的协同过滤)[5]：基于用户的协同过滤算法，使用 PCC 来计算不同用户间的相似度. 使用与当前活跃用户相似的其他用户的 QoS 体验为当前用户预测目标服务的 QoS 值；

(2) IPCC(基于服务的协同过滤)[11]：基于服务的协同过滤算法，使用 PCC 来计算不同 Web 服务间的相似度. 使用与当前 Web 服务相似的其他服务的 QoS 值预测目标服务的 QoS 值；

(3) UIPCC[14]：UIPCC 是一种混合了 UPCC 和 IPCC 的协同过滤方法. 结合 UPCC 和 IPCC 的 QoS 预测值，并加入一个参数来平衡两者的作用；

(4) UCAPK[21]：基于用户的可信度预测方法，采用 K-means 对用户进行聚类，识别不信任用户，然后根据信任用户预测未评价的 Web 服务 QoS；

(5) TrustWalker[24]：基于信任度和协同过滤的随机游走推荐方法；

(6) WSWalker-p[25]：基于地理位置和协同过滤的随机游走推荐方法；

(7) WSWalker-u[25]：结合用户个性化特征的基于地理位置和协同过滤的随机游走推荐方法；

(8) WSWalker-i[25]：结合服务个性化特征的基于地理位置和协同过滤的随机游走推荐方法；

(9) CRWA-p：使用式(11)计算随机游走的结果；

(10) CRWA-u：使用式(12)计算随机游走的结果；

(11) CRWA-i：使用式(13)计算随机游走结果；

(12) CRWA-ui：使用式(14)计算随机游走结果.

为了使实验部分更加完整，我们列出了本文方法及对比方法的重要参数和控制变量，如表 2 所示.

表 2　算法重要参数

方法	参数
UPCC	Top-k=10；相似度阈值 θ=0
IPCC	Top-k=10；相似度阈值 θ=0
UIPCC	Top-k=10；λ=0.2；相似度阈值 θ=0
UCAPK	Top-k=10；类的数量 k=7
TrustWalker	Top-k=20；终止条件 ε=0.0001
WSWalker	Top-k=20；终止条件 ε=0.0001
CRWA	Top-k=10；终止条件 ε=0.0001；λ=0.4

表 3 是采用 1%至 5%密度矩阵的不同方法的预测精度. 我们观察本文提出的方法——CRWA-p、CRWA-u、CRWA-i 和 CRWA-ui，与其它较经典的预测方法相比，无论在何种矩阵密度下均具有更小的 MAE 和 $RMSE$，特别是 CRWA-ui 与其它方法

相比具有最小的 *MAE* 和 *RMSE*，这表明本文方法具有最高的预测精度. 同时也可以发现协同过滤方法预测性能最低，特别在数据十分稀疏的情况下，预测精度远低于其他方法，这说明协同过滤方法不适合用于数据稀疏的推荐环境. 随着矩阵密度从1%增加到5%，所有方法的预测精度都显著提高. 然而基于协同过滤的方法 UPCC，IPCC 和 UIPCC 对数据稀疏性非常敏感. 基于随机游走的预测方法 TrustWalker，WSWalker 和 CRWA 对数据稀疏性则相对稳定，且本文方法最稳定. 通常用户项目矩阵的密度在真实情况下非常低，因此我们认为与其他方法相比，本文提出的方法更适用于真实服务推荐系统. 同时观察本文方法的四种计算方式，发现 CRWA-i 相比于 CRWA-u 具有更高的预测精度. 这说明，由于不同的服务在 QoS 值上有很大的不同，考虑到目标服务的平均 QoS 值而不是原用户评价过服务的平均 QoS 值，所以在 QoS 预测上精度更高. 而 CRWA-ui 结合了用户和服务的个性特征，在预测精度上比其它三种计算方式均有更高的预测准确性.

