

一种场景驱动的情境感知计算框架

徐步刊 周兴社 梁韵基 王海鹏 於志文

(西北工业大学计算机学院 西安 710129)

摘要 情境感知系统通常以情境信息直接驱动上层服务或应用。普适计算环境的情境信息在来源及种类上存在较大差异,情境信息的多样性导致由情境直接驱动的情境感知系统的扩展性、稳定性不足。对此提出一种场景驱动的情境感知计算框架。该框架以场景为基础,屏蔽了原始情境信息的异构性和多样性;以场景识别为核心,由系统中的基本情境信息识别当前用户的场景信息,由场景信息驱动相关的应用。该框架简化了系统设计的复杂度,有助于提高系统的扩展性和稳定性;同时场景识别采用神经网络的算法,避免了因采用推理所带来的知识库暴涨问题。所开发的原型系统验证了框架的有效性。

关键词 情境感知,场景识别,人工神经网络,普适计算

中图分类号 TP391 **文献标识码** A

Situation-driven Framework for Context-aware Computing

XU Bu-kan ZHOU Xing-she LIANG Yun-ji WANG Hai-peng YU Zhi-wen

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract Context-aware systems generally use contexts to drive higher applications. Because context information is diverse in types and values, the systems developed are poor in extensibility and stability. This paper proposed a situation-driven model for context-awareness computing. The framework is based on situation information, and it can shield the heterogeneity and diversity of original context information. Situation-recognition is the kernel of this framework. The situation is recognized from basic context information, and is used to drive related applications. This framework simplifies the processing on designing a system, and is helpful in improving the systems' extensibility and stability. The mechanism of situation recognition based on neural network avoids the problem of knowledge explosion when it uses the mechanism of reasoning. A prototype system was implemented, and it validated the proposed infrastructure.

Keywords Context-awareness, Situation recognition, Artificial neural network, Pervasive computing

1 引言

普适计算^[1,2]模式的提出,使得人们能够获得适时、适宜的计算服务,而不必了解具体是何种设备,如何提供服务等信息。普适计算的本质是要实现物理空间和信息空间的融合,在这个融合的空间中,人们可以随时随地、透明地获得数字化的服务。随时随地和透明性是普适计算的两个基本特征。

为了向用户提供随时随地、透明的服务, Schilit 提出了情境感知计算^[3] (Context-aware computing), 其逐渐成为普适计算的重要分支之一。情境感知计算通过获取用户相关情境信息为系统决策提供判断的依据,从而向用户提供适时、适地、无缝、透明的服务。情境感知计算所处理的数据对象被称为情境(Context)。“情境是环境本身以及环境中各实体所明示或隐含的可用于描述其状态(含历史状态)的任何信息”^[4]。其中,实体可以是人、地点等物理实体,也可以是诸如软件、程

序、网络等虚拟实体。

现有的多数情境感知计算系统采用以情境直接驱动应用的模型,即系统获取智能空间中环境和用户相关的情境信息,然后通过情境的推理确定系统需要执行的应用。然而智能空间中,情境信息的来源和种类多种多样。以情境直接驱动应用的设计模式势必导致系统的复杂度和稳定性随着规模的增大而急剧下降。

为了解决这个问题,提出了场景驱动的情境感知计算框架,其将情境信息进一步融合、抽象成场景,再依据场景做出决策,为用户提供最适于当前时间、地点、任务的服务。在场景的识别算法上,采用基于神经网络的场景识别算法,避免了采用情境推理时存在的知识库暴涨、逻辑复杂、设计困难等问题。

为了验证这一框架的有效性,实现了面向老年人的智能家庭原型系统,系统获取情境信息识别出老人所处的场景,从

到稿时间:2011-04-07 返修时间:2011-07-14 本文受国家自然科学基金项目(60803044, 60903125), 国家 863 基金项目(2009AA011903), 教育部“新世纪优秀人才支持计划”(NCET-09-0079)资助。

徐步刊(1986-),男,硕士生,主要研究领域为普适计算、上下文感知系统, E-mail: seekerxu@163.com; 周兴社(1955-),男,教授,博士生导师,主要研究领域为嵌入式计算、分布式计算、传感器网络; 梁韵基(1984-),男,博士生,主要研究领域为普适计算、上下文感知系统; 王海鹏(1975-),男,副教授,主要研究领域为普适计算、资源管理; 於志文(1977-),男,教授,主要研究领域为普适计算、智能信息处理、个性化技术。

