

# 基于属性子空间的孤立点内涵知识挖掘

王 越 刘亚辉 谭暑秋

(重庆理工大学计算机科学与工程学院 重庆 400050)

**摘 要** 孤立点通常都包含着重要的信息,挖掘出孤立点的内涵知识可以帮助用户更好地认知数据。通过给出的孤立点的原因属性子空间及其孤立度和孤立点的相似度等概念,提出了一个基于属性子空间的孤立点内涵知识挖掘算法,得到了每个孤立点的原因属性集,并结合聚类的思想把孤立点按照其相似性特征进行了分类,使每一类中的所有孤立点在一定精度下都具有相同的原因属性集。实验结果表明该算法是有效和实用的,且易用性较强。

**关键词** 孤立点,属性子空间,孤立点相似度,内涵知识

中图法分类号 TP181 文献标识码 A

## Finding Intentional Knowledge of Outliers Based on Attribute Subspace

WANG Yue LIU Ya-hui TAN Shu-qi

(College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400050, China)

**Abstract** Outliers usually contain important information, it can help improving the users' understanding of the data. New definitions of cause attribute subspace of outliers, degree of cause attribute subspace and similarity of outliers were given, and then an algorithm for finding intentional knowledge of outliers based on attribute subspace was proposed, the approach can obtain the cause attributes set of every outlier. Then the outliers were classified by their similarity combined with the thinking of clustering, all the outliers of every class have the same cause attributes set under certain precision. The experiment results show that the algorithm is effective and practical, and more ease of use.

**Keywords** Outliers, Attribute subspace, Outliers similarity, Intentional knowledge

孤立点(outlier)又叫做异常数据,是数据集中小部分新颖的、与常规数据模式显著不同的数据对象。在早期的数据挖掘中,孤立点通常被作为聚类过程的副产品,当作噪声处理或不被过多地关注。但是,一个人的噪声可能是另一个人需要的信号,孤立点通常包含着重要的信息,对孤立点进行检测和分析是数据挖掘中一个重要方面,也是一个非常有趣的数据挖掘课题。

孤立点的研究通常包含 3 个方面的子问题:①什么样的数据是不一致的,即孤立点的定义;②有效挖掘孤立点的方法;③孤立点的意义,即对孤立点的合理解释<sup>[1]</sup>。目前对孤立点的研究主要集中在对孤立点的检测上,即如何高效地挖掘出孤立点,研究人员已经提出了大量的孤立点挖掘算法<sup>[2]</sup>,而涉及到对挖掘出的孤立点进行分析、给出孤立点意义的文献还较少。然而,对挖掘出的孤立点进行分析,从而给出孤立点的意义(称为孤立点的内涵知识)将具有十分重要的意义。

孤立点分析是针对已挖掘出的孤立点数据研究其孤立特征或出现原因、孤立对象分类与聚类以及对时序数据的孤立趋势进行预测等。任何知识都是对象与属性的统一,对孤立点内涵知识的研究应以孤立点及其属性子空间为背景,探讨孤立点数据对象的特征或出现原因。文献[3]中把孤立点分为两类:非平凡孤立点(non-Trivial Outlier)和平凡孤立点

(Trivial Outlier)。其中,非平凡孤立点又分为强孤立点(Strongest Outlier)和弱孤立点(Weak Outlier)。文献[3-5]给出的孤立点内涵知识均是以非平凡孤立点为研究对象,即研究强孤立点和弱孤立点的分布特征及其挖掘算法。文献[6]针对这类非平凡孤立点给出了其用户意义,即给出了每个孤立点的原因属性集。这里的原因属性集是指所有能够解释某一数据对象成为孤立点的最小属性子空间组成的集合。本文在此基础上进一步做了更为精确的分析,重新定义了孤立点的原因属性集,另外给出了等价属性子空间、原因属性子空间及其孤立度和孤立点的相似度的概念,进而给出了一个挖掘孤立点的内涵知识的算法,最终把孤立点按照其相似度进行了分类,实验结果证明了该方法的有效性和实用性。