表 3 预测性能对比

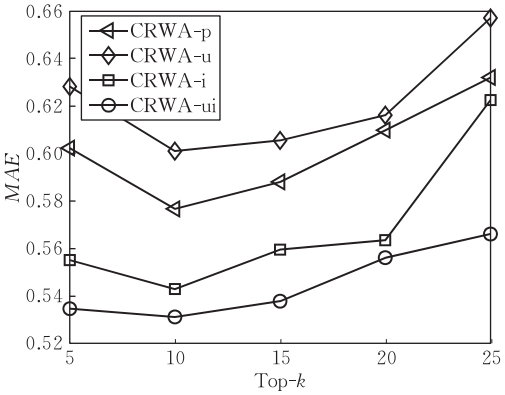
属性	方法	矩阵密度=1%		矩阵密度=2%		矩阵密度=3%		矩阵密度=4%		矩阵密度=5%	
		MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
RT	UPCC	1.2874	2.4753	0.9688	1.9896	0.8215	1.7220	0.6949	1.5030	0.6536	1.4246
	IPCC	1.0487	2.1024	0.8649	1.7340	0.7391	1.5434	0.7382	1.5366	0.7218	1.4802
	UIPCC	0.9514	1.9827	0.8584	1.6205	0.7193	1.5428	0.6631	1.4628	0.6487	1.4143
	UCAPK	0.8214	1.8796	0.7137	1.6157	0.6921	1.6427	0.6662	1.4796	0.6365	1.4162
	TrustWalker	0.6951	1.7395	0.6452	1.6361	0.6293	1.6752	0.6056	1.5918	0.5934	1.5284
	WSWalker-p	0.6205	1.6959	0.6151	1.6344	0.6079	1.5920	0.5987	1.6629	0.5878	1.5900
	WSWalker-u	0.6593	1.7170	0.6477	1.7221	0.6262	1.6837	0.6031	1.6432	0.5968	1.6475
	WSWalker-i	0.5927	1.5955	0.5869	1.5760	0.5664	1.6316	0.5625	1.5381	0.5496	1.5089
	CRWA-p	0.5767	1.5318	0.5551	1.4992	0.5446	1.5032	0.5036	1.4657	0.4792	1.4540
	CRWA-u	0.6011	1.4967	0.5857	1.4636	0.5710	1.4631	0.5331	1.4307	0.5182	1.4238
	CRWA-i	0.5429	1.4123	0.5225	1.4032	0.5079	1.3903	0.4776	1.3655	0.4770	1.3427
	CRWA-ui	**0.5311**	**1.3855**	**0.5054**	**1.3732**	**0.4881**	**1.3933**	**0.4735**	**1.3350**	**0.4687**	**1.3217**
TP	UPCC	77.4218	163.2898	44.3510	99.5287	32.3839	72.9393	28.4814	64.3338	27.9779	63.6001
	IPCC	38.9935	91.4431	34.8459	75.6790	32.0504	69.2848	31.1556	66.9538	30.3546	65.8589
	UIPCC	37.4635	79.0757	33.8406	73.4183	31.1110	67.5778	30.1980	65.1973	29.2668	65.3525
	UCAPK	36.9636	78.7845	33.4148	72.7845	29.2750	69.6025	25.4663	66.6338	23.7610	63.8590
	TrustWalker	36.8015	78.5572	31.0205	73.2834	28.3417	71.2060	24.7353	67.6069	22.0437	62.6271
	WSWalker-p	34.7635	78.7206	29.8967	72.8326	26.2457	70.9803	23.9786	67.6102	23.6947	65.5176
	WSWalker-u	36.5674	78.3351	31.4556	71.4896	27.0100	68.3350	25.1618	66.2850	22.9320	63.3669
	WSWalker-i	31.7664	76.3295	25.9922	69.3772	23.9089	68.0014	22.0464	65.6054	21.3632	61.9830
	CRWA-p	27.4013	70.9415	23.5378	65.9793	22.0113	63.6659	20.0402	60.9150	19.3666	59.0676
	CRWA-u	27.8434	68.4656	23.4571	63.3223	22.7382	61.0353	20.2859	60.2494	20.1514	58.4364
	CRWA-i	24.2750	63.8690	22.0614	60.1815	21.4989	59.2646	19.8416	58.0650	19.1812	57.5046
	CRWA-ui	**23.5843**	**60.8268**	**21.2387**	**58.6644**	**20.4527**	**57.6995**	**19.3969**	**57.1661**	**18.8009**	**56.5420**

