

基于二部图的服务推荐算法研究

姜 波^{1,2} 张晓筱¹ 潘伟丰^{1,3}

(1 浙江工商大学计算机与信息工程学院, 浙江 杭州 310018; 2 浙江大学计算机科学与技术学院, 浙江 杭州 310007; 3 武汉大学软件工程国家重点实验室, 湖北 武汉 430072)

摘要 通过考虑 Web 服务的兼容性, 提出了一个基于二部图的服务推荐方法(BIGSIR). BIGSIR 方法将 Web 服务及服务之间的关系抽象成一个二部图, 并基于二部图为用户推荐合适的 Web 服务; 使用^{my}Experiment 上的工作流服务和 Web 服务的真实数据来验证方法的有效性. 实验结果表明: 除了一些孤立 Web 服务结点和工作流结点, 执行 BIGSIR 方法的平均排名都在 0.184~0.281 之间, 与 GRM 方法相比具有更好的推荐性能, 同时还揭示了影响算法性能的因素; 针对当历史使用信息不充分时算法的推荐效果不佳的情况, 也提出了相应的拟解决方案.

关键词 服务计算; 兼容性; Web 服务推荐; 二部图; 历史信息

中图分类号 TP393 **文献标志码** A **文章编号** 1671-4512(2013)S2-0093-07

Bipartite graph-based service recommendation method study

Jiang Bo^{1,2} Zhang Xiaoxiao¹ Pan Weifeng^{1,3}

(1 Department of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China; 2 Department of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China; 3 State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

Abstract By taking the compatibility of Web service into full consideration, a Bipartite Graph based Service Recommendation (BIGSIR) method was proposed. BIGSIR adopted a bipartite graph to visual the Web services and the relationship between them. Based on the graph model, an effective recommendation algorithm was introduced to recommend the suitable Web services. Workflows and Web services from ^{my}Experiment were used as subjects to demonstrate the feasibility of the proposed approach. Experimental results demonstrate that apart from some isolated Web services or workflows, BIGSIR can obtain promising results that the average recommendation ranking rate of the recommended nodes is from 0.184 to 0.281. It has a better performance when compared with GRM (Global Recommendation Method). The factors that will influence the performance of BIGSIR are also explored. When the historical information is not such sufficient, the performance of our method will not be so good. The solution to address this problem was also presented.

Key words service computing; compatibility; Web service recommendation; bipartite graph; historical information

云计算是基于互联网的, 它依赖于对互联网上的计算资源进行共享来处理应用程序. 云计算中的资源都是以服务的形式通过互联网来提供给用户共享使用的. 服务是云计算中不可或缺的重要元素之一. Web 服务作为云计算中使用的

重要的服务类型之一, 可以被看作是一个小型的软件系统. 该系统支持计算机与计算机之间通过网络进行协同交互^[1]. 随着云计算的发展, 互联网上 Web 服务的个数不断增长, 且这些海量的服务中有很多具有相似的功能和非功能特性, 这使得

收稿日期 2013-07-25.

作者简介 姜波(1970-), 女, 教授, E-mail: nancybjiang@mail.zjgsu.edu.cn.

基金项目 国家自然科学基金资助项目(61202200).

如何推荐合适的 Web 服务对开发者来说成为了一个巨大的挑战。

近年来,服务质量^[2](quality of service, QoS)因子已经被考虑到 Web 服务推荐领域的研究中。QoS 是 Web 服务非功能特性的集合。还有很多的研究工作都将 Web 服务的 QoS 特性作为重要影响因子进行更好的 Web 服务发现^[3-5]、服务选择^[6-8]和服务推荐^[9-11]。文献[12]对 Web 服务行为的兼容性是否影响 Web 服务无缝集成与协作进行了充分的讨论,并指出 Web 服务的兼容性对于实施服务的动态组合和服务的动态替换具有重要的指导作用。因此,进行 Web 服务推荐相关研究时,必须考虑 Web 服务的可兼容性特性。

