

## 一种Web Service的服务质量预测方法\*

邵凌霄<sup>1,2</sup>, 周立<sup>1,2</sup>, 赵俊峰<sup>1,2+</sup>, 谢冰<sup>1,2</sup>, 梅宏<sup>1,2</sup>

<sup>1</sup>(北京大学 信息科学技术学院 软件研究所,北京 100871)

<sup>2</sup>(高可信软件技术教育部重点实验室,北京 100871)

### Web Service QoS Prediction Approach

SHAO Ling-Shuang<sup>1,2</sup>, ZHOU Li<sup>1,2</sup>, ZHAO Jun-Feng<sup>1,2+</sup>, XIE Bing<sup>1,2</sup>, MEI Hong<sup>1,2</sup>

<sup>1</sup>(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

<sup>2</sup>(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

+ Corresponding author: E-mail: zhaojf@sei.pku.edu.cn, http://www.pku.edu.cn

Shao LS, Zhou L, Zhao JF, Xie B, Mei H. Web service QoS prediction approach. *Journal of Software*, 2009, 20(8):2062-2073. <http://www.jos.org.cn/1000-9825/3375.htm>

**Abstract:** Consumers need to make prediction on the quality of unused Web services before selecting. Usually, this prediction is made based on other consumers' experiences. Being aware of different QoS (quality of service) experiences of consumers, this paper proposes a QoS prediction approach. This approach makes similarity mining among consumers and QoS data, and then predicts the QoS of the unused Web services from other consumers' experiences. Experimental results show that with this approach the preciseness of QoS prediction for Web services can be improved significantly.

**Key words:** Web service; quality of service (QoS); QoS (quality of service) prediction; similarity

**摘要:** 服务消费者在选择服务之前,通常需要基于其他消费者的经验对未使用过的服务的质量进行预测.考虑到不同服务消费者对同一服务的服务质量的感受之间可能存在较大的差别,提出了一种QoS(quality of service)预测方法.该方法以消费者的历史经验为基础,计算消费者之间以及服务之间的相似程度,并以此相似度为基础对消费者并未使用过的服务的QoS进行预测.实验结果表明,这种方法可以显著提高Web服务质量预测的准确性.

**关键词:** Web服务;服务质量(QoS);服务质量(QoS)预测;相似度

中图法分类号: TP311 文献标识码: A

近年来,互联网上出现了大量Web服务.这些服务以SOAP<sup>[1]</sup>等平台无关的通信协议为消费者提供功能,构成了未来大规模面向服务计算的基础.由于服务开发者众多,存在很多提供相似或相同功能的服务,如何在这些服务中进行合理的选择,一直是研究界广泛关注的问题之一.许多研究者认为,当服务消费者(service consumer)

\* Supported by the National Natural Science Foundation of China under Grant No.60803011 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2007AA010301 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2005CB321805 (国家重点基础研究发展计划(973)); the Science Fund for Creative Research Groups of China under Grant No.60821003 (国家创新研究群体科学基金)

Received 2007-12-07; Revised 2008-01-29; Accepted 2008-04-15

进行服务选择时,不仅应该考虑服务所能满足的功能性需求,而且也应该考虑服务所能满足的非功能需求(non-functional requirement,简称 NFR),即服务质量(quality of service,简称 QoS)需求.服务选择时将服务质量属性(如性能、可用性、可靠性等)考虑在内,对成功地构建面向服务应用(service oriented applications)具有非常重要的意义.因此,研究者们广泛认为,基于服务质量的服务查找与选择(QoS-aware service discovery and selecting)是保证基于服务的应用系统质量的重要技术之一<sup>[2-7]</sup>.

在大量服务广泛存在、候选服务众多的情况下,消费者在服务信息不完全情况下进行选择是普遍存在的<sup>[8]</sup>.也就是说,在众多候选服务中,单个消费者很可能仅仅使用过很小一部分候选服务.例如,10 个部署在互联网上的订票服务,消费者可能仅使用过两三个.同时,由于付费等因素的影响,消费者也不太可能对所有未使用过的服务逐一试用.在这样的情况下,消费者在选择服务之前,不可避免地需要对未使用过的服务的 QoS 进行预测.

在信息不完全的情况下,提高预测准确性的一个广泛使用的方法是复用其他消费者的使用经验.一个常用的机制是计算其他所有消费者提交的服务 QoS 数据的算术平均值.这种机制虽然简单且易于实现,但并没有考虑到消费者对 QoS 感受的差异.根据经验,消费者对相同服务的 QoS 的感受通常存在差异.这种差异是由多种因素造成的,如网络环境、编程语言、运行平台及主观感受的不同等.简单地对所有消费者提交的 QoS 数据进行平均而忽略消费者个体因素,极大影响了预测的准确性.

为了解决这个问题,我们曾在文献[9]中提出了一种基于协同过滤的个性化 QoS 预测方法:以消费者的历史经验为基础计算消费者之间的相似程度,根据相似消费者之间的经验对某消费者个体需要使用的服务进行个性化的 QoS 预测.然而,文献[9]中提出的方法存在一定的不足:在相似度计算中使用的 Pearson 相关系数一般用于度量数据之间的线性相关关系,并不能很好地解释客观 QoS 数据之间的相似性.此外,其实验验证方法还有待完善.针对上述不足,本文改进了相关的相似度计算算法,并基于此相似度技术方法提出了新的预测方法.同时,本文加强了实验验证方法.实验结果表明,相对于作者以往的工作以及其他方法,本文提出的方法可以进一步提高 QoS 预测的准确性.

本文的主要贡献包括:

- (1) 提出了一种基于欧几里德距离的 QoS 信息相似度计算方法;
- (2) 基于此相似度的 QoS 预测方法及其验证.

本文第 1 节介绍问题产生的背景.第 2 节阐述本文提出的 QoS 数据预测方法.第 3 节介绍本文的实验设计方案.第 4 节给出实验数据的分析.第 5 节是相关工作的比较.第 6 节给出本文得到的结论中对未来的工作进行展望.

