ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.10, October 2009, pp.2655−2666
doi: 10.3724/SP.J.1001.2009.03436

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

上下文感知系统中的规则生成与匹配算法*

刘 栋+, 孟祥武, 陈俊亮, 夏亚梅

(北京邮电大学 网络与交换技术国家重点实验室,北京 100876)

Algorithms for Rule Generation and Matchmaking in Context-Aware System

LIU Dong+, MENG Xiang-Wu, CHEN Jun-Liang, XIA Ya-Mei

(State Key laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing 100876, China)

+ Corresponding author: E-mail: liu.dong66@gmail.com

Liu D, Meng XW, Chen JL, Xia YM. Algorithms for rule generation and matchmaking in context-aware system. *Journal of Software*, 2009,20(10):2655−2666. http://www.jos.org.cn/1000-9825/3436.htm

Abstract: To solve the problem that rules in the existing context-aware systems are manually specified by developers or users, an automatic rule generation method is proposed. Context-Aware systems are regarded as decision systems, and context information are reduced with discernibility matrix so as to generate rules. Because the data that can be utilized are limited, the generated rules can not entirely cover the domains of contexts. The rule matches current context probably does not exist. A rule matching algorithm is presented as the solution to this problem. Finally, the efficiency and effectivity of the proposed method is verified.
Key words: context-aware; rule generation; semantic distance; rule matching

摘 要: 针对现有上下文感知系统中的规则主要依靠开发者或用户手工定义的问题,提出了一种基于粗糙集理论的自动规则生成方法.该方法将上下文感知系统视为一种决策信息系统,并利用可辨识矩阵对上下文信息加以约简,进而自动生成规则.由于可供使用的数据有限,所生成的规则无法完全覆盖上下文的取值范围,因此可能出现找不到与上下文状态相匹配规则的问题.为了解决这一问题,提出了一种基于语义距离的规则匹配算法.最后验证了所提出方法的有效性和效率.
关键词: 上下文感知;规则生成;语义距离;规则匹配
中图法分类号: TP18 **文献标识码**: A

随着计算机技术的发展和普及,特别是普适计算和普适服务等新技术的出现,很多用户可能同时拥有多个智能设备,如 PDA、智能手机等,也可能同时使用多种服务或与多个系统进行交互.在这种情况下,因为启动、配置、使用各种设备和服务都需要通过人机交互来完成,用户注意力就成为了一种稀缺资源.我们可以用分时系统模型来进行类比,在分时系统中多个用户通过各自的终端以交互方式共享一台主机的计算资源,计算机则以分时的方式轮流为每个用户服务.而在普适计算或普适服务环境中,一个用户被多个智能设备所“共享”,需要以

* Supported by the National Natural Science Foundation of China under Grant No.60432010 (国家自然科学基金); the National Basic Research Program of China under Grant No.2007CB307103 (国家重点基础研究发展计划(973))

Received 2007-07-06; Revised 2007-12-28; 2008-06-11; Accepted 2008-08-07

“分时”的方式与多个智能设备进行交互.这显然不符合人的行为习惯,特别是在设备和应用系统数量较多的时候将严重影响用户工作效率.为了解决这一问题,研究者们提出了上下文感知计算(context-aware computing)这种新的计算范式[1],其目标是使系统能够自动地发现并利用位置、周围环境等上下文信息前摄地(proactively)为用户提供服务和计算资源,从而减少人机交互,改善用户体验.上下文感知系统的核心是若干预先定义的推理规则,当上下文状态发生改变时,系统进行规则匹配操作,如果上下文状态满足某一规则所规定的条件,系统将自动执行相应的操作.这种基于规则的上下文感知系统主要涉及两个关键技术问题:规则生成和规则匹配.

上下文感知系统中的推理规则一般都是由系统开发者或用户人工定义的[2].一方面,由于开发者对用户需求的认识有限,很难定义出完全符合用户需要的规则;而另一方面,规则描述语言本身比较复杂,一般用户难以在短时间内掌握,为此,文献[3]还设计了一种图形用户界面,一定程度地简化了规则定义过程.但是,上下文感知计算的目标之一就是减少用户交互,而复杂的规则定义过程增加了用户交互的难度,显然与上下文感知计算的初衷相悖.为了解决这一问题,本文提出了一种以粗糙集理论为基础的自动规则生成方法,该方法可以利用少量的用户操作记录和上下文状态记录自动生成规则,无须人工参与.

另一方面,由于本文提出的规则生成方法所利用的用户交互记录和上下文记录数量有限,所生成的规则不能完全覆盖上下文的取值范围,可能出现无法找到与当前上下文相匹配的规则的情况.为了解决这一问题,本文提出了一种基于语义距离的规则匹配算法,该算法能够在找不到完全匹配的规则时,从规则集合中找出一条与当前上下文在语义上最近似的规则.

本文第 1 节介绍国内外相关工作.第 2 节介绍上下文的定义、分类和本文所采用的上下文建模方法.第 3 节介绍粗糙集理论和规则生成方法.第 4 节介绍基于语义距离的规则匹配算法.第 5 节介绍原型系统结构及实现方法.第 6 节是总结.

1 相关工作

文献[4]提出了一个名为 CoBrA(context broker architecture)的基于 Agent 的上下文感知系统,其中的推理功能主要包括:从传感器采集的原始数据中推导出含有明确语义信息的上下文以及维护和检查上下文库的语义一致性.这两种功能都是通过基于描述逻辑的本体推理实现的,CoBrA 没有将基于规则的推理与本体推理相结合,这个问题在文献[5,6]中得到了改进.

在文献[5,6]中,上下文推理被分为本体推理和用户定义规则推理两部分.其中,本体推理部分的功能与 CoBrA 相同.用户定义规则推理则为开发者提供了一种控制上下文感知系统行为的手段,例如,可以通过定义规则使移动终端在用户进入图书馆时自动静音.但是,文献[5,6]中的推理规则都是系统开发者手工定义的,而且采用的规则描述语言也比较复杂.

文献[3]定义了一种新的上下文感知规则描述语言 CADEL(context-aware rule description language),这种语言的语法与自然语言比较接近.另外,为了方便普通家庭用户使用,文献[3]还专门设计了一套图形用户界面以简化规则定义的过程.总之,目前,上下文感知领域中规则生成方法仍以手工方式为主,对规则自动生成问题的研究还比较少,因此,其他领域中的研究者在这方面取得的成果值得借鉴.