## 1 问题定义

给定一个数据集  $X = \{x_1, x_2, \dots, x_n\}$ , 其中  $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ ,  $(1 \leq i \leq n)$  为第  $i$  个数据对象, 属性集  $D = \{A_1, A_2, \dots, A_d\}$  称为  $d$  维属性集, 因此该数据集可以看作是一个  $n$  行  $d$  列的数据表, 共有  $n$  个数据对象,  $d$  个属性。定义  $D_d = A_1 \times A_2 \times \dots \times A_d$  为数据集  $X$  在  $D$  上的域空间,  $D_s = A_{i_1} \times A_{i_2} \times \dots \times A_{i_s}$  为  $D$  上的  $s$  维属性子空间, 其中,  $1 \leq i_j \leq d, (1 \leq j \leq s)$ , 则可知  $D_s \subseteq D_d$ ;  $O_{D_s} = \{o_1, o_2, \dots, o_m\}$  为  $D_s$  上的孤立点

到稿日期:2010-04-19 返修日期:2010-07-17 本文受重庆市科技攻关基金项目(CSTC, 2009AB2049; CSTC, 2009AC2068)资助。

王 越(1961-),男,博士,教授,主要研究方向为数据挖掘、数据库技术、嵌入式系统及应用, E-mail: wangyue@cqit.edu.cn; 刘亚辉(1984-),男,硕士,主要研究方向为数据库与数据挖掘; 谭暑秋(1982-),女,硕士生,主要研究方向为数据仓库与数据挖掘。

集,  $|O_{D_s}|=m$  为集合  $O_{D_s}$  的势。

**定义 1(等价属性子空间)** 给定一个输入参数  $\mu>0$ , 如果

$$\frac{|O_{D_s} \cap O_{D_d}|}{|O_{D_s} \cup O_{D_d}|} \geq \mu$$

则认为  $D_s$  是与域空间  $D_d$  关于  $D_d$  上的孤立点集的相似属性子空间。与  $D_d$  相似的  $D_s$  可能有多, 将等式左边值最大的  $D_s$  称为  $D_d$  的等价属性子空间 EAS(Equivalence Attribute Subspace), 记为  $D_s^d$ 。

这里的  $\mu$  是需要由用户给定的参数, 其值越大, 说明属性子空间  $D_s$  与域空间  $D_d$  具有产生相同孤立点集的能力越强。因此, 给定一个合适的  $\mu$ , 我们总可以找到一个与原始数据集的域空间等价的属性子空间  $D_s$ , 这样就把数据集的维度由  $d$  维降到了  $s$  维, 在最不理想的情况下, 也即  $s=d$ 。而在  $D_s^d$  上再作进一步的分析就可以在达到一定精度要求情况下减少计算复杂度。以下的研究均是在域空间的等价属性子空间上进行的。

文献[6]中给出的孤立点的原因属性集是指能够解释某一数据对象成为孤立点的所有最小属性子空间组成的集合。为了能够增加后续讨论的孤立点相似度问题的准确度, 下面给出某一孤立点的原因属性子空间的定义。

**定义 2(原因属性子空间)** 任何一个能够解释某数据对象成为孤立点的属性子空间叫做该孤立点的原因属性子空间, 也即, 给定一个孤立点  $o_i$ , 如果在  $s$  维属性子空间  $D_s$  上  $o_i$  仍为一个孤立点, 则称  $D_s$  是  $o_i$  的一个原因属性子空间 CAS(Cause Attribute Subspace), 记为

$$CD_s(o_i) = A_{i_1} \times A_{i_2} \times \dots \times A_{i_s}$$

式中,  $o_i \in O_{D_s^d}$ ,  $(1 \leq i \leq |D_s^d|)$ ;  $1 \leq i_j \leq |D_s^d|$ ,  $(1 \leq j \leq s)$ ,  $|D_s^d|$  表示  $D_s^d$  的属性的个数(注:  $D_s^d$  中的符号  $s$  与  $CD_s(o_i)$  中的符号  $s$  含义不相同)。