## 1 背景

在大量的基于服务的项目的开发过程中我们发现,当前,大多数实现了基于 QoS 的服务查找功能的 UDDI 系统(如 UX<sup>[10]</sup>,UDDIe<sup>[11]</sup>等)提供的 QoS 数据都是所有个体消费者提交的 QoS 数据的平均值(如响应时间、可用性等).这种使用平均值的计算方法虽然在一定程度上反映了 QoS 数据的客观性,但没有将消费者的个体因素考虑在内,而这些特征,如网络环境、编程语言、工具平台等,都对消费者感受到的服务的 QoS 造成很大的影响.

比如,消费者 A 向某支持 QoS 信息的 UDDI 系统请求查询一个提供网页搜索功能的 Web Service, UDDI 中已发布有 3 个提供该项功能的服务 1~服务 3,同时存放有消费者 A 及消费者 B~消费者 E 提交的针对这 3 个服务的 QoS 信息(如图 1 所示(数据中第 1 项表示服务响应时间,第 2 项表示服务可用性)).A 曾经尝试过使用 Service 1 和 Service 2,但未使用过 Service 3.目前的 UDDI 系统的实现在提供这种数据时普遍采用计算平均值的方式,即计算所有消费者已提交数据的平均值.这样,对 Service 3 而言, UDDI 描述其 QoS 为:平均响应时间  $(2.34+3.0+0.9)/3=3.12s$ ,平均可用性为  $(0.92+0.88+0.82)/3=3.12$ .

然而对 A 而言,这样由计算算术平均数所得到的数据不能很好地反映其可能感受到的 QoS.比如说,由于某些因素(如处于同一局域网中),相对于其他消费者, C 可能与 A 感受到的 QoS 比较接近.因此, A 在预测 Service 3 的服务质量时, C 提交的数据可能相对更为重要.因此,一种相对合理的计算方法是,根据消费者提交的 QoS 数

据,从所有消费者中找出与 A 在使用服务方面的经验较为相似的消费者,然后在对 A 所使用服务的 QoS 进行预测时,赋予这些相似消费者的 QoS 数据以较高的权重,从而从整体上提高预测 Web Service QoS 的准确程度。

|        | Service 1  | Service 2  | Service 3   |
|--------|------------|------------|-------------|
| User A | 0.4s, 0.92 | 1.6s, 0.8  |             |
| User B | 2.8s, 0.93 | 3.5s, 0.98 |             |
| User C | 0.9s, 0.77 | 0.6s, 0.85 | 2.34s, 0.92 |
| User D | 1.0s, 0.56 | 8s, 0.78   | 3.0s, 0.88  |
| User E | 0.8s, 0.9  |            | 0.9s, 0.82  |

Fig.1 QoS data stored in UDDI

图 1 UDDI 中保存的 QoS 数据

## 2 QoS数据预测方法

本方法基于以下基本思想:根据消费者提供的 QoS 数据,找出这些数据之间的相似关系;在需要对某消费者未使用过的服务的 QoS 进行预测时,参考与其历史 QoS 数据最为接近的消费者及服务的 QoS 数据值进行计算。

目前关于相似度挖掘有许多技术,如聚类<sup>[12]</sup>、基于内容的过滤<sup>[13]</sup>和协同过滤<sup>[14]</sup>等.这些技术中,协同过滤在解决许多问题上取得了良好的效果,且与本文的场景具有非常类似的问题空间.在我们以往的工作中,采用了标准的协同过滤技术对 QoS 数据进行预测并取得了一定的效果<sup>[9]</sup>.然而在进一步研究中我们发现,标准协同过滤技术对 QoS 数据预测不完全适合:基于协同过滤的系统往往用于对主观数据进行分析,其相似度计算方法通常只能用于度量消费者数据之间存在的线性关系;而在本文的目标问题中,消费者提交的是客观 QoS 数据,这些数据之间很可能存在非线性的数据相似关系.为解决这样的问题,本文引入了基于欧几里德距离的相似度计算方法.同时,基于该方法所得到的相似度,提出了相应的预测方法。

本文提出的方法主要包括 4 个步骤:

- 1) 数据预处理.
- 2) 正规化(normalization).
- 3) 相似度挖掘.
- 4) Web Service QoS 预测.

数据预处理用于将收集到的 QoS 数据进行初始化并剔除噪声数据;正规化用于将处于不同范围内的 QoS 数据转化到一个统一的范围.基于上述数据,本文使用欧几里德距离度量消费者之间的相似度和服务之间的相似度,并提出了一种基于此相似度对某消费者未使用服务的 QoS 数据进行预测的方法。

### 2.1 问题定义

为简化后续的描述,本文将待解决的问题进行如下的形式化:

1.  $S=\{s_1, s_2, \dots, s_m\}$  是一个具有相同功能的服务集合,  $s_i$  表示第  $i$  个服务.
2.  $U=\{u_1, \dots, u_l\}$  是服务消费者的集合,  $u_i (1 \leq i \leq l)$  表示第  $i$  个服务消费者个体.
3.  $Q_{i,j} = \langle p_{i,j}^1, p_{i,j}^2, p_{i,j}^3, \dots, p_{i,j}^n \rangle, p_{i,j}^k \in R \cup \{\emptyset, ?\}$ .

$Q_{i,j}$  是一个向量,表示消费者  $u_i$  感受到的服务  $s_j$  的 QoS.  $n$  为所需度量的 QoS 属性个数.  $p_{i,j}^k$  为  $u_i$  感受到的服务  $s_j$  的第  $k$  个 QoS 属性的值.  $p_{i,j}^k$  的值可以为实数值,  $\emptyset$  或者 ?. 当消费者对于某 QoS 属性没有提交数值时,该值表示为  $\emptyset$ ; 若  $p_{i,j}^k$  标为 ?, 则表示该项数值为目标预测对象。

1.  $D_i = \langle Q_{i,1}, Q_{i,2}, \dots, Q_{i,k} \rangle$  是一个向量,指代从消费者  $u_i$  获得的 QoS 数据;
2.  $T = \langle D_1, D_2, \dots, D_n \rangle$  是包含所有消费者 QoS 数据的一个向量.

