

基于用户社群关系的 Web 服务发现研究

田浩^{1,2}, 樊红¹, 杜武¹

(1. 武汉大学 测绘遥感信息工程国家重点实验室, 湖北 武汉 430079; 2. 湖北经济学院 信息工程学院, 湖北 武汉 430205)

摘 要: Web 服务发现是 Web 应用研究中的关键问题。研究了多个用户彼此之间的社群关系对 Web 服务发现结果的影响, 将用户之间的社群关系分解为偏好关系、聚类关系和信任关系, 给出了这 3 个关系的形式化方法。基于形式化后的社群关系提出了一个差分式服务发现策略, 构建了一个基于用户社群关系的 Web 服务发现系统框架, 按照 Web 服务与用户兴趣背景相关性由强到弱的原则来逐步检索或推荐候选 Web 服务以产生最终结果。实验表明, 该方法能有效提高服务发现的效率和查准率。

关键词: Web 服务发现; 社群关系; 策略; 形式化方法

中图分类号: TP393

文献标识码: A

Research on Web service discovery based on user community relations

TIAN Hao^{1,2}, FAN Hong¹, DU Wu¹

(1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China;

2. School of Information Engineering, Hubei University of Economics, Wuhan 430205, China)

Abstract: Web service discovery was the key problem in the study of the Web application. Multiple user community relations between each other on the result of Web service discovery was studied, the user community relations was decomposed into predilection relation, clustering and trust relation, while describing the methods of formalization. Based on the formalized community relations a differential service discovery strategy was put forward, a Web service discovery system framework was built, which can retrieve or recommend candidate Web services step by step to produce the final result according to the “from strong to weak” principle in terms of the correlation between Web service and the user interest background. The experimental results show that proposed method can effectively improve the efficiency and precision of Web service discovery.

Key words: Web service discovery; community relations; strategy; formalization methods

1 引言

Web 服务作为一种重要的解决方案, 大量地被开发并被应用到 Internet 上。而由于分布式网络环境的异构性和跨领域语义的复杂性, Web 服务发现问题成为了制约 Web 服务大规模应用的技术难题之一。目前, 大部分 Web 服务发现方法在本质上都是基于注册机制的, 为追求更精确的服务发现结果及更好的自动化过程, 大量的精力被耗费在描述服务发布和服务请求上。而且这些方法在注册机制中

基本不记录或考虑用户之间相互交互的历史操作, 只是将重点放在单个用户服务请求内容的解析上, 损失了许多额外的用户辅助信息, 致使计算资源和通信开销重复多、浪费大、效率低, 无法有效应对服务请求或候选服务对象数量增多的情况。

本文以用户的辅助信息为切入点, 考虑用户的偏好、用户的使用历史及用户的聚类关系、信任关系等社群关系因素对 Web 服务发现结果准确度和效率的影响, 研究用户社群关系的形式化方法和模型, 提出基于用户社群关系的 Web 服务发现方法,

收稿日期: 2014-10-14; 修回日期: 2015-07-07

基金项目: 国家自然科学基金资助项目 (41471323)

Foundation Item: The National Natural Science Foundation of China (41471323)

并在此基础上构建一个服务发现系统框架。实验结果表明,该方法有较好的稳定性,能有效提高 Web 服务发现的效率和查准率。

2 相关研究

目前已有大量的模型和方法来研究 Web 服务发现问题。它们有着不同的侧重点和性能表现,通常将服务描述、规则推理和服务 QoS 等作为研究切入点和解决重点。

通过服务描述和标注来凸显 Web 服务发布与服务请求的功能属性和语义特征是目前大多数方法的选择。Paliwal 等提出基于语义技术对服务描述进行标注和分类的自动服务发现解决方案^[1],利用多层本体概念以及一个改进的服务向量模型来对服务发布进行描述和分类,同时利用 LSI (latent semantic indexing) 来增强服务请求的语义。Bruno 等利用 SVM (support vector machine) 来实现标注后服务的自动分类^[2],并采用形式概念分析中的概念格技术,通过建立语义标注间的关系格来确定特定域中的概念和同类服务间的联系。文献[3]提出了一个语义标注模型,该模型可以将 WSDL 中的功能参数分别映射为 OWL-S 中的原子过程,也能将服务请求转化为一个基于 OWL-S 的语义 Web 服务。

语义技术能够支持逻辑推理和服务的自动执行。因此,构建一定的语义规则,通过逻辑推理来实现服务发现过程的自动化和智能化是 Web 服务发现研究中的一个典型做法。Di Noia 等^[4]为实现高效服务匹配提出了一个基于描述逻辑的发现框架,提出了概念外展和概念收缩 2 个非单调的推理,在这 2 个推理的基础上设计了相应的语义匹配算法。Alberti 等^[5]基于计算逻辑提出了一个推理引擎框架,通过基于本体的推理来解决服务发现问题。其推理主要针对服务的行为接口,该方法根据服务发布与服务请求之间的行为接口建立一个可能的交互计划。推理引擎框架通过对该计划的逻辑推理来实现最终的服务发现。Garcia 等^[6]在服务发现执行之前设计增加了一个基于 SPARQL 查询的预处理环节,使之能够在候选服务发布集合内滤除那些在功能性或非功能性属性方面均与用户请求无关的服务发布,从而缩小服务发现的范围,提高服务发现的准确度。

