

基于偏好推荐的可信服务选择^{*}

朱 锐¹⁺, 王怀民^{1,2}, 冯大为¹

¹(国防科学技术大学 计算机学院, 湖南 长沙 410073)

²(国防科学技术大学 并行与分布处理国防科技重点实验室, 湖南 长沙 410073)

Trustworthy Services Selection Based on Preference Recommendation

ZHU Rui¹⁺, WANG Huai-Min^{1,2}, FENG Da-Wei¹

¹(School of Computer, National University of Defense Technology, Changsha 410073, China)

²(Key Laboratory of Science and Technology for National Defense of Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: mruzce@gmail.com

Zhu R, Wang HM, Feng DW. Trustworthy services selection based on preference recommendation. *Journal of Software*, 2011, 22(5):852-864. <http://www.jos.org.cn/1000-9825/3804.htm>

Abstract: This paper presents a Trustworthy Services Selection Based on Preference Recommendation (TSSPR) method that assists users in selecting the right Web services, according to their own preferences. First, a group of recommenders that have similar preferences are found, and then the similarity rating is computed by using the Pearson correlation method. Second, filtering services based on the user's recommending level, relative domain degrees, and similarity ratings can improve the quality of recommendations. Experimental results show that given an appropriate setting, this method can effectively solve the weaknesses of recommendation systems, such as sparseness, cold starts, and inaccurate recommendations.

Key words: Web service; service selection; trust; service recommending; collaborative filtering

摘 要: 针对现有服务选择中服务推荐技术的不足,提出一种基于偏好推荐的服务选择(trustworthy services selection based on preference recommendation,简称 TSSPR)方法.首先搜索一组偏好相似的推荐用户,并通过皮尔逊相关系数计算用户的评价相似度,然后基于用户的推荐等级、领域相关度和评价相似度等对用户的推荐信息进行过滤,从而使推荐信息更为可信.模拟实验结果表明,通过正确的参数设置,该方法能够有效地解决推荐算法中冷启动、推荐信息不准确等问题.

关键词: Web 服务;服务选择;可信;服务推荐;协同过滤

中图法分类号: TP311 文献标识码: A

以 Web 服务为代表的软件服务及软件服务协同已经成为一种新型的 Web 应用形态.基于 Web 服务的服组合作为实现灵活、快速集成的重要方法,已经成为新的研究热点.随着部署在 Internet 上的 Web 服务不断丰富,这些可被公共访问和集成的服务构成了一个巨大的标准组件库.但是,面对大量功能相同或相近的 Web 服务,用

* 基金项目: 国家重点基础研究发展计划(973)(2005CB321800, 2005CB321801); 国家杰出青年基金(60625203)

收稿时间: 2009-06-15; 修改时间: 2009-09-11; 定稿时间: 2009-12-07

户在服务选择上无所适从,且用户对服务的质量要求很难被识别,一味地只满足响应时间或者降低开销并不能满足每一个用户的偏好需求.所以,根据用户的偏好定制个性化的服务选择指标很重要.推荐系统(recommender system)是个性化信息服务的主要技术之一.作为一种信息过滤的重要手段,推荐系统被广泛应用于电子商务领域,可以把用户潜在的、模糊的需求转化为现实的、明确的需求.此外,推荐还研究用户模型和用户的喜好,以满足个性化的需求^[1].特别是随着越来越多的用户访问 Web 服务所形成的经验积累,可以预见,服务推荐将成为 Web 服务选择的自然需求.

所谓可信,就是用户对于服务本身的信任,是其在参与或者使用服务过程中所形成的一种主观感受^[2].但是,人们的主观感受很难被客观地描述和获取,且对同一服务的不同使用者,由于自身偏好的不同而有着不同的使用体验,导致用户评价服务可信程度的准则也相应地不同.可信的概念为满足用户主观上的预期,而用户评价信息正是用户对于服务行为是否符合预期的一种打分,因此,评价价值的高低可以较好地反映用户使用服务的主观感受.同时,在收集评价信息的过程中要考察评价用户与自身评价准则的相似程度,这样便于用户通过赋予不同的权重表达对于评价信息不同的接受程度.

服务推荐不同于传统的推荐方法,用户具有明确的应用需求,只是在面对大量同种功能服务时很难判断服务符合自身偏好需求的程度.近年来,大量的研究工作已经或者正在围绕利用历史经验信息判断服务质量优劣的问题展开,但这些研究工作并未过多地考虑人的因素:多数研究依赖于不同用户对同一服务的反馈信息来评估服务可信程度,同等地看待每个用户的评价,或仅仅考虑了用户评分的相似度,并未考虑用户的领域相关性与推荐信息的可信程度.因此,在尽量避免用户的个人倾向或偏见造成的主观随意性的同时,不仅要考虑用户在评价方面的相似程度,还需要有一种方法来确定用户对领域知识的熟悉程度以及推荐信息的可信程度.

本文针对上述问题提出一种基于偏好推荐的可信服务选择(trustworthy services selection based on preference recommendation,简称 TSSPR)方法.该方法属于协同过滤(collaborative filtering)的方法范畴^[3].通过皮尔逊相关系数在一组相关用户中寻找与自己评价指标最相似、推荐信息最为可信的一组用户,再根据该组用户对服务的评价值加权计算出服务的可信度,以此作为服务选择的依据.模拟实验结果表明,该方法能够有效地解决推荐算法中冷启动、推荐信息不够准确等问题,有效地提高了推荐质量.

本文第 1 节介绍相关工作,同时进一步阐明本文与相关工作的差异和深入研究的意义.第 2 节描述皮尔逊相似度评价体系,并定义、计算用户评价相似度.第 3 节对 TSSPR 方法进行描述、分析.第 4 节是仿真实验与结果分析.最后一节对全文进行总结与展望.

1 相关工作

网络环境的开放性使可用的 Web 服务构成一个不断成长、动态变化的服务空间,这使得服务的选择成为既令人兴奋又令人头疼的问题.服务质量信息通常是由服务提供者给出,往往存在主观的和非公正的因素.同时,专业知识的匮乏性使得用户在可信服务的选取上无从下手.因此,当前大量研究工作都致力于帮助用户识别并选择最为可信的服务,通常有 3 种方式:① 通过可信的第三方;② 通过用户使用服务形成的历史经验信息;③ 通过从其他服务或服务使用者处间接获得的信息.

可信第三方通常基于监控代理和认证中心来收集注册表中所有可用服务的各类信息^[4,5],其中包括用户的反馈信息等,并基于此向用户提供服务 QoS 属性信息.文献[6]通过部署大量的传感器监视服务的运行状态并周期性地向中心节点报告,但是这类方法更适合于较小规模的封闭式系统,在开放的互联网环境下,部署传感器将产生极大的开销;同时,服务的动态加入和退出等特性也必将导致额外的开销用于安装和回收传感器.此外,集中式的信息管理方式可能使中心节点成为性能瓶颈和攻击对象.