而提供相应的服务。例如,当老人正在观看电视时,若有电话接入,系统采集环境和用户位置等情境信息,并对这些情境信息进行融合分析,判断出老人正处于来电提醒场景中,因此调用提醒服务对老人进行提醒,而提醒服务则会根据老人的位置自动把来电的信息转移到电视屏幕上对其进行提醒。

2 相关工作

情境感知计算的应用开发是一项十分繁杂和耗时的的工作。一个情境感知系统涉及情境信息的获取、表示、存储、解释推理,直至最后向用户提供适时、适宜的服务等内容。因此有必要采用合适的计算框架简化应用的设计、原型系统实现及测试等工作。

Context Toolkit^[5]是基于面向对象技术的情境感知计算框架,它将相关的情境收集到一个组件中,然后对这些情境信息进行处理,开发人员需要编写相关的情境处理代码。Gaia^[6]是类似传统操作系统的情境感知计算框架,通过将智能空间及所包含的资源抽象为可编程实体以方便开发人员构建相关应用。SOCAM^[7]是一个基于本体的情境感知计算框架,在情境获取、推理和访问等方面提供了支持。DCASI^[8]则综合了以往情境感知系统框架的优点,通过按需分散获取机制和双库集中共享机制,探索解决可扩展问题和效率问题的途径。以上框架都以情境信息驱动上层应用,虽在一定程度上简化了情境感知应用的开发,却忽视了情境信息的多样性对系统设计实现的影响。由于缺乏对原始情境数据的抽象,因此开发人员需要针对各种情境进行处理,这不仅浪费了系统中的计算资源,增加了通信负载,而且不利于系统的开发和维护。

一些研究人员意识到这种不足,他们将原始情境数据进行融合,得到高级情境数据,从而丰富了情境数据的语义性。利用高级情境数据来驱动上层应用或者服务,如用户的行为、意图等就是典型的高级情境数据。情境融合的研究主要有 Sentient Object Model^[9],其采用的是贝叶斯网络的方法对情境信息进行融合处理;Solar^[10]提出了一种情境融合网络等。但是用户的行为、意图等高级情境信息仍然存在多样性,有必要对其进行进一步的融合、抽象。

于是出现了一些场景感知的应用研究,文献[11-13]研究了场景感知的应用开发,并对场景进行了描述和建模。文献[12]提出了场景的接口描述语言,文献[13]则运用本体的思想对场景进行建模和推理。将情境信息融合为场景信息,可以进一步简化情境感知应用的开发。然而场景在这些框架中并不处于核心地位,而且在场景的识别上多采用基于规则的推理,当情境信息过于庞大时,规则的制定与维护将变得越来越困难。本文提出的场景驱动的情境感知计算框架将场景作为应用的驱动,在场景识别上采用了神经网络的算法,有效克服了以上的不足。

3 场景驱动的情境感知计算框架

智能空间是普适计算模式的体现,其本质是要以用户为中心,实现信息空间与物理空间的融合,为用户提供透明服务。对一个智能空间而言,用于采集用户和环境状态信息的设备是有限的,系统根据情境感知的结果所做出的动作也是有限的,可以用有限集和映射等形式化方法对这个智能空间进行描述。

每个设备所采集的状态信息用状态变量表示,则任一时

刻智能空间的状态可表示为一个 $n+1$ 维向量 $V=\{t, v_0, v_1, \dots, v_{n-1}\}$ (其中 n 是该智能空间中的状态总数; v_0, v_1, \dots, v_{n-1} 表示状态变量,可以是数值形式,也可以是逻辑形式; t 用于表示时间状态)。若用向量 $A=\{a_0, a_1, \dots, a_{m-1}\}$ 表示系统所提供的动作组合,其中 a_0, a_1, \dots, a_{m-1} 每一分量表示系统所执行的一次动作,则一个普适计算系统模型本质上是要建立 $V \xrightarrow{f} A$ 映射。然而设备所获取的原始状态信息大多是无意义的数值或者逻辑值,直接根据这些状态信息来调整系统动作,所进行的开发将是十分繁杂和耗时的。

情境感知计算框架着眼于智能空间下各个设备所明示或隐含的关于环境的信息即情境信息,将设备原始的无意义的状态信息映射为可理解的情境信息,再通过对情境信息的建模、存储、推理来达到目的,典型的如 Context Reasoning^[14]。若用 $C=\{t, c_0, c_1, \dots, c_{l-1}\}$ 表示一种情境属性集,则情境感知计算的数学模型可以表述为 $V \xrightarrow{f_1} C \xrightarrow{f_2} A$ 。这种方法符合人类的思维习惯,大大降低了开发难度。然而,智能空间中,情境的种类和取值几乎无穷无尽,随着系统规模的增加,系统的计算规模迅速增加,规则库急剧膨胀,从而大大降低了推理的效率与准确性,最终导致系统的可靠性与稳定性下降。

