基于定性多用户偏好的 Web 服务选择

周 宁,谢俊元
(1.计算机软件新技术国家重点实验室,江苏南京 210093; 2.南京大学计算机科学与技术系,江苏南京 210093)

摘 要: 随着 Web 服务数量逐渐增多,出现了许多服务提供者提供功能相同或相似服务的情况.如何在一些功能相似或相同的服务集合中,根据用户对服务质量的需求选出更符合用户要求的服务,是当前研究人员普遍关心的课题.本文提出一种定性的服务选择方法,用来解决涉及多个用户的 Web 服务选择问题.各用户对服务质量的需求被表示为偏好.提出的算法将能够综合考虑各个用户的偏好,使选出的服务尽可能满足大部分用户的偏好要求,并给出了一个实例说明该算法的应用.实验结果表明了算法的有效性.

关键词: Web 服务; 服务质量; CP-net; 偏好逻辑
中图分类号: TP311.52 **文献标识码:** A **文章编号:** 0372-2112 (2011) 04-0729-08

Select Web Services Based on Qualitative Multi-users Preferences

ZHOU Ning, XIE Jun-yuan
(1. *State Key Laboratory for Novel Software Technology, Nanjing, Jiangsu* 210093, *China*;
2. *Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*)

Abstract: Current solutions on web services selection based on preferences mainly consider single user's preferences, moreover, most of them are quantitative methods. However, many users propose different preferences on QoS attributes of web services in real applications. On the other hand, in many domains it is desirable to assess such preferences in a qualitative rather than quantitative way. This paper focuses on QoS-based qualitative service selection according to multi users' preferences. We propose an algorithm based on CP-net. The algorithm can deal with Web service selection in terms of multi users' preferences. Experimental results indicate that this method can obtain an optimal outcome which closely satisfies most users' preferences and show the performance of the method.

Key words: Web services; quality of services; CP-net; preference logic

1 引言

基于 QoS(Quality of Services, QoS)的服务选择已经得到越来越多的关注和研究[3,8].服务选择可以采用两种方法:定量的方法和定性的方法.定量的方法适用于用户对于服务的各种属性(如响应时间、价格、可用性等)及各属性的权重等能够给出具体数值的情况.然而在大多数情况下,这种要求对用户来说可能是苛刻的,因为用户可能自己都不知道该如何给定这些值才更符合自己的要求,他们更愿意表达自己对于各种属性的喜好关系,比如"在其它条件相同的情况下,我偏好于价格甚于响应时间".如何捕获用户的这种偏好关系并将其应用于服务的选择,以选择出更符合用户偏好的服务,这就是本文所要解决的主要问题.这种定性的方法在服务计算领域还缺乏深入的研究.其在人工智能和决策论领域有一些相关的研究,但主要集中于单用户情形,对于多用户的应用尚需要进一步的研究.但实际的情况是:在涉及多个用户的应用中,各个用户往往会提出不同的偏好需求,有时用户给出的偏好信息是不完全的,甚至有可能存在冲突.比如:某个 Web 服务有 A、B、C、D,E,F 六个属性,用户可能存在这样的偏好需求:"在其它条件相同的情况下偏好 A 甚于 B;在其它条件相同的情况下偏好 B 甚于 C;在其它条件相同的情况下偏好 C 甚于 A",用户只对属性 A、B 和 C 提出了自己的偏好,即用户的偏好是不完全的,可能导致服务选择时某些服务无法比较其优劣,而且这里用户对 A、B、C 的偏好序列间接构成环路,即偏好出现了冲突,在这种情况下是没有服务能满足用户偏好要求的,那么我们应该如何解决偏好序列的冲突问题呢? 以及如何综合考虑所有用户的偏好需求,选出尽可能满足大部分用户偏好需求的服

收稿日期:2009-11-24;修回日期:2010-04-14
基金项目:国家自然科学基金(No.60875038,No.60721002,No.60503021);江苏省科技支撑计划(No.BE2009142);教育部重点项目基金(No.108151)

务呢? 现有的只考虑单个用户偏好的方法和只考虑用户线性偏好序列的方法都不能解决这个问题.

我们在 mCP-nets[2]思想的基础上,提出相应的算法来解决这些问题.

本文的主要贡献在于:(1)使用了一种定性的方法来表示各个用户的偏好,其不需要用户给出具体的数值;(2)提出了多用户不完全偏好情形下的推理算法;(3)提出了用户的偏好序列中存在冲突时的处理方法和相应的算法;(4)扩展了 F. Rossi 等人提出的 Rank 机制,提出了 Subrank 机制;(5)给出了一个实例说明以上方法如何用于实际的服务选择中.

2 应用场景

为什么在很多情况下需要表示和推理多个用户的共同偏好呢? 一个典型场景是群用户决策,即多个用户进行服务选择,每个用户都描述他们的偏好,如果用一个 Agent 代表一个用户,那么这些 Agent 相互之间就会协作或竞争,最后就用户提出的偏好达成一致,然后根据达成一致的偏好去寻找那些最能满足所有用户偏好的服务或服务组合.

在该研究领域,一个很吸引人的现实例子就是企业信息管理系统.在这个场景中,在企业或各种部门广泛应用的可能是数据存储和数据访问服务.这些服务需要满足不同企业或部门的要求.例如,一个跨国企业的分公司在和总部进行联合市场销售活动时,需要访问位于总部的相关数据,则每个分公司都会对服务的 QoS(Quality of Services)表达自己的偏好,如分公司 A 可能偏爱安全性甚于其他属性(例如响应时间、价格等),而分公司 B 可能更关心服务的响应时间,其他分公司也可能偏爱其他不同的 QoS 属性(例如准确性,可靠性等).如果各个分公司不能说服其他的分公司,那么将没有合适服务能够满足所有的分公司的偏爱要求.而且,有些分公司出于某种原因考虑提出的偏爱要求可能是不完全的,即对服务的一部分属性描述其 QoS 要求,还有,由于各分公司并不是该领域的专家用户,因此各分公司基于自己的目标对服务提出的偏爱要求可能会出现冲突的情况.于是面临的问题是:在用户的偏爱要求不完全和可能出现冲突的情况下,应如何从大量功能相同或相似的 Web 服务中选择出尽可能满足大部分用户 QoS 偏好的服务呢? 本文中提出的方法将试图解决此类问题.