Maximilien^[7]认为,QoS 的不确定性、对服务运行环境认知的缺乏以及不可信任的服务提供商等因素,是服务选择所面临的最大难题.他提出了一种基于 agent 的体系结构,用户仅能与 agent 交互,agent 作为服务和应用的代表负责收集并共享服务信息.该方法的特点是,服务 QoS 值的描述并不由服务提供商或者用户决定,而是通过多个 agent 之间的协商来确定.服务的 QoS 属性被分为客观和主观两种,可信度(trustworthiness)作为一种主观

的服务 QoS 属性将随着时间和交互体验的增加而发生变化.Pinar^[7-9]带领的研究小组致力于在开放环境中寻找可信任的服务,并展开了大量的研究工作.他们认为,针对不同的任务,服务有着不同的可信程度,所以通过注册中心或者认证授权的方式确定一个服务的可信值通常并不适用.Pinar 提出了一个简单的服务提供者 and 使用者之间的交互图:大部分服务使用者缺乏专业知识,很难通过自身判断服务的可信度;但多个使用者之间却有着很强的沟通性,通过共享历史交互经验判断服务质量的优劣具有较好的可行性.因此,他认为在开放系统中基于推荐信息定位可信服务是一种最基本、最有效的手段.其推荐方法的特色之处在于基于用户的交互经验信息,而不仅仅是基于用户的评分.但是在选择推荐信息时,仅仅考虑了推荐用户的专家等级和社交性(sociability)的高低,对用户的偏好缺乏足够的支持;同时,对推荐信息的可信程度也未进行评估.

推荐系统根据推荐信息形成方式的不同可以分为 3 种,即基于规则的系统、基于内容过滤的系统和协同过滤系统.基于规则的推荐系统和基于内容过滤的推荐系统都只能为用户推荐过去喜欢的项目和相似的项目,并不能推荐用户潜在感兴趣的项目.协同过滤最先由 Goldberg^[10]在 1992 年提出,如今,协同过滤技术已经成为电子商务系统中最为重要的技术之一.协同过滤技术可以分为基于用户的协同过滤(user-based collaborative filtering)、基于物品的协同过滤(item-based collaborative filtering)和基于模型的协同过滤(model-based collaborative filtering)这 3 类.基于用户的协同过滤通过寻找邻居向用户推荐其可能感兴趣的物品;基于物品的协同过滤则通过用户曾经评过分的物品寻找相似的物品,并向用户推荐;为了避免协同过滤技术中稀疏矩阵和处理大量数据时无法满足时效性等问题,提出了基于模型的协同过滤技术.该方法先用历史数据得到一个模型,模型的建立可以使用各种机器学习的方法,如贝叶斯网络技术、聚类技术、人工神经网络、概率模型等,再用此模型进行预测.Karta^[11]提出了结合基于规则的和基于内容过滤的多维度推荐系统,从而避免和弥补各自推荐技术的弱点.为解决用户评分数据极端稀疏的问题,文献[12]通过奇异值分解(singular value decomposition)减少项目空间维数.但降维会导致信息损失,特别是在项目空间维数很高的情况下,推荐效果难以保证.张光卫等人^[13]利用云模型在定性知识表示以及定性、定量知识转换时的桥梁作用,提出了一种基于云模型的用户相似度比较方法,在一定程度上克服了用户评分数据极端稀疏的负面影响.Paolo 等人^[14]提出了基于信任的协同过滤技术,即根据用户之间的信任关系推荐服务:若用户 A 信任用户 B,那么用户 B 向其推荐的服务一定满足 A 的需求.但文献[4]中并未解决如何在成千上万的用户中寻找可信用户的问题,并且我们认为,可信的用户不一定可以推荐出满足用户偏好需求的服务.基于协同过滤的推荐系统与基于信誉的推荐系统相比更为注重用户间体验的类似^[15],而后者关注用户之间的信任关系.因此,协同过滤技术在满足用户偏好方面更加适合.

在用户主观感受的量化方面,Robert^[16]提出了与量化有关的 4 个相互关联的重要因素:品牌(branding)、可用性(usability)、功能性(functionality)和内容(content).其中,品牌主要映射到服务提供商的品牌效应,例如,规模较大的服务提供商提供的服务相对于规模较小提供商而言,更容易被用户所信任;可用性主要指易用性,即用户是否能够通过一种简单、友好的交互方式使用服务;功能性主要指服务能否满足用户所需的功能,以及使用服务过程中用户的隐私能否被保护、执行的进展情况能否被告知等;内容则指调用服务返回的内容是否结构清晰、内容准确,从而有利于用户的了解.此外,Zhou^[17]提出了一种基于可用性测试的旨在对产品使用过程中的用户体验质量进行综合评价的模型,针对可用性(usability)概念的模糊性及其综合评估中的问题,应用层次分析法(analytic hierarchy process,简称 AHP)确定了指标体系中相关因素的权重,并根据模糊隶属度函数对用户体验质量评价标准进行了模糊化处理.最后,采用模糊评价方法对用户体验质量进行评价.

2 用户评价相似度

2.1 搜集推荐用户

在现实社会中,大多数具有相同兴趣和知识背景的用户往往具有相同或相似的行为特征和评价准则.这意味着,相似用户在对某事务的认知和评价上应该处于一个相同或者相近的水平.因此在推荐系统中,当考察其他用户对服务的评价信息时,不能用平等对待的观点.理想的做法是,在一大群评价用户中找出与我们自身相似的一小群人,并确定这一小群人在品位方面与自身的相似程度.这样,就能找到对自身而言最为可信的评价信息.

定义 1(用户服务评价). U_i 是一个用户,其服务评价是一个二元组:

$$EV_i=(WS_i, \delta_i).$$

其中, WS_i 和 δ_i 分别定义如下:

- WS_i 是用户 U_i 所访问服务的有限集合, $WS_i=\{ws_{i1}, \dots, ws_{in}\}$. 其中, $|WS_i|=n$ 表示集合的大小为 n .
- $\delta_i: WS_i \rightarrow [1, 10]$ 是服务评价函数. 对于 $\forall ws_{ik} \in WS_i, \delta_i(ws_{ik})=d_{ik}, d_{ik} \in [1, 10], k \in [1, n]$, 表示用户 U_i 对服务 ws_{ik} 的评价值为 d_{ik} .

定义 2(偏好相似用户). 设用户 U_i 和 U_j 的服务评价分别是 $EV_i=(WS_i, \delta_i)$ 和 $EV_j=(WS_j, \delta_j)$, 如果 $WS_i \cap WS_j \neq \emptyset$ 且 $|WS_i \cap WS_j| \geq M$, 则称用户 U_i 与 U_j 为偏好相似用户. 其中, M 为条件因子.