由此,图 1 中定义的数据可以格式化为图 2,而本文所需解决的问题则根据图 2 提供的数据对  $Q_{3,1}$  进行预测。

|       | $s_1$                        | $s_2$                                  | $s_3$                                  |
|-------|------------------------------|--|--|
| $u_1$ | $\langle 0.4s, 0.92 \rangle$ | $\langle 1.6s, 0.8 \rangle$            | $\langle ?, ? \rangle$                 |
| $u_2$ | $\langle 2.8s, 0.93 \rangle$ | $\langle 3.5s, 0.98 \rangle$           | $\langle \emptyset, \emptyset \rangle$ |
| $u_3$ | $\langle 0.9s, 0.77 \rangle$ | $\langle 0.6s, 0.85 \rangle$           | $\langle 2.34s, 0.92 \rangle$          |
| $u_4$ | $\langle 1.0s, 0.56 \rangle$ | $\langle 8s, 0.78 \rangle$             | $\langle 3.0s, 0.88 \rangle$           |
| $u_5$ | $\langle 0.8s, 0.9 \rangle$  | $\langle \emptyset, \emptyset \rangle$ | $\langle 0.9s, 0.82 \rangle$           |

Fig.2 Formatted QoS data

图 2 格式化后的 QoS 数据

2.2 数据预处理

$D_i = \langle Q_{i,1}, Q_{i,2}, \dots, Q_{i,k} \rangle$  指代从消费者获取的 QoS 数据.在本文提出的方法中,QoS 数据的预测是针对单个属性进行的,即针对响应时间、可用性等属性单独进行预测.在数据预处理步骤中,我们将获取的数据转化为如下的向量:

- $D_i^k = \langle p_{i,1}^k, p_{i,2}^k, p_{i,3}^k, \dots, p_{i,m}^k \rangle$  表示从消费者  $u_i$  收集到的第  $k$  个属性的所有数据, $m$  表示服务的个数.
- $T'_k = \langle D_1^k, D_2^k, \dots, D_t^k \rangle$  是一个向量,表示从所有消费者收集到的关于第  $k$  个属性的 QoS 数据.其中, $t$  表示服务消费者的数目.
- $T' = \langle T'^1, T'^2, \dots, T'^l \rangle$  是一个向量,表示收集到的所有 QoS 数据.其中, $l$  表示 QoS 属性的数目.
- $T'$  是转化完成时 QoS 数据的形式.

举例来说,对图 2 中的 QoS 数据,按单个 QoS 属性划分的结果如图 3 所示.

|       | $s_1$ | $s_2$       | $s_3$       |
|-------|-------|-------------|-------------|
| $u_1$ | 0.4s  | 1.6s        | ?           |
| $u_2$ | 2.8s  | 3.5s        | $\emptyset$ |
| $u_3$ | 0.9s  | 0.6s        | 2.34s       |
| $u_4$ | 1.0s  | 8s          | 3.0s        |
| $u_5$ | 0.8s  | $\emptyset$ | 0.9s        |

(a) Response time  
(a) 响应时间

|       | $s_1$ | $s_2$       | $s_3$       |
|-------|-------|-------------|-------------|
| $u_1$ | 0.92  | 0.8         | ?           |
| $u_2$ | 0.93  | 0.98        | $\emptyset$ |
| $u_3$ | 0.77  | 0.85        | 0.92        |
| $u_4$ | 0.56  | 0.78        | 0.88        |
| $u_5$ | 0.9   | $\emptyset$ | 0.82        |

(b) Availability  
(b) 可用性

Fig.3 Data after format transformation

图 3 转换格式后的数据

2.3 正规化

数据预处理之后,QoS 数据已经按照 QoS 属性的不同进行了划分.一般而言,涉及到的 QoS 数据的类型可以分为数值型或者比率型.比率型的数据在一个有限的范围内变化,比如 0~100%.数值型数据可能在许多不同的范围内变化.一个典型的数值类型的属性是响应时间,对于消费者 A,它可能在[0,1]s 内变化;而对于消费者 B,它可能变化范围是[10,20]s.

数据正规化的作用在于,将处于不同区域的数据映射到一个统一的区间之中.对于数值类型的数据,若不进行正规化处理,则处于不同区域的数值对计算消费者的相似度的贡献将会产生差别.比如两组数据:1s 和 1.1s,数值差为 0.1s;10s 和 11s,数值差为 1s.虽然两组之间的数值差相差很大,但是相差的比率都是 10%.根据比率计算消费者之间的相似度显然比直接根据数值计算相似度更为合理.因此在计算相似度之前,需要对数值型数据进行正规化.

本文采用高斯法<sup>[15]</sup>对数据进行正规化.该方法可以将数据映射到[0,1]的区间之中.相对于其他正规化方法,如极值正规化法,该方法能够更好地避免异常值(特别大或特别小的数值)的影响.高斯正规化方法在本方法中的表达式为公式(1).其中,  $\bar{p}_i^k$  表示消费者  $u_i$  对所有服务在第  $k$  个 QoS 属性上感受到的平均值,  $\sigma_i$  表示数据的标准差.在公式(1)中,  $3\sigma_i$  是因为  $3\text{-}\sigma$  规则.该规则表明,对于高斯正规化而言,如果取值为  $3\sigma$  可以使 99% 的数据映射到[0,1]的区间之内<sup>[15]</sup>.对于正规化之后处于[0,1]之外的数据,可以赋值为 0 或者 1(根据其值与哪一端更为接近

而定).

$$p_{i,j}^k = 0.5 + \frac{p_{i,j}^k - \bar{p}_i^k}{2 \times 3\sigma_i} \quad (1)$$

## 2.4 相似度挖掘

在消费者缺少某服务的 QoS 的情况下,一个比较直观的办法就是复用其他消费者感受到的该服务的 QoS 信息.然而,由于消费者个体之间的差异性,一般而言,需要寻找那些差异较小的消费者所提交的 QoS 数据进行复用.在以往的工作中,我们提出了基于协同过滤的 QoS 预测方法<sup>[9]</sup>.该方法使用 Pearson 相关系数<sup>[16]</sup>度量消费者相似度.然而,Pearson 相关系数主要利用消费者数据与其均值的偏离程度衡量不同消费者间的相关关系.该方法在衡量主观数据的线性相似度时有很好的效果.随着进一步的研究我们发现,QoS 数据作为一种客观数据,其数据之间的相关关系可能并非线性的,用这种基于均值偏离度量的线性相似度可能并不合理.

同时,在进一步的研究中我们发现,不仅有些消费者之间的数据存在一定的相似性,而且有些服务之间的数据也存在一定的相似性.即在所度量的服务全体中,若干服务的质量可能比较接近.因此,若从服务 QoS 数据之间相似性的角度出发进行度量,则很有可能提高预测结果的准确性.