3 背景

3.1 服务质量(QoS)

在本文中我们所研究的偏好逻辑主要是针对 Web 服务 QoS 属性的偏好,参照大多数公开发表的研究资料[8],Web 服务主要考虑以下 QoS 属性:(1)响应时间(2)服务价格(3)可用性(4)可靠性(5)完整性(6)安全性.

我们在实验中采用的 QoS 属性兼有该服务所属专业领域的属性和上述的 QoS 通用属性,其数据是随机生成的,用数据库来模拟具体的 Web 服务.由于每个 QoS 属性的评估标准不同,取值范围也不一样,为了简化对本文实例中服务 QoS 属性的评估,实验中将按需要做一定的规格化处理.

3.2 CP-nets

CP-net(Conditional preferences networks)是一种以图形化的形式直观的表示用户偏好的方法.和其它表示方法相比,CP-net 方法更加直观和容易理解.简单的定义 CP-net 如下[1]:

定义 1 CP-net 是变量 $V=\{X_1,\cdots,X_n\}$ 上的有向图 G,其每个顶点 $X_i\in V$ 对应一个条件偏好表 $CPT(X_i)$.每个条件偏好表 $CPT(X_i)$ 定义了在给定 $Pa(X_i)=U$ 上的一个定值 u 时 $Dom(X_i)$ 上的偏序关系 $\geq_u^i$,$Dom(X_i)$ 是 X_i 的值域.

其中:$X_i\in V$ 的值域为:$Dom(X_i)=\{x_1^i,\cdots,x_n^i\}$.

$Pa(X_i)$是能够影响变量 X_i 取值的所有父变量的集合.

根据 $Pa(X_i)$可以构造出基于集合 V 的 CP-net:对于每一个结点 $X_i\in V$,使 $Pa(X_i)$中的结点成为其直接父结点,并且使一条有向边从 $Pa(X_i)$指向 X_i.

对集合 $X\subseteq V$ 的一个指派 x(或称为 X 的一个实例化)是为 X 中每个元素指定其值域内的某个值;对集合 $X\subseteq V$ 的所有指派记为 $Asst(X)$.如果 $X=V$,则 x 称为完全指派,否则 x 称为部分指派.

使 $Pa(X_i)=U$,对于每个 $u\in Asst(U)$,我们假设 $\geq_u^i$ 是 $Dom(X_i)$ 上的偏序关系:对任意 $x,x'\in Dom(X_i)$,则必有下列关系之一成立:$x\geq_u^i x'$,$x'\geq_u^i x$.每个结点 X_i 都有一个条件偏好表(CPT),其指明在给定集合 $U=Pa(X_i)$的一个指派 u 时变量 X_i 上的偏序关系 $\geq_u^i$.

定义 2 设 N 是变量集合 V 上的 CP-net,对于 $X_i\in V$,U 是 X_i 的父结点集合,令 $Y=V-(U\cup\{X_i\})$.$uxy\in Asst(V)$ 是任意一个结果(完全指派),这里 $x\in Dom(X_i)$,$u\in Asst(U)$,$y\in Asst(Y)$.结果 uxy 对应于变量 X_i 的一个改进(退化)是结果 $ux'y$,如果 $x'\geq_u^i x(x\geq_u^i x')$.(注如果在给定 u 的情况下 x 已经是 X_i 上的最优解,则在 X_i 上已不存在改进).

CP-net 可以捕获并表示用户的偏好.比如如下的偏好:我喜欢吃肉(M_{mc})甚于吃鱼(M_{fc});当我吃肉的时候我喜欢喝鱼汤(S_f)甚于喝菜汤(S_v);当我吃鱼的时候我喜欢喝菜汤(S_v)甚于喝鱼汤(S_f);当喝鱼汤时我喜欢喝

白酒(W_w)甚于喝红酒(W_r);当喝菜汤的时候我喜欢红酒(W_r)甚于喝白酒(W_w).可以用如图 1 所示的 CP-net 表示用户的偏好关系.

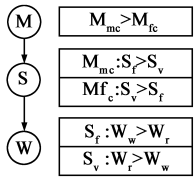


图1　CP-net

3.3　mCP-nets

一个 CP-net 代表一个用户的偏好,对于每个用户可以使用 CP-net 表示和推理一个用户的偏好.把几个 CP-net 放在一起(也称为 mCP-nets)来表示和推理多个用户的偏好.

4　投票语义

本文提出一种基于 Rank 语义的算法来推理多用户的偏好,其是投票语义机制的一种,用于推理 mCP-nets 表示的多用户的偏好.虽然 F.Rossi 等人[5]第一次提出将 Rank 投票语义机制应用于 mCP-net 中,但他们只是给出了其概念,并没有提出具体的算法[5].本文实现并扩展了投票语义机制来表示和推理基于服务 QoS 属性的多用户偏好,其能够找到一组服务集合,使得其能够最大限度地满足大部分人的需要(事实上已经有人证明多用户偏好的推理结果不可能是完全公平的,它只能满足大部分人的需求[4]).下面将介绍 Rank 投票语义机制的关键概念和对其进行的扩展.

给定一个 mCP-net 的两个结果 α 和 β,让 $S_{\succ}$,$S_{\prec}$,$S_{\approx}$和 $S_{\bowtie}$分别代表 $\alpha\succ\beta$,$\alpha\prec\beta$,$\alpha\approx\beta$,$\alpha\bowtie\beta$(分别表示 α 优于 β,β 优于 α,α 和 β 具有相同优先性,α 和 β 不具有可比性).