文献[7]提出了一种应用于模糊专家系统的规则生成算法,该算法将规则生成过程划分为两个步骤:第 1 步首先生成前件只包含一个连接词的规则,其他类型的规则在后续步骤中生成.测试结果表明,该算法生成的规则数量比较合理.另外,文献[7]还提出了代表性(representativeness)和有效性(effectiveness)两个规则生成质量度量指标.

文献[8]提出了一种基于粒计算(granular computing)的规则生成算法 RGAGC.粒计算能够将问题空间划分为基本粒,并通过对基本粒组合、拆分操作生成新的粒,最终形成粒空间.RGAGC 算法则能够从粒空间中生成规则粒,并利用规则粒构造规则.与经典的基于判定树的方法相比,RGAGC 算法不需要考虑属性选择问题,具有较好的通用性.

文献[9]提出了一种基于近似关系的规则生成算法,该算法用近似关系替代经典粗糙集理论中的不可分辨

关系,这样可以省去对数据型属性进行离散化的操作,并使生成的规则具有较强的健壮性.但是,该算法没有考虑与本体的结合问题,处理的上下文也以数据型为主,没有明确的语义描述.

在规则匹配方面,文献[5,6]使用 Jena 2 语义网工具包[10]提供的通用规则推理机进行规则匹配操作;而文献[11]则采用了 Jess 规则引擎.这两种推理机都是基于 Rete 算法实现的,该算法在进行匹配操作之前先将规则前件编译为一个判定网络,从而减少了比较操作的次数,提高了运行效率.但是,基于 Rete 算法的规则推理机只能进行精确匹配,不能进行近似匹配.

2 基于本体的上下文信息建模

本体是对某一领域中的概念所作的明确的形式化描述,在上下文感知系统中引入本体可在如下几个方面提高系统的推理能力:首先,引入本体以后,可以进行基于描述逻辑的本体推理,即从传感器采集或用户输入的显式上下文中推导出更多的隐式上下文.例如,将位置上下文 locateIn 定义为 owl:TransitiveProperty 型属性,若有“John locateIn MeetingRoom”以及“MeetingRoom locateIn Building1”,则可推导出“John locateIn Building1”.其次,本体可以为上下文信息提供明确的语义描述,这样基于规则的推理过程中的前件匹配操作将不仅仅是字符串匹配和数量关系的比较,而且还能够支持语义匹配.本节从上下文定义和分类出发介绍了上下文本体 CACO(context-aware computing ontology)及其与规则生成、匹配的关系.

2.1 上下文定义、分类

上下文这一概念目前还没有公认的定义,研究者采用较多的是 Dey 在文献[12]中给出的定义:“上下文是任何可用于刻画实体所处环境的信息.这里的实体可以是人,也可以是地点或其他在用户和应用程序交互过程中所涉及的对象,甚至还包括用户和应用程序本身”.从这个定义我们可以看出,实体与上下文之间存在如下关系:上下文是描述实体属性的信息,同时实体与上下文也是相对的概念,在某些情况下可以相互转换.例如,位置可以作为上下文来描述人或设备,此时,人、设备是两种实体;温度、湿度等也可作为上下文来描述某个位置,此时位置又作为实体出现.根据上述分析,我们进一步给出本文所采用的实体和上下文定义如下:

定义 1(实体(entity)). 实体是用户和应用程序交互过程中所涉及的具有相同属性的对象,每一种实体都可用一个 owl:Class 来表示.

定义 2(上下文(context)). 上下文是刻画实体的某种属性的信息,可用 4 元组 $C=\{E,V,S,T\}$来表示.其中,E 表示上下文 C 所描述的实体;V 表示上下文的取值;S 表示上下文的来源,我们将上下文的来源分为 3 种,即人工输入、传感器采集和推理机推理;T 表示上下文更新时间.

对上下文分类方法的研究不仅是构造上下文本体的基础,同时也是选择上下文近似度计算方法的依据.首先,以层次化形式给出某一领域内的概念和实例的明确定义及其相互关系是构造本体目标之一,而对领域内的基本概念的分类问题的研究是实现这一目标的前提.因此,为了构造上下文本体,需要首先研究上下文的分类方法.其次,不同类型上下文的近似度计算方法往往有所不同,本文第 4 节中介绍的规则匹配算法根据上下文类型选择适当的近似度计算方法.

研究者从不同的角度出发,已经提出了多种上下文分类方法[1,13,14].本文根据取值范围的不同将上下文划分为两类:数据型上下文和实例型上下文.其中,数据型上下文的取值范围是 RDF 文字或 XML Schema 中内建的简单类型数据.例如,噪音水平、光照强度、温度、湿度、速度等属于数据型上下文.实例型上下文的取值范围为上下文本体中的实例.例如,用户位置、用户周围人、资源、设备以及用户的目标、任务和行为等属于实例型上下文.数据型上下文可以进一步划分为布尔型、数值型、时间日期型以及字符串型等.在上下文本体中,数据型和实例型上下文可分别用 owl:DatatypeProperty 和 owl:ObjectProperty 两种类型的属性来表示.

2.2 上下文本体CACO

根据第 2.1 节中的本体、上下文等概念的定义以及上下文分类方法,我们构造了一个本体 CACO 以描述上下文感知计算领域的知识,CACO 本体的片段如图 1 所示.

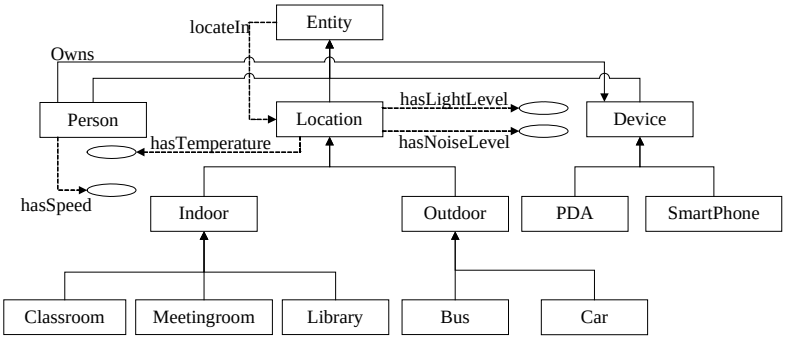

Fig.1 Part of context-aware computing ontology

图 1 上下文本体 CACO 片段

图 1 中的矩形表示实体.实线箭头表示实体之间的继承关系,可用 owl:subClass 表示.椭圆表示数据型上下文.虚线箭头则表示实体与上下文之间的关系,如前所述,数据型和实例型上下文可分别用 owl:DatatypeProperty 和 owl:ObjectProperty 来表示.