目前,针对 Web 服务非功能属性的 Web 服务

发现已经成为研究的热点问题,大部分已有研究利用服务质量 QoS 来过滤和选择那些有相同功能属性的 Web 服务。Diamadopoulou 等^[7]就从 Web 服务的功能性属性和非功能性属性 2 个方面来实现 Web 服务发现。他们提出了一个 Web 服务发现模型,该模型能够搜集隐含在 UDDI 服务器或者独立 URL 中关于 Web 服务质量的信息,模型的选择模块根据这些质量信息在众多具有相同功能属性的 Web 服务中选择出含有最大质量信息值的服务。Kritikos 等^[8]专门开发了一种可扩展的丰富本体语言来实现基于 QoS 的服务描述,在 Web 服务的文本中定义了 QoS,并提出一个扩展现有 Web 服务标准技术的路线图^[9],以此来实现功能属性和 QoS 方面均表现良好的 Web 服务发现。Raj 等^[10]也进行了类似的研究,提出一个基于输入输出接口的 QoS 感知 Web 服务发现方法。

总体来说,以上方法本质上都是基于发布-请求这种注册机制的,这些传统的 Web 服务发现方法一般只关注服务请求内容,将主要工作集中在解析服务请求上,孤立地看待每个用户及每次用户操作,没有充分挖掘和利用服务请求之外的用户潜在辅助信息。虽然这些方法在具体细节实现上有诸多差别,但总体而言其性能差别不大,而且方法复杂度的增加往往不能带来理想的效果提升。

从最近研究来看,在 Web 服务发现过程中挖掘和运用用户信息来改善 Web 服务发现的性能是一个可行且有效的途径。文献[11]专门分析了利用用户行为信息来进行 Web 服务发现和组合所能带来的潜在优势,并讨论了应用这些行为信息所需的代价以及如何达到最优的性价比。Maamar 等^[12,13]提出了 social computing 的概念,将人们日常经历的信任度、公平性等因素组合构建成一个 social network 的形式展现出来,并利用这个 social network 来进行 Web 服务发现。文献[14]在 Web 服务发现中将终端用户的角色从单一的消费者升级为生产者 (producer) 和消费者 (consumer) 的综合体 Prosumer。文献[15]构建了一个具有信任和推荐双重关系服务网络模型,并基于此模型提出了一种可信服务发现算法。文献[16]设计了一个适合大规模开放、动态网络环境的自组织服务推荐网络模型,为服务发现提供新的思路。本文的主要目的就是通过对多个用户之间行为逻辑与联系的抽象、建模和演变,提出一个基于用户社群关系的服务发现方

法，在保证服务发现结果准确度的前提下提高 Web 服务发现的效率。

3 用户社群关系形式化

本文认为用户的偏好、用户的使用历史以及多个用户之间、用户与服务之间、服务彼此之间的社群辅助信息都可用来解决 Web 服务发现问题。用户可以从与他兴趣背景相同并且是他所信任的其他用户那里得到推荐与启发，从而通过确定优先查找对象和动态限定查找范围来完成 Web 服务发现，提高服务发现的效率与精度。本文在前期研究^[17]基础上，将 Web 服务发现过程涉及到的用户社群关系形式化表示为一个三元组，即

$$UR=(P,C,T) \quad (1)$$

其中， UR 是形式化后的用户社群关系， P 表示用户对 Web 服务的偏好关系， C 表示多个用户间的聚类关系， T 表示用户间的信任关系。图 1 所示为用户 U_a 的社群关系模型实例。

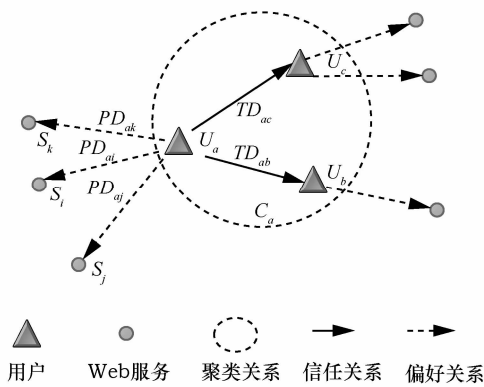


图 1 用户社群关系形式化模型实例

3.1 偏好关系

偏好关系是针对用户与那些已经被该用户调用并有评价值的 Web 服务而言的。用户对一个 Web 服务的偏好程度就是其使用该服务后的满意度，也就是服务体验后的评价。本文将偏好关系形式化如下

$$P=\{(u_a, s_i, PD_{ai})|u_a \in U, s_i \in S, PD_{ai} \in (0,1]\} \quad (2)$$

其中， U 表示全部用户集合； S 表示被 U 中的用户调用并被评价过的全部 Web 服务； PD_{ai} 表示用户 u_a 对服务 s_i 的偏好程度，其值越大表示用户越偏好此服务，而且该 Web 服务被推荐给其他用户的几率也越大，其值可以用式 (3) 来量化。

$$PD_{ai} = \frac{\sum_{q_k \in Q} v_{aik}}{\max_{s_j \in S_a} (\sum_{q_k \in Q} v_{ajk})}, \sum_{q_k \in Q} v_{aik} > \tau_p \quad (3)$$