Rank　如果一个结果是某 CP-net 中的最优结果,则该结果的 rank 值为 0,否则该结果的 rank 值是在该 CP-net 的导出偏好图中该结果到最优结果的最短路径.$\alpha\succ_r\beta$(α rank 优于 β)当且仅当 α 在所有 CP-net 中的 rank 值之和小于 β 在所有 CP-net 中的 rank 值之和.如果二个结果 rank 值之和相同,则这二个结果具有相同的优先关系.任何二个结果要么具有相同优先关系,要么一个优于另一个.

本文对 Rank 投票语义机制进行了一些扩展,即增加子 Rank(sub-Rank)的概念.我们首先使用 rank 投票语义机制将所有的元素分配到不同的集合中,记作 $E=(E_1,\cdots,E_n)$,在同一个集合中的所有结果都具有相同的 Rank 值,那么我们就可以根据集合的 rank 值来选择结果.

5　偏好的表示和推理技术

首先用 CP-nets 表示单个用户的不完全偏好,所有用户的 CP-nets 构成 mCP-nets,当然,这些偏好可能存在冲突.在多人多偏好中,必须综合考虑每个人的偏好,通过协商、折中每个用户偏好的办法找到一个或一组能够最大限度满足大部分用户偏好的结果,下面将介绍服务选择的执行过程和算法.

5.1　偏好的表示

令 $V=(X_1,X_2,\cdots,X_n)$是 mCP-nets 中 n 个变量的集合.每个变量 X_i 的值域为 $Dom(X_i)=\{v_1^i,\cdots,v_m^i\}$.一个可能的结果,记作 w,是对 mCP-net 中的所有变量 X_i 指派其定义域 $Dom(X_i)$中的某个值.令 Ω 是所有可能结果的集合.

5.2　服务选择的执行框架

服务选择的整个执行框架如图 2 所示.

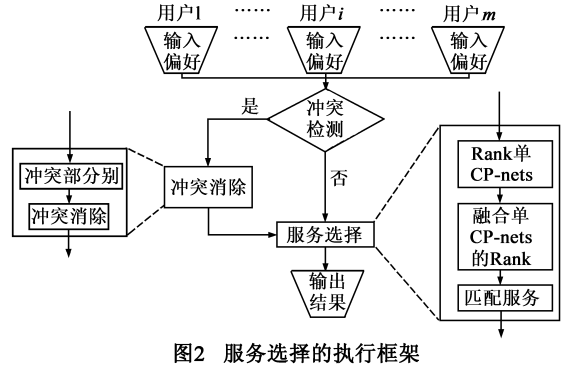


图2　服务选择的执行框架

首先由 m 个用户(Agent)分别输入其对服务的偏好,我们用 CP-nets 建模用户的偏好,得到 m 个用户的 CP-nets 及其偏爱导出图,然后对每个 CP-nets 及其偏爱导出图进行冲突检测,如果某个 CP-nets 及其偏爱导出图存在环路(必须是 CP-nets 和偏爱导出图同时存在环路,原因上面已分析),即出现了偏爱冲突,则须进行冲突消除,先识别偏爱导出图中涉及冲突的环路,然后用下文设计的算法消除环路(算法详见 5.3.1),冲突消除后便可进入到无冲突不完全偏爱的服务选择阶段(算法详见 5.3.2),首先使用 rank 投票语义机制将每个 CP-nets 所有的元素分配到不同的集合 E_i 中,那么我们就可以根据集合的 rank 值来选择结果,然后应用 Trade-off CP-net(tCP-net)的概念,根据每个用户偏好序列中偏好变量的重要程度可以得到集合 E_i 中所有元素的偏好优先序列,就可以对 E_i 中的每一个元素计算其相应的 SRank 值,进而排序之;在用 Rank 及 SRank 语义机制计算并排序好每个 CP-nets 的 E_i 之后,就可以利用 mCP-nets 的思想融合多个用户的的结果而得到 E_i 的最终排序结果;最后根据最终的排序结果匹配服务,选择出能最大程度满足多用户的服务或服务集.

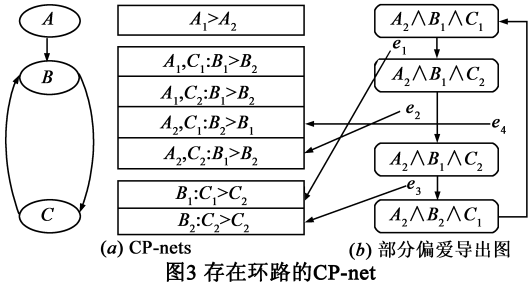
5.3　偏好的推理技术

首先检测用户提出的偏好是否会产生冲突,如果产生冲突则须消除冲突,若无冲突,则可直接进入无冲突不完全的服务选择阶段.

5.3.1 冲突消除

为了从 CP-net 中移除冲突,首先需要检测用户的

偏好序列是否存在冲突.由于已经有一些算法可以检测 CP-net 中是否存在冲突[6,7],系统将直接重用它们来识别冲突.如果用户的偏好序列中存冲突,系统将进入冲突移除阶段.假定应用场景中的数据存储服务的部分 QoS 信息可由如下六个属性描述:A:服务平台,取值为 $A1$:数据库系统,$A2$:文件系统;B:服务位置,取值为 $B1$:美国,$B2$:中国;C:服务提供者,取值为 $C1$:私立的,$C2$:公立的;D:响应时间,取值为 $D1$:响应时间 <20ms,$D2$:响应时间 $>=20$ms;E:价格,取值为 $E1$:价格 <20 \$,$E2$:价格 >20 \$;F:可用性,取值为 $F1$:可用性 >80,$F2$:可用性 <80.某用户在表达对 Web 服务 QoS 属性偏好的时候可能会说"在其它条件相同的情况下,我总是偏爱服务平台为数据库系统的服务;在服务平台为数据库系统的情况下,不管提供者是私立还是公立,我偏爱美国的服务甚于中国的服务;而在服务平台为文件系统的情况下,若提供者是私立的,我偏爱中国的服务甚于美国的服务,若提供者是公立的,我偏爱美国的服务甚于中国的服务;在服务位置在美国时,我偏爱私立的服务甚于公立的服务;在服务位置在中国时,我偏爱公立的服务甚于私立的服务".则该用户的不完全偏好陈述用 CP-net 表示如图 3 左半部分所示.



