一种物联网设备自动描述方法

李 勐[1,2],王晓峰[1],崔 莉[1]
(1. 中国科学院计算技术研究所,北京 100190;2. 中国科学院大学,北京 100049)

摘 要: 物联网感知设备的服务描述文件为海量资源的发现与检索提供了有效的支持,是面向服务的物联网架构的基础.当前服务描述文件主要通过开发人员手工撰写完成,工作量大.现有研究SPITFIRE提出了一种半自动方法协助开发人员撰写服务描述文件,但方法本身为集中式方法,配置较复杂且精度过度依赖人工参数调优,不适合大规模部署.针对物联网海量设备的描述问题,本文提出了一种基于度量学习的分布式的物联网感知设备自动描述方法.该方法使用设备的多种数值特征作为输入,利用一种分布式的DBSCAN聚类算法对设备进行归类与推导,设备通过归类结果可自动生成自身描述文件.该方法利用度量学习优化聚类的度量函数以保障精度,以分布式方式进行灵活快速的配置,可减少人工干扰.仿真实验表明,与使用单一属性作为度量方式的SPITFIRE相比较,本文方法在获得对设备聚类相当的查全率的同时,查准率提高了20.4%,更适合于物联网海量设备使用场景.

关键词: 物联网;海量设备;描述文件;分布式;优化;聚类
中图分类号: TN911.23 **文献标识码:** A **文章编号:** 0372-2112 (2016)05-1055-09
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.05.007

An Automatic Device Describe Method for Internet of Things

LI Meng[1,2], WANG Xiao-feng[1], CUI LI[1]
(1. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190, *China*;
2. *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: Service description file, which is fundamental for the service-oriented architecture, can be used by the IoT sensing devices to accelerate resource discovery and searching processes. Currently, these files are mostly written manually by the device developers, this process is inefficient and fallible. The state-of-art method SPITFIRE can generate the devices´ description in a semi-automatic way, but its configuration can be trivial and its accuracy still can be improved. In this paper, we proposed a novel automatic device description method, with which devices can automatically generate their individual description. We designed a clustering algorithm based on DBSCAN to infer the description of sensing device, taking advantages of existing descriptions and the data features of series gained by data sampling. A metric learning algorithm is also implemented to optimize the parameter used by the clustering algorithm. All the routines run independently on different devices, and no manual intervention is needed during the self-description process. Through simulation, we show that this method has a prominent advantage of precision over other state-of-art methods, making our method more suitable for the massive IoT devices.

Key words: internet of things; massive device; description file; distributed; optimization; clustering

1 引言

随着物联网技术的发展与设备计算能力的提升,越来越多的感知设备开始以水平化的方式接入网络.与传统传感网有所区别,物联网常包含海量与异构的感知设备.为了便于管理,现有研究与应用开始通过人工定义与撰写设备功能描述文件,将感知设备所提供的功能封装成为服务.面向服务的物联网架构可以有效地对资源进行组织,提高设备的共享程度,解决不同厂商设备间的连通性问题,用户通过服务搜索的方式,可以快速定位物联网中的设备与资源.由于设备的海量性,现有手工或半自动服务描述方式工作量较大,本

收稿日期:2014-07-22;修回日期:2015-07-24;责任编辑:马兰英
基金项目:中国科学院战略性先导科技专项(No. XDA06010403);国家国际科技合作专项(No. 2013DFA10690);国家自然科学基金(No. 61202211)

文针对该问题,提出了一种感知设备服务描述文件的自动生成方法.

面向服务的架构包含服务描述语言、协议、注册与管理方法三部分技术.现有用于物联网的服务描述语言继承了传统的 Web 服务描述方法,其中常用的是 WSDL[1]、uPnP XML[2],以及 OGC 组织下的 Sensor ML[3].这三类建模语言分别定义了设备描述规范,开发人员编写对应设备的描述文件,可以完成对物联网设备的描述.但在物联网场景下,这种传统的人工编写方式会带来可操作性问题.当前,物联网设备具有异构、海量的特点,人工撰写设备描述文件会导致较大的工作量,整个过程是繁杂、易错与低效的;此外不同开发人员的评价标准不一,最终得到的设备描述文件的可用性会受影响.文献[4]提出了半自动的设备描述框架 SPITFIRE,利用感知设备捕获的数据序列的特征.该方法通过聚类将特征类似的设备归为一类,同类设备给予相同的设备描述标签.这种方式可以有效辅助开发人员进行设备描述,但在实现过程中需要人工给定聚类参数;同时,SPITFIRE 本身是一种集中式算法,需要辅助的计算设备,会带来额外的人工工作量和设备成本.