4.4 Top-*k* 对预测精度的影响

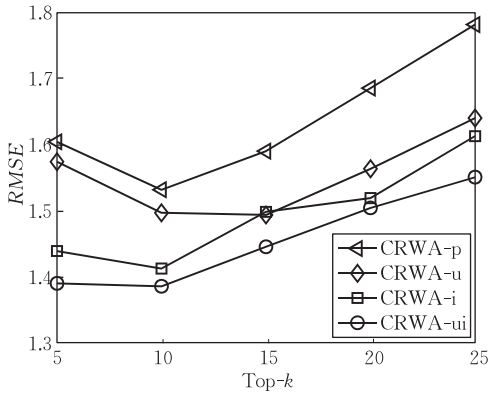
本节对 *RT* 和 *TP* 属性均采用1%的密度矩阵来评估 Top-*k* 对本文方法预测精度的影响. 参数 Top-*k* 控制着信任用户集合大小. *k* 值越小，用户的信任用户集合就会越小，反之越大. 设置参数 $\lambda=0.4$，*k* 值分别为 5，10，15，20，25. 如图 4 所示，本文的四种方法在开始时，随着 Top-*k* 值的增加，*MAE* 和 *RMSE* 的值随之降低. 但是当 Top-*k* 超过某个阈值时，*MAE* 和 *RMSE* 的值随之上升. 这说明 Top-*k* 取适当的值有利于提高预测精度. 因为当 Top-*k* 取值过小时，用户的信任用户集合过小，该用户的某些信任用户会被忽略，没有充分利用信任用户的信息，从而降低了预测精度. 而当 Top-*k* 取值过大时，用户的信任用户集合就会过大，可能包含某些不信任用户，这些用户的 QoS 值其实是噪声数据，降低了预测精度. 图 4 显示当 $k=10$ 时，所有方法的 *MAE* 和 *RMSE* 均达到最低值，预测精度最高，且 CRWA-ui 方法具有最高的预测精度.

4.5 矩阵密度对预测精度的影响

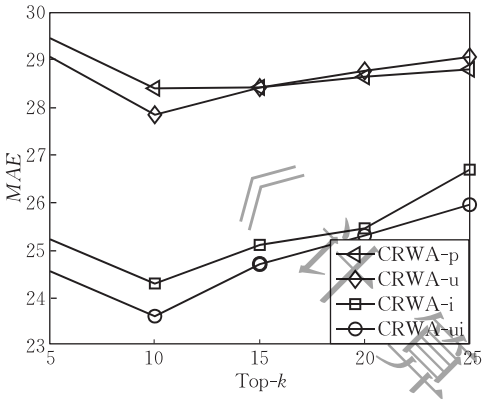
为进一步评估矩阵密度对 QoS 预测性能的影响，我们设置 Top-$k=20$，$\lambda=0.4$，并将 *RT* 和 *TP* 属性的密度从1%变化到5%，步长设为1%. 从图 5 可以看出，随着矩阵密度的增加 *MAE* 和 *RMSE* 值一直下降，(b)图中的 *RMSE* 在密度为3%时略有增大，但趋势同样是在逐渐减少，表明预测精度上升. 结果表明，较多的 QoS 训练数据有利于提高 QoS 预测精度. 因此在现实中可以收集更多的 QoS 数据来提高预测精度.