(a) CP-nets (b) 部分偏爱导出图

图3 存在环路的CP-net

很明显,在基于用户 CP-net 的基础上构造出部分偏爱导出图存在如图 3 右半部分的环路.在这种情况下,基于以下分析设计了有效的去除冲突的算法(图 4).算法的基本思想是:首先将所有结点标记为未访问,接着选择一个入度为零的结点作为初始结点,如果没有入度为零的结点,则随机选择一个入度最小的结点,假设为 v,并标记上已访问.设 w 是邻接于 v 的任意一个结点,把 w 标记为已访问并且前进到另一个结点,设为 x;若 x 已被标记为已访问,则删除边 $<w,x>$,否则再把 x 标记为已访问,并且前进到另一个邻接于 x 且标记为未访问的结点,这个选择邻接于当前结点的一个未访问过的结点的过程尽可能深的继续,直到我们找到一个接点 y,邻接于它的所有结点都已被标上已访问.在这点处,返回到最后访问的结点,比如设为 z,如果还有邻接于它的任何未访问结点,就访问它.继续这种方法,直到最终返回到起始结点 v.该算法的伪代码如图 4 所示.

```
Begin
  Initial stack S;      //初始化栈 S
   Parent[M][M];      //M 为结点的个数
   Explored[M];
   S.push(v);   //初始化 S 为具有一个元素 v 的栈
   While S≠Empty   //S 非空
    u←S.pop();//从 S 中取出一个结点 u
      If Explored[u] = false then
       Explored[u] = true;
        For each <u,w>    //每条与 u 相关联的边
       S.push(w);
       Parent[w][u] = u;
        End for
      End if
      If Explored[u] = true
        For each <u,w>    //每条与 u 相关联的边
          If Explored[w] = false then
            S.push(w);
            Parent[w][u] = u;
                Else if <u,w> 构成回路 then
            delete <u,w>;    //删除边 <u,w>
          End if
        End if
      End for
    End if
  End While
End
```

图 4 消除环路的算法

该算法的时间复杂度为 $O(M+N)$,其中 M 为结点数,N 为边数.

5.3.2 无冲突不完全偏爱的服务选择

给定一个 mCP-nets,有这样的问题:如何找到一个或一组结果其能够最大限度地满足所有用户的偏好需求? 这里,可以使用 Rank 方法模型来解决这个问题,其返回的结果集记为 $E=(E_1,\cdots,E_n)$.我们假设 mCP-nets 所有结果的集合为 Ω,Ω 中元素的个数记为 $|\Omega|$.$Opt(N_i)$代表 N_i 的最优结果($1\leqslant i\leqslant m$,N_i 代表 mCP-nets 中的第 i 个 CP-net),$len(O_{ij})$代表在 N_i 的偏爱导出图中结果 O_j 到 $Opt(N_i)$之间的最短路径.

算法的基本思想是:对于 Ω 中的每个元素,计算其在每一个 CP-net 中的 Rank 值.如果某结果是一个 CP-net 中的最优结果,则此结果的 Rank 值为 0,否则,此结果的 Rank 值为从该结果到用户的偏爱导出图中最优结果的最短路径加上一个阈值(阈值的设置见算法图 7 第七行),以使得到的结果集合中各子集的 Rank 值能和子集内元素的 Rank 值区分开,这样,按结果的 Rank 值就将所有结果分配到有顺序的集合 $E_i(0<i<L,L$ 是集合 E 的个数)中,最终,在计算出每个结果的 Rank 值后,将该结果在 mCP-nets 中的所有 Rank 值相加作为此结果的最终 Rank 值.

图 5 是计算集合 E 的伪代码.算法的时间复杂度为 $O(m2^n)$(m 是 CP-nets 的个数,n 是 mCP-nets 中变量的个数).

```
Input:mCP-nets,偏爱导出图(induced preferences graph)
Output:E=(E1,E2…EL)
|Ω|:mCP-nets 所有结果的个数,其中每一个结果记为 Oj
Opt(Ni):Ni 的最优结果(1≤i≤m,Ni 代表mCP-nets中的第 i 个CP-net)
len(Oij):在 Ni 的偏爱导出图中结果 Oj 到 Opt(Ni)之间的最短路径
1.  Begin
2.    i←1;
3.    j←1;
4.   For j=1 to |Ω|
5.     For i=1 to m
6.       if Oj=Opt(Ni) Rank(Oj)=0;
7.       else Rank(Oj)=(i-1)*100+len(Oij);
8.     End for
9.   End for
10.   L=1
11.   While(Ω≠ϕ) do
12.     For each O∈Ω do
13.           If Rank(O)=Min{Rank(Oi)| Oi∈Ω}
14.           EL←O;
15.         End for
16.       Ω←Ω-EL;
17.       L←L+1;
18.   End While
19. Return E=(E1,E2…EL);
20. End
```

图 5　计算集合 E 的算法

为了得到每个集合 E_i 内各元素的优先顺序,可使用 Trade-off CPnet(tCP-net)的概念:违反父变量偏好约束的结果要差于破坏任何子变量偏好约束的结果[17,19,1],对 $E_i(1\leqslant i\leqslant L)$ 中的所有元素根据每个用户偏好序列中偏好变量的重要程度来得到这些元素的偏好优先序列,然后可根据这些元素偏好的优先序列来计算每个元素的 sub-Rank(SRank)值,即根据这些元素的优先序列,在该集合 E_i 的 Rank 值基础上依次加 1 作为该 E_i 集合内元素的 Rank 值.从而为用户在同一个 E_i 集合内选择所需的偏好提供依据.

5.4　将偏好推理应用于 web 服务选择

在本节中,将展示如何将基于服务的 QoS 的多用户偏好推理应用于 Web 服务的选择过程中.