孤立点  $o_i$  的所有原因属性子空间  $CD_s(o_i)$  组成的集合称为  $o_i$  的原因属性集, 记为  $S\_CD(o_i)$ 。

**定义 3(原因属性子空间的孤立度)** 孤立点  $o_i \in O_{D_s^d}$  在其某一原因属性子空间  $CD_s(o_i)$  上与其它所有数据对象的距离和称为该孤立点的该原因属性子空间的孤立度, 记为

$$W\_CD_s(o_i) = \sum_{j=1, j \neq i}^n D(o_i, o_j)$$

式中,  $n$  为数据集  $X$  的数据对象的个数。

为了使各孤立点的原因属性子空间的孤立度之间的可比性更加精确, 在某一属性子空间下计算两个数据对象之间的距离时, 采用变形后的欧几里得距离。设两个数据对象之间的距离为  $d_{ij}$ , 则其计算式如下

$$d_{ij} = \sqrt{(|x'_{i1} - x'_{j1}|^2 + |x'_{i2} - x'_{j2}|^2 + \dots + |x'_{is} - x'_{js}|^2) / s}$$

式中,  $i = (x'_{i1}, x'_{i2}, \dots, x'_{is})$  和  $j = (x'_{j1}, x'_{j2}, \dots, x'_{js})$  是经过标准化预处理后的数据集的两个  $s$  维的数据对象。

某一孤立点的原因属性子空间的孤立度表示该原因属性子空间对数据对象成为孤立点的影响程度。  $W\_CD_s(o_i)$  越大, 说明该  $CD_s(o_i)$  越有可能是导致  $o_i$  成为孤立点的原因, 也即该原因属性子空间对孤立点的影响程度越大。

**定义 4(孤立点的相似度)** 给定两个孤立点  $o_i, o_j \in O_{D_s^d}$ , 其所有原因属性子空间组成的集合分别记为  $S\_CD_s(o_i)$  和  $S\_CD_s(o_j)$ , 将集合  $S\_CD_s(o_i), S\_CD_s(o_j)$  中的元素(即孤

立点的原因属性子空间)按其孤立度大小降序排列, 并为排序后的每个元素按其在集合中的排列次序分配一个自然数作为该元素的权值  $w_i = i, i \in N$ 。例如, 集合中的第一个元素的权值为  $w_1 = 1$ , 第二个元素的权值为  $w_2 = 2$ , 依次类推。

记  $Int\_SCD(o_i, o_j) = S\_CD(o_i) \cap S\_CD(o_j)$ ,  $W(v, S)$  为某一值  $v$  在集合  $S$  中的权值。则  $WMIN_k = \text{Min}(W(Int\_SCD(o_i, o_j)_k, S\_CD(o_i)), W(Int\_SCD(o_i, o_j)_k, S\_CD(o_j)))$ ,  $WMAX_k = \text{Max}(W(Int\_SCD(o_i, o_j)_k, S\_CD(o_i)), W(Int\_SCD(o_i, o_j)_k, S\_CD(o_j)))$ , 由此, 定义

$$SIM(o_i, o_j) = \frac{\sum_{k=1}^{|Int\_SCD(o_i, o_j)|} WMIN_k}{|S\_CD(o_i) \cup S\_CD(o_j)|}$$

为孤立点  $o_i, o_j$  的相似度 SIM(Similarity)。

由定义式可知  $0 \leq SIM(o_i, o_j) \leq 1$ 。

文献[4]中提到了属性集相似度的概念, 它只是从两个属性集的交集占其并集个数的比例大小来定义的, 而这里的孤立点的相似度根据其各原因属性子空间对孤立点本身的影响程度不同, 将其各原因属性子空间按其孤立度大小排序并赋予相应权值, 进而计算两个孤立点的相似度, 这样更符合其实际意义, 提高了问题的准确性。

## 2 孤立点的内涵知识挖掘

根据以上定义, 我们可以把得到孤立点内涵知识的方法描述如下:

根据给定的数据集  $X$  和孤立点集  $O_{D_d}$  得到域空间的等价属性子空间  $D_s^d$ , 在  $D_s^d$  上计算每一个孤立点  $o_i$  的原因属性集, 同时给出该属性集中每一个原因属性子空间的孤立度, 然后将这些原因属性子空间按照其孤立度大小降序排列, 并按照定义 4 中的方法为其设置权值, 然后计算  $O_{D_d}$  中两两孤立点的相似度  $SIM(o_i, o_j)$ 。

给定一个输入参数  $0 < \alpha \leq 1$ , 我们把  $SIM(o_i, o_j) \geq \alpha$  的两个孤立点归为一个类, 可以认为该类中的所有孤立点在显著性水平  $\alpha$  下关于原因属性子空间具有相同的特征, 即可认为这些孤立点是由于相同的原因引起的。

下面给出挖掘此类孤立点内涵知识过程的相关算法描述。

## 3 算法描述

input: 数据集  $X$ , 孤立点集  $O_{D_d}$ , 参数  $\alpha$

output: 由  $O_{D_d}$  中的元素组成的若干个簇

/\* 调用 EAS 算法, 查找域空间  $D_d$  的关于孤立点集  $O_{D_d}$  的等价属性子空间  $D_s^d$  \*/

$D_s^d = \text{EAS}(X, O_{D_d}, \mu)$ ;

FOR  $i$  FROM 1 TO  $|O_{D_s^d}|$

/\* 调用文献[4]中 DSCM 算法求出  $o_i \in O_{D_s^d}$  的原因属性集 \*/  $S\_CD(o_i) = \text{DSCM}(X, o_i)$ ;

/\* 将  $S\_CD(o_i)$  中的所有元素按其孤立度大小降序排列 \*/  $\text{DESC}(S\_CD(o_i))$ ;

/\* 为  $S\_CD(o_i)$  中排序后的元素设置权值 \*/

FOR  $j$  FORM 1 TO  $|S\_CD(o_i)|$

$w_j = \text{SetWeight}(S\_CD(o_i)_j)$ ;

END FOR;

END FOR;

```

FOR i FROM 1 TO | $O_{D_d}^d$ |
  FOR j FROM i+1 TO | $O_{D_d}^d$ |
    /* 对于每一对孤立点  $o_i, o_j \in O_{D_d}^d$  */
    IF SIM( $o_i, o_j$ )  $\geq \alpha$ 
      将  $o_i, o_j$  归为一个簇簇内;
    END IF;
  END FOR;
END FOR;
END

```

经过以上步骤后,得到的由  $O_{D_d}$  中的元素组成的若干个簇簇作为算法的输出。

EAS( $X, O_{D_d}, \mu$ )算法

input:数据集  $X$ , 孤立点集  $O_{D_d}$ , 参数  $\mu$

output: $X$  的域空间  $D_d$  的关于  $O_{D_d}$  的等价属性子空间  $D_d^d$

/\* 将  $D_d$  的所有子集放入一个数组  $A$  中 \*/

FillInQueue ( $D_d, A$ );

FOR  $i$  FROM 1 TO A.Count

/\* 调用与在  $D_d$  上挖掘出  $O_{D_d}$  相同的算法在  $A[i]$  上挖掘孤立点 \*/

$O_{A[i]} = \text{FindOutliers}(A[i]);$

$P_i = \frac{|O_{A[i]} \cap O_{D_d}|}{|O_{A[i]} \cup O_{D_d}|}; \text{MaxP} = 0;$

IF  $P_i \geq \mu$

IF  $P_i \geq \text{MaxP}$

MaxP =  $P_i$ ;

END IF;

END IF;

$D_d^d = \text{MaxP}$  对应的  $D_d$  的子集;

END FOR;

RETURN  $D_d^d$ ;

END

## 4 实验及结果分析

实验的数据采用质量管理中的“产品硬度”数据表作为数据源  $X$ , 该数据表共有 200 条记录, 6 个属性: “产品硬度”、“材料硬度”、“加工温度”、“添加量”、“材料及类型”、“班次”。