现有的推荐技术主要从以下两方面进行研究:a. 从相似度的角度出发,协同过滤方法是在 Web 服务推荐领域的最为成功并被广泛应用的推荐方法之一。

GroupLen^[13]是最早的采用协同过滤方法进行网络新闻推荐的系统。亚马逊^[14]采用基于物品的相似度计算进行商品推荐。类似地,文献[15]采用相似度计算,用历史非功能特性数据计算每两个消费者之间的相似度来进行相关推荐工作。文献[10]也采用了相关的方法,提出一个可扩展的混合协同过滤方法进行个性化 Web 服务推荐。然而,当用户的数目不断增长时,相似度的计算时间会变长,进而影响方法的性能。

b. 从历史信息的角度出发,进行服务推荐的基础是 Web 服务的历史使用信息,即服务的历史调用和组合信息。文献[11]充分利用历史使用信息,为基于服务的应用程序开发者做更好的服务推荐。文献[16]基于历史数据、用户兴趣度进行 Web 服务推荐。然而,当历史数据信息稀疏时,此类推荐方法的性能就会很差。同时,现有的方法忽略了兼容性特征。

选择合适的兼容的 Web 服务有助于开发者缩短开发周期、节约成本,同时提高开发质量。

1 基于二部图的推荐算法

为解决推荐领域存在的不足,本研究提出一个基于二部图的服务推荐方法(BIGSIR),从 Web 服务历史信息^[17]的角度出发,引入二部图^[18-19]的概念,最后列出前 k 个符合条件的 Web 服务列表进行服务推荐。不同于其他的基于历史信息进行推荐的 Web 服务推荐方法,BIGSIR 将历史信息抽象成一个二部图,并充分利用二部图中的结点

和边的信息为用户进行高效的推荐。同时,特别考虑了 Web 服务的兼容性特性。BIGSIR 的步骤描述如下。

步骤 1 数据预处理。本文从 Web 服务搜索引擎上爬取符合实验条件的工作流数据,并对源文件进行解析构建数据集。工作流即可以看作是一个组合的复杂服务。经过分析,本文构建了一个二部图来表示工作流结点和简单 Web 服务结点以及二者之间的组合关系信息。

步骤 2 方法实现。该部分是核心内容,通过方法实现,计算所有的用于推荐的结点资源值。

步骤 3 结果与分析。通过步骤 2 中 BIGSIR 方法计算出所有结点的资源值后,将用于推荐的结点进行排序,通过一定的指标进行相关的结果分析。

1.1 定义

与 BIGSIR 方法相关的定义如下。

定义 1 二部图。二部图是一类特殊的图,图中结点根据特性分为两类,其中被归到同一类中的结点都具有相似的特性。边则连接来自不同类的两个结点,表示结点之间的关联。对二部图的定义为

$$B_G = (W_F, W_S, E). \quad (1)$$

式中: W_F 为图的一类结点的集合, $W_F = \{w_{f_i}\}$, w_{f_i} 为两类结点集合的元素; W_S 为图的另一类结点的集合, $W_S = \{w_{s_i}\}$, w_{s_i} 为两类结点集合的元素; E 为图的边的集合, $E = \{e_{i,j}\}$, $e_{i,j}$ 为集合的元素。

定义 2 二部图邻接矩阵 A 用于表示二部图中的信息,邻接矩阵中的元素 $a_{i,j}$ 定义为

$$a_{i,j} = \begin{cases} 1 & (e_{i,j} \in E); \\ 0 & (\text{其他}) \end{cases} \quad (0 \leq i < m, 0 \leq j < n), \quad (2)$$

式中 m, n 分别表示二部图中两类结点的数量。

定义 3 边的权重指的是边两端的结点之间的关联的重要程度。权重的定义为

$$w_{i,j} = x \quad (x \geq 0), \quad (3)$$

式中权重值 x 的取值由研究人员根据具体的情况定义。本文 x 值表示结点与结点之间的调用关联的重要程度, x 越大表示完成组合服务的任务须要调用的简单服务的次数越多,关联性越强。

定义 4 结点初始化资源值指的是为用于推荐的结点所在的集合的结点赋初值,赋值规则式为

$$r_{w_{s_j}} = \begin{cases} 1 & (a_{i,j} = 1); \\ 0 & (a_{i,j} = 0), \end{cases} \quad (4)$$