另外,我们还利用 RDF 具体化(RDF reification)来表示上下文来源和更新时间,所谓 RDF 具体化就是利用具体化词汇对 RDF 陈述(RDF statement)进行再描述.RDF 具体化词汇包括:rdf:Statement,rdf:subject,rdf:predicate 和 rdf: object.图 2 为使用 OWL 语言描述的 CACO 本体片段.其中 rdf:ID 为“speedOfJohn”的陈述内容是“John hasSpeed 3”.这一陈述又被“caco:updateTime”和“caco:source”具体化为“于 2007 年 12 月 17 日 18:51:25”,“由 Dromometer1 采集”.

```
〈owl:Class rdf:ID=“Bus”〉
  〈rdfs:subClassOf rdf:resource=“#Outdoor”/〉
〈/owl:Class〉
〈owl:DatatypeProperty rdf:ID=“hasSpeed”〉
  〈rdfs:domain rdf:resource=“#Person”/〉
  〈rdfs:range
rdf:resource=“http://www.w3.org/2001/XMLSchema#int”〉
〈/owl:DatatypeProperty〉
〈owl:ObjectProperty rdf:ID=“locateIn”
  rdf:type=“http://www.w3.org/2002/07/owl#TransitiveProperty”〉
  〈rdfs:domain rdf:resource=“#Entity”/〉
  〈rdfs:range rdf:resource=“#Location”/〉
〈/owl:ObjectProperty〉

〈caco:Bus rdf:ID=“Bus1”/〉
〈caco:Person rdf:ID=“John”/〉
  〈caco:hasSpeed rdf:ID=“speedOfJohn”〉3〈/caco:hasSpeed〉
  〈caco:locateIn rdf:resource=“#Bus1”/〉
〈/caco:Person〉
〈rdf:Description rdf:about=“#speedOfJohn”〉
  〈caco:updateTime〉2007-12-17T18:51:25〈/caco:updateTime〉
  〈caco:source〉Dromometer1〈/caco:source〉
〈/rdf:Description〉
```

Fig.2 Context representation using OWL

图 2 基于 OWL 的上下文表示

2.3 与其他本体的比较

与其他上下文本体相比较,本文所提出的 CACO 本体规范了实体和上下文的表示方法,容易实现与现有的

各种领域本体的融合.在文献[4]所提出的本体 CoBrA-ONT 中,owl:Class 不仅被用来表示 Person,Agent,Place 等实体概念,同时也用于描述形如 PersonInRoom 的概念,并将这类概念称为上下文.这种表示方法的可扩展性较差,因为每添加一个实体都需要手动添加多个与之对应的上下文.例如要将 Professor 和 Student 这两个实体加入 CoBrA-ONT,则需要增加上下文 ProfessorInRoom 和 StudentInRoom 等.而将这两个概念加入 CACO 本体只需将它们作为 Person 概念的子类即可,因此可以根据需要将各种领域本体方便地添加到 CACO 本体中.

文献[5]提出的上下文本体 CONON 中的所有概念都继承自上下文实体(context entity)这一概念,没有明确指出实体和上下文的区别和联系;同时也没有指出对上下文更新时间和来源等附加信息的描述也支持,在 CACO 本体中我们通过引入 RDF 具体化解决了这个问题.

3 基于粗糙集理论的规则生成方法

由于用户使用服务或应用程序的习惯不同,并不是每种上下文都能够对用户行为产生影响,有些上下文甚至是冗余的,例如,用户一般不会根据湿度来设置手机振铃方式.另外,过多的上下文也会增加规则匹配的计算复杂度.为了实现根据用户行为习惯选择上下文,本节提出了一种利用粗糙集理论对上下文进行约简并生成规则的方法.

3.1 粗糙集理论

粗糙集理论(rough set theory)是波兰数学家 Pawlak 于 1982 年提出来的,可用于发现数据之间的依赖关系以及约简数据集中属性的数量[15].与证据推理、模糊逻辑等方法相比,粗糙集理论的优点在于不需要数据集以外的其他信息,下面我们简要介绍粗糙集理论中的基本概念.

定义 3(信息系统). 称 $I=(U,A,V)$为一个信息系统,其中 U 为非空有限集,称作论域;A 为形如 $a:U\to V_a$ 的映射构成的集合;$V=\bigcup V_a$ $(a\in A)$.

定义 4(决策信息系统). 若信息系统 $I=(U,A,V)$满足条件: $A=C\cup D$ 且 $C\cap D=\varnothing$,则称 I 为决策信息系统,其中 C 又称为条件属性,D 为决策属性.

定义 5(不可分辨关系). 对于$\forall P\subseteq C$,等价关系 $IND(P)$称为不可分辨关系,其中 $IND(P)=\{(x,y)\in U\times U\,.\,|\forall a\in P,a(x)=a(y)\}$.

定义 6(粗糙集及其近似). 不可分辨关系可以将论域 U 划分为若干等价类,如果 $X\subseteq U$ 不能用某些等价类的并集来表示,则称 X 为粗糙集.粗糙集可用上下近似集来近似表示,集合 $\underline{P}X=\{x|[x]_P\subseteq X\}$ 和 $\overline{P}X=\{x|[x]_P\cap X\neq\varnothing\}$ 分别称为 X 的下近似集和上近似集.

定义 7(正区域). 设 P,Q 是论域 U 上的等价关系,则集合 $POS_P(Q)=\bigcup_{X\in U/Q}\underline{P}X$ 称为正区域.

在决策信息系统中,决策属性对不同的条件属性的依赖程度不同,有些条件属性甚至是冗余的,即去掉该属性对决策信息系统不会产生影响,粗糙集理论中的一项重要研究内容就是找出并去掉冗余的条件属性,为此,我们再来介绍另外两个概念:约简和核.