本文引入场景的概念,进一步将情境信息融合为场景,根据场景决定系统的动作,即 $V \xrightarrow{f_1} C \xrightarrow{f_2} s_i \xrightarrow{f_3} A$,其中 s_i 为场景,并在此基础上构建了场景驱动的计算系统框架。由于智能空间中的场景数极少,因此以场景来驱动应用能有效地降低开发的难度和复杂度。同时系统复杂度的降低也为系统的可靠性与稳定性带来了提升。

3.1 场景的定义

普适计算环境下,系统根据用户意图做出的动作往往具有持续性,即用户在某一时期内的意图是不变的。例如在一个面向老年人的智能家庭中,当发现老人的身体状况处于异常状态时,系统就会向其家属或医生发出警报,而这一动作在异常解除前会一直持续下去。普适计算环境的这种性质导致了场景这一概念的产生。

场景(Situation)的概念在早期的一些普适计算研究中已有所提及,但并未引起足够重视。在这些研究中,场景可以视为一类高级情境(高级情境是指不能由感知设备直接获取,而由基本情境通过情境融合得到的情境数据),同时场景又有体现用户意图,以用户为中心的特点。如一些研究将场景用于表示系统针对用户意图执行的一系列任务或提供的相关服务^[15];另一些研究则把场景视作环境情境^[16]。

但是,对一个智能空间下的计算系统而言,为了向用户提供普适的服务,必须同时考虑服务提供的时机和方式,无论忽略其中的任何一部分,都将违背普适计算的理念。为此,本文综合考虑场景下所包含的情境信息和所需执行的动作,对智能空间中的场景作如下定义:

场景是智能空间在某一状态下所包含的情境信息及其所需执行动作的集合。用形式化的方式描述为 $\langle C, f, A \rangle$,其中 C 表示该场景下相关情境信息的集合; A 表示系统在该场景下所做的动作集合; f 表示集合 C 与集合 A 之间的映射关系。

在本文的定义中,场景是普适计算系统中状态和执行的抽象实体,一个场景包含了用于表征其状态的基本情境信息集合和在场景下系统根据用户需求所执行的动作集合。一个

智能空间可以通过某种算法划分为若干场景,系统通过对场景的识别,最终完成将环境及用户相关的情境映射到具体服务上的任务。

3.2 场景驱动的情境感知计算系统层次结构

场景驱动的普适计算系统需要完成环境及用户相关情境的采集、场景识别、服务调度等任务。本文采用层次化结构来描述场景驱动的情境感知计算系统。根据计算系统中各计算实体所需处理数据和完成任务的不同,系统可以划分为物理层、情境基本层、场景抽象层、应用层等4个层次,如图1所示。各层的功能描述如下。

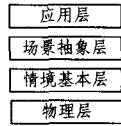


图1 场景驱动的情境感知计算系统层次结构

物理层直接操作环境中的物理设备,与各种感知设备交互;通过各种传感器采集普适空间的原始状态数据,并为上层提供相关数据;可以屏蔽底层的硬件差异,实现物理设备的透明性,有助于提高系统的可扩展性。

情境基本层负责管理相关的情境信息,将经由物理层获取的原始数据封装为情境实例,实现情境数据的统一封装,提高情境信息的可用性。

场景抽象层包含用于识别具体场景的逻辑或规则等信息,负责对一个或多个情境实例进行处理、融合。场景抽象层从大量的原始情境中抽取识别出具体的场景,并将场景信息映射到应用层具体的服务上。

应用层是普适空间中所有可用的服务的集合,提供对服务请求进行响应的功能。服务是智能环境下系统所执行的、具有某些共性的动作经过抽象而来的。应用层在接收到服务请求时,将从系统提供的服务集合中选择相关服务来响应服务请求。

3.3 场景驱动的情境感知计算框架

场景驱动的普适计算框架以图1所示的4层结构为基础进行扩展,根据每层的功能设计了各层的计算实体,如图2所示。

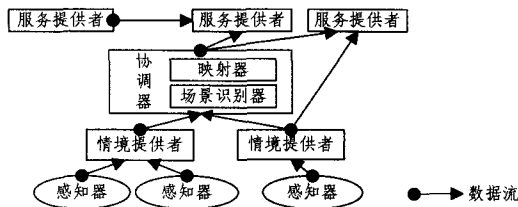


图2 场景驱动的计算框架

感知器位于物理层,采集环境及用户相关的原始数据并做一定的处理后供情境提供者使用。一般的传感设备不具有感知器的全部能力,需要由其他计算设备帮其完成数据处理及传输等任务。感知器是对于传感设备和相关计算实体的抽象。系统中存在大量的感知器,它们实时地监控系统状态的变化,并对原始数据进行预处理和实时传输。感知器可以被动态地创建和删除,这一特点有助于提高系统的可扩展性。