采用文献[7]中基于距离的孤立点挖掘算法对数据集  $X$  进行孤立点挖掘, 共检测出 12 个孤立点, 即  $O_{D_d} = \{x_4, x_{15}, x_{100}, x_{110}, x_{114}, x_{130}, x_{141}, x_{154}, x_{156}, x_{157}, x_{160}, x_{180}\}$ 。

把数据集  $X, O_{D_d}$  和参数  $\mu$  作为 EAS 算法的输入(这里设置参数  $\mu=0.85$ ), 得到  $X$  的域空间  $D_d$  的关于  $O_{D_d}$  的等价属性子空间  $D_d^d$ , 其共包含 4 个属性: “产品硬度”、“材料硬度”、“加工温度”、“添加量”。而数据集中的“材料及类型”和“班次”则被约去。

表 1 列出了算法执行过程中得到的各个孤立点的所有原因属性子空间及其对应的孤立度。

根据表 1 中得到的数据计算  $O_{D_d}$  中两两孤立点的相似度 SIM, 设置参数  $\alpha=0.9$ , 得到表 2 所列的 5 个簇簇。

从以上实验结果看出, 通过对挖掘出的孤立点进行相似度分析, 得到了各个孤立点的原因属性集, 从中找出孤立点产生的相似性, 利用这种相似性特征结合聚类分析的思想对孤立点进行了分类, 使每一类中的所有孤立点在一定精度下具有相同的原因属性集, 并给出了每一类中所有孤立点的共享的原因属性子空间, 所有这些信息都能够帮助用户更好地认

知这些孤立点数据。

表 1 孤立点的原因属性子空间及其孤立度

数据对象	原因属性子空间	原因属性子空间的孤立度
$x_4$	{加工温度}, {产品硬度, 加工温度}, {材料硬度, 加工温度}, {加工温度, 添加量}, {产品硬度, 材料硬度, 加工温度}, {产品硬度, 加工温度, 添加量}, {材料硬度, 加工温度, 添加量}	1534.90, 1124.23, 1110.02, 1106.18, 937.50, 934.52, 922.69
	{产品硬度}, {产品硬度, 材料硬度}, {产品硬度, 加工温度}, {产品硬度, 添加量}, {产品硬度, 材料硬度, 加工温度}, {产品硬度, 材料硬度, 添加量}, {产品硬度, 加工温度, 添加量}, {材料硬度}	4963.76, 3527.61, 3521.70, 3521.38, 2889.84, 2889.61, 2884.79, 437.49
	...	...
$x_{180}$	{产品硬度}, {产品硬度, 加工温度}, {产品硬度, 添加量}, {产品硬度, 材料硬度}, {产品硬度, 加工温度, 添加量}, {产品硬度, 材料硬度, 加工温度}, {产品硬度, 材料硬度, 添加量}	1616.68, 1178.82, 1165.19, 1157.80, 980.09, 973.28, 962.44

表 2 孤立点聚类结果

序号	簇簇	共享的原因属性子空间
1	$x_4, x_{130}, x_{156}, x_{157}$	{加工温度}, {加工温度, 添加量}, {产品硬度, 加工温度}, {材料硬度, 加工温度}, {产品硬度, 加工温度, 添加量}, {材料硬度, 加工温度, 添加量}, {产品硬度, 材料硬度, 加工温度}
2	$x_{15}$	
3	$x_{100}, x_{114}, x_{160}$	{产品硬度}, {加工温度}, {产品硬度, 添加量}, {产品硬度, 材料硬度}, {产品硬度, 加工温度}, {产品硬度, 材料硬度, 添加量}, {产品硬度, 加工温度, 添加量}, {产品硬度, 材料硬度, 加工温度}
4	$x_{141}, x_{154}, x_{180}$	{产品硬度}, {产品硬度, 添加量}, {产品硬度, 材料硬度}, {产品硬度, 加工温度}, {产品硬度, 材料硬度, 添加量}, {产品硬度, 加工温度, 添加量}, {产品硬度, 材料硬度, 加工温度}
5	$x_{110}$	