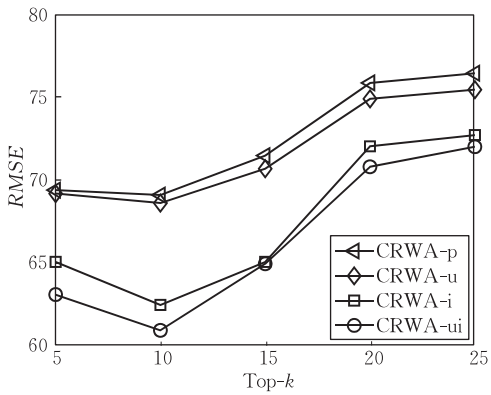
(a) *RT*属性



(b) *RT*属性

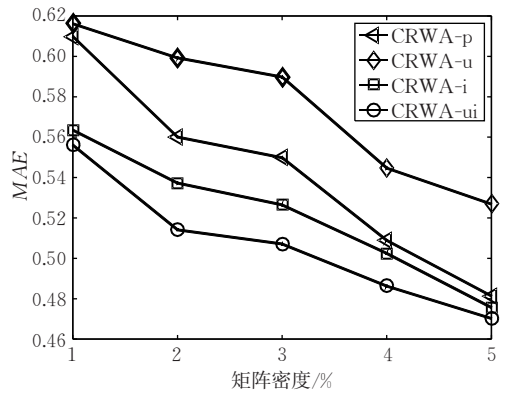


(c) *TP*属性

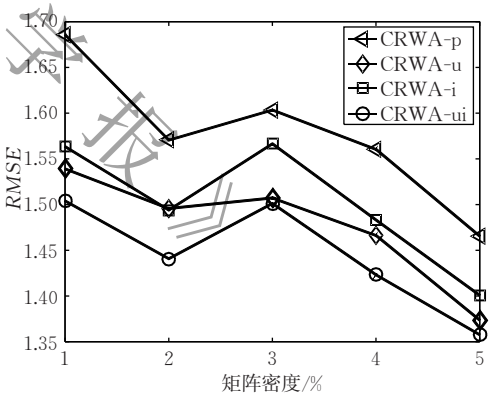


(d) *TP*属性

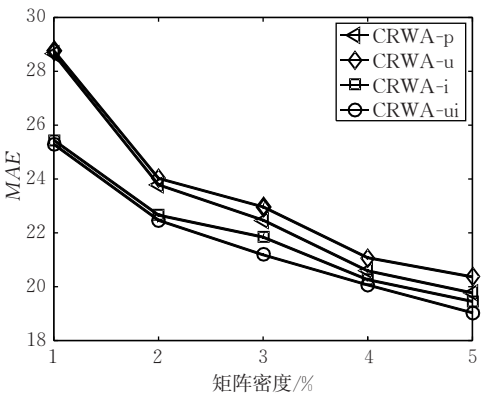
图 4　Top-k 对预测精度的影响



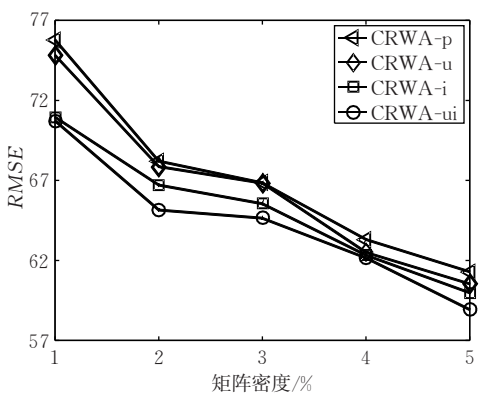
(a) *RT*属性



(b) *RT*属性



(c) *TP*属性



(d) *TP*属性

图 5　矩阵密度对预测精度的影响

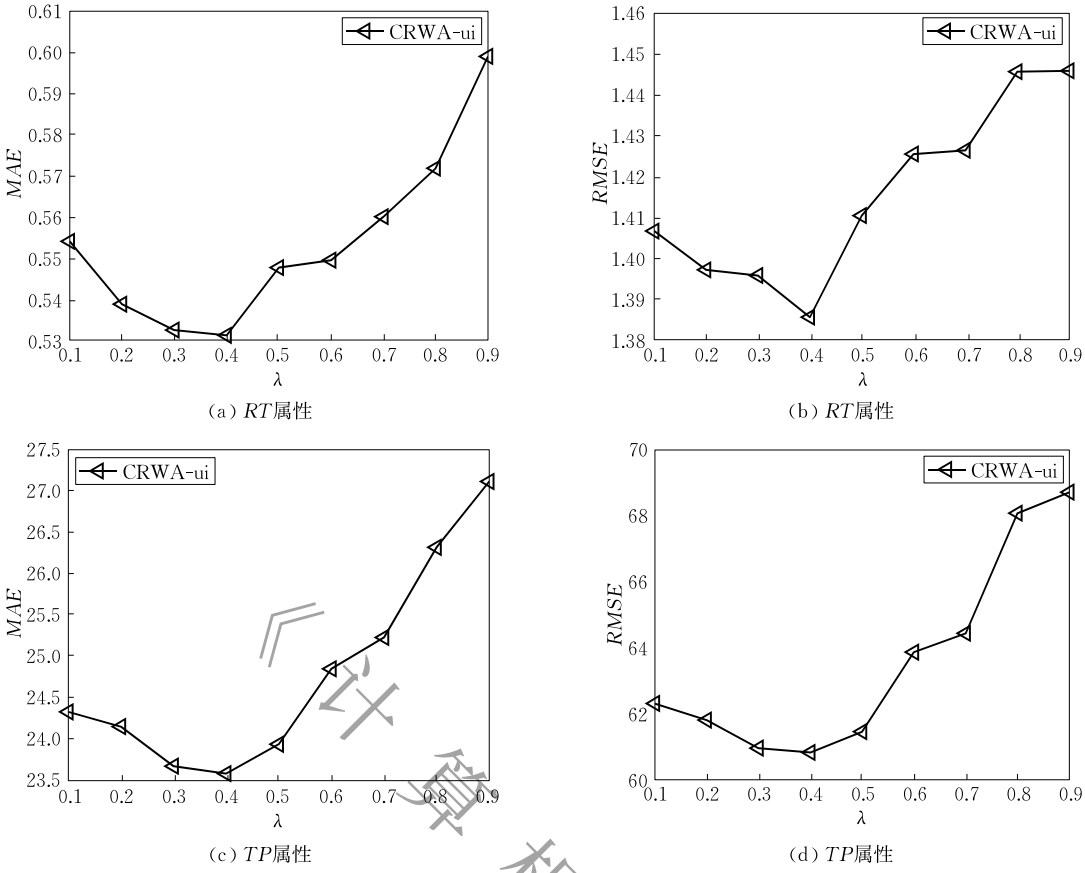


图 6 λ对预测精度的影响

4.6 λ对预测精度的影响

在为用户预测服务的QoS值时，参数λ决定依赖用户和服务个性化特征的程度. 4.3节的实验结果表明，通过结合用户和服务的个性化信息，CRWA-ui实现了比CRWA-u和CRWA-i更好的预测精度. 在本节中，我们评估参数λ值对CRWA-ui的影响. 当$\lambda=0$时，CRWA-ui等同于CRWA-i，因为只考虑服务的个性化特征. 同样，当$\lambda=1.0$时，CRWA-ui相当于CRWA-u. 因此，本文设置矩阵密度为1%，Top-$k=10$，λ值从0.1到0.9来评估其对预测精度的影响.