本文提出了一种基于欧几里德距离的相似度度量方法.该方法将  $T$ (收集到的所有 QoS 信息,见第 2.1 节)中每一行(列)的 QoS 数据视为空间中的一个点,这样,两点间的相似度可以由两点间的距离来反映.两点间的距离越大,表示两点间相似度越低;反之,则相似度越高.基于欧几里德距离计算的基本算式为公式(2).

$$w_{k,i} = \frac{1}{\left( \sum_{i=1}^n (X_i - Y_i)^2 / n \right)} \quad (2)$$

在公式(2)中, $X$ 与 $Y$ 都是向量,与图3(a)中的一行(或一列)相对应. $X_i, Y_i$ 分别代表向量 $X$ 与 $Y$ 的第 $i$ 维数据. $n$ 代表向量的维度. $w_{k,i}$ 为计算得到的相似度.

这种基于欧几里德距离的方法虽然较为直观,但是它需要解决维度缺失的问题,即如何处理原始数据中缺失的数据问题.倘若直接忽略这些数据而进行计算,则很有可能导致那些数据缺失较多的向量与其他向量之间具有非常高的相似度,这样显然会影响相似度计算的准确性.

为了避免这种由于数据维度缺失带来的计算误差,我们给出一种处理缺失数据的方法,这种方法以相邻数据的平均值作为填充数据.以图3(a)的数据为例,需要计算消费者 $u_1$ 与其他消费者的 QoS 数据之间的相似度.由于 $p_{1,3}^1$ 的数据未知(图中标为?),需要先对其进行处理.我们的方法是,将其他所有消费者在对此服务的数值的平均值填入此未知数据中(如将消费者 $u_3, u_4, u_5$ 的数据 $p_{3,3}^1, p_{4,3}^1, p_{5,3}^1$ 的平均值填入 $p_{1,3}^1$ 后进行计算).

## 2.5 QoS预测

承前所述,设待预测的数据为 $p_{a,i}^r$ ,即消费者 $u_a$ 对服务 $s_i$ 感受到的其在第 $r$ 个 QoS 属性的值.公式(3)是标准的协同过滤公式,这是一种基于 $p_{a,b}^r (b \neq a)$ 与其所在向量的均值的偏差预估 $p_{a,i}^r$ 的方法.也就是说,根据其他消费者在服务 $s_i$ 上的 QoS 数据与其均值的偏差,预估消费者 $u_a$ 在此服务上感受到的 QoS 与 $u_a$ 在其他服务上感受到的 QoS 的均值之间的偏差.

$$p_{a,i}^r = \bar{p}_a^r + \frac{\sum_{u=1}^n (p_{u,i}^r - \bar{p}_u^r) \times w_{a,u}^r}{\sum_{u=1}^n w_{a,u}^r} \quad (3)$$

在我们以往的工作中<sup>[9]</sup>,曾经指出该公式的两个不合理之处:

- 1) 在 $n=1$ 时,该公式无法得到有效值.
- 2) 对 $w_{a,b}^r$ 的符号没有分正负分别考虑(在标准的协同过滤中, $w_{a,b}^r$ 可以取正值或者负值).

然而,以往的工作中提出的修正方法取得的效果并不显著,其原因在于,协同过滤算法一般用于对主观数据的预测,而对客观数据(如本文中的 QoS 数据)并不完全适用.

本文提出的方法与标准协同过滤方法和我们在以往的工作中提出的方法<sup>[9]</sup>有所不同,是针对在第 2.4 节提出的相似度挖掘基础上的一种计算方法——基于消费者之间的 QoS 相似度和服务之间的 QoS 相似度进行 QoS 预测.这种方法的好处在于,同时考虑了上述两种相似度,并具有直观且易于实现的优点.综合来说,我们将第 2.4 节公式(2)中计算得到的相似度作为权重,从消费者角度计算得到的预测值是对其他消费者针对待预测服务提交的 QoS 数据的加权平均.其计算公式如下:

$$U_{k,j} = \sum_{i=1}^n X_i w_{k,i} \quad (4)$$

其中: $U_{k,j}$ 表示从消费者角度对消费者  $u_k$  可能感受到的服务  $s_j$  的质量的估计值; $X_i$ 表示消费者  $u_i$  感受到的服务  $s_j$  的质量; $w_{k,i}$ 表示消费者  $u_k$  与  $u_i$  的相似度(由公式(2)得到),在公式(4)中表示加权平均式中的权重.

同样,我们可以从服务角度预测其质量数据,计算公式如下:

$$S_{k,j} = \sum_{i=1}^n X_i w'_{i,j} \quad (5)$$

其中: $S_{k,j}$ 表示从服务角度对消费者  $u_k$  可能感受到的服务  $s_j$  的 QoS 的估计值; $X_i$ 表示消费者  $u_k$  感受到的服务  $s_j$  的 QoS; $w'_{i,j}$ 表示服务  $s_j$  与  $s_i$  之间的相似度,在公式(5)是加权平均式中的权重.

最后,我们需要将从两方面计算得到的数据进行加和,如公式(6), $Q_{k,j}$ 即为我们最后的预测值.

$$Q_{k,j} = (U_{k,j} + S_{k,j}) / 2 \quad (6)$$

### 3 实验设计

我们设计了以下的实验以验证本文提出的方法.该实验设计的目标是模拟真实情况下服务消费者对 Web 服务的调用.在该设计中,我们尽量避免干涉消费者设定调用服务的参数.这样的设计尽可能地反映实际应用中 QoS 数据的真正情况.

#### 3.1 服务消费者

我们对部署在 Internet 上真实的 Web Services 开发了客户端程序,并将这些客户端程序分发给志愿者.这些志愿者的地理位置大多数都在亚洲,另外有一些在美国和欧洲.这些志愿者使用不同的操作系统,如 Windows 和 Linux,并运行在不同的网络环境,如校园网、公司和广域网.