情境提供者是向上层提供情境实例的抽象计算实体。情境实例是由感知器传输的数据经过预处理、统一封装而成的。一个情境提供者仅负责一种情境,但向其提供原始数据的感知器可以是一个或多个。在具体的应用开发实现时,多个提

供者可能被设计到一块,我们把其中具体负责每种情境的部分都视为一个提供者。情境提供者的主要数据接收方是协调器,但是有些高层的服务也需要某些低层情境的支持(如基于位置的服务等),因此情境提供者也向某些服务直接提供情境实例。情境提供者也具有动态创建及加入的功能。

协调器包括场景识别器和映射器,是系统的核心,负责由多个情境实例融合得到相应的场景信息,并由相关的场景映射到需要调用的服务上。场景识别器对下层的情境实例进行融合、抽象,判断所处的场景。场景识别器需要根据具体的场景识别算法进行设计,第4节将给出本文所用的场景识别算法。映射器则负责将识别出的具体场景映射到上层提供的服务上,以完成服务调度的功能。

服务提供者是系统动作的执行实体,实现某种具体的服务。所实现的服务是由智能环境下系统针对各场景所执行的具有某些共性的动作经过抽象而来的,一个场景所执行的动作可以由一个或多个服务来完成,某些服务提供者还需要低层的情境提供者提供情境信息的支持,部分服务提供者之间还存在相互协作的关系。服务提供者同样具有动态创建及加入的功能。

在系统运行过程中,感知器实时采集环境及用户相关的原始数据并对其进行一定的预处理;情境提供者则根据需求从一个或多个感知器那里获取相关的原始数据,封装为可理解的情境实例(如一个感知器负责感知用户的位置状态,其所采集到的信息是无意义的原始的三维坐标,负责位置情境的情境提供者去该感知器处获取三维坐标信息,将其封装为可理解的位置情境实例,用来表示用户所处的具体位置);情境提供者封装后的情境实例被送往协调器中的场景识别器进行处理,场景识别器处理的结果为具体的场景。当映射器发现场景改变时,根据识别出的场景调用所需的服务提供者。框架中的各计算实体通过松耦合的方式进行通信,彼此各有分工又相互协作。松耦合机制使得系统便于扩展和维护。

4 场景识别算法

情境信息的融合普遍采用规则推理的方式。本文在前面已经指出,这种方式在系统规模增加时,规则库将急剧膨胀,从而大大降低了推理的效率与准确性,最终导致系统的可靠性与稳定性下降。为此,需要引入更高效的算法,避免类似的知识膨胀问题。由于一个智能空间下的场景数是有限的,在应用开发的设计阶段,场景基本已经确定,因此一个场景识别问题完全可以转化为场景分类问题。基于此,提出采用神经网络进行场景分类,完成场景识别的功能。

4.1 基于神经网络的场景识别模型

神经网络^[17]具有信息存储与处理一体化的特点,美国神经网络科学家 Hecht Hielsen^[18]对其的一般定义是:“神经网络是由多个非常简单的处理单元彼此按某种方式相互连接而形成的计算系统,该系统是靠其状态对外部输入信息的动态响应来处理信息的”。

神经网络的基本组成单元是感知器^[19],采用 BP(Back Propagation)算法的多层感知器(Multilayer Perceptron)模型^[20]是至今为止应用最广泛的神经网络。利用神经网络可以高效地完成场景识别的工作。图3是本文采用BP神经网络进行场景识别的模型。

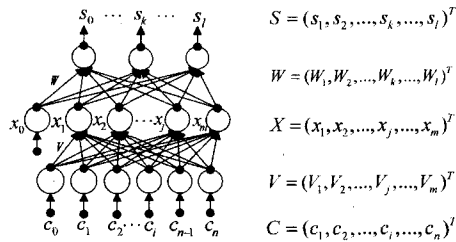


图3 场景识别模型

输入向量 $C = \{c_1, c_2, \dots, c_i, \dots, c_n\}^T$ 是神经网络的输入, 其中 $c_1, c_2, \dots, c_i, \dots, c_n$ 每一分量均表示一种情境的属性值。有些情境属性可能不是数值形式, 必须先转化为数值表示形式, 再通过转换函数将各个指标的数值转换到 $[0, 1]$ 区间上。 $c_0 = -1$ 是为隐层神经元引入阈值而设置的。