式中: $r_{w_{s_j}}$ 为在为结点 w_{f_i} 进行结点 w_{s_j} 推荐时的

初始资源分配值; $a_{i,j}$ 的值为1表示在当前情况下两个结点是相连的,为0表示两个结点互不关联。

定义5 实验中随机去边的数量 R_{EdgeNum} 定义为

$$R_{\text{EdgeNum}} = \alpha E_{\text{dgeNum}} \quad (5)$$

式中: E_{dgeNum} 为当前的边的数量; α 为去边的比率。

定义6 引入排名比率指标作为 BIGSIR 方法的衡量指标.排名比率的计算式为

$$R_{R,i,j} = R_{\text{ank}_{w_{s_j}}} / U_{w_{f_i}} \quad (6)$$

式中: $R_{\text{ank}_{w_{s_j}}}$ 为 Web 服务 w_{s_j} 在用于推荐的服务列表中推荐排名; $U_{w_{f_i}}$ 为与 w_{f_i} 不相连的 Web 服务结点总数。

用 $F_k(R_R)$ 表示一次实验的平均排名比率,计算式为

$$F_k(R_R) = \sum_{i,j}^N R_{R,i,j} / R_{\text{EdgeNum}} \quad (7)$$

1.2 算法流程

BIGSIR 方法的算法流程下。

步骤1 输入带权重二部图 B_G 。

步骤2 初始化二部图的结点 n ,邻接矩阵 a ,资源矩阵 r ,权重矩阵 w 和度矩阵 d 。

步骤3 遍历工作流结点。

步骤4 判断工作流结点是否孤立;若为孤立,则相关资源值置-1,重复步骤3;否则,进行步骤5。

步骤5 进行第一次资源流动分配;通过遍历每一个工作流结点与 Web 服务结点权重,计算累加的 $a[i][j]r[k][j] \times w[i][j]/w[m][j]$ 的值,遍历结束后赋值给 $r[i][n]$ 。

步骤6 进行第二次资源流动分配;通过遍历每一个 Web 服务结点与工作流结点权重,计算累加的 $a[i][j] \times r[i][n] \times w[i][j]/w[i][n]$ 的值,遍历结束后赋值给 $r[k][j]$;重复步骤3,直到结束遍历。

步骤7 返回 $r[][]$ 的值。

由此得到方法的时间复杂度为 $O(mn)$ 。

2 算法实例

2.1 问题描述

通过一个简单的例子来进一步举例说明方法的实现过程.通过对历史信息的解析,采用式(1)和式(3)构建二部图,根据式(4)计算 Web 服务的初始资源值.由此构建并初始化二部图实例如图1所示。

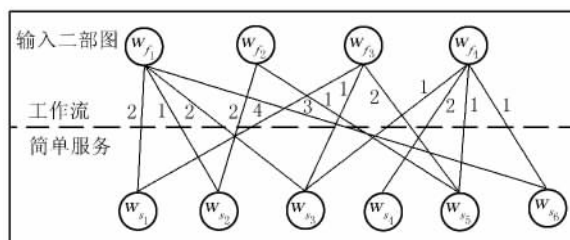


图1 构建并初始化二部图

如图1所示: $w_{1,1}=2$,即表明 w_{f_1} 功能的实现调用了 Web 服务 w_{s_1} 两次.两个结点集合 $W_F = \{w_{f_1}, w_{f_2}, w_{f_3}, w_{f_4}\}$, $W_S = \{w_{s_1}, w_{s_2}, w_{s_3}, w_{s_4}, w_{s_5}, w_{s_6}\}$. 本例的目标是为 w_{f_1} 推荐结点.初始化的结果为 $R = \{1, 1, 1, 0, 0, 1\}$ 。

2.2 算法实现过程

BIGSIR 的核心是资源的两次分配,实现过程实例如图2所示。

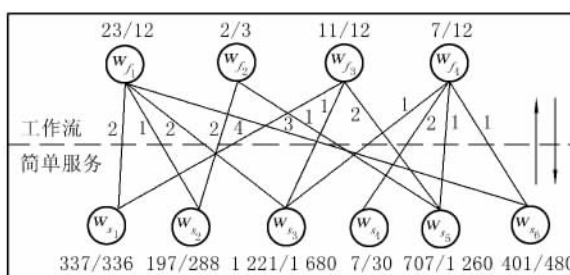


图2 两次源流动分配的过程及结果

a. 进行第一次资源流动分配.将 Web 简单服务层的资源值投影到工作流层,计算工作流结点层每一个结点的资源值.投影规则按照边的权值设定.权值根据式(3)设定.本文通过结点相连的边的权值在所有与结点相连边的权值总和中所占的比例进行资源流动分配.例如本例中 w_{f_1} 资源值计算公式为

$$r_{w_{f_1}} = (2/6)r_{w_{s_1}} + (1/3)r_{w_{s_2}} + (2/4)r_{w_{s_3}} + (3/4)r_{w_{s_6}},$$