合理调整 M 的大小, 可以控制与用户具有相似偏好用户的数量.

定义 3(推荐用户). 设用户 U_i 和 U_j 为偏好相似用户, 且其服务评价分别是 $EV_i=(WS_i, \delta_i)$ 和 $EV_j=(WS_j, \delta_j)$, 如果 $\exists ws_{jk} \in WS_j$ 且 $ws_{jk} \notin WS_i$, 同时, 用户 U_i 希望获取服务 ws_{jk} 的评价信息, 则称用户 U_j 为 U_i 关于服务 ws_{jk} 的推荐用户.

这里, 我们需要考虑推荐用户的领域相关度和推荐等级对推荐权重的影响. 推荐用户的领域相关度是指用户对某类服务的关注程度和了解程度. 如果用户是某个服务的领域专家或者对相关知识领域有深入的了解, 我们将相应增加其推荐的权重. 由于验证用户领域相关度的方法较复杂, 因此我们简单地认为, 若用户对某类服务的评价次数较多, 则用户为该类服务的领域相关用户. 因此, 我们需要对服务进行分类, 以便识别用户是否为领域相关用户以及确定相关度. Web 服务分类已经有了一些成熟的方法, 本文采用文献[18]中的算法, 通过分析 WSDL 文件和 UDDI 提供的信息对服务的类别进行划分, 具有较好的可行性、通用性和易操作性. 推荐用户的推荐等级是指推荐用户推荐信息被采纳的概率, 其主要依赖于用户对推荐信息的反馈.

由定义 2、定义 3 可知, 如果两个用户存在推荐关系, 那么他们必然是偏好相似用户, 反之则不成立. 在下一节中, 我们将计算用户间的评价相似度, 计算评价相似度仅仅考虑两个用户共同评价过的服务项集. 因此, 无论用户间是否存在推荐关系, 都不会影响评价相似度的计算.

2.2 计算用户评价相似度

在获得一组可以提供推荐信息的用户后, 就要确定哪些用户与自己的评价指标(rating level)最为相似. 我们认为, 一个用户评价指标与自己越相似, 在评价服务的可信度时该用户的评价值也会与自己越相似. 为此, 需要比较推荐用户与自己的评价相似度.

在计算评价相似度时, 可能会面临稀疏矩阵等问题, 导致推荐信息不准确, 较容易产生动荡, 甚至无力产生推荐. 传统的推荐系统在产生推荐信息时, 需要将用户与所有其他用户进行比较, 其速度是难以忍受的. 本文所提出的推荐算法中, 我们假设用户 U_i 希望获取一组功能相似服务 $\{ws_1, ws_2, \dots, ws_n\}$ 的推荐信息, 并从中选取最可能符合自身需求的服务进行访问. 因此, 对用户产生的推荐限定于某个特定的集合中, 而并非盲目的推荐. 这不但符合现实中互联网用户选取服务的实际特征, 而且能够较好地解决稀疏矩阵带来的问题.

本文利用皮尔逊相关系数(Pearson correlation coefficient)来指导计算用户评级指标的相似度. 皮尔逊相关系数是一种度量两个变量间相关程度的方法. 它是一个介于 1 和 -1 之间的值, 其中, 1 表示变量完全正相关, 0 表示无关, 而 -1 则表示完全负相关. 我们认为, 若皮尔逊相关系数小于 0, 则等同于无关. 下面给出皮尔逊相关系数的计算公式:

$$r_{uv} = \frac{\sum_{i \in I_{uv}} (R_{u,i} - \bar{R}_u) \times (R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{uv}} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_{uv}} (R_{v,i} - \bar{R}_v)^2}} \quad (1)$$

其中, u, v 表示任意两位偏好相似用户, 且 $u \neq v$; r_{uv} 表示 u 和 v 之间的评分相似性; I_{uv} 表示 u 和 v 的共同评分项集; $R_{u,i}$ 与 $R_{v,i}$ 为用户 u, v 对服务 i 的评价值, \bar{R}_u 和 \bar{R}_v 表示 u, v 在 I_{uv} 上的平均评分.

举例来说, 如果有 4 个偏好相似用户 U_A, U_B, U_C, U_D , 其对服务 ws_1, \dots, ws_4 的评价值见表 1. 根据公式(1)计算得到用户 U_A 和 U_B 的相似度约为 0.759, U_A 与 U_C 的相似度约为 0.405, U_A 与 U_C 的相似度为 0.135. 这意味着, 对用户 U_A 而言, U_B 的评价更符合服务 U_A 的偏好, 那么从 U_B 预测 U_A 评分的精度就越高. 更直观地讲, 如图 1 所示, 图 1(a)~

图 1(c)分别描述了用户 U_A 与 U_B 、 U_A 与 U_C 以及 U_A 与 U_D 的评分结果比较。 U_A 对 ws_1 的评价为 4,而 U_B 对 ws_1 的评价为 2,则在图 1(a)中坐标(4,2)被标识出来.图中直线为一条线性拟合线,其绘制的原则是尽可能地靠近图上所有的标识点.可以想象,如果两个用户对所有服务的评价价值相同,就会得到一条对角线;同时,所有的标识点都将与这条对角线相交,而此时这两个用户的相似度为 1.