定义 8(约简和核). 设有决策信息系统 $I=(U,C\cup D,V)$,且有 $P\subseteq C,P\neq\varnothing$,若 P 满足条件 $POS_P(D)=POS_C(D)$,则称 P 是 C 的一个约简.记 $RED(C)$为 C 的所有约简,则称 $CORE(C)=\cap RED(C)$ 为 C 的核.

在得到了决策信息系统属性约简结果之后可以构造规则,即将约简结果作为规则的前件(predecessor),同时将对应的决策属性作为规则的后件(successor),本文第 3.2 节将具体介绍属性约简和规则生成算法的原理.

3.2 规则生成算法原理

本节我们利用粗糙集理论对上下文感知系统进行分析,并给出一种规则生成算法.假设有 n 种不同类型的上下文构成了条件属性集合 $A=\{C_1,C_2,...,C_n\}$,由定义 2,我们可以用向量 $CV=\{V_1,V_2,...,V_n\}$ 来表示这 n 种上下文在某一时刻的取值,其中 V_i 为上下文 C_i 的取值.将上下文向量构成的集合 $U=\{CV_1,CV_2,...,CV_m\}$ 作为论域,由于上下文包括数据型和实例型两种类型,因此我们定义不可分辨关系如下:

$$IND(P)=\left\{(cv_i,cv_j)\in CV\times CV\middle|\forall a\in P,a(cv_i)=a(cv_j)\vee class\left(a(cv_i)\right)=class\left(a(cv_j)\right)\right\},$$

其中 $P\subseteq\{C_1,C_2,...,C_n\}$,$a(x)$表示向量 x 在属性 a 上的取值,$class(v)$表示个体 v 所属的类.另外,我们将用户进行的操作或使用的服务作为决策属性,这样就建立了一个决策信息系统,可以利用粗糙集理论对属性集进行约简.属性约简的方法有很多种,本文采用了基于可辨识矩阵的方法,下面是相关概念的定义:

定义 9(可辨识矩阵). 矩阵 $(c_{ij})=\begin{cases}\left\{a\in A\middle|a(x_i)\neq a(x_j)\vee class\left(a(x_i)\right)\neq class\left(a(x_j)\right)\right\}, & D(x_i)\neq D(x_j)\\ 0, & D(x_i)=D(x_j)\end{cases}$, $i,j=1,2,\ldots,m$ 称为可辨识矩阵.

定义 10(属性分辨能力). 我们用属性在可辨识矩阵中的出现频率与属性取值范围集合的基数之比来刻画属性的分辨能力.

算法基本步骤是:① 利用可辨识矩阵求核(可辨识矩阵中基数为 1 的元素所包含的属性就是核),将核加入到属性约简结果 R 中;② 用约简结果 R 化简可辨识矩阵;③ 计算非核属性的分辨能力,选择分辨能力最强的非核属性加入 R 中;④ 重复②、③,直到可辨识矩阵为零矩阵;⑤ 利用约简结果生成规则.用伪代码描述算法 1 如下,其时间复杂度与上下文记录数 m、上下文数 n 相关,为 $O(nm^2)$.

算法 1. 上下文规则生成算法.

输入:$CV=(CV_1,CV_2,\ldots,CV_m)$:上下文状态记录表;$O$:用户操作记录;CACO:本体;$A=(C_1,C_2,\ldots,C_n)$:上下文集合.

输出:Rules:上下文感知规则集合,core:核,red:约简结果.

1) 求可辨识矩阵和核

```
for (i=0;i<m;i++){
    for (j=i+1;j<m;j++){
        if (O[i]==O[j])
            DM[i][j]=0;
        else 如果 CV[i]与 CV[j]的分量 V_k不相等或不是同一个类的实例,则将上下文 C_k加入 DM[i][j];
        if (DM[i][j]中只有一个上下文 C)
            将上下文 C 加入核 core;}}
```

2) 将可辨识矩阵 DM 中含有核的项置为 0,并将核 core 加入约简结果 red;

3)
```
while (DM !=0){
        计算未加入 red 的上下文的分辨能力,将分辨能力最强的上下文加入 red;
        将 DM 中含有 red 中的上下文的项置为 0;}
```

4) 根据约简结果生成规则.

3.3 算法举例

本节我们通过一个实例来进一步说明算法 1 的原理.现有一个上下文感知信息服务系统能够根据用户位置、运动速度、光照条件、噪音水平等上下文信息以文本、图像或语音方式为用户提供新闻信息.例如,当用户处于交通工具上,运动速度较快不方便阅读较长信息时,用户可能只需要简短的新闻摘要;而当光照条件较差时,用户可能希望系统能够以语音方式播放新闻.如前所述,即使在图形用户界面的帮助下,用户自行定义这些规则也是相当繁琐的.本文所提出的自动规则生成方法不改变现有的人机交互模式,只利用少量的用户交互记录和上下文记录自动生成规则.表 1 给出了一组上下文记录以及用户所选择的信息服务形式,为了简化问题,我们将光照强度、噪声强度、温度和速度分别离散化为 3 个等级.由表 1 求得的可辨识矩阵如图 3 所示.利用算法 1,可得核$\{c_1,c_2\}$,用核化简可辨识矩阵后可辨识矩阵为零矩阵,因此,$\{c_1,c_2\}$就是属性约简结果.将上下文记录作为前件,并将信息服务形式作为后件,可得如图 4 所示的 9 条规则.