为了更加形象具体的说明如何将多用户偏好推理应用于 Web 服务的选择过程中,本节仍以上文的应用场景中的数据存储与访问的服务为例,来解释偏好推理算法的具体实现过程,并展示如何根据推理结果来指导服务选择.实验环境是 2.13 GHz Intel Core2 工作站,2GB 内存,Windows XP 系统.假设存在很多 Web 服务其均能提供数据存储与访问服务,该系统通过随机方法产生了大量数据存储的抽象服务及其 QoS.这里,数据存储与访问 Web 服务仍由上述六个属性描述:A:服务平台(数据库系统,文件系统),B:服务位置(美国,中国),C:服务提供者(私立的,公立的),D:响应时间(0 – 100ms),E:价格(0 – 100 $),$F$:可用性(0 – 100%),其取值为:$a$ 代表数据库系统,$\bar{a}$ 代表文件系统;b 代表美国,$\bar{b}$ 代表中国,c 代表私立的,$\bar{c}$ 代表公立的,d 代表响应时间 < 20ms,$\bar{d}$ 代表响应时间 > = 20ms;e 代表价格 < 20 $,$\bar{e}$ 代表价格 > = 20 $;$f$ 代表可用性 > 80%,$\bar{f}$ 代表可用性 < = 80%.表 1 给出了一些候选服务的例子.

表 1　候选服务的例子

服务	A	B	C	D	E	F(%)
S_1	数据库	美国	私立	12	19	86
S_2	文件	美国	公立	56	16	35
S_3	文件	中国	公立	8	15	90
S_4	数据库	美国	私立	69	43	52
S_5	数据库	中国	公立	26	48	40
S_6	文件	美国	私立	9	15	80

现在,假设一个用户 X,对提供该功能的服务的 QoS 属性的提出这样的偏好:服务平台≥服务位置≥服务提供者≥响应时间≥价格≥可用性;而另一个用户 Y,对提供该功能的服务的 QoS 属性提出不同的偏好:服务提供者≥响应时间≥服务位置≥价格≥可用性≥服务平台.根据请求,用户 X 的偏好的 CP-net 和用户 Y 的偏好的 CP-net 如图 6(a)和图 6(b)所示:

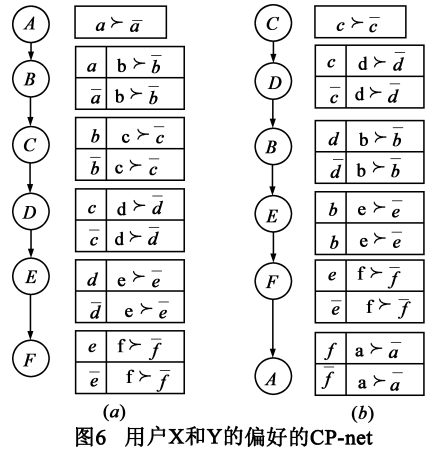


图6　用户X和Y的偏好的CP-net

根据上面提出的两个偏好序列,我们对其应用图 5 中的算法可以得到:

$E=(E_1,\cdots,E_n)$,

$E_1=\{abcdef\}$, $Rank(E_1)=0+0=0$;

$E_2=\{abcdef\bar{}, abcde\bar{f}, abc\bar{d}ef, ab\bar{c}def, a\bar{b}cdef, \bar{a}bcdef\}$,

$Rank(E_2)=100+1=101$

$E_3=\{abcde\bar{f}\bar{}, abcd\bar{e}f\bar{}, abcd\bar{e}\bar{f}, abc\bar{d}ef\bar{}, abc\bar{d}e\bar{f}, abc\bar{d}\bar{e}f, ab\bar{c}def\bar{}, ab\bar{c}de\bar{f}, ab\bar{c}d\bar{e}f, a\bar{b}cdef\bar{}, a\bar{b}cde\bar{f}, a\bar{b}cd\bar{e}f, \bar{a}bcdef\bar{}, \bar{a}bcde\bar{f}, \bar{a}bcd\bar{e}f, \bar{a}\bar{b}cdef, \bar{a}\bar{b}cde\bar{f}, \bar{a}\bar{b}cd\bar{e}f\}$,

$Rank(E_3)=200+2=202$;

$E_4=\{abcd\bar{e}\bar{f}, abc\bar{d}e\bar{f}, abc\bar{d}\bar{e}f, ab\bar{c}de\bar{f}, ab\bar{c}d\bar{e}f, ab\bar{c}d\bar{e}\bar{f}, a\bar{b}cde\bar{f}, a\bar{b}cd\bar{e}f, a\bar{b}cd\bar{e}\bar{f}, \bar{a}bcde\bar{f}, \bar{a}bcd\bar{e}f, \bar{a}bcd\bar{e}\bar{f}, \bar{a}\bar{b}cde\bar{f}, \bar{a}\bar{b}cd\bar{e}f, \bar{a}\bar{b}cd\bar{e}\bar{f}, \bar{a}\bar{b}\bar{c}def, \bar{a}\bar{b}\bar{c}de\bar{f}, \bar{a}\bar{b}\bar{c}d\bar{e}f, \bar{a}\bar{b}\bar{c}d\bar{e}\bar{f}\}$,

$Rank(E_4)=300+3=303$;

$E_5 = \{\overline{abcdef}, a\overline{bcdef}, \overline{abc}\overline{def}, \overline{abcd}\overline{ef}, \overline{abcde}f, \overline{a}bc\overline{def}, \overline{a}b\overline{cdef}, \overline{a}b\overline{cd}\overline{ef}, \overline{a}b\overline{cde}f, \overline{a}\overline{bc}def, \overline{a}\overline{bc}d\overline{ef}, \overline{a}\overline{bc}\overline{de}f, \overline{a}\overline{bcd}\overline{ef}, \overline{a}\overline{bcde}f, \overline{a}\overline{b}\overline{cdef}\}$,

$Rank(E_5) = 400 + 4 = 404$;