其他三个工作流结点的计算原理类似。

b. 资源的第二次流动分配.从 Web 服务结点层投影到工作流结点层,分配的规则同 a.例如 w_{s_1} 的资源值计算公式为: $r_{w_{s_1}} = (2/8)r_{w_{f_1}} + (4/7)r_{w_{f_3}}$,其余 Web 服务结点资源计算原理类似。

通过结点资源的两次流动分配,服务层的所有结点得到了最终资源值.从本例中可以得出与 w_{f_1} 不相连的 w_{s_4}, w_{s_5} 的资源值.通过 BIGSIR 方法后,再根据计算的结果值对 w_{s_4} 和 w_{s_5} 的排名比率根据式(6)进行计算,从而从小到大进行排序.因此,推荐给工作流 w_{f_1} 的 Web 服务列表为 $\{w_{s_5}, w_{s_4}\}$ 。

3 实验及性能分析

3.1 实验

3.1.1 数据来源

为了评估 BIGSIR 方法,采取从^{my} Experiment^[20]上获取数据来构建数据集.在^{my} Experiment 上包含有两类 Web 服务,分别为工作流和简单 Web 服务,工作流即组合服务.^{my} Experiment 中的工作流是通过简单 Web 服务和其他组合服务的调用来协同完成的.这一特性正好满足 BIGSIR 方法的需求.

^{my}Experiment 主要包含了 2 000 个工作流,这些工作流可以分为 10 类,分别为:Bioclipse Scripting Language, BioExtract Server, GWorkflowDL, Kepler^[21], KNIME, LONI Pipeline, RapidMiner, Tesla, Taverna 1, Taverna 2.由于 Taverna 1 和 Taverna 2 两个类型具有比较清晰且完整的文档结构,因此本文重点考虑了这两类工作流,并对其源文件进行解析,共解析出 843 个工作流和 Web 服务.这些 Web 服务可以分为如下几个类型,如 beanshell, rshell, biomart, stringconstant, local, soaplab 和 wsdl.表 1 详细阐述了这些 Web 服务类型的具体含义.

表 1 一些简单的 Web 服务类型的意义

Web 服务类型	意义
beanshell	采用 beanshell 脚本编写,且有相应的依赖库的服务
rshell	在 R 服务器上运行,采用 R 脚本语言编写的服务
biomart	该服务是对数据集查询的业务的封装,这些数据集部署在 biomart 平台上的 mart 类型的服务库中
stringconstant	字符串常量
local	由本地包、类库封装成的服务
soaplab	此类服务为 Soaplab 类型 Web 服务,可以通过相关的接口对其进行访问、调用
wsdl	部署在搜索引擎、服务注册库等平台的 Web 服务,该类服务提供 wsdl 文档供用户进行访问、调用
biomoby	部署在 MOBY 系统中的服务,该类服务用于生物学数据和分析服务的相互操作
workflow	由简单服务组合而成,并被封装起来的复杂 Web 服务

本文实验构建的二部图网络结构如图 3 所示.这个二部图包含 382 个工作流结点、461 个

Web 服务结点和 1 170 条边.内圈结点表示工作流、外圈的结点表示 Web 服务.

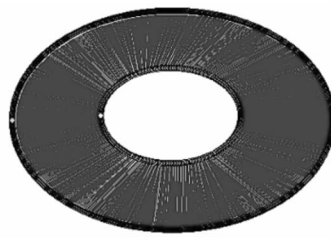


图 3 由工作流和 Web 服务组成的二部图

3.1.2 实验结果

BIGSIR 方法是从历史使用数据的角度出发的,所以本文根据一定的比率随机去除图 3 所示二部图的边来构建二部图的一个历史场景.去除的边的数量由式(5)来计算.