隐层输出向量为 $X = \{x_1, x_2, \dots, x_j, \dots, x_m\}^T$, 其中 $x_0 = -1$ 是为输出层神经元引入阈值而设置的。

输出层向量为 $S = \{s_1, s_2, \dots, s_k, \dots, s_l\}^T$, 输出向量是神经网络的输出, 可以表示成数值形式也可以表示成编码等其他形式。神经网络的输出经过解释就是识别出来的具体场景。

输入层到隐层之间的权值矩阵用 V 表示, $V = \{V_1, V_2, \dots, V_j, \dots, V_m\}^T$, 其中列向量 V_j 为隐层第 j 个神经元对应的权向量; 隐层输出层之间的权值矩阵用 W 表示, $W = \{W_1, W_2, \dots, W_k, \dots, W_l\}^T$, 其中列向量 W_k 为输出层第 k 个神经元对应的权向量。

对各变量有以下的表达式:

输出层, 有:

$$s_k = f(\text{net}_k) \quad k=1, 2, \dots, l \quad (1)$$

$$\text{net}_k = \sum_{j=0}^m \omega_{jk} x_j \quad k=1, 2, \dots, l \quad (2)$$

隐层, 有:

$$x_j = f(\text{net}_j) \quad j=1, 2, \dots, m \quad (3)$$

$$\text{net}_j = \sum_{i=0}^n v_{ij} c_i \quad j=1, 2, \dots, m \quad (4)$$

以上各式中, 变换函数 $f(x)$ 均为单极性 Sigmoid 函数:

$$f(x) = \frac{1}{1 + e^{-x}}$$

式(1)~式(4)共同构成了场景识别的数学模型。

4.2 神经网络的训练学习

神经网络在应用前需要经过一定的训练, BP 神经网络采用的是误差反向传播算法 (Error Back Propagation)^[21]。其基本思想是, 学习过程由信号的正向传播与误差的反向传播两个过程组成。正向传播时, 输入样本从输入层传入, 经各隐层逐层处理后传向输出层。若输出层的实际输出与期望的输出不符, 则转入误差的反向传播阶段, 即以某种形式将输出误差向输入层逐层反传, 对各单元权值进行修正。权值调整的过程就是网络训练学习的过程, 当网络输出的误差减少到可接受的程度, 或进行到预先设定的学习次数时, 训练过程才算结束。

图4所示的框图是BP神经网络训练调整的信号流向。由图4可以看出, 前向过程是: 输入信号 X 从输入层进入后, 通过隐层各节点的内星权向量 V_j 得到该层的输出信号 Y ; 该信号向前输入到输出层, 通过其各节点内星权向量 W_k 得到该层输出 O 。反向过程是: 在输出层期望输出 d 与实际输出 O 相比较得到误差信号 δ^o , 由此可计算出输出层权值的调整

量; 误差信号 δ^o 通过隐层各节点的外星权向量反传至隐层各节点, 得到隐层的误差信号 δ^h , 由此可计算出隐层权值的调整量。

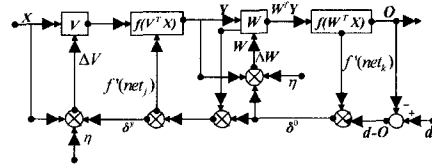


图4 BP神经网络训练框图

5 应用案例

本文基于场景驱动的情境感知计算框架实现了面向老年人的智能家庭原型系统 (见图5), 并在老年人安居与智能辅助技术实验室进行了相关实验。原型系统着眼于向老年人提供安全舒适的生活环境, 设计了来电提醒和突发事件提醒 (如煤气没有及时关掉、水壶烧开没有被取走) 两个典型的场景。智能家庭中布置有 UWB (Ultra-Wideband) 定位系统、来电检测设备及温湿度等环境传感器。当系统识别出特定的场景时, 将通过多媒体提醒的方式向用户发出适时、适宜的提醒。

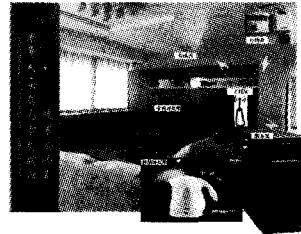


图5 面向老年人的智能家庭原型系统

系统的工作流程为: 实验室中所布置的传感设备将采集到的数据送到计算设备 (服务器) 进行处理, 以识别当前所处的场景。系统根据场景调用提醒服务, 显示该场景的提醒内容。提醒服务则通过判断用户所处的位置, 将提醒内容显示在相应的设备上。例如, 在一个来电场景中, 当老人在客厅看电视时, 若有电话接入, 则会在电视上显示来电信息; 若出现水壶烧开却没有被及时取走的突然事件提醒场景时, 系统也会根据老人的位置在电视或触摸屏上显示水开提醒。