理想情况下,设备的开发人员无需了解设备数据的具体意义,且无需编写设备的服务描述文件,仅需为设备添加统一的自动描述程序,设备的服务描述文件会随自动描述程序的执行而自动生成.这种设想在物联网感知设备上具有可行性,原因在于设备捕获的诸如温度、湿度、光照等环境信号有明显的数值特征差异,利用特征的差异与相关性可能推断出数据属性甚至地理位置的近似程度.如果能正确获得不同感知设备之间的相关性,则在已知少量设备描述"标签"的情况下,大量的缺失功能描述文件的设备可以利用其他有相似数据特征的设备的描述文件进行补全.本文设想通过物联网前端感知设备的自主协商,分布式地对感知设备输出的数据数值上的相似性进行判断,利用已有的少量设备描述信息补全缺失描述,以达到自动撰写自身的描述文件的目的.

为此,本文在文献[4]的基础上,设计了一种自协商、易扩展的物联网设备自动描述方法.设备利用自身与其他设备采集到的原始数据的相关性,通过分布式地协商获得最优参数,通过在设备自身上执行聚类,对数据本身的意义进行推断,补全和创建缺失的设备的文本描述.

本文的主要贡献有以下两点:(1)基于度量学习思想[5],提出了一种可扩展的设备相似性度量策略.该策略使用线性规划方法,从所有候选度量方式中选择最优组合,线性加权得到一种区分能力较强的度量方式,同时推断出聚类的参数;(2)设计了一种分布式的、基于 DBSCAN[6]的分布式设备属性聚类方法,通过此方法,物联网感知设备可以对自身的描述进行推断.仿真实验表明,本文的设备自动描述方法与 SPITFIRE 方法相比较,在功能描述的查准率上有明显优势,在查全率上基本持平.

2 相关工作

现有的物联网服务描述的研究主要集中于对服务搜索与设备发现功能的支持与设备服务运行能耗的控制上.uPnP 是最早用于搭建面向服务的物联网的技术之一,通过重新定义 uPnP 中的 XML 来描述物联网设备的静态信息,uPnP 可应用于物联网的设备发现.针对物联网设备资源受限的特点,IrisNet[7]提出了一种整体描述物联网子网的方法,考虑到传感器节点自身的能量与计算能力,IrisNet 将子网的结构、功能特性进行描述之后,存储于物联网网关上.文献[8]针对传感网中 Web 服务描述资源占用过多这一现象,提出了一种轻量级的传感器描述方法.Sensor ML[3]是 OGC 组织推广的一种传感器描述方法以及一套资源定位与搜索工具,最大特点是可以利用语义工具进行设备地理位置的推理.Sensor ML 规定一套完整的描述规则和查询机制,可以对包含 Sensor ML 描述的物联网系统中的设备与数据进行快速查询.文献[9]提出了一种语义物联网搜索框架,为了获得更好的搜索效果,研究使用本体对物联网进行建模,用户可使用不同的优先级进行定制化搜索.但以上方法均假设物联网开发人员手工定义设备描述文件,无法从根本上解决海量感知设备的描述问题.

物联网设备感知数值特征的表示方面的研究对于解决设备自动描述问题也具有借鉴意义.Yan 等人[10]提出了一种基于 SIFT 算法的图像传感器的数据特征提取的方法,通过对不同图像特征进行聚类,避免了较大的计算开销,设计了一种高效的索引,用于相似图片的快速查找.Truong 等人[11]为了估计传感器实时数据流数值上的相似程度,提出了一种基于 Fuzzy Set 的特征提取方法.以上两种方法可以较好地解决特定领域内数据特征的提取,从而获得数据之间的相似度,但设备自动描述任务面对了更多类型的原始输入,需要一种扩展性更强的方式.