其中， v_{aik} 、 v_{ajk} 是用户 u_a 对服务 s_i 、 s_j 的第 k 个 QoS 参数的具体评价；服务 s_i 、 s_j 均来自于用户 u_a 评价过的 Web 服务集合 S_a ； q_k 是 QoS 参数集 Q 的第 k 个参数； τ_p 是评价总分阈值。

为后续讨论的需要，用 $P_a.SSet$ 表示用户 u_a 偏好关系中的 Web 服务集合，用 $|P_a.SSet|$ 表示集合中服务个数。

3.2 聚类关系

本文将用户间的兴趣背景作为用户聚类的依据，也就是说，将有着类似兴趣背景的用户作为一个用户聚类集合，越是兴趣背景相似的用户在这个集合中就越是邻近。

用户间的兴趣背景可从用户的历史操作和调用记录等信息中挖掘得出。本文设想基于某一领域的分类本体来建立用户兴趣背景模型。因为对于一个特定的用户来说，通过分析和提炼他的历史操作和调用记录等信息就能揭示他的兴趣点，而且这些兴趣点反映在领域本体上就会表现为本体的一个局部概念分支。也就是说，用户的兴趣背景模型可视领域本体树的子树。为分析的方便，本文假设用户的兴趣点仅限定在某个特定的应用领域内，暂不考虑用户兴趣跨领域的情况。本文以本体形式化方法中的六元组法^[18]为基础来构建用户兴趣背景模型 IB

$$IB=(C, A^C, R, A^R, H, X, I) \quad (4)$$

其中， C 表示领域本体中各个层次的概念集合； A^C 是对应着每个本体概念的属性集合； R 是本体关系的集合； A^R 是对应着每个本体关系的属性集合； H 代表着一个针对概念 C 的层次结构，它实质上是一组描述 C 中的概念间特定关系的集合，这种关系一般是父-子（超类-子类）关系； X 则代表着一个公理的集合，其中的每个公理就是一个约束； I 表示本体概念被用户所关注的程度，具体为

$$I=\{(c_i, d_i)|c_i \in C, d_i \in [0,1]\} \quad (5)$$

$$d_i = \frac{cnt(c_i) + \sum_{c_j \in C} cnt(c_j)}{\sum_{c_k \in C} cnt(c_k)}, (c_i, c_j) \in H \quad (6)$$

其中， $cnt(c_i)$ 是用户对本体概念 c_i 的调用次数； $cnt(c_j)$ 是概念 c_i 的子类概念 c_j 的调用次数； $cnt(c_k)$ 是用户

对本体所有概念中任意一个概念 c_k 的调用次数。

本文采用前期研究中提出的“从顶到底，逐层匹配剪枝”的方法^[17]来计算用户之间兴趣背景模型 IB 的相似度。因此，聚类关系可以形式化为

$$C = \{(u_a, u_b, IB_{ab}) | u_a \in U, u_b \in U, IB_{ab} \in (0, 1]\} \quad (7)$$

同样地，为后续讨论的需要，用 $C_a.USet$ 表示用户 u_a 聚类关系中的用户集合，用 $|C_a.USet|$ 表示集合中用户个数。

3.3 信任关系

用户间的信任关系是通过用户间的交互行为逐渐建立的，本文将推荐行为作为用户间主要的交互行为。即若用户 u_b 推荐 Web 服务给用户 u_a 并被其调用和认可，那么 u_a 对 u_b 的信任程度就会增加。

信任关系可以通过信任程度来表征，信任程度又取决于用户的兴趣背景、每次推荐的效果和用户的评价倾向等多个因素。

推荐的效果取决于用户每次调用被推荐服务后的服务体验。若用户 u_b 将 Web 服务 s_i 推荐给用户 u_a 并被其调用和评价，那么评价此次推荐的效果 $E_{ba}(i)$ 的计算方法为

$$E_{ba}(i) = \frac{\sum_{q_k \in Q} (v_{aik} - v_{bik})}{v_m |Q|} \quad (8)$$

其中， $|Q|$ 代表 Web 服务被评价的 QoS 参数总数； v_m 表示每个参数项的最高评分值； v_{aik} 、 v_{bik} 分别是用户 u_a 和用户 u_b 对服务 s_i 的第 k 个 QoS 参数的具体评价价值。

用户的评价倾向也会影响 2 个用户之间的信任关系。本文认为，若用户 u_a 和用户 u_b 对他们共同

调用过的每一个 Web 服务的每个 QoS 参数项的评分值都完全相同，则他们的评价倾向相似度 TS_{ab} 为 1，否则 TS_{ab} 可由式 (9) 计算得出。

$$TS_{ab} = Pr_{ab} Jc_{ab} \frac{\sum_{s_i \in S_0} \prod_{q_k \in Q} \left(1 - \frac{|v_{aik} - v_{bik}|}{v_m} \right)}{|S_0|} \quad (9)$$

其中， Pr_{ab} 和 Jc_{ab} 分别是根据 Pearson 方法^[19]和 Jaccard 方法^[20]计算出的用户 u_a 和 u_b 之间的相似度； S_0 是 u_a 和 u_b 共同调用并评价过的 Web 服务集合， $|S_0|$ 是该集合中 Web 服务的总数。

在以上因素量化的基础上，用户 u_a 对用户 u_b 的信任程度 TD_{ab} 可表示为

$$TD_{ab} = IB_{ab} \left(1 + \sum_{s_i \in S_{ba}} E_{ba}(i) TS_{ab} \right) \quad (10)$$