Table 1 User scoring matrix
表 1 用户评分矩阵

User	WS			
	ws_1	ws_2	ws_3	ws_4
U_A	4	6	2	8
U_B	2	7	3	6
U_C	6	6	4	5
U_D	5	3	4	5

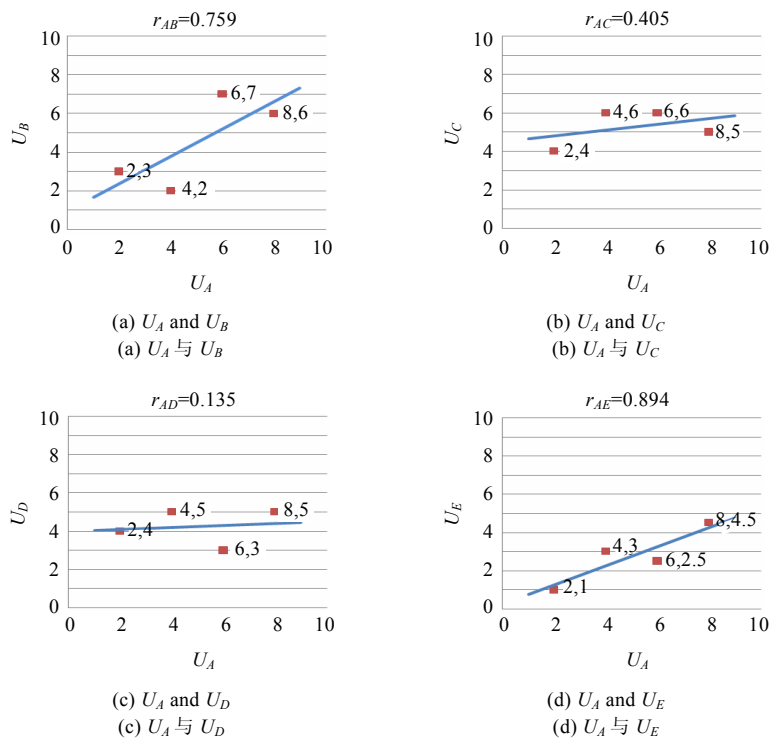


Fig.1 Comparing scoring
图 1 评分结果比较

除了皮尔逊相关系数以外,传统度量用户相似性的方法还有余弦(cosine)相似性和修正的余弦相似性(adjusted cosine)等方法.余弦相似性实现较简单,且计算速度快,但是未体现出用户评分的统计特征;修正的余弦相似性方法则更多地体现了用户的相关性而不是相似性.皮尔逊相关系数不仅考虑了项目的平均评分,还可以有效地识别如下的情况:若某用户总是倾向于给出比另一个用户更高的评价,而二者的分差又始终保持一致,那么我们依旧认为他们具有很好的相似性.也就是说,某用户可能比其他用户评分更为严格或者更为宽松,但是两个用户却有着非常相似的特性,如图 1(d)所示.

定义 4(用户评价相似度). 设两个偏好相似用户 U_i 和 U_j 的服务评价分别是 $EV_i=(WS_i,\delta_i)$ 和 $EV_j=(WS_j,\delta_j)$,其评价相似度定义为

$$S_{U_i \leftrightarrow U_j} = \begin{cases} r_{ij}, & r_{ij} \geq 0 \\ 0, & \text{else} \end{cases} \quad (2)$$

其中, r_{ij} 为两用户在共同服务访问项集 $WS_i \cap WS_j$ 评价上的皮尔逊相关系数。

针对互联网的动态和开放特性,在查考服务评价信息时需要考虑时间因素对推荐信息或结果的影响。因此在实验小节,我们会对过于陈旧的评价信息产生一定的衰减。

定义 5(用户推荐度). 若用户 U_j 为 U_i 关于服务 ws_{jk} 的推荐用户,则用户 U_j 针对 U_i 对服务 ws_{jk} 的推荐度为

$$R_{U_j \Rightarrow U_i}(ws_{jk}) = \delta_j(ws_{jk}) \times S_{U_i \leftrightarrow U_j} \times (1 + \lambda + \mu) \quad (3)$$

其中, λ 是用户的领域相关度所产生的增值系数, $0 < \lambda < 1$; μ 是根据用户的推荐等级所产生的增值系数, $0 < \mu < 1$ 。在确定 λ 和 μ 时,需要体现群体协同的思想。假设候选服务类型为 T , 且如果推荐用户 U_j 历史评价信息中 T 类服务的评价总数为 N_j , 则

$$\lambda = N_j / N \quad (4)$$

N 为所有推荐用户评价 T 类服务的总数。同样,在确定由于推荐用户等级而产生的增值系数 μ 时,也采用相同的策略,见公式(5)。

$$\mu = A_j / A \quad (5)$$

其中, A_j 为推荐用户 U_j 推荐信息被采用的次数, A 为所有推荐用户推荐信息被采用次数的总和。

从增值系数可以看出,推荐的用户对该服务领域的了解程度越高或用户的推荐等级越高,那么其所产生推荐信息的权重系数也必然越高,这符合现实中互联网用户选取服务的实际特征。

必须看到,对服务的评分是用户的一种主观感受,因此很可能存在非公正的因素。Sensoy^[9]针对用户打分主观性的问题,提出了基于经验的服务选择方法。其工作的特点是不仅要记录用户主观评分,还要客观地记录用户实际使用服务的情况(例如服务响应时间),从而便于其他使用者对服务进行评估。类似地,大量的研究工作在服务选择上客观地基于服务 QoS 属性(通用的 Web 服务 QoS 属性包括执行价格 $q_{pay}(ws)$ 、执行时间 $q_{time}(ws)$ 、执行成功率 $q_{rat}(ws)$ 以及可靠性 $q_{av}(ws)$),以衡量用户在多个维度上的不同需求。本文所提出的评价相似度公式并未考虑这些维度,而仅仅通过用户对服务的总体评价指标进行评价相似度计算。但是,这种方法完全可以根据需求任意扩展。举例来说,我们可以计算用户在多个 QoS 属性维度上的评价相似度,最后作加权和,即

$$r_{uv} = w^{(p)} \times r_{uv}^{pay} + w^{(t)} \times r_{uv}^{time} + w^{(r)} \times r_{uv}^{rat} + w^{(a)} \times r_{uv}^{av}, w^{(p)} + w^{(t)} + w^{(r)} + w^{(a)} = 1 \quad (6)$$

如前所述,在计算用户评价相似度时,我们并不区分两个用户是否存在推荐关系。至此,可以根据用户对服务的历史评价信息搜集出该用户的推荐用户,并通过皮尔逊相关系数计算出每个用户的评价相似度。下面,基于这样的前提引入 TSSPR 算法。

3 TSSPR:基于偏好推荐的可信服务选择

3.1 TSSPR算法思想和模型

互联网环境下,许多 Web 站点为各种用户提供服务评价机制。随着服务评价信息的日益丰富与积累,这类可被公共获取和访问的网络资源为 TSSPR 算法走向应用提供可行性上的保证。例如,目前运营状态良好的 Web Service 搜索引擎 seekda(<http://seekda.com/>)和 WebserviceList(<http://webservicelist.com/>)网站都提供了方便的服务评价机制。分享用户访问服务的经验并对服务打分。为方便阐述算法的思想,我们依旧假设用户 U_i 希望获取一组功能相同服务 $\{ws_1, ws_2, \dots, ws_n\}$ 的推荐信息,并从中选取最可能符合自身需求的服务进行访问。此时,用户 U_i 已经寻找到一组推荐用户,计算每个推荐用户与用户 U_i 的评价相似度后,按照数值由高至低排序,并取前 K 个用户

$$S_{U_1 \leftrightarrow U_i} \geq S_{U_2 \leftrightarrow U_i} \geq \dots \geq S_{U_K \leftrightarrow U_i}$$

其中,用户 U_1, \dots, U_K 关于该组服务的评价值见表 2,表中 $null$ 表示用户对服务没有评价, R_{ij} 为服务的评价值。

由公式(3)可以计算出每个用户对服务的推荐度,若有多个用户对服务 $ws_j (j \leq n)$ 评价过,则总的推荐度为这些推荐用户对服务 ws_j 的推荐度之和。为防止多位用户访问同一个服务而导致的推荐度增加,我们需要对推荐度