如图6所示，在*RT*和*TP*属性上，λ值对CRWA-ui的预测精度均有显著的影响. 在预测精度达到最佳之前，随着λ值的增加*MAE*和*RMSE*的值随之减小，说明预测精度提高. 然而，当λ值超过某个阈值时预测精度反而降低. 这表明合适的λ值提供最佳的预测精度，因为它能够对用户和服务的个性化特征信息进行适当的整合. 图6中显示，当$\lambda=0.4$时，*MAE*和*RMSE*均达到最低值，预测精度最高.

4.7 实验结论

本文方法在真实的Web服务QoS数据集上进行了一系列实验. 通过4.3节可以看出无论在响应时间还是吞吐量方面，本文方法与以往方法相比都有最小的*MAE*和*RMSE*，这表明本文方法具有最高的预测精度. 从该节中我们还能发现，在数据极度稀疏的情况下，本文方法最稳定. 从4.5节中可以观察虽然本文方法在数据极度稀疏的环境下比较稳定，但是随着数据密度的增加预测精度还是有比较明显的提高. 因此，我们可以理解，较大的矩阵密度有利于Web服务的QoS预测. 根据4.4节和4.6节发现适当的Top-k值和λ值有利于提高QoS预测精度. 综上所述，本文提出的CRWA方法与现有方法相比具有最佳的预测精度，适合于数据极度稀疏的场景.