本实验的目标是检验二部图方法是否正确、高效地推荐去除边所连接的 Web 服务结点给相应的工作流结点.为了测试方法的性能,本文设置去边比率 $\alpha=0.1$,且做了 1 000 次实验来验证方法是否高效、稳定.设置 $\alpha=0.1$,表示将 90%的数据作为训练集,剩下 10%的数据作为测试集.训练集中的数据作为已有的 Web 服务结点和工作流的历史数据信息,测试集中的数据是用来预测的、没有被用过的 Web 服务结点.根据去边比率,本实验中训练集中的边的数量为 1 053,测试集中为 117.本实验测试集的选择以及二部图的构建过程共重复了 1 000 次.在每一次实验中,本文计算每一个工作流对应的没有连接的 Web 服务结点的资源值,并求出测试集中的与该工作流相连的结点的排名比率.排名比率的根据式(6)计算,平均排名比率根据式(7)进行计算.累积平均排名比率是本实验衡量方法的性能指标,具体计算见 3.2.1 节.

本文执行了 1 000 次实验,实验结果记录每一次实验 Web 服务结点的平均排名比率以及累积平均排名比率.

实验结果图 4 记录了每一次实验结束后的此次实验结果中所有工作流的平均排名比率 F 从最低的 0.183 到 0.231(平均排名比率最后趋于 0.227).从图中可以看出:随着实验次数的增加,方法的平均排名比率较低,即推荐性能好,且趋于稳定.

实验结果图 5 表示随机抽取的一次实验,去除的边所连的 Web 服务结点的平均排名比率.同时将本方法与主流的推荐方法 GRM(global recommendation method)进行了比较.

实验结果图 6 显示了 1 000 次实验后,在不

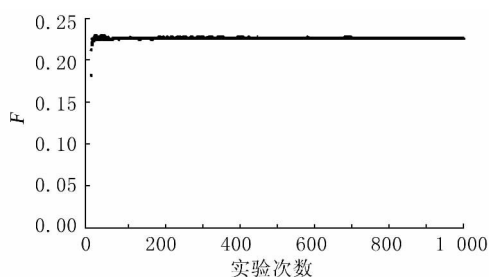


图4 与去除边相连接的 Web 服务结点的平均排名比率

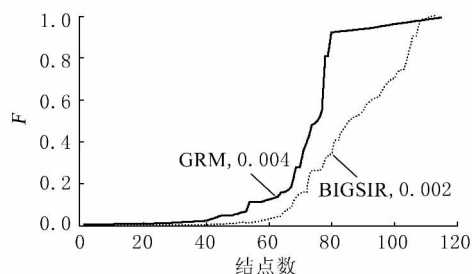


图5 BIGSIR 方法和 GRM 方法一次实验平均排名比率对比

同区间内 BIGSIR 方法和 GRM 方法推荐的服务的准确度. 每一个区间增加的结点格式为 10 个. 由图可以看出: BIGSIR 方法的推荐性能比 GRM 方法要好. 执行 1 000 次实验后, BIGSIR 推荐方法的累积平均排名比率 S_F 为 0.228, 而 GRM 推荐方法为 0.353. 另外, 若将排名比率作为二部图方法的性能评价指标, 则它应该收敛于一个稳定的值. 对平均排名比率的方差 E_F 进行了 1 000 次的计算, 结果如图 7 所示.

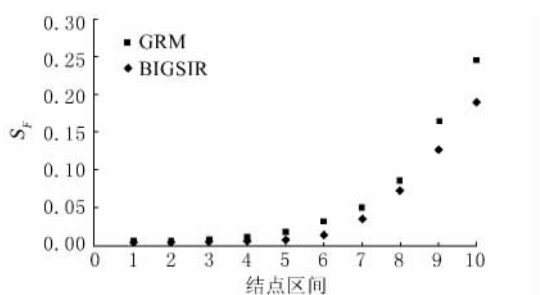


图6 BIGSIR 和 GRM 在不同区间内服务累积平均排名比率

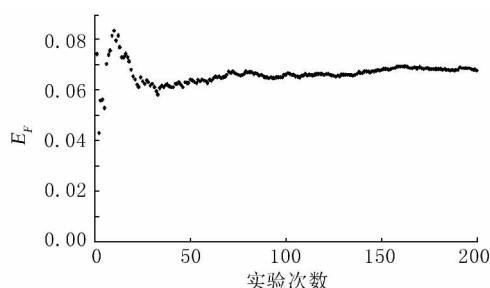


图7 每 5 次实验的平均排名比率方差

3.2 性能分析

3.2.1 性能指标

BIGSIR 方法的性能通过定义 6 排名比率进行衡量, 排名比率越低, 方法的性能越好. 在本文中, 并不是通过单独每一个结点的排名比率来分析的, 而是以每一次实验为单位, 通过计算本次实验的平均排名比率进行分析. 平均排名比率由式 (7) 计算得到. 另外, 引入了累积平均排名比率来衡量实验方法的性能是否稳定, 计算公式为

$$A_{RR} = \sum_{i=0}^{C_{TS}} F_i(R_R), \quad (8)$$

式中: $F_i(R_R)$ 为一次实验的平均排名比率; C_{TS} 为当前为止的实验次数.