对比表 1 给出的孤立点的原因属性子空间及表 2 给出的聚类结果中每一簇簇的共享原因属性子空间, 可以看出, 从引起这些孤立点产生的原因方面考虑, 某些孤立点之间存在着很大的相似性, 因此对数据集中检测出的孤立点进行以上聚类分析是非常有必要的。

## 5 算法性能分析

本文算法由于先使用 EAS 算法找出原始数据集域空间的等价属性子空间, 使实验中的数据集由 6 维降到了 4 维, 减小了下面计算的复杂度, 提高了算法的效率。从这方面考虑, 只要通过查找数据集域空间的等价属性子空间可以达到降低数据集的维度, 那么就能提高算法的执行效率, 且数据集的规模越大就越明显。下面以一个实验对比来说明。

实验采用“某工厂 3 月份高炉数据”数据表作为数据集, 选取其中的 8 个属性(硅 Si, 锰 Mn, 磷 P, 硫 S, 钛 Ti, 铬 Cr, 镍 Ni, 铁 Fe)进行测试。图 1 显示了当数据集的规模  $n=200$ 、已知孤立点集  $O$  中孤立点个数为 15、属性维数不同时采用 EAS 算法和不采用 EAS 算法, 整个算法的运行时间情况。

实验中当数据集维度为 4 和 5 时, 采用和不采用 EAS 算法, 整个算法的执行时间几乎相同, 这是因为采用 EAS 算法得到的数据集域空间的等价属性子空间并没有约去相关属性维。在维度为 6, 7 和 8 时, 可以看到由于首先采用 EAS 算法约去了一个属性维, 从而使算法的执行时间大大缩短。理论和实验皆可看出原始数据集的维度越高, 算法的执行效率就越可能有明显的提高。

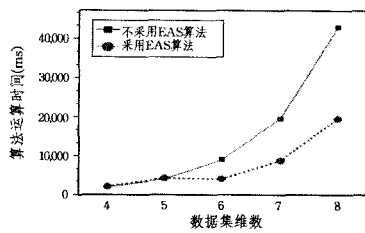


图1 不同维度时算法执行时间比较

另外,算法在进行查找孤立点的原因属性集的计算中,需要对数据集域空间的等价属性子空间的所有子空间进行操作,这样如果原始数据集的维数较大,算法的执行效率就显得较低,但是却能得到更加精确的聚类结果。如何在保证可以得到较为精确的聚类结果的情况下,提高算法的执行效率是作者进一步工作的重点。

**结束语** 孤立点内涵知识挖掘目前在孤立点分析领域中研究较少,孤立点通常包含着重要的信息,对孤立点进行分析,挖掘出隐含在其中的有用信息有助于用户更好地理解数据,获取有价值的信息。本文首先定义了数据集的等价属性子空间的概念,提高了在此基础上的计算效率,而后通过给出的孤立点的原因属性子空间及其孤立度和孤立点的相似度等概念,获得了有关孤立点的原因属性集及孤立点之间的相似性等内涵知识,并结合聚类的思想最终把孤立点按照其相似性特征进行了分类,每一个类中的所有孤立点在一定精度下具有相同的原因属性集。本文对算法的效率进行了分析,实