为了模拟更多服务的消费者,我们为每个志愿者提供了一个代理服务器列表(不同志愿者的代理服务器名单存在少量的重复).这些代理服务器的地址都是从 Internet 上获取的,我们并不关注这些代理服务器的确切地理位置和运行状况.志愿者可以自行在代理服务器列表中增加或者删除.运行在志愿者机器上的客户端程序能够自动地在这些代理服务器之间切换,并通过这些代理服务器调用 Web Services.对于每一个独立的(Volunteer, Proxy)组合,我们将其视为一个独立的服务消费者.使用这样的方法,可以在志愿者并不太多的情况下模拟大量服务消费者的情况.

服务消费者数据由 28 个志愿者和 63 个代理服务器产生,平均每个志愿者都试用了 5 个左右的代理服务器(那些长期不可访问的代理服务器的记录被自动删除).我们收集了 136 名服务消费者对 20 个真实部署在互联网上的 Web 服务的调用记录,这些 Web 服务一部分是由 Google, Yahoo!, Amazon 等公司发布,另一部分是由一个 UDDI 注册中心 www.xmethods.net 发布.整个数据收集过程耗时 3 周,总共收集了约 40 万条调用记录(平均每个消费者、服务对提交约 200 条调用记录).通过这些调用记录,我们得到了一个类似于图 2 的数据表格作为初始数据.

客户端程序都采用 Java 语言开发(JDK 1.5),SOAP 引擎使用的是 Apache 1.4.志愿者可以设定这些客户端运行时的参数.第 3.2 节详细描述了这些参数的设定情况.

#### 3.2 参数设定

在本文设计的实验中,有两种类型的参数:1) 服务调用参数,这些参数用于定义 SOAP 消息中传递的参

数;2) 客户端程序运行参数,用于指明客户端程序运行的时间、频率等。

服务调用参数的数据类型包括字符串、整数和浮点数等。为了减少志愿者手动设置参数的负担,客户端程序能够自动生成上述类型的参数。当参数为整数或浮点型时,客户端程序将随机生成这些数据。虽然本文提出的方法和实验方案并不依赖于志愿者的偏好,但是我们仍然希望字符串型参数能够在某种程度上反映志愿者的部分使用偏好。对字符串参数的自动产生,是由客户端程序自动读取消费者以往访问的网页中的内容得到的。

运行参数在志愿者运行客户端程序之前设置。这些参数包括:

- 1) 客户端程序在每天的什么时刻开始执行。
- 2) 两次客户端程序执行之间的时间间隔。
- 3) 客户端程序每次运行时调用服务的次数。

### 3.3 QoS属性的采集与计算

调用记录定义为一个向量  $R, R=(IPc;IPp;sn;st;et;input;output)$ 。其中,  $IPc$  指代志愿者的 IP 地址,  $IPp$  指代理服务器的 IP 地址,  $sn$  指代被调用服务的名称,  $st$  指代 SOAP 请求发送的起始时刻,  $et$  指代 SOAP 请求处理结果从服务端返回的时刻,  $input$  和  $output$  是客户端程序发送和接受的 SOAP 消息。具有相同  $IPc$  和  $IPp$  标识的调用记录被认为是从同一服务消费者发送的数据。

本实验中考虑两个 QoS 属性:平均响应时间和服务可用性。响应时间是指服务消费者发出消息和接收到响应之间的时间间隔<sup>[17]</sup>。对每条记录,其响应时间的计算公式为  $et-st$ 。假设  $p_{i,j}^{rep}$  表示消费者  $u_i$  度量服务  $s_j$  获得的平均响应时间,  $p_{i,j}^{rep}$  的计算是调用记录中得到的响应时间的算术平均值<sup>[4]</sup>。

可用性定义为服务可供使用的概率。可用性的计算公式可以归纳为基于访问次数的和基于时间区间的。考虑到在服务计算领域服务消费者对服务的访问不可能非常频繁,因此本文采用的可用性的计算公式是基于访问次数的,具体定义为公式(7)<sup>[18]</sup>:

$$Ava_{i,j}=(N_a-N_f)/N_a \quad (7)$$

在此公式中,  $Ava_{i,j}$  表示的是消费者  $u_i$  感受到的服务  $s_j$  的可用性,  $N_a$  表示  $u_i$  总共曾经对服务进行调用次数,  $N_f$  表示未能访问到服务的次数。

## 4 实验分析

### 4.1 验证方案

我们将本文提出的基于欧几里德距离的服务预测方法(以下简称 OD)与两种其他方法进行了比较:

- (1) 使用算术平均值进行预测的结果(以下简称 AVG);
- (2) 使用我们以前在文献[9]中提出的算法得到的结果(以下简称 CF)。

使用平均值方法进行预测的公式如下:

$$Average = \frac{\sum_{i=1}^n p_{i,j}^r}{(n-n_e)} \quad (8)$$

此公式中,  $p_{i,j}^r$  的含意与上文相同,表示第  $i$  个消费者对第  $j$  个服务在第  $r$  个 QoS 属性上的度量值;  $n$  表示所有的服务消费者的数量,  $n_e$  表示这些服务消费者中未对服务  $s_j$  未提交数据的个数。

验证方案的基本思路是,将获取的 QoS 数据中的一部分数据排除在外(下文中将此部分数据称为真实值),然后使用保留下来的数据预测这些被排除的数据,根据不同方法的预测值与真实值之间的相近程度判断不同方法之间的优劣。本实验中,我们从获取的初始 QoS 数据中选择  $x\%$  的数据( $x$  以 5 为步长,从 50 增长到 90)。为防止一次随机排除的偶然性对预测结果的干扰,对于每一个固定的  $x$ ,此排除-预测的过程重复进行 10 次。

在每一次重复中,每一个被排开的位置都有一个唯一的编号  $i$ (假设该位置表示了消费者  $u_k$  感受到的服务  $s_k$  的质量)。对于每一个这样的位置,一次预测完成后记录以下几个数值:

- (1)  $rd_i$ :此位置的真实值(也就是消费者  $u_k$  实际感受到的服务  $s_k$  的质量).
- (2)  $pd_{1i}$ :使用本文提出的方法得到的预测值.
- (3)  $pd_{2i}$ :使用扩展的协同过滤方法得到的预测值.
- (4)  $ad_i$ :使用平均值方法得到的预测值.