总和进行整理:

$$\sum_{ws_j \in WS_i} R_{U_i \Rightarrow U_i}(ws_j) / \sum_{ws_j \in WS_i} R_{U_i \Leftarrow U_i}$$

(7)

即多个推荐用户的推荐度之和与评价相似度之和相除.需要说明的是,本文暂不考虑推荐用户的数量对推荐度贡献的大小.然而在实际情况中,若某服务被评价的次数比其他服务多,则可能该服务更受好评或者差评,因此很难通过评价次数准确判断贡献度的走向.但是也必须认识到,若一个服务被评价的次数越多,那么基于公式(7)所获得的服务推荐度也越为精确和可信.此外,还有一种推荐用户的选择方法,即选择评价相似度大于某一数值的所有推荐用户,我们可以根据应用情况确定选取何种方式.通常认为,皮尔逊相关系数在 0.8~1.0 为极强相关,0.6~0.8 为强相关,0.4~0.6 为中等程度相关,0.2~0.4 为弱相关,0.0~0.2 为极弱相关或无相关.

Table 2 Recommendatory user scoring matrix
表 2 推荐用户评分矩阵

User	WS			
	WS ₁	WS ₂	...	WS _n
U ₁	R _{1,1}	null	...	R _{1,n}
U ₂	null	R _{2,2}	...	R _{2,n}
...
U _k	R _{k,1}	R _{k,2}	...	R _{k,n}

在介绍算法之前,还需要解决冷启动的问题:当新用户加入系统时,由于缺乏历史评分记录,很难对其产生推荐.文献[15]认为,解决冷启动的办法是对用户进行训练.但用户的多样性会导致训练样本数据难以确定,甚至造成用户放弃系统的使用.基于我们的假设,用户在寻找推荐用户前已有一组希望获取推荐信息的服务列表,如果没有寻找到足够的推荐用户,那么可以通过减少 M 值的大小降低偏好相似用户的门槛,或者直接对多个用户对服务的评分求均值.这样不仅避免了传统推荐系统需要将一个用户和其他所有用户进行比较的要求,而且较好地解决了冷启动的问题.虽然在初始阶段降低了推荐信息的质量,但该策略简单且有效.随着用户评价信息的增加,其所获得的推荐信息质量将逐渐增加.

图 2 为本文提出的服务推荐模型.在该模型中,通过用户的反馈实现对推荐用户等级的影响,从而遏制不诚实用户对系统的破坏.同时,每个用户都由于其推荐等级和领域相关度的不同而有着不同的推荐权重系数.

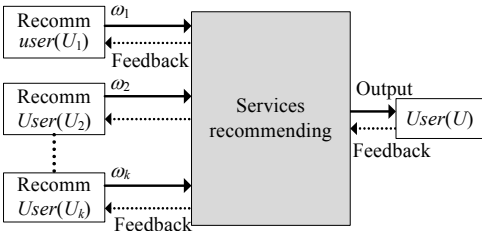


Fig.2 Services recommending model
图 2 服务推荐模型

3.2 过程描述

根据上述的思想和模型,本节设计了可信服务的选择的偏好推荐算法;在接受用户的查询请求后,比较过滤出一组推荐用户,最终产生一组服务推荐列表,用户可以根据特定需求从中选择最满意解;未被选中的非最满意解可以在服务访问发生异常时作为冗余服务使用.

算法 1. TSSPR 算法.

- 输入:(1) 用户 U_i 的服务评价信息 $\langle WS_i, \delta_i \rangle$;
- (2) 希望获取推荐值的一组功能相似候选服务 $\{ws_1, ws_2, \dots, ws_n\}$;
- (3) 条件因子 M ;

(4) 推荐用户数量 K .

输出:按照推荐值大小排序的服务推荐列表.

步骤 1. 对候选服务的服务类别进行识别.

步骤 2. 若用户的访问服务列表大小 $|WS_i| < M$, 则直接搜索访问过服务 $\{ws_1, ws_2, \dots, ws_n\}$ 的一组用户, 并对服务评分求均值, 跳至步骤 7. 其中, 在计算每个服务评分时需要考虑评分用户的领域相关度、推荐等级和时间因素对评分的影响.

步骤 3. 按照用户 U_i 的历史访问服务集合 WS_i 和候选服务集合 $\{ws_1, ws_2, \dots, ws_n\}$ 搜索推荐用户.

步骤 4. 计算所有推荐用户与用户 U_i 的评价相似度, 并取评价相似度最高的前 K 个用户.

步骤 5. 计算推荐用户领域相关程度和推荐等级所产生的增值系数 λ 和 μ .

步骤 6. 根据公式(7), 计算服务集合 $\{ws_1, ws_2, \dots, ws_n\}$ 中每个服务的推荐度, 形成服务推荐度列表.

步骤 7. 对推荐列表进行排序, 并返回推荐列表.

步骤 8. 获取用户使用推荐服务后的反馈信息, 并作用于推荐用户的推荐等级.

4 实验与结果分析

本节通过一组仿真实验比较 3 种服务选择策略的效果, 并对结果进行分析和评价. 3 种服务选择策略分别为随机服务选择策略、基于随机用户推荐信息的服务选择策略和基于推荐信息的可信服务选择策略:

- 随机服务选择策略(简称 RSS): 用户在服务需求列表中随意选取并访问, 不借鉴任何推荐信息.
- 基于随机用户推荐信息的服务选择策略(简称 RRSS): 随机选取一组用户, 并通过该组用户的评价值对预访问服务的评分求均值, 用户选取评分最高的服务访问.
- 基于偏好推荐的可信服务选择策略(简称 TSSPR): 该策略为本文所提出的服务选择方法, 根据用户需求协同过滤出一组推荐用户, 并通过推荐用户的领域相关度和推荐等级调整推荐信息的权重.

4.1 实验准备

由于该方法基于用户的评分记录, 因此必须建立用户评分数据集. 为使模拟实验更为真实, 我们参考注册在 WebServiceList 站点(<http://webservicelist.com/>)上服务的分类情况, 并按照文献[18]的分类方法对服务进行划分. 服务总数为 100, 服务分类情况为: 访问控制/安全类服务——6, 地址/定位类服务——13, 金融商业类服务——21, 开发工具类服务——12, 目录/数据库类服务——5, 政府/政策类服务——13, 在线验证类服务——6, 证券类服务——7, 搜索/探测类服务——5, 自动销售类服务——5, 零售类服务——7.