Table 1 Records of contexts and user operations

表 1 上下文及其用户操作记录

No.	Location (c_1)	Illumination (c_2)	Noise (c_3)	Temperature (c_4)	Speed (c_5)	Service delivery form
1	Classroom1	3	1	3	1	Content
2	Meetingroom1	2	2	3	1	Abstract
3	Library	3	1	2	1	Content
4	Bus1	3	3	1	3	Audio
5	Bus2	1	3	1	3	Audio
6	Library	2	1	2	1	Content
7	Meetingroom2	3	2	1	1	Abstract
8	Bus2	2	3	3	3	Audio
9	Classroom2	1	1	3	1	Audio
10	Meetingroom1	3	1	3	1	Abstract

$$\begin{pmatrix}
0 & c_1c_2c_3 & 0 & c_1c_3c_4c_5 & c_1c_2c_3c_4c_5 & 0 & c_1c_3c_4 & c_1c_2c_3c_5 & c_2 & c_1 \\
 & 0 & c_1c_2c_3c_4 & c_1c_2c_3c_4c_5 & c_1c_2c_3c_4c_5 & c_1c_3c_4 & 0 & c_1c_3c_5 & c_1c_2c_3 & 0 \\
 & & 0 & c_1c_3c_4c_5 & c_1c_2c_3c_4c_5 & 0 & c_1c_3c_4 & c_1c_2c_3c_4c_5 & c_1c_2c_4 & c_1c_4 \\
 & & & 0 & 0 & c_1c_2c_3c_4c_5 & c_1c_3c_5 & 0 & 0 & c_1c_3c_4c_5 \\
 & & & & 0 & c_1c_2c_3c_4c_5 & c_1c_2c_3c_5 & 0 & 0 & c_1c_2c_3c_4c_5 \\
 & & & & & 0 & c_1c_2c_3c_4 & c_1c_3c_4c_5 & c_1c_2c_4 & c_1c_2c_4 \\
 & & & & & & 0 & c_1c_2c_3c_4c_5 & c_1c_2c_3c_4 & 0 \\
 & & & & & & & 0 & 0 & c_1c_2c_3c_5 \\
 & & & & & & & & 0 & c_1c_2c_3c_4 \\
 & & & & & & & & & 0
\end{pmatrix}$$

Fig.3 Discernibility matrix

图 3 可辨识矩阵

(1) (c_1,Classroom) and (c_2,1), then form=Audio

(2) (c_1,Classroom) and (c_2,3), then form=Content

(3) (c_1,Meetingroom) and (c_2,2), then form=Abstract

(4) (c_1,Meetingroom) and (c_2,3), then form=Abstract

(5) (c_1,Library) and (c_2,2), then form=Content

(6) (c_1,Library) and (c_2,3), then form=Content

(7) (c_1,Bus) and (c_2,1), then form=Audio

(8) (c_1,Bus) and (c_2,2), then form=Audio

(9) (c_1,Bus) and (c_2,3), then form=Audio

Fig.4 Rules for context-awareness

图 4 上下文感知规则

3.4 算法有效性验证

为了验证本文提出的规则生成算法的有效性,我们按照下述方法进行了实验:首先手工定义若干条上下文感知规则,并随机生成一定量的上下文记录,将手工定义的规则应用于这些上下文记录以得到操作记录;这些上下文记录与对应的操作记录共同构成测试数据集.从测试数据集中任取一部分作为本文第 3.2 节描述的算法 1 的输入,同时用测试数据集中剩余的上下文记录及其对应操作记录所组成的集合作为比照数据集.最后将算法 1 生成的规则应用于比照数据集中的上下文记录上,并将得到的操作记录与比照数据集进行比较,从而得出算法 1 所生成的规则的有效性(effectiveness).如果用 *RD* 表示手工定义的规则构成的集合;*RG* 表示算法 1 生成的规则;*C* 表示比照数据集中上下文记录构成的集合;*apply*(*R*,*c*)表示将规则集 *R* 应用到上下文记录 *c* 上得到的操作记录,则生成规则的有效性可定义为

$$Effectiveness(RG)=\frac{\left\|\{(c,apply(RG,c))|c\in C\}\cap\{(c,apply(RD,c))|c\in C\}\right\|}{\|C\|}.$$

在具体的实验过程中我们手工定义了 3 条规则:(1) (c_1,Bus),则 form=Audio;(2) (c_1,Classroom)且(c_2,1),则 form=Abstract;(3) (c_1,Meetingroom)且(c_2,1),则 form=Abstract,并随机生成了 10 000 条上下文记录,包括用户位置、光照强度、噪声强度、温度、速度等 10 种上下文.选取其中 9 000 条记录作为比照数据集,并从另外 1 000 条记录中选择 n 条记录作为规则生成算法的输入.图 5 显示了论域基数 n 与生成规则的有效性之间的关系,可以看出生成规则的有效性随着论域基数的增加而增加,并逐步逼近最大值 1.换言之,随着论域基数的增加,算法 1 所生成的规则逐渐逼近手工定义的规则.当数据量充分大时,生成规则将与手工定义的规则等价.

然而,规则生成算法执行时间也会随着论域基数的增加而增加,为了定量地研究二者之间的关系,我们在下述环境中测试了算法 1 的时间复杂度:Intel Pentium 4 CPU(主频 3.0GHz)、512M 内存、Windows XP 操作系统,算法实现语言为 Java.图 6 显示了上下文数分别为 10,25 和 50 时算法执行时间和论域基数之间的关系,可以看出,算法 1 的时间复杂度随着论域基数的增加而增加,并且上下文越多,其增长速度越快.因此,在实际应用中,确定论域基数时不仅要考虑其对规则有效性的影响,同时也要考虑系统效率问题,合适的论域基数是保证生成规则的有效性和算法执行效率的关键.

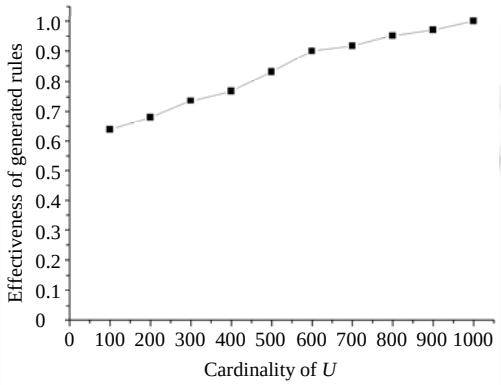

Fig.5 Relationship between cardinality of universe and the effectiveness of generated rules

图 5 论域基数与生成规则有效性的关系

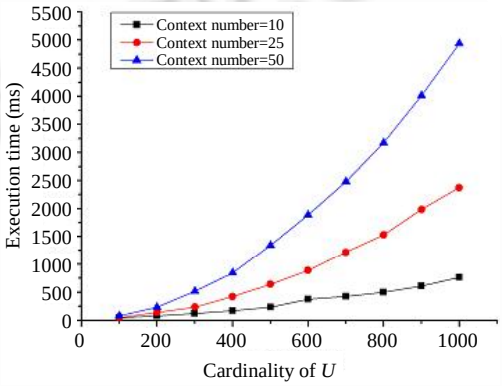

Fig.6 Time complexity of rule generation algorithm

图 6 规则生成算法时间复杂度

4 基于语义距离的规则匹配算法

规则匹配失败的根本原因是可用于生成规则的数据规模有限,不能完全覆盖上下文取值范围,为了在不增加论域规模的前提下提高规则匹配的正确率,本节给出了一种基于语义距离的规则匹配算法,该算法能够在找不到与当前上下文状态完全匹配的规则时,从规则集合中选择一条最近似的规则.