验结果证明了该算法是一个高效实用的方法;另外,算法需要设置的参数较少,减少了用户的参与,易用性较强。

## 参考文献

- [1] Han Jia-wei, Kamber M. Data Mining: Concepts and Techniques [M]. Academic Press, 2001
- [2] Ramaswamy S, Rastogi R, Shim K. Efficient Algorithms for Mining Outliers from Large Data Sets [C] // Proc of the ACM SIGMOD International Conference on Management of Data. [S. l.]: ACM Press, 2000
- [3] Knorr E, Ng R. Finding Intensional Knowledge of Distance-based Outliers [C] // Proc. of the 25th VLDB Conference. Scotland; Edinburgh, 1999: 211-222
- [4] Chen Z, Tang J, Fu A. Modeling and efficient mining of intentional knowledge of outliers [C] // Proc of the 7th Int'l Database Engineering and Applications Symposium. Los Alamitos, CA: IEEE Computer Society Press, 2003: 1-10
- [5] Papadimitriou S, Kitagawa H, Gibbons P B. LOCI: Fast outlier detection using the local correlation integral [C] // Proc. of the 19th Int'l Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society Press, 2003: 315-326
- [6] 王越, 刘亚辉, 徐传运. 孤立点用户意义分析在质量管理中的应用研究 [J]. 计算机应用, 2009, 29(11): 3077-3079
- [7] Angiulli F, Basta S, Pizzuti C. Distance-Based Detection and Prediction of Outliers [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(2): 145-160

(上接第 181 页)

2. 任务:

```

查询任务集合;
If 任务存在
    启动本地服务;
Else
{
    初始化超时计数器;
    初始化服务请求数据缓冲区域;
    向服务请求模块发送服务请求;
    While(未超时)
    {
        If 读取服务请求结果成功 Break;
        计数器减一;
    }
    初始化界面生成器;
    传送数据到界面生成器;
}

```

算法分析:若服务与任务列表采用存储法,则算法中的查询均采用  $O(1)$  算法,由启动模块产生一个标号,根据标号采用地址计算方式得到查询位置。若位置存在则成功,否则失败。界面生成器在服务提供时采用层次界面产生方式。此时外部资源请求本地服务,外部资源有可能显示资源受限,因此在界面产生时应该采取多级界面标准,根据外部资源情况返回合适的界面指标。这个合适界面的顺序查询时间最坏情况下为  $O(\ln)$ ,其中  $\ln$  表示系统中存储的界面数,这个数目通常情况下很少(小于 10),也可以采用更合适的算法优化这个查询时间。算法中最坏情况下为任务调度时的 while 循环,此时的时间复杂度为网络返回数据超时约为  $O(\text{time})$ 。因此该算法在最坏情况下的时间复杂度为  $O(\text{time} * \ln)$ 。当界面

数目小于或等于 10 时约为  $O(\text{time})$ ,亦即  $O(n)$ 。这样,在嵌入式系统中这种时间消耗是可以接受的。

**结束语** 在当前普适计算快速发展的时代,嵌入式系统已经成为普适计算中一个不可或缺的重点组成部分。本文给出了面向普适计算的嵌入式系统的硬件模型并重点设计了底层服务提供模型。文中还给出了两种系统的行为,最后分析了调度模块算法的效率。本模型的特色有助于支持用户界面自动生成(界面自适应)与任务级的物理设备交互,有助于面向普适计算的嵌入式系统进行服务设计。

## 参考文献

- [1] Ponnkanti S R, Lee B, Fox A, et al. A Service Framework for Ubiquitous Computing Environments [C] // Proceedings of the Ubiquitous Computing Conference. Lecture Notes in Computer Science, Vol. 2201. 2001: 56-57
- [2] 张德干, 徐光佑, 史元春, 等. 面向普适计算的扩展的证据理论方法 [J]. 计算机学报, 2004, 27(7)
- [3] 徐光佑, 陶霖密, 史元春, 等. [J]. 计算机学报, 2007, 30(7)
- [4] 陈锋, 黄剑. 支持普适计算的多嵌入式系统 [J]. 计算机科学, 2007, 34(11)
- [5] 姜丽芬, 卢桂章, 辛运韩. 普适计算软件体系结构研究 [J]. 计算机科学, 2007, 43(9)
- [6] 徐文控, 辛运韩, 卢桂章, 等. 普适计算系统架构的研究 [J]. 计算机科学, 2009, 36(4)
- [7] 牛连强, 王金东, 赵海. 普适计算体系结构的研究 [J]. 小型微型计算机, 2005, 25(6)
- [8] Muhlhauser M. Ubiquitous Computing and Its Influence on MSE [C] // Proceedings of International Symposium on Multimedia Software Engineering. 2000: 48-55