$E_6 = \{\overline{b}cdef, \overline{ac}def, \overline{ab}def, \overline{abc}ef, \overline{abcd}f, \overline{abcde}\}$,

$Rank(E_6) = 500 + 5 = 505$

$E_7 = \{\overline{abcdef}\}$, $Rank(E_7) = 600 + 6 = 606$;

在每个集合 E_i(在本例中 $1 \leqslant i \leqslant 7$)中,应用 Trade-off CP-net(tCP-net)的概念.根据每个用户偏好序列中偏好变量的重要程度可以得到集合 E_i 中所有元素的偏好优先序列.比如把上面的概念应用于集合 E_2,根据第一个用户的偏好序列可以得到(为了更加清晰我们仅用一个结果中的非字符表示该结果):$\bar{f} > \bar{e} > \bar{d} > \bar{c} > \bar{b} > \bar{a}$,其中 $SRank_X(\bar{f}) = 101$, $SRank_X(\bar{e}) = 102$, $SRank_X(\bar{d}) = 103$, $SRank_X(\bar{c}) = 104$, $SRank_X(\bar{b}) = 105$, $SRank_X(\bar{a}) = 106$;根据第二个用户的偏好序列我们可以得到(为了更加清晰,仅用一个结果中的非字符表示该结果):$\bar{a} > \bar{f} > \bar{e} > \bar{b} > \bar{d} > \bar{c}$,其中 $SRank_Y(\bar{a}) = 101$, $SRank_Y(\bar{f}) = 102$, $SRank_Y(\bar{e}) = 103$, $SRank_Y(\bar{b}) = 104$, $SRank_Y(\bar{d}) = 105$, $SRank_Y(\bar{c}) = 106$;根据两个用户的偏好序列为集合 E_2 中的每个元素定义 $sub-Rank(SRank)$,有:

$SRank(\bar{a}) = SRank_X(\bar{a}) + SRank_Y(\bar{a}) = 106 + 101 = 207$,

$SRank(\bar{b}) = SRank_X(\bar{b}) + SRank_Y(\bar{b}) = 105 + 104 = 209$,

$SRank(\bar{c}) = SRank_X(\bar{c}) + SRank_Y(\bar{c}) = 104 + 106 = 210$,

$SRank(\bar{d}) = SRank_X(\bar{d}) + SRank_Y(\bar{d}) = 103 + 105 = 208$,

$SRank(\bar{e}) = SRank_X(\bar{e}) + SRank_Y(\bar{e}) = 102 + 103 = 205$,

$SRank(\bar{f}) = SRank_X(\bar{f}) + SRank_Y(\bar{f}) = 101 + 102 = 203$.

首先选择与集合 E_1 中的结果 $abcdef$ 相对应条件的服务,即选择其 QoS 属性满足服务平台是数据库系统,服务位置在美国,服务提供者是私立的,响应时间小于 20ms,价格小于 20 $,可用性大于 80% 的服务;如果没有服务能满足该集合中的结果所对应的条件,那么接着选择与集合 E_2 中的结果相对应条件的服务,依此类推.在一个集合中,也许有很多元素,每次根据元素的 SRank 来选择一个结果.首先选择具有最小 SRank 值的元素,如果没有服务满足该结果所对应的条件,那么接着选择具有次小 SRank 值的元素,依此类推.作为一个例子我们来考虑集合 E_2,因为 $SRank(\bar{f}) = 203$ 在集合 E_2 中具有最小的 SRank 值,所以首先选择元素 $abcdef$ 作为候选条件,其对应的服务满足条件:服务平台是数据库系统,服务位置在美国,服务提供者是私立的,响应时间小于 20ms,价格小于 20 $,可用性小于 80%;如果没有服务满足该条件,那么接下来依次选择 $abcdef$, $\bar{a}bcdef$, $ab\bar{c}def$, $a\bar{b}cdef$, $ab\bar{c}def$(它们的 SRank 值分别为 205,207,208,209,210)作为候选结果.上面所述的选择服务的过程,也就是如何根据推理结果来指导服务选择的过程.

下面通过三组实验来展示我们设计的算法的效率.

第一组实验演示算法在选择尽可能满足大多数用户偏好的最好的服务方面的效率.我们注意到算法产生的集合 E 中 E_i 的个数随服务属性个数而变化,但由于 E 中 E_i 之间有这样的关系:$E1 > E2 \cdots > En$,且用户总是关心最能满足其偏爱要求的服务,所以在实验中仅显示集合 E 中前 5 个子集合.实验随机产生 1000,5000,8000 个服务,每个服务分别有 6、9、12、15 个属性,每个属性取二元值.在 Agent 个数为 2 个的情况下,图 10 的(a)、(b)、(c)、(d)分别显示了属性个数为 6、9、12、15 的情况下,E_i 中服务个数随候服务个数在 1000、5000、8000 的情况下的变化图.从图 10 可得知,随着候选服务个数的增多,相应的 E_i 中选择到的服务个数也增加,这是符合算法思想的,因为候选服务个数的增加,符合 E_i 中元素模式的服务个数必然增加;还可得知,当服务的属性比较多,而候选服务个数却不是很多时,可能并不能得到最优的满足用户偏爱要求的服务,这也是可以理解的,因为服务属性很多时,则用户可以对一个服务提出很多的偏爱要求,而在有限的候选服务中可能并不存在"完美地"满足这样"苛刻"条件的服务,例如在图 10 (d)中,在服务个数为 1000 的情况下,$E1$ 为空集,即满足用户的最优服务不存在,这时,只能退而求其次,将匹配 $E2$ 中服务模式的服务提交给用户,若 $E2$ 也是空集,则考虑 $E3$,依此类推.