其中， S_{ba} 表示用户 u_b 推荐给用户 u_a 的全部 Web 服务的集合。从上式可以看出，用户聚类关系是信任关系的前提基础。

因此，信任关系可以形式化为

$$T = \{(u_a, u_b, TD_{ab}) | u_a \in U, u_b \in U, TD_{ab} > 0\} \quad (11)$$

为后续讨论的需要，用 $T_a.USet$ 表示用户 u_a 信任关系中的用户集合，用 $|T_a.USet|$ 表示集合中用户个数。

4 Web 服务发现系统框架

在上述形式化关系的基础上，本文提出一个基于用户社群关系的 Web 服务发现系统框架，如图 2 所示。

此框架分为 4 个主要部分：客户端、语义支持模块、Web 服务预处理模块、服务发现执行模块。

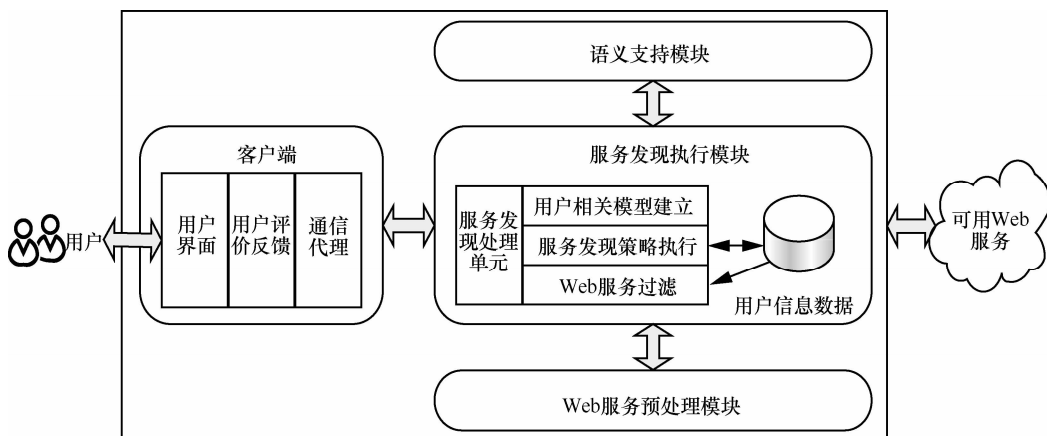


图 2 基于用户社群关系的 Web 服务发现框架

客户端主要实现 3 个功能：1) 提供用户操作界面，显示发现结果；2) 收集用户对 Web 服务的服效果评价，记录用户的操作行为；3) 实现客户端与服务器之间的通信。

语义支持模块提供相应的领域本体模型并将用户请求进行语法分词和语义标注，使之能够与本体概念对应起来。如前所述，本文讨论的只是单一领域本体，暂不考虑跨领域问题。目前已有 Assam^[21]等语义注解工具可以实现 Web 服务的自动语义标注，且实际应用效果反映较好。本文假设所有服务请求的语义信息都能够被解析对应到本体中的概念，且所有的可用 Web 服务都能被语义注解工具标注成为 OWL-S 服务。

Web 服务预处理模块的功能为：1) 用来监测、登记和管理当前网络环境下可用的 Web 服务，包括服务的功能属性和 QoS 参数等；2) 根据用户兴趣背景模型对 Web 服务进行分类和排序；3) 实现相关 Web 服务的调用。

服务发现执行模块是整个服务发现框架的核心。它主要功能为：1) 保存和管理用户操作历史、用户评价等信息数据；2) 利用用户信息数据建立用户的社群关系模型；3) 响应客户端的服务请求并返回最终结果；4) 执行特定的服务发现策略；5) 根据用户的需求对发现结果进行过滤与推荐。

5 差分式服务发现策略

为适应此 Web 服务发现系统框架的特点，有效利用用户社群关系，本文设计了一个差分式服务发现策略(DS2, differential service discovery strategy)。该策略可以分为以下 4 个阶段，这些阶段涉及到的 Web 服务对象集合内容上存在着差分关系。

5.1 服务发现预处理

此阶段的主要工作为后续服务匹配等环节提供模型和对象支持。在此阶段，需要依次完成以下任务。

首先根据用户信息数据建立起每个用户的社群关系模型 UR 。

其次监测和捕捉当前网络环境下可用 Web 服务，形成可用 Web 服务集合 S ，按照式 (12) 记录它们的状态和参数。

$$S = (sn, O, \rho) \quad (12)$$

其中， sn 是 Web 服务的名称； O 是 Web 服务的具体操作集，用来表明服务的功能属性； ρ 是检测到

的 Web 服务的 QoS 性能集合。

最后，对用户偏好服务集合 $P.SSet$ 中的每个服务 $s_i \in P.SSet$ ，根据式(13)从 Web 服务之间的功能属性和服务名 2 个方面来计算集合 S 中每个服务 s_j 与它的相似度 $CSim_{ij}$ ，并记录相似度超过阈值 τ_k 的前 k 个服务，将它们集合记为 $s_i.kSet$ 。

$$CSim_{ij} = \alpha NSim(s_i.sn, s_j.sn) + \beta FSim(s_i.O, s_j.O), \quad (\alpha > 0, \beta > 0, \alpha + \beta = 1) \quad (13)$$