由于系统的场景识别器是基于神经网络设计的, 在应用前需要经过一定的训练。神经网络的输入为数值化的情境实例, 包括电话状态、环境温度、环境湿度、压力值等, 输出为识别的结果, 以数值编码表示所处的场景。训练样本通过人为观察获得, 实验中的场景数设置为 12 个 (分别为 4 个基本的来电场景、水开场景、服药场景、起床场景, 基本场景通过组合得到的 7 个扩展场景, 及无任何动作的空场景)。在训练样本数足够时, 识别的准确率接近 85%。原型系统所设计的提醒服务根据不同的场景负责把提醒的内容显示在相关的设备上。由于提醒是随用户的位置转移的, 提醒服务还需要位置情境的支持。

6 实验结果

实验对原型系统的可行性与系统性能进行了评估。由于场景识别是本文框架的核心, 因此系统的可行性主要是针对本文提出的基于神经网络的场景识别准确率进行分析, 系统性能则通过系统响应时间 (主要是场景识别时间) 及系统消耗的资源来衡量。

6.1 场景识别准确率分析

神经网络参数:输入神经元6个,分别为温度、湿度、智能床压力、智能药箱压力、位置、电话状态等6种情境实例,其中温湿度传感器布置于电热水壶旁,电热水壶在烧开时会鸣叫示意水开;输出神经元为4个,分别对应来电提醒场景、水开提醒场景、服药场景和起床场景,用逻辑数值表示。当输出值为1时,表示处于对应场景,输出值为0时,表示不处于该场景(如输出1100表示处于来电及水开提醒场景中,0000表示不需处理的空场景)。

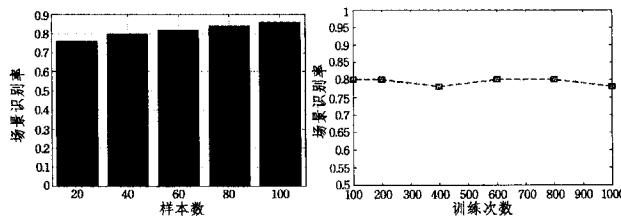
实验样本来自于老年人安居与智能辅助实验室,如表1所列。表1中的湿度与压力值是传感器采集到的数据经过处理得到的,未换算成标准的量纲;电话状态用0、1表示,1表示有来电;位置用1~6的数值表示6个不同的位置;场景的识别结果通过实验人员观察判断得到。实验样本中的前100个作为神经网络的训练样本集,后50个作为测试样本集。

表1 实验样本

序号	温度	湿度	压力(智能床)	压力(智能药箱)
1	32	1060	69	329
2	25	739	70	327
3	18	572	71	329
4	29	964	70	71
5	40	1327	70	330
...
150	54	1444	69	329

序号	电话状态	位置	水开场景	服药场景	起床场景
1	0	1	0	0	0
2	1	1	0	0	0
3	0	5	0	0	0
4	0	6	0	0	0
5	0	2	0	1	0
...
150	0	3	0	0	0

场景识别准确率与训练样本、训练次数的关系分别如图6所示。



(a) 样本数与场景识别率的关系 (b) 训练次数与场景识别率的关系

图6 场景识别率

图6(a)是在训练次数为1000时场景识别准确率与样本数的关系。结果显示,当训练样本集增大时,场景识别准确率有稳固的提高。在训练样本足够大时,场景识别准确率接近85%。图6(b)是在训练样本数为40的条件下场景识别率与训练次数的关系。图中反映的场景识别率随训练次数的增加没有较大的提高,这表明在采用神经网络进行场景识别时,可以在较少的训练次数下达到较为理想的识别准确率。因此,通过选择适当的训练次数可以在一定程度上缩短训练过程,从而减少系统的开销,提高实用性。

隐层神经元的个数与场景识别准确率的关系如表2所列。结果显示隐层神经元个数的选择与场景识别准确率影响较大,合理选择隐层神经元个数能较大提高场景识别的准确率。在隐层神经元个数的确定上,尚无理想的方法,在实际应

用中需要通过实验确定。

表2 隐层神经元个数与场景识别率的关系

隐层神经元个数	3	4	5
场景识别率	0.86	0.84	0.72

6.2 系统性能评估

本文的原型系统在老年人安居与智能辅助实验室实现,关键技术的实现已在以上各节做了相应介绍。后台计算机的配置如下: Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz 2.99GHz,内存2GB,操作系统为Windows XP sp3。实验中,重点考察系统在运行过程中的响应时间,通过50次实验取平均值获得。