4.1 规则匹配算法原理

如前所述,上下文的取值有两种类型:一种是本体 CACO 中的实例,另一种则是具体数据.对于数据型上下文,我们可以利用公式:

$$dist(v_1,v_2)=1-\frac{|v_1-v_2|}{range} \quad (1)$$

来计算两个取值之间的近似度,其中 $range$ 为该上下文的取值范围大小.对于其他数据型上下文,如布尔型、时间日期型等,需要先进行转换才能利用公式(1)计算近似度.以时间日期型上下文为例,若将某一时刻(如 1970 年 1 月 1 日 0:00)作为原点,并以某一粒度的时间单位(如 s,ms 等)为度量手段,则可将时间日期型数据转化为整数型数据,这样就可以代入公式(1)计算数据项的近似度了.特别地,对于字符串型上下文,我们用两个字符串的最大公共子串长度与两个字符串的平均长度之比来刻画其近似度.

对于实例型的上下文则需要利用语义距离来刻画取值之间的差异.目前,研究者已经提出了多种语义距离定义[16],本文采用文献[17]提出的 GCSM 距离,其具体定义如下:

定义 11(最低共同祖先(lowest common ancestor,简称 LCA)). LCA 是指本体中两个概念的深度最大的共同祖先.

定义 12(GCSM(generalized cosine-similarity measure)距离). 若 c_1,c_2 是本体 O 中的两个概念,则 c_1,c_2 的 GCSM 距离为 $GCSM(c_1,c_2)=\dfrac{depth(c_1)+depth(c_2)}{2\times depth\left(LCA(c_1,c_2)\right)}$.

例如,对于如图 7 所示的本体片段,概念 Classroom 和 Library 的深度为 2,而两者的最低共同祖先 Indoor 深度为 1,代入定义 12 中的公式,$GCSM(Classroom,Library)=\dfrac{2+2}{2\times 1}=2$.

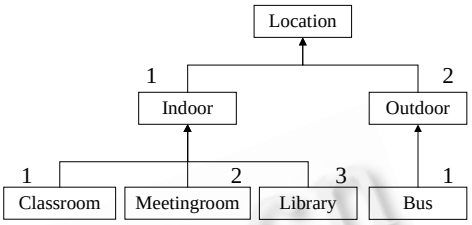

Fig.7 Partial ontology

图 7 本体片段

为了提高效率,我们在系统运行前遍历本体,并为通向不同子类的路径赋予不同的编码,这样就得到从根节点到达每个概念的路径序列.例如,概念 Classroom 和 Library 的编码分别为(1,1)和(1,2),在系统运行时可以直接利用这些编码来计算概念之间的语义距离.具体方法是:首先找出两个编码序列的最大前缀,最大前缀的长度与最低共同祖先深度相等,而概念的编码长度与深度相等,带入定义 12 中的公式即可得到概念的语义距离.概念 Classroom 和 Library 的编码长度为 2,最大前缀(1)长度为 1,计算得到语义距离仍为 2.由于上下文感知规则可能涉及到多种上下文,因此首先需要计算上下文向量与规则前件的各个分量的近似度的加权平均值,并将这一数值作为上下文与某一规则前件的近似度.这里我们将属性重要性作为权重,其定义如下:

定义 13(属性依赖度). 参数 $k=\gamma_P(Q)=\dfrac{|POS_P(Q)|}{|U|}$ 称为属性集 Q 对属性集 P 的依赖度.

定义 14(属性重要性). 称 $\sigma_{(C,D)}(a)=1-\dfrac{\gamma_{C-\{a\}}(D)}{\gamma_C(D)}$ 为属性 a 的重要性.

将公式(1)的计算结果作为数据型上下文取值近似度,同时将 GCSM 距离的倒数为实例型上下文的近似度,则上下文向量与规则前件的近似度计算公式为 $sim(CV,R)=\sum_{i=1}^{n}W_{C_i}\cdot sim(V_{C_i},R_{C_i})$,其中 W_{C_i} 为上下文 C_i 的权重;V_{C_i} 和 R_{C_i} 分别表示上下文向量 CV 和规则前件 R 在 C_i 上的取值;$sim(V_{C_i},R_{C_i})$ 表示 V_{C_i} 和 R_{C_i} 的近似度.规则匹配算法的伪代码描述见算法 2,其时间复杂度为 $O(rc)$,其中,r 为规则数,c 为约简结果中的上下文数.

算法 2. 规则匹配算法.

输入:Context:当前上下文状态;Rules:上下文感知规则集合;CACO:上下文感知计算本体;Reduct:属性约简结果;Weight:权重.

输出:*matched_rule*:匹配规则.

```
max_sim=0; //最大近似度
for (int i=0;i<Rules.length;i++){
    //计算每条规则条件部分与当前上下文状态的近似度
    sim=0;
    for (int j=0;j<Reduct.length;j++){
        //计算每种上下文的近似度
        if (Reduct[j]取值为实例)
            sim+=Weight[j]*(1/GCSM(Context[j],Rules[i].getCondition(j)));
```

else if (*Reduct*[*j*]取值为一般数据)

sim+=*Weight*[*j*]*(1−*abs*(*Context*[*j*]−*Rules*[*i*].*getCondition*(*j*))/*Reduct*[*j*].*getRange*());}

if (*sim*>max_*sim*)

matched_rule=*Rules*[*i*];}

4.2 算法举例

本节我们通过一个实例来进一步说明规则匹配算法的原理.设当前上下文状态为(Car1,3,2,2,3),显然,图 4 所示的生成规则集合中没有与之完全匹配的规则,需要利用算法 2 从中选择一条与当前上下文在语义上最接近的上下文感知规则.根据图 1 所示的 CACO 本体可得,概念 Car 与 Classroom,Meetingroom,Library 和 Bus 这 4 个概念之间的 GCSM 距离分别为 2,2,2 和 1.333;根据表 1 中的上下文记录和操作记录可知,属性 c_1,c_2,c_3,c_4 和 c_5 的属性重要性依次为 0.2,0.2,0,0 和 0.规则匹配算法的计算结果表明,图 4 中的 9 条规则与当前上下文的匹配程度依次为 0.167,0.3,0.233,0.3,0.233,0.3,0.217,0.283 和 0.35,显然第(9)条规则的前件与当前上下文最近似,因此,这条规则可作为匹配结果.上述实例表明,在上下文出现新的取值或传统的规则匹配操作失败时,利用本文提出的规则匹配算法能够在规则库中选择一条适合当前上下文环境的规则.