上式中的 $Nsim$ 与 $Fsim$ 采用文献[17]中提出的相似度计算方法。

以上工作可以由后台在用户离线时完成，并定时进行更新，直至用户登录为止。

5.2 服务匹配

在这个阶段，当用户提交服务请求 R 后，语义支持模块通过语法分词和语义标注，将其转化为一个三元组形式

$$R = (N, O, E) \quad (14)$$

其中， N 是一个由本体概念组成的概念集，用来归纳和表达用户对目标 Web 服务的功能需求； O 用来在语义层面上指定用户对目标服务中操作的输入和输出的要求； E 是用户的实时偏好，表明用户对所需 Web 服务 QoS 参数的要求。因此，可以计算 Web 服务 s_i 与服务请求 R 的匹配度 $RSim_i$

$$RSim_i = \alpha NSim(s_i.sn, R.N) + \beta FSim(s_i.O, R.O) \quad (15)$$

其中， $\alpha > 0, \beta > 0, \alpha + \beta = 1$ 。

用户请求标注完成后，开始执行服务匹配流程。在此过程中，本文利用用户社群关系中的偏好关系和聚类关系，按照候选 Web 服务与用户兴趣背景相关性由强到弱的次序来逐步扩大匹配范围。这个阶段的算法伪代码如算法 1 所示。

算法 1 服务匹配算法

输入：用户名 u_a ，服务请求描述 R ，匹配阈值 τ_m

输出：初步候选服务列表 $List_1$

$List_0 = \phi$

$List_1 = \phi$

FOR $s_i \in P_a.SSet$ //查询用户 u_a 偏好关系中的服务集合

IF $RSim_i > \tau_m$ //计算并比较匹配程度

Add s_i to $List_1$ //超过阈值则加入到候选列表

ELSE

Add s_i to $List_0$ //否则加入到排除列表

```

END IF
END FOR
IF  $List_1 \neq \emptyset$ 
RETURN  $List_1$ //候选列表不为空则结束查询并
返回
QUIT
END IF
FOR  $s_i \in P_a.SSet$ //对用户  $u_a$  偏好关系中的每个
服务  $s_i$ 
    FOR  $s_j \in s_i.kSet \wedge s_j \notin List_0$ //查询与服务  $s_i$ 
最相似且不在排除列表中的前  $k$  个服务
        IF  $RSim_j > \tau_m$ //计算并比较匹配程度
            Add  $s_j$  to  $List_1$ //超过阈值则加入到候选列
表
        ELSE
            Add  $s_j$  to  $List_0$ //否则加入到排除列表
        END IF
    END FOR
END FOR
IF  $List_1 \neq \emptyset$ 
RETURN  $List_1$ //候选列表不为空则结束查询并
返回
QUIT
END IF
FOR  $u_k \in C_a.USet$ //对于用户  $u_a$  聚类关系中的每
个用户  $u_k$ 
    FOR  $s_m \in P_k.SSet \wedge s_m \notin List_0$ //查询用户  $u_k$ 
偏好关系中且不在排除列表中的服务
        IF  $RSim_m > \tau_m$ //计算并比较匹配程度
            Add  $s_m$  to  $List_1$ //超过阈值则加入到候选列
表
        ELSE
            Add  $s_m$  to  $List_0$ //否则加入到排除列表
        END IF
    END FOR
END FOR
IF  $List_1 \neq \emptyset$ 
RETURN  $List_1$ //候选列表不为空则结束查询并
返回
QUIT
END IF
FOR  $s_n \in S \wedge s_n \notin List_0$ //查询排除列表之外的余
下服务

```

```

    IF  $RSim_n > \tau_m$ //计算并比较匹配程度
        Add  $s_n$  to  $List_1$ //匹配程度超过阈值则加入到候
选列表
    ELSE
        Add  $s_n$  to  $List_0$ //否则加入到排除列表
    END IF
END FOR
RETURN  $List_1$ //返回候选列表

```

算法 1 依次通过用户的偏好关系和聚类关系来逐步缩小服务发现过程中的候选对象范围。首先查询用户调用过并偏好的服务集合；若得不到满意结果则扩展到集合中的每个服务临近的 k 个服务；同理再查询聚类关系集合中每个用户的偏好服务集；仍得不到满意结果的话就采用泛洪算法查询余下的可用服务。整个发现过程按照候选服务与用户需求由强到弱的次序来查询，提高了服务发现的针对性，也为提高整个算法的精度提供了保证。

5.3 结果生成

以上阶段的算法和步骤执行完成后，可能因为服务请求条件限制过多或者是服务请求表述等方面的原因没有查询到所需服务，致使服务列表 $List_1$ 仍为空。那么此时就需要根据用户的信任关系产生最终的返回结果。具体而言，可以通过算法 2 来得到最终候选服务列表 $List_2$ 。