3.2.2 结果分析

本文分析实验结果如图 5 所示, GRM 方法的执行效率的执行结果没有 BIGSIR 方法高, 其中排名比率在 $[0.9, 1.0]$ 的结点有 40 个, 而 BIGSIR 方法中属于这种情况的结点只占 11 个. 由图 5 所示, 根据 Web 服务结点特性的分析, 根据平均排名比率的值将 Web 服务结点分成两类.

a. $0 < R_{R,i,j} < 0.046$. 平均排名比率在这一区间的 Web 服务结点具有比较高的推荐效率. 从图 5 可以看出有 61 个结点排名属于这一区间. 在这一区间内的结点比较多, 超过要推荐结点的半数以上, 且它们的平均排名比率都比较稠密地分布在 $[0, 0.046]$ 内. 它们中绝大部分是属于 local 类型的 Web 服务, 也有一部分是属于二级工作流或者 wsdl 类型的 Web 服务. 因此, 本研究可以得到如下结论: local 类型 Web 服务对于工作流开发者来说是非常重要的, 它们能够被很好地重用和共享.

b. $R_{R,i,j} \geq 0.046$. 在这个区间内的 Web 服务结点的排名比率从 0.076 到 1.000. 这些 Web 服务结点平均排名率显得比较离散. 可以从如下方面的原因进行分析: 在除去一些边后, 有些工作流变成了孤立的结点, 即没有与之相连的 Web 服务结点, 这使得 BIGSIR 方法的核心过程资源流动分配无法进行; 这些工作流所调用的 Web 服务可能来自不同的领域, 特定领域的 Web 服务不能很好地被开发者共享应用到其他领域. 从构建的二部图中可以看出: 有很多服务只被某些工作流调用、组合了一次, 因此在采用 BIGSIR 方法进行服务推荐时, 两端结点的资源不能通过边进行流动分配, 然而随着越来越多的开发者将 Web 服务应用到交叉领域, 这种 Web 服务资源不共享的情况被很好地改善了.

为了验证本文提出的排名比率性能指标对本文提出的基于二部图的服务推荐方法衡量的精准性,本文计算并记录了每 5 次实验的平均排名比率的方差,结果如图 7 所示.可以看出:在 200 次实验之前,方差曲线的波动比较剧烈.而经过 800 次实验后,方差收敛于 0.068.本实验的总体方差为 2.13×10^{-5} ,表明本性能是比较稳定的.

5 结论

实验结果表明:除了一些孤立的 Web 服务或 workflow 结点外,BIGSIR 得到的推荐效果都是比较理想的.即 BIGSIR 方法具有比较高的精度,同时也具有很可观的前景.本文的二部图推荐方法还可以应用到服务发现、服务组合领域来进一步提高效率.在服务发现、服务组合中,也会遇到与服务推荐类似的场景:互联网上所需要的服务在功能或者非功能特性上,都存在着很多的相似之处.此时,采用 BIGSIR 方法,通过历史信息从功能或非功能特性上相似的服务中为用户(主要指开发者)推荐更适合、更有可能感兴趣的服务供开发者选择,从而进行更高效地服务发现与组合.

此外,本文避免了其他关于 Web 服务推荐研究相关方法的不足,将 Web 服务的兼容性充分考虑到了本文中.进一步工作将考虑把 Web 服务的相似度加入到本文方法中进一步提高推荐方法的精确度,这样便可以解决当历史使用数据稀疏的情况所导致的方法推荐效果的不理想问题.本文另一个可以扩展的是通过对 workflow 所调用的其他 Web 服务类型的分析来扩展数据集的规模.在实用性方面可以进一步优化的是将该算法扩展应用到 Web 服务组合领域,为组合 Web 服务推荐可组合对象.