在实验中, 我们模拟了 300 个服务使用者, 且初始阶段每个用户都对 20~40 个服务做过评价. 为了体现不同用户的评价指标对评分结果的影响, 我们假设用户仅根据服务的响应时间作为评分的标准. 也就是说, 用户的评价指标是时间. 举例来说, 某类用户可能对响应时间在 1 秒内的服务感到满意, 而某些用户可能认为 3 秒~4 秒内返回结果的服务是满足自身需求的. 在实验中, 我们根据该评价指标对用户和服务进行划分. 我们将评价指标分为 10 个类别, 即 1~10, 我们假设所有服务的响应时间也按照该类别均匀地分布在这 10 个跨度中. 例如, 用户的评分指标为 8, 则对应不同响应时间服务的用户评价价值见表 3.

Table 3 User's rating (rating level=8)

表 3 用户评分表(评分指标为 8)

Response time of service	1	2	3	4	5	6	7	8	9	10
Rating	3	4	5	6	7	8	9	10	9	8

实验环境为 PC 机, 硬件为 Intel Core(TM)2 Duo CPU 2.66GH, 4GB 内存; 软件环境为 Windows XP Professional Service Pack 2, 仿真程序用 Python 语言编写.

仿真实验采用用户满意度作为评价指标, 若预测的用户评分集合为 $\{p_1, p_2, \dots, p_n\}$, 而实际评分集合为 $\{q_1, q_2, \dots, q_n\}$, 则用户满意度通过公式(8)得到. 在本文的实验中, 用户满意度即为返回服务的响应时间是否符合用户

自身的需求.

$$Sat(U)=\frac{\sum_{i=1}^n|p_i-q_i|}{n}$$

(8)

4.2 实验结果和分析

4.2.1 有效性实验

该实验用于验证推荐算法的有效性.如图 3 所示,在初始阶段,用户并没有任何访问记录(rating num=0),因此 TSSPR 算法的效果等同于基于随机用户推荐信息的服务选择策略(RRSS);同时可以看到,随着迭代次数的增加,用户的满意度随着用户的评分次数的增长而呈增加趋势,这反映出 TSSPR 算法在寻找满足用户偏好服务方面是可行的.

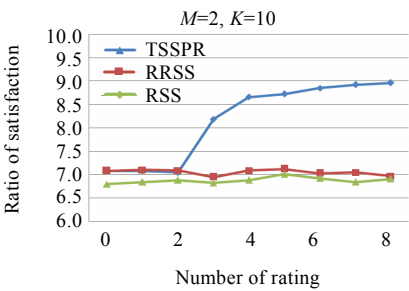


Fig.3 Ratio of satisfaction comparison with different selection methods

图 3 不同选择方法的用户满意度比较

在实验中为了简化对不同用户评分指标的描述,我们把用户和服务划分为 10 类且均匀分布在这 10 个跨度中.因此从图中可以看出,在迭代次数大于 4 后,算法就得到了较好的效果.在实际的应用中,用户的评分指标可能千差万别,为了算法能够达到理想的效果,需要更多的迭代次数.为此,我们把用户评价指标和服务的响应时间最大划分到 100 种,并考察迭代次数对用户满意度的影响,见表 4.

Table 4 Effect of ratio of satisfaction with different iterations

表 4 迭代次数对用户满意度的影响

Num of category	M	K	Rating level	Iterations with better result	Ratio of satisfaction with the iterations	Percent (%)
5	2	10	4	3	4.1 (5)	82
10	2	10	8	4	8.69 (10)	86.9
30	2	10	24	5	26.31 (30)	87.7
50	2	10	40	5	44.4 (50)	88.8
100	2	10	80	6	89.5 (100)	89.5

用户和服务的类别数越高,说明存在越多的用户评分指标;同样,服务的响应时间也存在更多的跨度.这里我们没有为“较好效果”下定义,我们认为,随着迭代次数的增加,当用户的满意度不会再有明显的增加时,则认为达到了较好的效果,而对应的迭代次数也是我们所关注的.可以看到,算法收敛的速度比较快,达到较好效果的迭代次数较低.例如,当用户和服务的类别数达到 100 时,只需迭代 6 次即可达到 89.5 的用户满意度(最高满意度为 100).也就是说,冷启动问题将随着用户评价次数的少量增加而不复存在.

4.2.2 参数调整

在算法中,针对不同的应用场景和用户的评价次数合理地设置 M 和 K,可以使算法尽快达到理想的效果.通常,M 越小,寻找到的偏好相似用户越多,但也会出现公共评分项集过少的现象,从而导致评分相似度的计算不准确;如果 M 设置过大,则难以找到足够多的偏好相似用户,会产生找不到推荐用户或推荐用户的推荐信息不够准确的情况.图 4 描述了在 K 值锁定的情况下,用户的满意度随着迭代次数的增加在不同 M 值下的走势.正如我们所讨论的,如果 M 取值过小,例如 M=1,则用户的满意度将一直保持在较低的水平,为 8 左右;如果 M 取值过大,

例如 $M=7$,则在迭代的开始阶段由于找不到足够的推荐用户而导致用户满意度较低,只有在迭代次数大于 18 后才能返回比较满意的推荐信息。

K 值的设置决定了参与推荐的用户数量。我们模拟了访问评价次数为 40 次的用户在不同 K 值下的用户满意度。由图 5 可见,随着 K 的增加,其满意度呈总体上升趋势,这是因为推荐用户的增加会使推荐信息更为准确。另一方面,在 $K \geq 9$ 之后,用户的满意度基本保持在一个较稳定的水平上;但是在 $K \geq 15$ 后,用户的满意度会有微小的下降。这是因为,模拟实验中选取推荐用户的策略是按照评价相似度的高低取前 K 个用户,当推荐用户较少时,排名靠后的推荐用户的评价相似度可能较低,从而导致推荐信息不够准确。用户满意度仅有较小幅度的下降,是因为我们在计算用户推荐度时考虑了用户间的评价相似度(见公式(3)),从而减少了不确定推荐信息对推荐结果的影响。

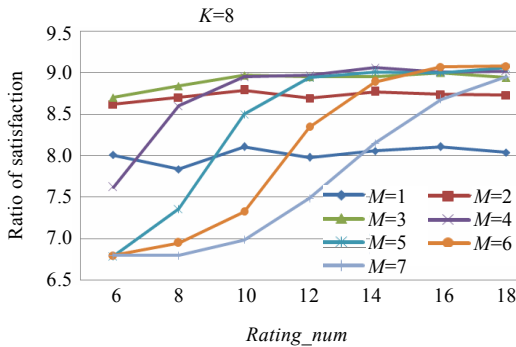


Fig.4 Ratio of satisfaction comparison with different Ms

图 4 不同 M 值的满意度比较

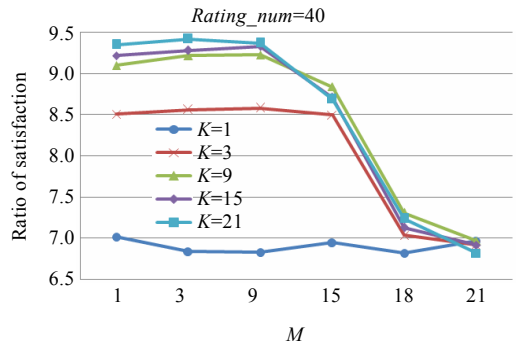


Fig.5 Ratio of satisfaction comparison with different Ks

图 5 不同 K 值的满意度比较

如第 3.1 节所述,还有一种通过选择评价相似度(皮尔逊相关系数)大于某数值的用户作为推荐用户的选取策略。为此,我们模拟了选取评价相似度大于 0.2,0.4,0.6 和 0.8 的推荐用户产生推荐后的用户满意度。如图 6 所示,用户满意度并未受评价相似度的影响。究其原因,一是在计算用户推荐度时考虑了评价相似度,二是在该模拟场景中寻找到了足够的推荐用户,如图 7 所示。比较图 6 和图 7 可以看出,用户的满意度与能够寻找到的推荐用户的数量具有较紧密的联系,基本上当推荐用户数量大于 10 以后,就可以达到较好的推荐效果。