4.3 算法有效性验证

我们仍采用本文第 3.4 节提出的方法验证生成规则的有效性,测试数据量仍为 10 000,比照数据集大小也保持在 9 000 的水平上,所不同的是,在规则匹配阶段,当生成规则中不存在完全匹配的规则时,执行算法 2,选择一条语义上最接近的规则作为匹配结果.从图 8 我们可以看出,在用于生成规则的数据量相同的情况下,在规则匹配阶段采用算法 2 能够提高生成规则的有效性,尤其是在论域基数较小的情况下,生成规则有效性的提高更为明显.另外,为了验证算法的时间复杂度,我们在与第 3.4 节相同的环境中测试了上下文数分别为 10,25 和 50 时算法执行 100 次所消耗的时间与规则数之间的关系,结果如图 9 所示.算法执行时间随着生成规则数的增加而增加,并且上下文数越多,其增长速度越快.在上下文数为 50 且生成规则数为 1 000 时,执行 1 次算法 2 的时间约为 45ms.

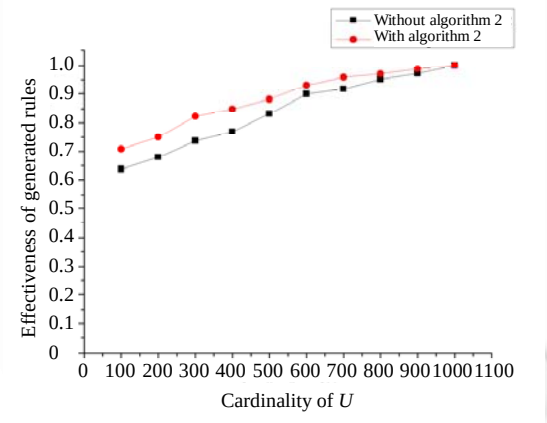

Fig.8 Comparison of the effectiveness of generated rules
图 8 生成规则有效性比较

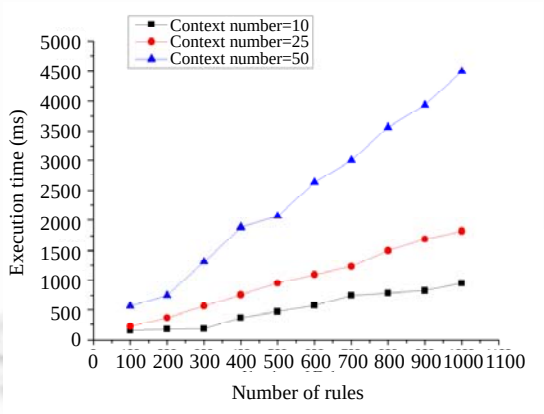

Fig.9 Time complexity of rule matchmaking algorithm
图 9 规则匹配算法的时间复杂度

5 原型系统及其实现方法

本文所提出的方法能够应用于综合信息服务领域,我们在“综合智能业务平台”[18]的研究基础上实现了一个原型系统,其目的是使系统能够根据上下文自动选择适当的信息表现形式为用户提供各种信息.具体应用场景如下:内容提供商将新闻、天气预报等内容发布到业务平台上,用户向平台订阅信息服务,平台根据上下文选择适当的形式为用户提供信息,如用短信发送新闻摘要、用彩信发送图片或全文、通过语音呼叫播送音频信息.

原型系统结构如图 10 所示,在逻辑上可划分为传感器层、上下文层和应用层这 3 个层次.传感器层主要包括传感器和上下文收集器两类逻辑实体.上下文收集器对传感器采集的原始数据进行预处理、离散化等操作后送入上下文层的上下文库.上下文层是本系统的核心,主要由上下文库、本体库、规则库、应用程序日志库、规则生成器、规则匹配器和日志采集器所组成.规则生成器定期地从上下文库和应用程序日志库中提取相关记录,并将利用本文所提出的方法生成的规则放入规则库中.当上下文状态改变时,规则匹配器利用上下文状态和规则库中的规则进行匹配操作,并根据结果自动调用应用程序中的相关接口函数.日志采集器则主要负责记录用户操作.另外,各种上下感知应用,如"综合智能业务平台"上的信息服务等工作在应用层.

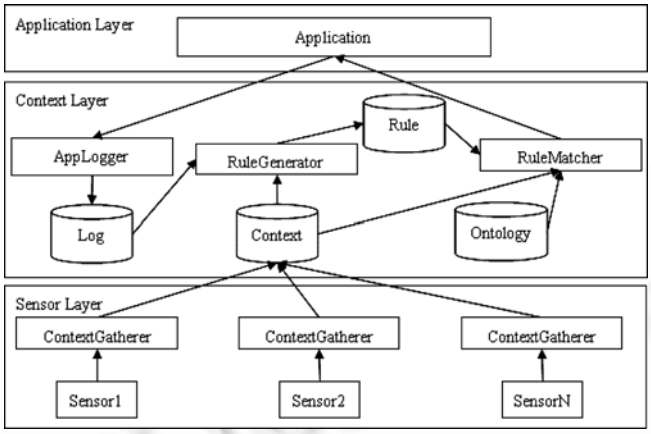

Fig.10 Architecture for prototype

图 10 原型系统结构图

规则生成器基于本文第 3 节提出的算法实现.规则匹配器结合了 Jena 2 语义网工具包提供的通用规则推理机和本文第 4 节提出的算法来实现.在进行规则匹配操作时,首先利用通用规则推理机进行精确匹配操作,如果匹配失败,就调用本文提出的基于语义距离的规则匹配算法,进行近似匹配操作.