由此,我们可以得到在每一个被排开位置上的真实值与几种不同的预测值之间的差值——亦即每种预测方法在该点上的误差值.设共有  $n$  个位置的数据被排开,某一被排开的位置编号为  $i$ ,  $i$  位置的真值为  $rd_i$ ,某种预测方法在  $i$  位置得到的预测值为  $pd_i$ .在文献[9]中,我们提出了一种基于误差值之和的验证标准,即,设在第  $i$  个点上,第  $k$  种预测方法( $1 \leq k \leq 2$ )的误差值  $e_k^i = |pd_{ki} - rd_i|$ ,则第  $k$  种方法在第  $r$  次重复时,在所有待预测点上的平均误差值为  $E_k^r, E_k^r = \sum_{j=1}^n e_k^j$ .  $E_k^r$  值越大,说明其总体的误差越大.然而,这种基于误差值之和的预测方法有其不足之处:某种方法可能在大多数待预测的点上都优于其他方法,然而由于在少数点的误差过大,导致其在最终的“平均误差值”大于其他方法.

本文增加了一个指标——相对误差排名,以比较各种不同的预测方法的效果.将 3 种预测方法在某一点  $i$  上的误差值进行排序,误差最小者记为 1,其次记为 2,最次记为 3.最后统计每种方法获得名次 1 的次数作为相对误差排名的指标.本指标用于比较 3 种方法在预测点个数一定时接近于真实值的次数.此指标可以很好地弥补误差值之和这一指标的不足.设第  $k$  种方法在第  $r$  次重复下得到的指标为  $C_k^r$ ,表示该方法在当次循环中获得排名第一(误差最小)的总次数.

最后,将在某一固定的  $x$  值下重复 10 次分别得到的指标进行算术平均,即得到我们最后的验证指标,即误差值之和指标:  $E_k = \sum_{r=1}^{10} E_k^r / 10, C_k = \sum_{r=1}^{10} C_k^r / 10$ .

## 4.2 实验结果分析

图 4(a)和图 4(b)是对 3 种不同方法相对误差排名的比较.图中的横轴代表初始数据缺失的比率,纵轴代表某种方法在各个预测点上总共获得第一名的次数.从中我们可以看出,本文提出的方法和扩展的协同过滤方法,其预测相对成功的次数都远远大于基于平均值的预测结果.当数据排除率小于 80%时,本文提出的方法在该指标上的上升趋势明显好于算术平均方法和协同过滤方法;当数据排除率大于 80%时,由于大量数据被排开,本文提出的方法的预测效果下降得较快.在预测响应时间时,当排开率达到 90%时,其结果与扩展的协同过滤方法计算得到的结果较为接近.在预测可用性时,当排除率达到 85%时,本文提出的方法的预测效果均不及平均法和协同过滤方法.

在数据缺失较多时,本文方法的效果下降是由于目前实验使用的数据量并不庞大;在排除数据较多时,剩余的数据已不能很好地反映消费者和服务的特征.从以上分析可以看出,在数据缺失率在 85%以下时,本文提出的方法和扩展的协同过滤方法都比目前普遍采用的算术平均法有较大的提高;但在数据缺失过多时,本文提出的方法在预测可用性时不如协同过滤方法有效.

图中在数据排开率大于 80%时,由于大量数据被排开,大量消费者的数据不足以被预测(我们要求每个消费者至少提交对 3 个服务的数据,才能对其未使用的服务 QoS 进行预测).因此在数据排开率大于 80%时,总的可预测数据点的个数反而会小于排开率在 75%时.

图 5 分别是对以上 3 种不同的方法在对响应时间和可用性进行预估时的误差值之和的比较.从此图我们可以看出,在大多数情况下,本文提出的方法比扩展的协同过滤方法有较大的提高;而协同过滤方法比平均值方法也有一定的提高.当数据排除量大于 85%时,本文提出的方法准确率下降较多.

以上分析说明,本文提出的方法在大多数情况下优于广泛使用的算术平均方法,并且能够对我们在以往工作中采用的扩展协同过滤方法有很大的优化.我们有理由相信,本文提出的相似度挖掘算法及相应的 QoS 预测方法可以很好地弥补协同过滤算法在预测客观数据数值上的不足.