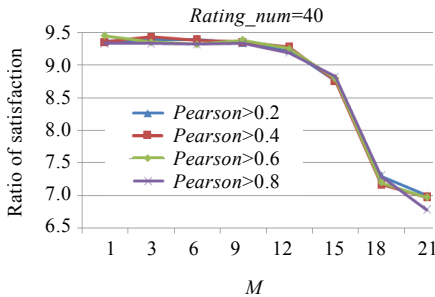


Fig.6 Ratio of satisfaction comparison with different Ms and correlation

图 6 不同 M 值和相关系数下满意度比较

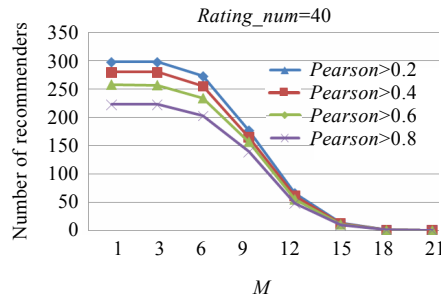


Fig.7 Number of recommenders with different Ms and correlation

图 7 不同 M 值和相关系数下的推荐数量

4.2.3 服务响应时间分布调整

前两节的模拟实验基于服务响应时间均匀分布的假设,因此,即使是随机选择服务(RSS)的策略也有接近 6.8 的用户满意度。在接下来的模拟中,我们将指定部分服务的响应时间,然后比较 3 种服务选择策略的优劣。本

次模拟实验中参数设置为: $M=4, K=10$, 用户已评价次数为 20, 评价指标为 8. 在第 1 个实验中, 我们分别观察系统中在响应时间为 1 (用户对该类服务的满意度为 3) 的数服务量从 5 递增至 100 的情况下, 3 种服务选择策略的用户满意度, 如图 8(a) 所示. 实验结果表明, 随着系统中响应时间为 1 的服务数量的增加, 3 种选择算法的效果都呈下降趋势. 但是, 在该类服务数量达到 50 后, TSSPR 算法产生的推荐服务的用户满意度仍高达 8.3 左右. 这说明, 只要系统中存在满足用户需求的服务, TSSPR 算法就可以产生较好的推荐效果.

第 2 个实验用于验证服务响应时间的分布在极端情况下对推荐质量的影响. 该实验假设服务的响应时间只有两类, 分别为 8 和 1, 即用户的评价值分别为 10 和 3, 实验结果如图 8(b) 所示. 从图中可以观察到, 当两类服务的数量各占一半时, TSSPR 算法产生推荐的用户满意度依然接近 10, 而 RRSS 和 RSS 算法已经急速下降到了 7.8 和 6.4. 即使响应时间为 1 的服务数量已经达到 80 (此时, 系统中满足用户需求的服务数量为 20), TSSPR 算法的推荐效果依旧可以达到 8.2 左右的用户满意度.

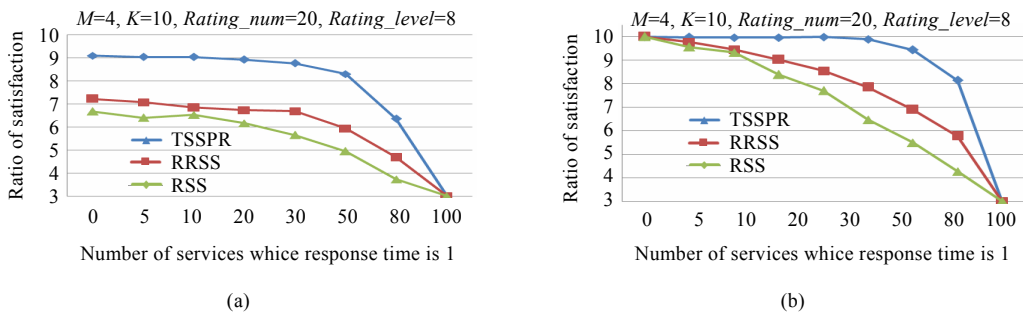


Fig.8 Ratio of satisfaction comparison with specified services' response time

图 8 指定服务响应时间场景下满意度比较

4.2.4 不诚实评价

推荐系统中存在一类需要特别关注的问题, 即不诚实评价的问题. 不诚实用户可以不按照实际服务的结果提交评价, 从而影响推荐的效果. 对于我们提出的服务推荐模型来说, 由于不诚实用户并不知道用户的评价指标, 导致该类不诚实用户很难被选为推荐用户. 此外, 我们在推荐算法中引入了用户评价等级, 不诚实用户由于推荐信息质量不高导致其推荐权重的下降, 因此并不会对本文所提出的推荐算法效果产生较大影响. 在模拟实验中, 我们依旧假设响应时间为 1 和 8 的服务各占一半, 如图 9 所示. 随着不诚实用户数量的增加, TSSPR 算法产生推荐服务的用户满意度几乎未被影响.

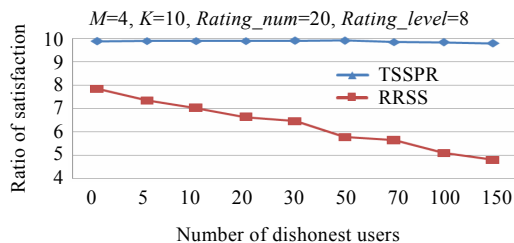


Fig.9 Ratio of satisfaction comparison with dishonest users

图 9 不诚实用户参与的满意度比较

5 结束语

服务的本质在于开放, 不能期望服务提供方和服务使用方相互熟知, 也不能要求服务使用方具备专业知识. 因此在开放系统中, 基于推荐信息的可信服务选取是一种最为合理、有效的手段, 已成为目前主流的研究方向之一. 本文面向开放的 Internet 环境, 提出基于用户推荐信息的可信服务选择方法. 该方法根据用户的查询请求

搜寻一组推荐用户,并通过推荐用户的领域相关度、推荐等级和评价相似度对用户进行过滤和筛选,力求推荐信息的可信且满足用户的偏好.实验结果表明,与传统的推荐系统相比,该方法不仅避免了需要将一个用户与所有其他用户进行比较的要求,而且较好地解决了冷启动等问题;随着用户评价信息的增加,其所获得的推荐信息质量也将逐渐增加.