5 总 结

本文提出一种基于覆盖随机游走算法的服务质量预测方法. 该方法首先利用覆盖算法为每个用户寻找其信任用户；其次连接每个用户与其信任用户构建用户信任网；最后在用户信任网进行多次随机游走，汇总随机游走返回的所有QoS值来预测Web服务QoS值. 本文在真实的Web服务数据集上进行一系列实验，实验结果表明本文方法的QoS预测

精度与以往方法相比有显著提高，不仅能解决用户信任问题，同时也解决了数据稀疏性高的服务质量预测问题. 在接下来的研究工作中，我们将考虑 Web 服务的动态特征，研究基于 Storm 大数据平台的服务质量实时预测方法，并开发相应的原型系统.

致　谢　非常感谢审稿专家为提高本文质量提出的建设性意见，同时要特别感谢编辑部老师为本文所做的全部工作！

参　考　文　献

[1] Zhang L J, Cai H, Zhang J. Services Computing. Beijing: Tsinghua University Press, 2007

[2] Zhang Y, Zheng Z, Lyu M R. WSExpress: A QoS-aware search engine for Web services//Proceedings of the 8th International Conference on Web Services. Miami, USA, 2010: 91-98

[3] Yau S S, Yin Y. QoS-based service ranking and selection for service-based systems//Proceedings of the 8th International Conference on Services Computing. Washington, USA, 2011: 56-63

[4] Kang G, Liu J, Tang M, et al. Web service selection for resolving conflicting service requests//Proceedings of the 9th International Conference on Web Services. Washington, USA, 2011: 387-394

[5] Shao L, Zhang J, Wei Y, et al. Personalized QoS prediction for Web services via collaborative filtering//Proceedings of the 2007 IEEE International Conference on Web Services. Salt Lake City, USA, 2007: 439-446

[6] Chen X, Liu X, Huang Z, et al. RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized Web service recommendation//Proceedings of the 8th International Conference on Web Services. Miami, USA, 2010: 9-16

[7] Zheng Z, Ma H, Lyu M R, et al. Collaborative Web service QoS prediction via neighborhood integrated matrix factorization. IEEE Transactions on Services Computing, 2013, 6(3): 289-299

[8] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003, 7(1): 76-80

[9] Xu Lei, Yang Cheng, Jiang Chun-Xiao, Ren Yong. Game analysis of user participation in collaborative filtering systems. Chinese Journal of Computers, 2016, 39(6): 1176-1189(in Chinese)
(徐蕾，杨成，姜春晓，任勇. 协同过滤推荐系统中的用户博弈. 计算机学报，2016，39(6)：1176-1189)

[10] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering//Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, USA, 1998: 43-52

[11] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms//Proceedings of the 10th International Conference on World Wide Web. Hong Kong, China, 2001: 285-295

[12] McLaughlin M R, Herlocker J L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience//Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffield, UK, 2004: 329-336

[13] Liu R R, Jia C X, Zhou T, et al. Personal recommendation via modified collaborative filtering. Physica A: Statistical Mechanics and its Applications, 2009, 388(4): 462-468

[14] Zheng Z, Ma H, Lyu M R, et al. WSRec: A collaborative filtering based Web service recommender system//Proceedings of the 7th International Conference on Web Services. Los Angeles, USA, 2009: 437-444

[15] Jiang Y, Liu J, Tang M, et al. An effective Web service recommendation method based on personalized collaborative filtering//Proceedings of the 9th International Conference on Web Services. Washington, USA, 2011: 211-218

[16] Wu J, Chen L, Feng Y, et al. Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(2): 428-439

[17] Ma Y, Wang S, Hung P C K, et al. A highly accurate prediction algorithm for unknown Web service QoS values. IEEE Transactions on Services Computing, 2016, 9(4): 511-523

[18] Qiu W, Zheng Z, Wang X, et al. Reputation-aware QoS value prediction of Web services//Proceedings of the 2013 IEEE International Conference on Services Computing. Santa Clara, USA, 2013: 41-48