算法 2 结果生成算法

输入：用户名 u_a ，服务请求描述 R ，初步候选服务列表 $List_1$

输出：最终候选服务列表 $List_2$

$List_2 = \emptyset$

IF $List_1 \neq \emptyset$

$List_2 = List_1$ //候选列表不为空则就是最终结果

RETURN $List_2$

QUIT

END IF

FOR $u_i \in T_a.USet$ //对于用户 u_a 信任关系中的每个用户 u_i

FOR $s_j \in P_i.SSet$ //查询用户 u_i 偏好关系中的服务集

Calculate $RSim_j$ //计算服务集中每个服务 s_j 与请求的匹配程度

END FOR

Add s_k which has $\max(RSim)$ to $List_2$ //将匹配程度最高的服务加入到最终候选服务列表

END FOR

RETURN $List_2$ // 返回最终候选服务列表

算法 2 是对算法 1 的补充, 当算法 1 执行结果为空时, 算法 2 会根据用户的信任关系推荐潜在的可能目标服务给用户。对于当前用户信任集合中每个其他用户, 算法 2 从他们各自的偏好关系集合中找出最接近当前用户需求的服务, 汇总到最终候选服务列表推荐给当前用户, 增加了算法的灵活性。

5.4 选择与评价

此阶段, 服务发现执行模块首先根据 Web 服务预处理模块监测到的 QoS 性能, 并结合用户 u_a 的实时偏好, 依据式 (16) 计算出列表 $List_2$ 中每个 Web 服务 s_i 与用户请求 R 的最终相似度 Sim_i

$$Sim_i = \omega_0 RSim_i + \sum_{q_k \in Q} \omega_k \frac{v_{ik}}{v_m} \quad (16)$$

其中, $\omega_0 > 0, \omega_k > 0, \omega_0 + \sum \omega_k = 1$ 。 v_{ik} 是系统监测到的服务 s_i 的第 k 个 QoS 参数值, v_m 表示每个参数的最高评分值。系数 ω_0 和 ω_k 由用户 u_a 在提交服务请求时指定。

其次, 将结果列表返回给客户端, 用户 u_a 根据候选 Web 服务列表来选择、调用和评价目标服务。

最后, 系统收集此次发现过程中新增的用户 u_a 信息数据, 并在后台更新用户 u_a 的社群关系模型。

6 实验与分析

为检测本文提出的框架、算法和模型的性能, 本文构建了一个原型系统。客户端基于 Bootstrap 框架并用 JSP 来实现, 部署在 10 台 PC 机上; 服务器端采用基于 Win7 64 位操作系统的 Tomcat 服务器, 硬件环境为 CPU: Intel Core 双核四线程 2.20 GHz, 内存: 16 GB; 指定本体为 OWLS-TC3 中的 travel 本体并用 Protégé3.5 来实现; 用户信息数据用 MySQL5.6 来管理; Web 服务用 Jena TDB 来登记和管理。

为增加实验的可信度, 实验数据主要来自于香港中文大学提供的 WS-DREAM 数据集^[22]和标准测试集 OWL-TC4^[23], 从中选取部分并采用实际标注方式产生 Web 服务 1 000 余个, 同时随机补充生成 200 余个, 这些 Web 服务基本覆盖了 Travel 本体的所有概念。为获得足够的用户数据, 本文在建立了原型系统后, 模拟了 50 位用户进行具体操作, 这些用户分别在不同时段独立使用此系统, 共产生了 1 500 多条操作记录。

实验将用户评价 Web 服务效果的 QoS 参数设定为 4 个, 包括 price、responsetime、reliability 和 availability, 采用 1~5 分评分机制; 设定相关系数为: $\tau_p=10, \tau_k=0.5, \tau_m=0.5, s_i.kSet$ 的规模 $k=10; \alpha=0.2, \beta=0.8$; 限定 $|P_a.SSet|, |C_a.USet|$ 和 $|T_a.USet|$ 的最大值均为 10。

图 3 和图 4 是原型系统 2 种不同的返回结果。图 3 表明, 对应的服务发现过程只利用了用户的偏好关系和聚类关系就得到了最终结果; 而从图 4 中则可以看到, 针对此次用户请求, 服务发现过程在执行了算法 1 之后仍未得到满意结果, 因此算法 2 就要根据用户的信任关系来产生最终的返回结果。

服务id	信任用户	信任用户评分	相似度	操作
16			0.75016606	查看详情 评分选择

本次搜索耗时: 9ms

图 3 系统运行结果实例 1

服务id	信任用户	信任用户评分	相似度	操作
66	u101601008	["time":2,"price":3,"reliability":5,"availability":2]	0.8610179	查看详情 评分选择
72	u101601010	["time":3,"price":2,"reliability":4,"availability":4]	0.76635814	查看详情 评分选择
15	u101601005	["time":1,"price":2,"reliability":5,"availability":5]	0.75016606	查看详情 评分选择
61	u101601031	["time":4,"price":4,"reliability":1,"availability":3]	0.7109439	查看详情 评分选择
45	u101601014	["time":4,"price":5,"reliability":3,"availability":1]	0.67719257	查看详情 评分选择