系统的运行过程大致可以分为3个阶段,每个阶段分别计时得到的结果如下:初始化包括神经网络参数的预读、各硬件设备的初始化等,耗时3369ms;对各计算单元的注册响应时间为242ms;场景识别时间为47ms,响应场景变化的时间为126ms,运行过程中的最大内存消耗仅为10980kB。实验结果表明,系统初始化过程稍长,但在合理的范围内;系统的响应时间为毫秒级,系统的响应速度与资源消耗完全满足在实际应用中为用户提供无缝、自然的服务。

结束语 本文引入并讨论了智能空间下场景的概念,提出了场景驱动的情境感知计算框架。框架将情境信息进一步抽象成场景,由场景驱动应用的设计模式能有效减少系统设计的复杂度和系统开发的难度;同时框架采用松耦合的设计方法,提高了系统的扩展性和稳定性。在场景识别的算法上,本文利用神经网络具有的信息存储与处理一体化等特点,避免了传统的基于情境推理方法所带来的知识爆炸等问题。原型系统验证了本框架的可行性,初步体现了其优越性及应用的前景。

框架在场景识别上所采用的神经网络方法由于其自身的特点,在应用前需要进行训练且训练数据获取不易。未来将对这一方法进行改进,使其能在应用过程中具备自学习、自适应等能力。

参考文献

- [1] Weiser M. The computer for the 21st century[J]. Scientific American, 1991, 265(3): 66-75
- [2] 徐光祐,史元春,谢伟凯. 普适计算[J]. 计算机学报, 2003, 26(9): 1042-1050
- [3] Schilit B N, Adams N, Want R. Context-aware Computing Applications[C]//IEEE Workshop on Mobile Computing Systems and Applications. Washington, USA: IEEE Computer Society, 1994
- [4] Dey A, Abowd G. Towards a better understanding of context and context-awareness[C]// Workshop on the what, who, where, when and how to context-awareness at CHI 2000. New York, ACM Transactions on Computer Systems, 2000
- [5] Salber D, Dey A K, Abowd G D. The Context Toolkit: Aiding the Development of Context-enabled Applications[C]//Proc. of Conference on Human Factors in Computing Systems. Pittsburgh, USA, 1999: 434-441
- [6] Román M, Hess C, Cerqueira R, et al. Gaia: A Middleware Infrastructure to Enable Active Spaces[J]. IEEE Pervasive Computing, 2002
- [7] Gu Tao, Pung H-K, Zhang Da-qing. A service-oriented middleware for building context-aware services [J]. Network and Com-

puter Applications, 2005, 28(1): 1-18

[8] 於志勇, 周兴社, 王海鹏, 等. 动态上下文知识的获取与共享[J]. 计算机科学, 2009, 36(9): 218-223

[9] Biegel G, Cahill V. A Framework for Developing Mobile, Context Aware Applications[C]//Proc. of the 2nd IEEE Conference on Pervasive Computing and Communications. Orlando, USA, 2004

[10] Chen G, Li M, Kotz D. Design and Implementation of a Large-scale Context Fusion Network[C]//1st International Conference on Mobile and Ubiquitous Systems. 2004

[11] Yau S S, Gong H D, Seth H S. Development and runtime support for situation-aware application software in ubiquitous computing environments[C]//28th Annual International Computer Software and Application Conference. Hong Kong, 2004

[12] Yau S S, Wang Y, Huang D, et al. Situation-aware contract specification language for middleware for ubiquitous computing[C]//Proc. of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems. California, 2003

[13] Yau S S, Liu J. Hierarchical Situation Modeling and Reasoning for Pervasive Computing[C]//Proc. of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems. 2006

[14] Guo Bin. Sixth-sense: Context Reasoning for Potential Objects

Detection in Smart Sensor Rich Environment[C]//Proc. of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology. Hong kong, China, 2006

[15] Marti P, Gabrielli F, Petroni L, et al. Situated Interactions in Art Settings[C]//Proc. Workshop on Situated Interaction in Ubiquitous Computing. April 2000

[16] Schiele B, Starner T, Rhodes B, et al. Situation Aware Computing with Wearable Computers [J]. Augmented Reality and Wearable Computers, 1999

[17] 韩力群. 神经网络教程[M]. 北京: 北京邮电大学出版社, 2006

[18] Hecht-Nielsen R. Kolmogorov's mapping neural network existence theorem[C]//Proc. 1987 IEEE International Conference on Neural Networks. New York; IEEE Press, 1987