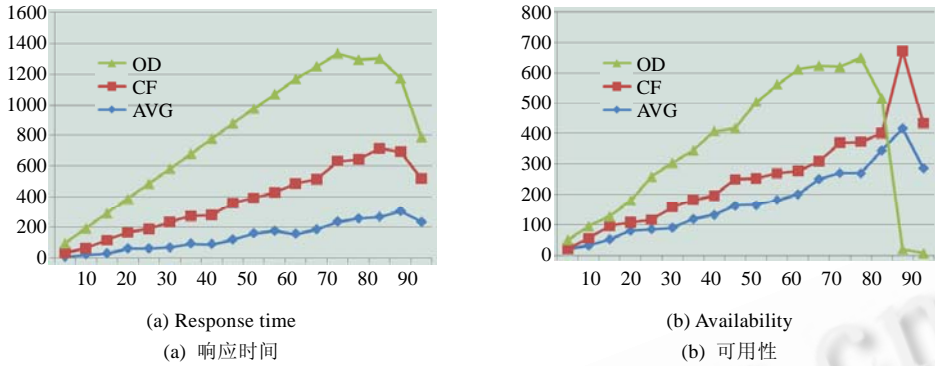


Fig.4 Relatively error rank for three prediction methods

图 4 3种预测方法的相对误差排名比较

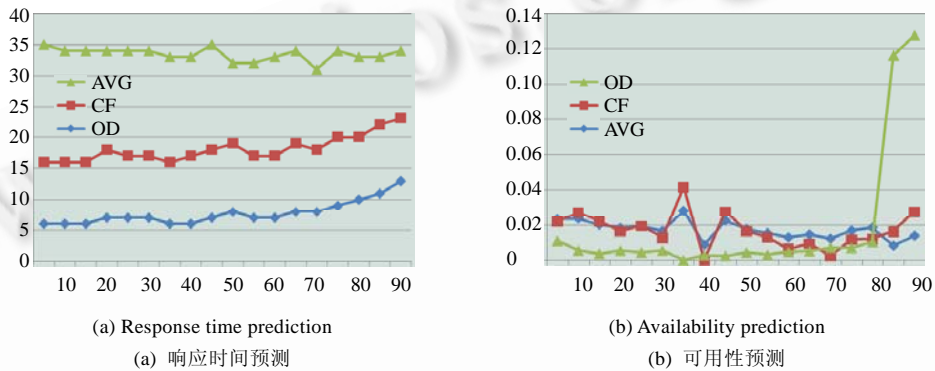


Fig.5 Average error rank for three prediction methods

图 5 3种预测方法的平均误差

### 5 相关工作

基于 QoS 的服务选择与发现是研究界的热门话题之一,许多研究者已对此提出了相关的框架和解决思路.本文的工作建立在这些研究的基础之上.Ran 曾提出扩建 UDDI 注册中心以支持基于 QoS 信息的服务发现<sup>[19]</sup>,在这项研究中,RAN 扩展了 UDDI 中的 tModel 来描述 Web 服务的 QoS 信息,并提出了一系列扩展接口以支持服务发现.UDDIe<sup>[11]</sup>是 Cardiff 大学提出的一个扩展 UDDI 注册中心,在 UDDIe 中,QoS 信息被描述为 business service 的扩展属性(UDDI 数据模型的核心元素之一).同时,UDDIe 扩展了 UDDI 原有的标准接口以支持基于 QoS 的服务发现和 QoS 信息发布.Zhou 等人提出了 UX 模型以扩展的 UDDI<sup>[10]</sup>.UX 与以往研究最显著的区别在于,QoS 数据是在服务消费者与服务交互时动态采集的.杨胜文等人提出了一种支持 QoS 约束的服务发现模型.该模型对 UDDI 中的 Web 服务的 QoS 信息进行约束,并支持基于此约束的服务方法.其模型将一般的 Web 服务 QoS 模型扩展到了 Operation 级,并在 UDDI 体系结构上进行了一些扩展<sup>[7]</sup>.

本文提出的预测机制是对上述服务发现方面的研究的一种补充.用户在使用上述基于 QoS 的服务查找系统时,可以使用本文提出的预测机制,以提高用户查询结果的准确率.

本文在验证过程中用到了若干已有的服务质量度量方法,如 Zeng 等人提出的使用算数平均的方法计算响应时间等<sup>[4]</sup>.在已有研究工作中,除了上述使用的方法以外,Doshi 等人还提出了一种考虑服务质量动态变化的 QoS 度量方法<sup>[5]</sup>.该方法在服务质量发生变化时,能够较快地得到相对准确的预测值.对已采集到调用记录的服务的 QoS 的度量是本文预测机制的基础.但本文的预测机制不依赖于具体的度量方法,即本文的预测机制并不

是针对某一特定的度量方法才能进行.因此,本文的 QoS 属性计算是采用较为简单且被广泛认可的方法<sup>[4]</sup>,对已采集到调用记录的服务的 QoS 进行度量.

许多学者提出,服务选择应将用户偏好考虑在内.Balke 和 Wagner 提出了将用户使用模式、需求以及用户偏好综合在一起的服务查询方案<sup>[20]</sup>.Balke 还进一步提出了基于用户偏好的 Web 服务协同发现机制<sup>[21]</sup>.Maamar 等人提出了关于 Web Service 交互的模型<sup>[22]</sup>,除了用户偏好以外,这一工作还突出了 Web Service 交互的资源.这些工作的重点在于提供一个机制以形式化服务消费者的偏好、历史、情景和服务提供商的资源.Balke 和 Maamar 等人提出的这些机制都是基于预设的语义模型和本体模型,建立这样的一些模型需要很大的手工工作.本文提出的方法是基于消费者采集得到 QoS 的数据进行预测,即自动收集消费者与服务之间的交互信息,在这一点上与上述方法有很大的不同.

协同过滤是一种广泛使用的信息检索方法.协同过滤有许多成功应用的案例,如向消费者推荐新闻、电影、书籍和音乐产品<sup>[23]</sup>.Jonathan 等人提出了一种协作过滤算法框架并提出了一些优化机制<sup>[14]</sup>.本文的工作基于 Jonathan 提出的算法框架.从我们调研到的情况来看,目前尚无研究工作提出使用协同过滤技术预测 Web 服务 QoS 数据.同时,本文对普通的协作过滤模型进行了较大的扩展(包括预处理、相似度计算和预测技术).

Maximilien 对 Web Service 的声誉提出了一个概念模型<sup>[24]</sup>.在此模型中,用户可以对服务评级,并与其他消费者分享他们的经验.Kokash 提出了一种基于用户以往经验的服务发现方法.这一方法将服务和消费者之间的交互形式化为隐式文化,以此向新注册的服务消费者提供很好的服务推荐<sup>[17]</sup>.这些研究工作从消费者的喜好方面发掘相似性,同样需要很多手工性工作.本文提出的方法是从 QoS 数据中挖掘用户相似性,这些 QoS 数据是客观的,并且可以从服务与消费者的大量交互中自动获取.

本文作者曾在文献[9]中提出了一种基于协同过滤的个性化 QoS 预测方法:以消费者的历史经验为基础计算消费者之间的相似程度,根据相似消费者之间的经验对某消费者个体需要使用的服务进行个性化的 QoS 预测.然而,文献[9]中提出的方法存在一定的不足:在相似度计算中使用的 Pearson 相关系数一般用于度量数据之间的线性相关关系,并不能很好地解释客观 QoS 数据之间的相似性.本文提出的方法是对作者以往的工作的进一步优化,具体体现在:

- (1) 改进了以往消费者线性相似度的计算方法,并较好地回避了消费者数据较少时对相似性计算产生干扰的问题;
- (2) 基于上述相似度计算的预测方法;
- (3) 弥补了以往工作在验证上的欠缺.

实验结果表明,相对作者以往的工作以及其他方法,本文提出的方法可以进一步提高 QoS 预测的准确性.

## 6 结论和未来的工作

Web 服务 QoS 预测是 Service 选择中的一个非常重要的步骤.本文提出了一种基于相似度挖掘的 QoS 预测方法.该方法以消费者的历史经验为基础计算消费者之间以及服务之间的相似程度,以此相似度为基础对消费者并未使用过的 QoS 进行预测.实验结果表明,这种方法可以显著提高 Web Service QoS 预测的准确性.