6 总 结

现有的上下文感知系统中的规则是系统开发者或用户手工定义的,难以适应用户多样化的使用习惯.本文从上下文的概念和分类方法出发提出了一种基于本体的上下文表示方法,并与已有的上下文表示方法进行了比较.在此基础上,提出了一种基于粗糙集理论的上下文感知规则生成方法,该方法将上下文感知系统视为一种决策信息系统,并利用基于可辨识矩阵的方法对上下文信息进行约简,进而生成规则.为了便于与本体论结合,本文直接使用粗糙集理论进行规则生成,今后工作中将考虑与 ID3、C4.5 等决策树生成算法进行融合,以提高规则生成的效率,但是首先需要解决本体论与 C4.5 算法及决策树剪枝操作的结合问题.由于在规则生成阶段所能利用的数据有限,生成的规则可能难以完全覆盖上下文取值范围,为了增加规则匹配的准确率,本文还设计了一种基于语义距离的规则匹配算法.另外,为了验证所提出算法的有效性,本文实现了一个原型系统.我们下一步将对本文所提出的方法加以改进,以应用于文献[19]中提出来的新的支持普适服务的一体化可信网络环境中.

References:

[1] Chen G, Kotz D. A survey of context-aware mobile computing research. Technical Report, TR2000-381, Hanover: Department of Computer Science, Dartmouth College, 2000.

[2] Gu T, Pung HK, Zhang DQ. Toward an OSGi-based infrastructure for context-aware applications. Pervasive Computing, 2004,3(4): 66−74.

[3] Nishigaki K, Yasumoto K, Shibata N, Ito M, Higashino T. Framework and rule-based language for facilitating context-aware computing using information appliances. In: Martin DC, ed. Proc. of the 25th IEEE Int'l Conf. on Distributed Computing Systems Workshops. Los Alamitos: IEEE Computer Society, 2005. 345−351.

[4] Chen H, Finin T, Joshi A. Using OWL in a pervasive computing broker. In: Cranefield S, Finin TW, Tamma, VAM, Willmott S, eds. Proc. of the Workshop on Ontology in Open Agent Systems. Aachen: CEUR Publications, 2003. 9−16.

[5] Gu T, Pung HK, Zhang DQ. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications, 2005,28(1):1−18.

[6] Ejigu D, Scuturici M, Brunie L. An ontology-based approach to context modeling and reasoning in pervasive computing. In: Ceballos S, ed. Proc. of the 5th IEEE Int'l Conf. on Pervasive Computing and Communications, Workshops (PerCom 2007). Los Alamitos: IEEE Computer Society, 2007. 14−19.

[7] Dmitry K, Dmitry V. An algorithm for rule generation in fuzzy expert systems. In: Werner B, ed. Proc. of the 17th Int'l Conf. on Pattern Recognition. Los Alamitos: IEEE Computer Society, 2004. 212−215.

[8] An J, Wang G, Wu Y, Gan Q. A rule generation algorithm based on granular computing. In: Hu X, Liu Q, Skowron A, Lin TY, Yager RR, Zhang B, eds. Proc. of the IEEE Int'l Conf. on Granular Computing. Piscataway: IEEE Computer Society, 2005. 102−107.

[9] An L, Tong L. Decision rule generation based on similarity relation. In: Proc. of the 6th World Congress on Intelligent Control and Automation. Piscataway: IEEE Computer Society, 2006. 5915−5918. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1714213

[10] Jena—A Semantic Web Framework for Java. 2006. http://jena.sourceforge.net/index.html

[11] Yang SJH, Zhang J, Chen IYL. A JESS-enabled context elicitation system for providing context-aware Web services. Expert Systems with Applications, 2008,34(4):2254−2266.

[12] Dey AK. Understanding and using context. Personal and Ubiquitous Computing, 2001,5(1):4−7.

[13] Dey AK, Abowd GD. Towards a better understanding of context and context-awareness. In: Gellersen HW, ed. Proc. of the 1st Int'l Symp. on Handheld and Ubiquitous Computing. LNCS 1707, London: Springer-Verlag, 1999. 304−307.

[14] Dix A, Rodden T, Davies N, Trevor J, Friday A, Palfreyman K. Exploiting space and location as a design framework for interactive mobile systems. ACM Trans. on Human Computer Interaction, 2000,7(3):285−321.

[15] Jensen R, Shen Q. Fuzzy-Rough sets assisted attribute selection. IEEE Trans. on Fuzzy Systems, 2007,15(1):73−89.

[16] Zhang P. The research and implementation of semantic based Web services discovery [MS. Thesis]. Beijing: Tsinghua University, 2005 (in Chinese with English abstract).

[17] Ganesan P, Garcia-Molina H, Widom J. Exploiting hierarchical domain structure to compute similarity. ACM Trans. on Information Systems, 2003,21(1):64−93.

[18] Xu M, Meng XW, Chen JL, Mei X. The research of load balancing for integrated service platform. Journal of Beijing University of Posts and Telecommunications. 2006,5(Supl.):94−97 (in Chinese with English abstract).

[19] Zhang HK, Su W. Fundamental research on the architecture of new network-universal network and pervasive services. Acta Electronica Sinica, 2007,35(4):593−598 (in Chinese with English abstract).

附中文参考文献:

[16] 张钋.基于语义的网络服务匹配机制的研究与实现[硕士学位论文].北京:清华大学,2005.

[18] 徐萌,孟祥武,陈俊亮,等.综合业务平台负载均衡算法的研究.北京邮电大学学报,2006,5(增刊):94−97.

[19] 张宏科,苏伟.新网络体系基础研究——一体化网络与普适服务.电子学报,2007,35(4):593−598.

刘栋(1981－),男,北京人,博士生,主要研究领域为语义 Web 服务,上下文感知.

陈俊亮(1933－),男,教授,博士生导师,中国科学院与中国工程院院士,CCF 高级会员,主要研究领域为网络智能化.

孟祥武(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为通信软件,网络服务.

夏亚梅(1976－),女,博士,主要研究领域为语义网.