[19] Minsky M L, Papert S. Perceptrons[M]. Cambridge, MA; MIT Press, 1969

[20] Rumelhart D E, Hinton G E, Williams R J. Learning internal representation by error propagation [C]//Parallel Distributed Processing. Cambridge, MA; MIT Press, 1986

[21] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by back-propagation errors[J]. Nature, October 1986

(上接第 211 页)

的能量, ecgHrv 表示心率变异性, ecgHrvDistr 表示心率分布, ecgdeltaNN 表示相邻正常心动周期差值均方的平方根, SDNN 表示正常心动周期的标准差, ecgTTDWON 表示 T 波下降沿间距, ecg_rmsd 表示相邻正常周期差值的平方和的平方根; max 为最大值, min 为最小值, mean 为均值, median 为中值, std 为方差, range 为幅值变化范围。

表 4 最优特征子集

情感	最优特征子集	维数
Anger	ecgPQ-mean, ecgQS-std, ecgQS-min, ecgPampl-std, ecgHrvDistr-median, ecgabsRdiffampl-mean, ecgdeltaNN-mean, ecgTenergy-mean	8
Disgust	ecgR-mean, ecgQS-median, ecgRampl-min, ecg-Sampl-std, ecgPampl-median, ecgHrvDistr-std, ecgPT-median, ecgdeltaNN-mean	8
Fear	ecgP-range, ecgQS-mean, ecgRampl-min, ecgPT-std, ecg_rmsd, ecgdeltaNN-mean, ecgQRSenergy-mean	7
Grief	ecgP-min, ecgPQ-min, ecgPampl-std, ecgTampl-std, ecgdeltaNN-mean, ecgdeltaNN-median, ecgSDNN/SD Delta NN, ecgPenergy-median	8
Joy	ecgR-std, ecgP-min, ecgPQ-std, ecgST-mean, ecg-Sampl-min, ecgHrv-std, ecgHrvDistr-mean, ecgTTDWON-mean, ecgPT-std, ecgTampl-median, ecgabsTdiffampl-std, ecgdeltaNN-mean, ecgdeltaNN-median, ecgSDNN/SD Delta NN	14
Surprise	ecgP-range, ecgQ-min, ecgST-range, ecgPampl-mean, ecgPampl-median, ecgPampl-range, ecgHrvDistr-median, ecgTTDWON-mean, ecgTdiffampl-mean, ecg_rmsd	10

结束语 本文通过处理采集的 ECG 信号来提取原始特征集。通过分析特征间的相关性, 实现对原始特征集的降维。针对基本的二进制量子行为粒子群优化算法的不足, 提出了改进算法, 通过比较标准函数的测试结果, 改进算法比 BQP-SO 算法具有更好的全局搜索能力和稳定性。

情感识别实验结果表明, 通过相关性分析对 ECG 信号的原始特征集进行降维, 结合 SBQPSO 算法与 Fisher 分类器在

降维后的空间中选择特征具有可行性。在后期工作中, 重点将智能算法和不同的分类器结合起来进行特征选择, 以达到更好的识别效果。

参 考 文 献

[1] Picard R W, Vyzas E, Healey J. Toward Machine Emotional Intelligence: Analysis of Affective Physiological State[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(10): 1175-1191

[2] Kim K H, Bang S W, Kim S R. Emotion Recognition System Using Short-term Monitoring of Physiological Signals[J]. Medical Biology Engine Computer, 2004, 42: 419-427

[3] Haag A, Goronzy S, Schaich P, et al. Emotion Recognition Using Bio-sensors; First Step Towards an Automatic System [J]. Affective Dialogue Systems, 2004, 3068: 36-48

[4] 余祖龙, 周旭欣, 艾信友, 等. 基于离散平稳小波变换的心电信号去噪方法[J]. 科技创新导报, 2008, 04: 31-32

[5] Xu Ya, Liu Guang-yuan, Hao Min, et al. Analysis of Affective ECG Signals Toward Emotion Recognition[J]. Journal of Electronics (China), 2010, 27(01): 8-14

[6] Cheng Y Q, Zhuang Y M, Yang J Y. Optimal discriminant analysis using the rank decomposition[J]. Pattern Recognition, 1992, 25(1): 101-111

[7] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]//Proceedings of 2004 Congress on Evolutionary Computation. 2004: 235-331

[8] 孙俊. 量子行为粒子群优化算法研究[D]. 南京: 江南大学, 2009

[9] Jone K A D. An Analysis of the Behavior of a Class of Genetic Adaptive Systems [D]. No. 76-9381. University of Michigan, 1975

[10] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern Recognition Society, 1997, 30: 1145-1159