下一步的工作将主要从以下 3 个方面展开:

- (1) 恶意推荐问题,恶意用户可以获取用户的历史评分记录,并通过该评价记录假扮成推荐用户进行恶意推荐.文献[14]虽然指出这类恶意用户进行类似攻击得不偿失(没有动机),但有必要在分布式系统中进一步完善用户的反馈和惩罚机制,从而提高推荐系统的推荐质量.
- (2) 激励机制,如何鼓励更多的用户在使用服务后对服务做出客观的评价.
- (3) 如何在推荐服务时考虑组合服务的需求,服务推荐不仅要满足局部要求,也要考虑全局约束,从而提高组合服务的执行成功率和用户满意度.

References:

- [1] Xu HL, Wu X, Li XD, Yan BP. Comparison study of Internet recommendation system. *Journal of Software*, 2009,20(2):350–362 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/20/350.htm> [doi: 10.3724/SP.J.1001.2009.00350]
- [2] Wang HM, Tang YB, Yin G, Li L. Trustworthiness of Internet-based software. *Science in China (Series F: Information Sciences)*, 2006,49(6):759–773. [doi: 10.1007/s11432-006-2024-4]
- [3] Segaran T. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media, Inc., 2007.
- [4] Ran SP. A framework for discovering Web services with desired quality of service attributes. In: Zhang LJ, ed. *Proc. of the Int'l Conf. on Web Services (ICWS 2003)*. 2003. 208–213. <http://dblp.uni-trier.de/db/conf/icws/icws2003.html#Ran03>
- [5] Tian M, Gramm A, Naumowicz T, Ritter H, Schiller J. A concept for QoS integration in Web services. In: *Proc. of the 4th Int'l Conf. on Web Information Systems Engineering Workshops (WISEW 2003)*. 2003. 149–155. <http://citeseer.ist.psu.edu/old/tian03concept.html> [doi: 10.1109/WISEW.2003.1286797]
- [6] Truong HL, Samborski R, Fahringer T. Towards a framework for monitoring and analyzing QoS metrics of grid services. In: *Proc. of the 2nd IEEE Int'l Conf. on e-Science and Grid Computing*. 2006. <http://portal.acm.org/citation.cfm?id=1192634> [doi: 10.1109/E-SCIENCE.2006.261149]
- [7] Maximilien ME, Singh MP. Toward autonomic Web services trust and selection. In: *Proc. of the 2nd Int'l Conf. on Service Oriented Computing (ICSOC 2004)*. 2004. 212–221. <http://portal.acm.org/citation.cfm?id=1035198> [doi: 10.1145/1035167.1035198]
- [8] Yolum P, Singh MP. An agent-based approach for trustworthy service location. In: *Proc. of the 1st Int'l Workshop on Agents and Peer-to-Peer Computing (AP2PC 2002)*. 2002. 45–56. <http://www.springerlink.com/content/mtut3abakrrnv0vn/> [doi: 10.1007/3-540-45074-2_5]
- [9] Sensoy M, Pembe FC, Zirtiloğlu H, Yolum P, Bener A. Experience-Based service provider selection in agent-mediated e-commerce. *Engineering Applications of Artificial Intelligence*, 2007,20(3):325–335. [doi: 10.1016/j.engappai.2006.06.003]
- [10] Goldberg D, Nichols D, Oki BM, Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992,35(12):61–70. [doi: 10.1145/138859.138867]
- [11] Karta K. An investigation on personalized collaborative filtering for Web service selection. Technical Report, 2005. <http://www.csse.uwa.edu.au/%7Ewei/honours/2005/cshonours-ken.pdf>
- [12] Sarwar BM, Karypis G, Konstan JA, Riedl JT. Application of dimensionality reduction in recommender system—A case study. In: *Proc. of the ACM WebKDD Workshop*. 2000. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.744>
- [13] Zhang GW, Li DY, Li P, Kang JC, Chen GS. A collaborative filtering recommendation algorithm based on cloud model. *Journal of Software*, 2007,18(10):2403–2411 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/2403.htm> [doi: 10.1360/jos182403]
- [14] Massa P, Bhattacharjee B. Using trust in recommender systems: An experimental analysis. In: *Proc. of the 2nd Int'l Conf. on Trust Management*. Berlin: Springer-Verlag, 2004. 221–235. <http://www.informatik.uni-trier.de/~ley/db/conf/itrust/itrust2004.html>

- [15] Wang Y, Vassileva J. A review on trust and reputation for Web service selection. In: Proc. of the 27th Int'l Conf. on Distributed Computing Systems Workshops (ICDCSW 2007). 2007. <http://portal.acm.org/citation.cfm?id=1270925> [doi: 10.1109/ICDCSW.2007.16]
- [16] Robert R. How to quantify the user experience. 2009. <http://www.sitepoint.com/article/quantify-user-experience/>
- [17] Zhou RG. How to quantify user experience: fuzzy comprehensive evaluation model based on summative usability testing. In: Aykin N, ed. Usability and Internationalization, Part II, HCII 2007. LNCS 4560, Springer-Verlag, 2007. 564–573. <http://dblp.uni-trier.de/rec/bibtex/conf/hci/Zhou07>
- [18] Saha S, Murthy CA, Pal SK. Classification of Web services using tensor space model and rough ensemble classifier. In: Proc. of the 17th Int'l Symp. on Methodologies for Intelligent Systems (ISMIS 2008). 2008. 508–513. <http://www.springerlink.com/content/f16571u668n34l46/> [doi: 10.1007/978-3-540-68123-6_55]

附中文参考文献:

- [1] 许海玲,吴潇,李晓东,阎保平.互联网推荐系统比较研究.软件学报,2009,20(2):350–362. <http://www.jos.org.cn/1000-9825/20/350.htm> [doi: 10.3724/SP.J.1001.2009.00350]
- [13] 张光卫,李德毅,李鹏,康建初,陈桂生.基于云模型的协同过滤推荐算法.软件学报,2007,18(10):2403–2411. <http://www.jos.org.cn/1000-9825/18/2403.htm> [doi: 10.1360/jos182403]



朱锐(1980—),男,北京人,博士,主要研究领域为分布计算,可信计算.



冯大为(1985—),男,博士生,主要研究领域为分布计算.



王怀民(1962—),男,博士,教授,博士生导师,主要研究领域为分布计算中间件,软件Agent,网络与信息安全.