本次搜索耗时: 273ms

图 4 系统运行结果实例 2

为进一步验证本文提出的方法性能, 将 DS2 同文献[16]中基于推荐网络和蚁群算法的服务发现方法 ABSDA 以及随机游走方法 (REM, random ergodic matching) 三者进行比较。为保证实验结果客观, 消除噪声数据, 本文构建了一个理想的实验环境, 首先创建了 10 条不同的用户请求, 设定初始候选 Web 服务为 300 个, 然后每次增加 100 个直至可用 Web 服务个数达到 1 200, 最后记录并比较 DS2、ABSDA 和 REM 在不同个数可用 Web 服务情况下分别响应这些服务请求的平均完成时间和平均查准率。这里的完成时间指的是根据解析后的服务请求开始匹配服务直至返回结果列表的时间。查准率 R_C 定义如下

$$R_C = \frac{N_T}{N_T + N_F} \quad (16)$$

其中, N_T 是返回结果中符合用户功能需求和实时偏好的 Web 服务个数; N_F 是不能完全满足用户需求的 Web 服务个数。

图 5 所示为 DS2、ABS DA 和 REM 三者的平均完成时间对比。从图中可以看到, 当候选 Web 服务的数量小于 500 时, DS2 的平均完成时间比 ABS DA 和 REM 都要高。究其原因是因为在候选 Web 服务数目较少时, 针对某些服务请求无法找到满足需求的对象, 这时 ABS DA、REM 会返回空的结果列表; 而 DS2 则会利用用户的信任关系产生一个推荐列表, 其实现逻辑和执行流程要比另外两者算法都复杂, 因此其完成时间也比它们多。

而当候选 Web 服务的数量从 500 开始逐渐增多时, DS2 的平均完成时间始终要低于 ABS DA 和 REM, 而且随着 Web 服务的数量的增大, DS2 的表现愈加稳定, 效率优势体现愈加明显。这是因为随着候选服务数目与用户操作历史条数的增加, 用户的社群关系模型会逐渐清晰与稳定, DS2 在服务发现过程中的针对性也会逐渐增强; 而 ABS DA 和 REM 在原理上都需遍历所有候选服务, 因此需要更多的完成时间。

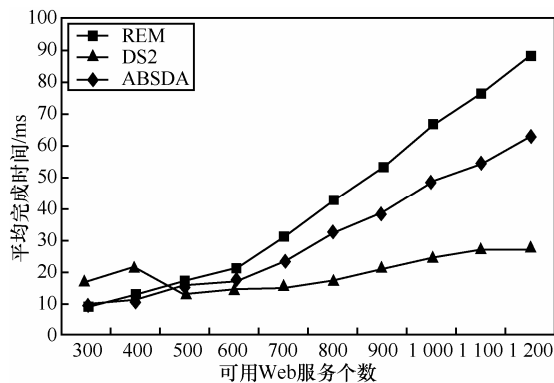


图 5 平均完成时间比较

图 6 所示为三者的平均查准率对比结果。从图中结果可以看到, DS2 的查准率始终要大幅高于 ABS DA 和 REM。这是因为 DS2 按照候选 Web 服务与用户兴趣背景相关性由强到弱的次序来匹配候选服务, 在服务选择阶段考虑到了用户的实时偏好, 因此其检索到的 Web 服务结果要接近用户实际需求。而且根据其算法原理可知, DS2 的查准率稳定性高, 受服务对象数量的影响小。而 ABS DA 和 REM 仅从服务名和服务功能属性上对候选服务进行匹配, 不能保证结果能完全被用户接受, 随服务对象数量的增加, 查准率呈下降趋势。

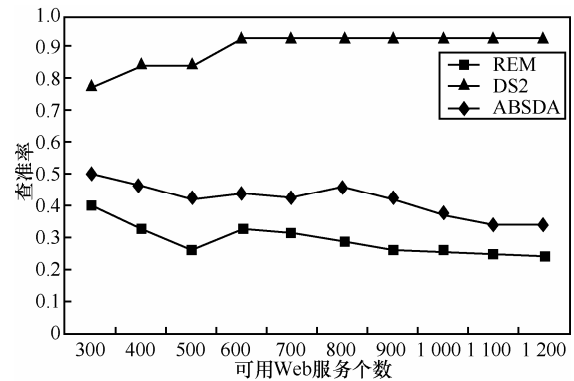


图 6 平均查准率比较

虽然 DS2 表现出了良好的服务发现效率和查准率, 但这是以牺牲查全率指标为代价的。因为在其算法原理上, DS2 找到符合需求的服务就会终止算法流程, 其返回的服务个数一般都比 ABS DA 和 REM 少, 除非利用用户信任关系产生最终结果。

7 结束语

本文为有效利用用户的辅助信息, 提高 Web 服务发现的效率与查准率, 提出了一个基于用户社群关系的 Web 服务发现方法。其核心思想是通过对用户社群关系的抽象与刻画, 在 Web 服务发现过程中依次利用用户社群关系中的偏好关系、聚类关系和信任关系, 按照 Web 服务与用户兴趣背景相关性由强到弱的原则来逐步检索或推荐产生最终结果。为检验方法性能, 本文还构建了原型系统, 开展了验证实验。结果显示, 相比已有类似方法, 本文提出的方法在 Web 服务发现效率和查准率方面具有明显优势且表现稳定。

在后续研究工作中, 如何在保持该方法现有优点的前提下提高服务发现的查全率, 增强用户实时偏好处理的灵活性是需要解决的主要问题。同时也需考虑进一步优化 DS2 算法性能来应对候选 Web 服务数量偏少的情况, 减少此条件下算法的平均完成时间。

参考文献:

- [1] PALIWAL A V, SHAFIQ B, VAIDYA J, *et al.* Semantics-based automated service discovery[J]. IEEE Transactions on Services Computing, 2012, 5(2):260-275.
- [2] BRUNO M, CANFORA G, DI PENTA M, *et al.* An approach to support web service classification and annotation[A]. 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2005) [C]. Hong Kong, China, 2005. 138-143.
- [3] TALANTIKITE H N, AISSANI D, BOUDJLIDA N. Semantic annota-

- tions for web services discovery and composition[J]. *Computer Standards & Interfaces*, 2009, 31(6):1108-1117.
- [4] DI NOIA T, DI SCIASCIO E, DONINI F M. Semantic matchmaking as non-monotonic reasoning: a description logic approach[J]. *Journal of Artificial Intelligence Research*, 2007, 29:269-307
- [5] ALBERTI M, CATTAFI M, CHESANI F, *et al.* A computational logic application framework for service discovery and contracting[J]. *International Journal of Web Services Research*, 2011, 8(3):1-25.
- [6] GARCIA J M, RUIZ D, RUIZ-CORTES A. Improving semantic web services discovery using SPARQL-based repository filtering[J]. *Journal of Web Semantics*, 2012, 17:12-24.
- [7] DIAMADOPOULOU V, MAKRIS C, PANAGIS Y, *et al.* Techniques to support Web service selection and consumption with QoS characteristics[J]. *Journal of Network and Computer Applications*, 2008, 31(2):108-130.
- [8] KRITIKOS K, PLEXOUSAKIS D. Mixed-integer programming for QoS-based web service matchmaking[J]. *IEEE Transactions on Services Computing*, 2009, 2(2):122-139.
- [9] KRITIKOS K, PLEXOUSAKIS D. Requirements for QoS-based web service description and discovery[J]. *IEEE Transactions on Services Computing*, 2009, 2(4):320-337.
- [10] RAJ R, SASIPRABA T. Web service recommendation framework using QoS-based discovery and ranking process[A]. *International Conference on Advanced Computing (ICAC2011)*[C]. Chennai, India, 2011.371-377.
- [11] BROGI A. On the potential advantages of exploiting behavioral information for contract-based service discovery and composition[J]. *Journal of Logic and Algebraic Programming*, 2011,80(1S1):3-12.
- [12] MAAMAR Z, FACI N, WIVES L K, *et al.* Using social networks for Web services discovery[J]. *IEEE Internet Computing*, 2011, 15(4): 48-55.
- [13] MAAMAR Z, WIVES L K, BADR Y, *et al.* LinkedWS: a novel Web services discovery model based on the metaphor of social networks[J]. *Simulation Modelling Practice and Theory*, 2011, 19(1):121-132.
- [14] YU J, SHENG Q Z, HAN J, *et al.* A semantically enhanced service repository for user-centric service discovery and management[J]. *Data & Knowledge Engineering*, 2012, 72:202-218.
- [15] 刘迎春, 郑小林, 陈德人. 基于信任和推荐关系的可信服务发现[J]. *系统工程理论与实践*, 2012, 32(12):2789-2795.
LIU Y C, ZHENG X L, CHEN D R. Trustworthy services discovery based on trust and recommendation relationships[J]. *Systems Engineering -Theory &Practice*, 2012, 32(12):2789-2795.
- [16] 谢晓芹, 宋超臣, 张志强. 一种基于推荐网络和蚁群算法的服务发现方法[J]. *计算机学报*, 2010, 33(11):2093-2103.
- XIE X Q, SONG C C, ZHANG Z Q. A service discovery method based on referral network and ant-colony algorithm[J]. *Chinese Journal of Computers*, 2010, 33(11):2093-2103.
- [17] 田浩. 以用户为中心的 Web 服务发现方法及其在金融服务中的应用研究[D]. 武汉: 武汉大学, 2014.
TIAN H. Research on user-centered Web service discovery and its application in financial services[D]. Wuhan: Wuhan University, 2014.
- [18] NAING M, LIM E, HOE-LIAN D G. Ontology-based Web annotation framework for hyperlink structures[A]. *Web Information Systems Engineering(WISE 2002)*[C]. Singapore, 2002. 184-193.
- [19] UPENDRA S, MAES P. Social information filtering: algorithms for automating “word of mouth”[A]. *Computer Human Interaction (CHI'95)*[C]. Moscow, Russia, 1995. 210-217.
- [20] CANDILLIER L, MEYER F, FESSANT F. Designing specific weighted similarity measures to improve collaborative filtering systems[A]. *Industrial Conference on Data Mining (ICDM2008)*[C]. Pisa, Italy, 2008. 242-255.
- [21] HESS A. Assam WSDL annotator[EB/OL]. <http://andreas-hess.info/projects/annotator/index.html>.
- [22] ZHENG Z B. WS-DREAM[EB/OL]. <http://www.wsdream.net>.
- [23] SemWebCentral. OWLS-TC version 4.0[EB/OL]. http://projects.semwebcentral.org/frs/?group_id=89&release_id=380.

作者简介:



田浩 (1977-), 男, 湖北赤壁人, 博士, 湖北经济学院副教授, 主要研究方向为网络与通信、语义 Web。



樊红 (1967-), 女, 湖南益阳人, 武汉大学教授、博士生导师, 主要研究方向为 GIS 应用、语义 Web 等。

杜武 (1987-), 男, 江西南昌人, 武汉大学博士生, 主要研究方向为语义 Web、并行与分布式系统等。