本文仅对两个 QoS 属性进行了验证,而实际上有大量非常重要的 QoS 属性(如可靠性等)并未涉及到.本文的后续工作包括对这些重要属性的验证.同时,本文所采集数据客观上仍然属于模拟数据(主要是由于部分运行参数是由程序自动生成,而不是用户具体指定),这样得到的数据可能与实际的消费者调用 Web 服务所产生的数据有一定的差异.本文的后续工作还包括采集大量真实的交互数据以验证方法的可行性.

**致谢** 本文的撰写得到张路副教授和潘颖博士的悉心指导,在此我们表示衷心的感谢!

## References:

- [1] W3C. SOAP specification. 2007. <http://www.w3.org/TR/soap/>

- [2] Bilgin AS, Singh MP. A DAML-based repository for QoS-aware semantic Web service selection. In: Jeckle M, Zhang LJ, eds. Proc. of the Int'l Conf. on Web Services. Orlando: IEEE Computer Society, 2005. 368–375.
- [3] Yan YH, Liang Y. Composing business processes with partial observable problem space in Web services environments. In: Proc. of the Int'l Conf. on Web Services. Salt Lake City: IEEE Computer Society, 2006. 541–548.
- [4] Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. *IEEE Trans. on Software Engineering*, 2004,30(5):311–327.
- [5] Doshi P. Dynamic workflow composition using Markov decision processes. In: Proc. of the Int'l Conf. on Web Services. Chicago: IEEE Computer Society, 2004. 368–375.
- [6] Yang SW, Shi ML. A model for Web service discovery with QoS constraints. *Chinese Journal of Computers*, 2005,28(4):589–594 (in Chinese with English abstract).
- [7] Hu CM, Huai JP, Sun HL. Web service-based grid architecture and its supporting environment. *Journal of Software*, 2004,15(7): 1065–1073 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/1065.htm>
- [8] Yue K, Wang XL, Zhou AY. Underlying techniques for Web services: A survey. *Journal of Software*, 2004,15(3):428–442 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/428.htm>
- [9] Shao LS, Zhang J, Wei Y, Zhao JF, Xie B, Mei H. Personalized QoS prediction for Web services via collaborative filtering. In: Proc. of the Int'l Conf. on Web Services. Salt Lake City: IEEE Computer Society, 2007. 439–446.
- [10] Zhou C, Chia LT, Silverajan B, Lee BS. UX: An architecture providing QoS-aware and federated support for UDDI. In: Zhang LJ, ed. Proc. of the Int'l Conf. on Web Services. Las Vegas: IEEE Computer Society, 2003. 171–176.
- [11] ShaikhAli A, Rana OF, Al-Ali R, Walker DW. UDDIe: An extended registry for Web services. In: Zhang LJ, ed. Proc. of the 2003 Symp. on Applications and the Internet Workshops. IEEE Computer Society, 2003. 85–89.
- [12] Jain AK, Murty MN, Flynn PJ. Data clustering: A review. *ACM Computing Surveys*, 1999,31(3):264–323.
- [13] Marko B, Yoav S. Fab: Content-Based, collaborative recommendation. *Communications of the ACM*, 1997,40(3):66–72.
- [14] Herlocker JL, Konstan JA, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In: Proc. of the 22nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'99). New York: ACM Press, 1999. 230–237.
- [15] Ortega M, Rui Y, Chakrabarti K, Mehrotra S, Huang TS. Supporting similarity queries in MARS. In: Proc. of the ACM Multimedia. 1997. 403–413.
- [16] Pindyck RS, Rubinfeld DL. *Econometrics Models and Economic Forecasts*. México: McGraw-Hill, 2000.
- [17] Kokash N, Birukou A, Andrea VD. Web service discovery based on past user experience. Technical Report, DIT-06-098, Informatica e Telecomunicazioni, University of Trento, 2007.
- [18] Mikic-Rakic M, Malek S, Medvidovic N. Improving availability in large, distributed component-based systems via redeployment. In: Proc. of the 3rd Int'l Working Conf. on Component Deployment (CD 2005). LNCS 3798, Berlin: Springer-Verlag, 2005. 83–98.
- [19] Ran S. A model for Web services discovery with QoS. *ACM SIGecom Exchanges*, 2003,4(1):1–10.
- [20] Balke WT, Wagner M. Cooperative discovery for user-centered Web service provisioning. In: Zhang LJ, eds. Proc. of the Int'l Conf. on Web Services. Las Vegas: IEEE Computer Society, 2003. 399–404.
- [21] Balke WT, Matthias W. Towards personalized selection of Web services. In: Proc. of the Int'l World Wide Web Conf. New York: ACM Press, 2003. <http://www.l3s.de/apis/paper/www03.pdf>
- [22] Maamar Z, Mostefaoui SK, Mahmoud QH. Context for personalized Web services. In: Proc. of the 38th Annual Hawaii, Int'l Conf. on System Sciences. IEEE Computer Society, 2005. 166.2.
- [23] Zeng C, Xing CX, Zhou LZ. A survey of personalization technology. *Journal of Software*, 2002,13(10):1952–1961 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/13/1952.htm>
- [24] Maximilien EM, Singh MP. Toward autonomic Web services trust and selection. In: Proc. of the 2nd Int'l Conf. on Service Oriented Computing. New York, 2004. 212–221.

附中文参考文献:

- [6] 杨胜文,史美林.一种支持 QoS 约束的服务发现模型.计算机学报,2005,28(4):589-594.
- [7] 胡春明,怀进鹏,孙海龙.基于 Web 服务的网格体系结构及其支撑环境研究.软件学报,2004,15(7):1065-1073. <http://www.jos.org.cn/1000-9825/15/1065.htm>
- [8] 岳昆,王晓玲,周傲英.Web 服务核心支撑技术:研究综述.软件学报,2004,15(3):428-442. <http://www.jos.org.cn/1000-9825/15/428.htm>
- [23] 曾春,邢春晓.个性化服务技术综述.软件学报,2002,13(10):1952-1961. <http://www.jos.org.cn/1000-9825/13/1952.htm>



邵凌霜(1979-),男,湖北武汉人,博士生,主要研究领域为构件技术,软件复用.



谢冰(1970-),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件复用,软件工程.



周立(1982-),男,硕士,主要研究领域为构件技术,软件复用.



梅宏(1963-),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,软件复用,软件构件技术,分布对象技术.



赵俊峰(1974-),女,博士,副教授,CCF 会员,主要研究领域为软件复用,软件工程.

www.jos.org.cn