第二组实验分析冲突消除算法的效率.由于冲突

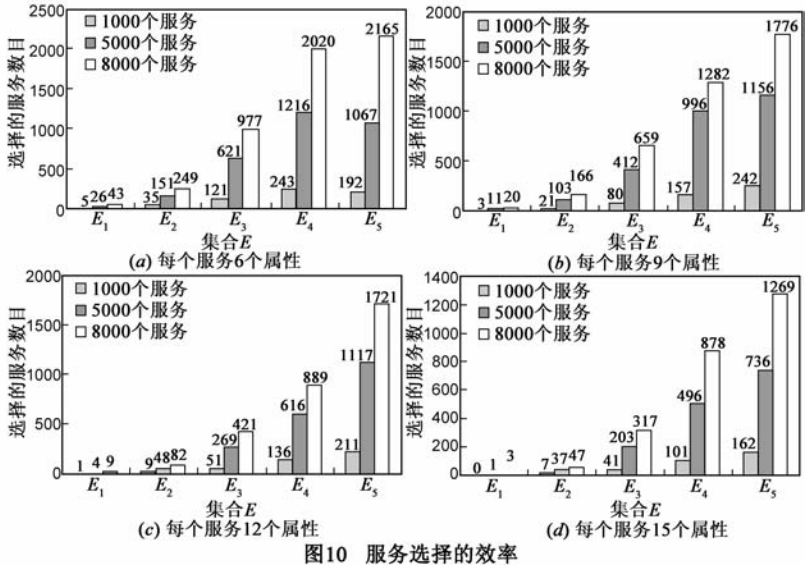


图10 服务选择的效率

消除算法主要用于遍历用户 CP-nets 的偏爱导出图检测出其中的环路,并去除环路,而用户 CP-nets 的偏爱导出图的规模与对应服务类别的属性个数和每个属性的取值个数有关,由于本文仅讨论服务属性取二元值的情况,因此,主要考察算法随服务属性个数增长,其执行时间的变化情况.当服务的属性个数在 6,9,12,15 的情况下,算法的执行时间如图 11.由图可看出,随服务属性个数的增加,冲突去除算法的时间增长较快,这是显然的,因为如 5.3.1 所述,算法的时间复杂度是与偏爱导出图的结点和边的个数之和成线性关系,而由 5.3.2 可知,偏爱导出图的结点和边的个数与服务属性的个数增加是成指数关系的(当属性取二元值时,是 $O(2^n)$),因此实验结果是符合上文理论分析的,也看到,在属性达到 15 个时,去除冲突的算法的时间是 10.6ms, 这在实际应用中也是可以接受的.

第三组实验分析算法在不同服务个数、用户个数和属性个数时的执行时间.图 12 显示了服务个数在 1000,5000,8000,服务属性个数为 6 个,每个属性取二元值,用户(Agent)数为 2 个的情况下,算法的执行时间,由图 12 可看出,算法的执行时间随候选服务的个数增加呈线性增加.为了说明用户(Agent)的个数对该算法性能的影响,我们固定属性个数为 6 个,每个属性取二元值,候选服务个数同样在 1000,5000,8000 个变化时,图 13 显示了在用户个数分别在 2,4,8,16 个的情况下算法的执行时间,由图可看到,算法的执行时间随用户个数的增加呈线性增加,这也是符合上文述及的算法的时间复杂度的理论分析的.上面两个实验中,候选服务涉及的属性个数仅为 6 个,然而,在实际应用中,Web 服务常用的 QoS 属性除了本文中 3.1 所述的常用的 6 个属性外,一些专业领域的 Web 服务,如应用场景中所述及的数据存储或访问服务,还可能还具有一些如第二部分应用场景中所述的服务平台、服务位置、服务提供者等专业领域的属性.因此,为了分析算法在这种情况下的性能,我们试验算法在服务的属性个数大于 6 个的情况下,这里设置属性个数为 6,9,12,15 个,并且固定用户(Agent)个数为 16 个,候选服务个数为同样在 1000,5000,8000 个变化的情况下的效率,其执行时间如图 14 所示,由图可看出,随候选服务属性个数的增加,算法的执行时间增长的很快,这也是符合算法的时间复杂度 $O(m2^n)$(m 是 CP nets 的个数,n 是 mCP nets 中变量的个数)的.最后,考察算法在本次实验中"最坏"的情况下的平均时间,即在候选服务个数为 8000 个,每个服务涉及 15 个属性,每个属性取二元值,用户个数在 16 个的情况下,我们执行算法 30 次,其结果如图 15 所示,并计算得到平均时间是 69.1ms,这个时间在实用中是可以接受的,并由图可看出,算法稳定性也很好,有其统计上的意义.

由上述三组实验结果我们可知:(1)本文所使用的定性的 Rank 语义方法能够根据多个用户的偏好选择出较优的 Web 服务集;(2)算法执行的平均时间随着用户个数增加而线性增长;(3)Web 服务属性的个数对算法的执行时间有较大影响,当属性个数不是非常多时算法的执行效率较高.在实际应用中,一般所选取的 QoS 属性的个数是 6 个,但实验也显示,QoS 属性的个数是增加到 15 个时,其执行时间也是可以接受的;(4)在属性个数达到 15 个时,去除冲突的算法的执行时间是 10.6ms,这在实际应用中也是可以接受的.因此,本文中所使用的算法能够有效地用于实际的服务选择应用.

本文的方法可用于单纯的服务选择应用(如本文中给出的应用场景),也可应用于组合逻辑确定后,再实施服务选择的应用场景.必须指出,本文使用定性的方法表示和推理多用户的偏好,其主要目的是弥补定