参 考 文 献

- [1] Has H, Brown A. Web services glossary[EB/OL]. [2013-01-15]. <http://www.w3.org/TR/ws-gloss>.
- [2] Ran S. A model for Web service discovery with QoS[J]. ACM SIGecom Exchanges, 2003, 4(1): 1-10.
- [3] Al-Masri E, Mahmoud Q H. QoS-based discovery and ranking of Web services[C]//International Conference on Computer Communications and Networks. Honolulu: IEEE, 2007: 529-534.
- [4] Kyriakos K, Plexousakis D. Requirements for QoS-based Web service description and discovery[J]. IEEE Transactions on Services Computing, 2009, 2(4): 320-337.

- [5] Ali R J A, Rana O F, Walker D W, et al. G-QOSM: grid service discovery using QoS properties[J]. Computing and Informatics, 2002, 21(4): 1-15.
- [6] Yu Tao, Zhang Yue, Lin K J. Efficient algorithms for Web services selection with end-to-end QoS constraints[J]. ACM Transactions on the Web (TWEB), 2007, 1(1): 1-26.
- [7] Wang Heichia, Lee Changshing, Ho Tsunghsien. Combining subjective and objective QoS factors for personalized Web service selection[J]. Expert Systems with Applications, 2007, 32(2): 571-583.
- [8] Huang A F M, Lan Ciwei, Yang S J H. An optimal QoS-based Web service selection scheme[J]. Information Sciences, 2009, 179(19): 3309-3322.
- [9] Zheng Zibin, Ma Hao, Lyu M R, et al. QoS-aware Web service recommendation by collaborative filtering[J]. IEEE Transactions on Service Computing, 2011, 4(2): 140-152.
- [10] Chen X, Liu X D, Huang Z Ch, et al. RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized Web service recommendation[C]//IEEE International Conference on Web Services. Miami: IEEE, 2010: 9-16.
- [11] Blanzieri E, D'Andrea V, Giorgini P, et al. Improving Web service discovery with usage data[J]. IEEE Software, 2007, 24(6): 47-52.
- [12] 邓水光,李莹,吴健,等. Web 服务行为兼容性的判定与计算[J]. 软件学报, 2007, 18(12): 3001-3014.
- [13] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]//Computer Supported Cooperative Work (CSCW). New York: ACM, 1994: 175-186.
- [14] Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [15] Zheng Z B, Ma H, Lyu M R, et al. WSRec: a collaborative filtering based Web service recommender system[C]//IEEE International Conference on Web Services (ICWS). Los Angeles: IEEE, 2009: 437-444.
- [16] Kang G Sh, Liu J X, Tang M D, et al. AWSR: active Web service recommendation based on usage history[C]//IEEE International Conference on Web Service (ICWS). Honolulu: IEEE, 2012: 186-193.
- [17] 潘伟丰,李兵,邵波,等. 基于软件网络的服务自动分类和推荐方法研究[J]. 计算机学报, 2011, 34(12): 2355-2369.
- [18] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physi-

- cal Review; E, 2007, 76(4): 1-7.
- [19] 邓水光,尹建伟,李莹,等. 基于二分图匹配的语义 Web 服务发现方法[J]. 计算机学报, 2008, 31(8): 1364-1375.
- [20] de Reoure D, Goble C, Stevens R, et al. The design and realization of the my experiment virtual research environment for social sharing of workflows[J/OL]. [2013-02-15]. <http://eprints.soton.ac.uk/265709/>.
- [21] Kepler. Kepler Web site[EB/OL]. [2013-02-16]. <http://kepler-project.org>.

.....

(上接第 67 页)

- [12] Vrabie M, Ma J, Chen J, et al. Scalability, fidelity, and containment in the potemkin virtual honey farm [J]. ACM SIGOPS Operating Systems Review, 2005, 39(5): 148-162.
- [13] Box G E P, Jenkins G M, Reinsel G C. Time series analysis: forecasting and control [M]. Hoboken: Wiley, 2013.
- [14] Roy N, Dubey A, Gokhale A. Efficient auto scaling in the cloud using predictive models for workload forecasting[C]//2011 IEEE International Conference on Cloud Computing (CLOUD). Athens: IEEE, 2011: 500-507.
- [15] Sobel W, Subramanyam S, Sucharitakul A, et al. Cloudstone: multi-platform, multi-language benchmark and measurement tools for web 2.0[C]//Proc of CCA. Gleacher: CAA, 2008: 1-7.