[19] Zhang M, Liu X, Zhang R, et al. A Web service recommendation approach based on QoS prediction using fuzzy clustering//Proceedings of the 9th International Conference on Services Computing. Honolulu, USA, 2012: 138-145

[20] Yu C, Huang L. CluCF: A clustering CF algorithm to address data sparsity problem. Service Oriented Computing and Applications, 2017, 11(1): 33-45

[21] Wu C, Qiu W, Zheng Z, et al. QoS prediction of Web services based on two-phase k-means clustering//Proceedings of the 22nd International Conference on Web Services. New York, USA, 2015: 161-168

[22] Zhang Z, Zeng D D, Abbasi A, et al. A random walk model for item recommendation in social tagging systems. ACM Transactions on Management Information Systems, 2013, 4(2): 8

[23] Zhou Y, Liu L, Perng C S, et al. Ranking services by service network structure and service attributes//Proceedings of the 20th International Conference on Web Services. Santa Clara, USA, 2013: 26-33

[24] Jamali M, Ester M. TrustWalker: A random walk model for combining trust-based and item-based recommendation//Proceedings of the 15th ACM SIGKDD International Conference

on Knowledge Discovery and Data Mining. Paris, France, 2009: 397-406

[25] Tang M, Dai X, Cao B, et al. WSWalker: A random walk method for QoS-Aware Web service recommendation// Proceedings of the 22nd International Conference on Web Services. New York, USA, 2015: 591-598

[26] Zhang L, Zhang B. A geometrical representation of McCulloch-Pitts neural model and its applications. IEEE Transactions on Neural Networks, 1999, 10(4): 925-929

[27] Zheng Z, Zhang Y, Lyu M R. Investigating QoS of real-world Web services. IEEE Transactions on Service Computing, 2014, 7(1): 32-39

ZHANG Yi-Wen, born in 1976, Ph. D., associate professor. His research interests include service computing, recommender system and swarm intelligence.

WANG Kai-Bin, born in 1993, M. S. candidate. His research interest is service computing.

YAN Yuan-Ting, born in 1986, Ph. D., lecturer. His research interests include machine learning and granular computing.

CHEN Jie, born in 1982, Ph. D., associate professor. Her research interests include machine learning and granular computing.

HE Qiang, born in 1982, Ph. D., senior lecturer. His research interests include service computing and software engineering.

LI Wei, born in 1969, Ph. D., professor. Her research interests include software engineering and data mining.

Background

Service-oriented architecture (SOA) has become a major framework for building complex distributed software systems by discovering and composing loosely coupled Web services provided by different organizations. The development and popularity of e-Business, eCommerce, especially the pay-as-you-go business model promoted by cloud computing have fueled the rapid growth of Web services. The quality of a service-oriented system (SOS) relies on its component services. To ensure the quality of an SOS, we must be able to predict the quality of the Web services used to compose the SOS, e.g., their response time, throughput, etc. With the ability to predict the quality of Web services, Web service recommendation systems can be built and employed to recommend Web services with appropriate quality values that fulfill system engineers' quality requirements. Thus, quality prediction for Web services has been an extremely active research field in recent years.

In recent years, the collaborative filtering (CF) techniques, such as item-based and user-based methods, have been widely employed in the recommender systems for Web services. However, the traditional CF technique cannot be directly employed in the big data scenario for Web service recommendation due to two major issues: (1) Data extremely sparse; (2) User trust relationship. To address these issues, we propose a service quality prediction approach based on covering random walk algorithm. This approach firstly employs an improved covering-based clustering algorithm to select the Top-k trusted users for each user based on the user service historical quality experiences, the clustering approach does not require the number of clusters; secondly connecting all users and their trusted users to build user trust network to address the issue of user trust relationship; finally, the value of QoS is predicted by random walk on the user trust network. Experiments conducted on real-world Web service datasets show that our approach outperforms the existing approaches in QoS prediction accuracy, and the extremely data sparse and user trust relationship issues can be addressed well by the proposed approach.

This work is supported by the National Key Technology R&D Program of China (2015BAK24B01), the National Natural Science Foundation of China (61602003), and the Natural Science Foundation of Anhui Province (1808085MF197).