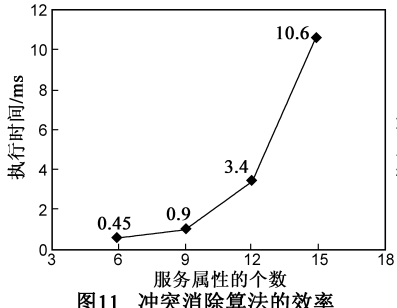


图11　冲突消除算法的效率

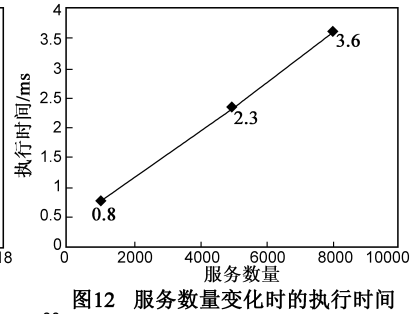


图12　服务数量变化时的执行时间

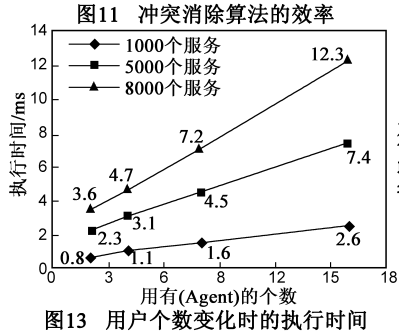


图13　用户个数变化时的执行时间

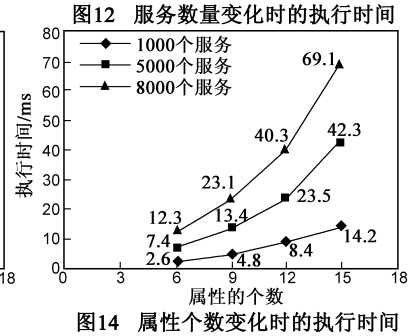


图14　属性个数变化时的执行时间

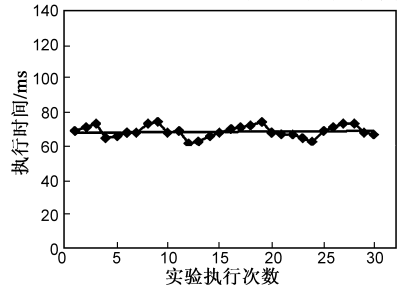


图15　平均执行时间

量方法的不足,并非要完全取代定量方法.

6 相关工作

用定性的方法处理偏好在服务选择方面尚缺乏深入系统的研究.文献[5]提出过一种定性的方法用于服务的选择,与本文的区别在于前者是针对单个用户的偏好,并且在处理用户偏好时假定偏好序列没有环路.在人工智能和决策领域,已经有一些定性的方法用来表示和推理单个用户的偏好.但对于不完全偏好还缺乏相应的研究.

用定性的方法表示和推理多用户偏好在人工智能领域有一些初步的研究.

F.Rossi 等[2]使用 mCP-nets 来建模和处理多个用户定性的条件偏好.文章给出了 5 种不同的投票语义机制来推理多用户偏好,这些投票机制描述了如何比较结果的优先性,论文也分析了完成这些任务的复杂性.虽然在这篇文章中,F.Rossi 等人第一次提出将 Rank 投票语义机制的思想应用于 mCP-nets 上,但他们并没有提出具体的实现方法和相应的推理算法.同时,包括 F.Rossi 在内的已有研究均假设偏好序列没有环路.本文与 F.Rossi 等人的工作的区别在于:(1)我们不但实现了 Rank 投票语义机制,提出了具体的推理算法,而且使用子 Rank(sub-Rank)概念对 Rank 投票语义机制进行了扩展,然后,将以上方法用于 Web 服务选择;(2)考虑了当用户的偏好序列中存在环路时,如何将有环 CP-net 转化为无环 CP-net,并提出了具体的算法来处理这种情况.

7 总结和未来工作

本文提出 Rank 投票语义方法解决多用户的服务选择问题.给出了 Rank 方法的实现机制和相应的算法.另外,我们还提出了子 Rank 的思想,并在本文中得到应用.文章还讨论了当用户的偏好序列中存在环路时,如何将有环偏好序列拆解为无环偏好序列,使其能够应用图 3 中的算法对多用户有环偏好进行处理.与其它一些定性的偏好推理方法相比,Rank 投票语义方法的优势在于:其不但可以推理单个用户偏好,还可以推理多个用户的偏好;不但可以对多个用户的完整偏好进行推理,还可以对多个用户的部分偏好进行推理.本文用其来解决服务选择问题,事实上,该方法可以被扩展用于处理多用户决策问题.

参考文献

[1] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, et al. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements[J]. Journal of Artificial Intelligence Research, 2004, 21(2): 135 – 191.

[2] F Rossi, K B Venable, T Walsh. mCP nets: representing and reasoning with preferences of multiple agents[A]. Proceedings of AAAI Conference on Artificial Intelligence[C]. California: AAAI Press, 2004, 729 – 734.

[3] San-Yih Hwang, Ee-Peng Lim et al. Dynamic Web Service Selection for Reliable Web Service Composition[J]. IEEE Transactions on Services Computing, 2008, 1(2): 104 – 116.

[4] Francesca Rossi, Kristen Brent Venable, Toby Walsh. Aggregating preferences cannot be fair[J]. Intelligenza Artificiale, 2005, 2(1): 30 – 38.

[5] Hongbing Wang, Junjie Xu, Peicheng Li, Patrick Hung. Incomplete preference-driven Web Service Selection[A]. 2008 IEEE International Conference on Services Computing[C]. Hawaii: SCC, 2008, 75 – 82.

[6] Wilson, N. Extending cp-nets with stronger conditional preference statements[A]. Proceedings of the 19th national conference on Artifical intelligence[C]. California: AAAI Press, 2004, 735 – 741.

[7] Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in cp-nets[J]. Journal of Artificial Intelligence Research, 2005, 19(1): 144 – 149.

[8] 邱田,李鹏飞,林品.一个基于概念语义近似度的 Web 服务匹配算法[J],电子学报,2009,37(2):429 – 432.
Qin tian, Li Pengfei, Lin Pin. A Web service matching algorithm based on semantic similarity of concepts.[J]. Acta Electronic Sinica, 2009, 37(2): 429 – 432. (in Chinese)

作者简介

周 宁 男,1979 年生于江苏南京.现在南京大学计算机系攻读博士学位,研究方向为 web services,人工智能等.
E-mail: zn@ketai-inc.com

谢俊元 男,1961 年生,南京大学计算机科学与技术系教授,博导,主要研究领域为智能系统,智能信息处理等.