与本文最为相近的工作是由 Kay Romer 等人主导的 SPITFIRE 物联网搜索项目[4].该项目完成了一套完整的物联网搜索平台,其中为了防止进行搜索时设备描述信息过少,引入了一种集中式的半自动的设备描述框架.SPITFIRE 采用一种数值数据与属性的绑定的方式,通过 rdf 查询方法 SPARQL,快速定位物联网中的

资源.由于此项目需要对大量定义用于设备描述的XML,为了减小工作量,文献[4]设计了一种基于数据数值特征聚类的半自动的描述框架.该框架可以为开发人员提供备选的设备描述方式,减小了人工撰写描述文件的工作量.但 SPITFIRE 存在两点较为突出的问题:(1)由于基于聚类方法,针对不同类型的感知设备,开发人员需对聚类参数进行调优,产生了额外的工作量;(2)该方法是一种集中式算法.在物联网的海量设备场景下通讯开销较大,同时需要一个计算能力较强的中心设备进行聚类计算,增加了设备成本.本文在 SPITFIRE 搜索框架的基础上,针对以上两个问题,设计了一种全新的分布式设备自动描述方案.

3　自动描述方法流程

前端感知设备捕获到的数据序列本身包含大量的特征信息.以室内环境监测传感器为例,环境监测传感器采集温度、湿度、光照三类环境数据,这三类数据在数值特征上有明显的差异.图 1 显示了 Intel Lab 数据集[12]一个时间片段内不同类型数据的数值分布规律.其中,室内温度的绝大多数采样在 18 摄氏度到 20 摄氏度之间波动,湿度和光照均可能在 0 到最大量程之间波动.由于遮挡、灯光变化等扰动或突发事件,光照传感器可能捕获到较前两者变化频率更高的数据序列.如果能找到合适的区分标准,则可以区分不同类别的数据序列,为鉴别数据的属性提供依据.

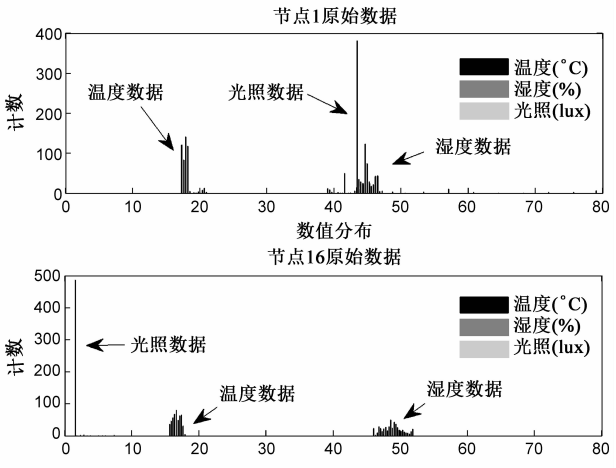


图1　Intel Lab数据集某段时间内数值分布特征

本文建立在以下假设上:(1)设备描述只考虑数据采集功能,对于非数据采集功能,是否可以使用自动描述取决于是否存在一种有效的训练标签获取方式,本文暂不讨论;(2)每个设备只提供一种数据的输出,每种数据输出对应一个实时的数据序列,该序列是用于决策的最小单元;(3)设备两两之间均连通,不考虑由于设备之间的连通性导致的问题;(4)设备在进行自动描述过程期间没有休眠.

感知设备的自动描述可以形式化为一个半监督的机器学习问题.原始设备包括少量含有描述信息的设备与大量不含描述信息的设备.设备的描述信息可以看为设备训练标签,每个感知设备的采样信息为一个数据样点.规定原始状态下包含标签的设备对应的采样信息为训练数据样点,不包含标签的采样信息为测试数据样点.

本文的自动描述方法分两个步骤,首先利用已有描述信息的设备数值特征,对数据整体的参数特征进行估计.之后,通过原始数据的数值特征对不同设备进行区分,根据区分情况与同类设备已有的描述信息,对无描述信息的设备添加描述,完成设备自动描述.

图 2 展示了自动描述方法的整体过程:(1)开发人员给定采样时长,提供候选的特征提取方式,此外如果原始设备均不包含描述信息,则需要人工给定少量描述信息作为推断依据;(2)感知设备启动之后,统一进行指定长度时间的数据采集,并分别将数据序列使用(1)中规定的特征提取方式进行提取;(3)包含标签的设备进行协商,确定最优度量,同时确定聚类参数;(4)所有设备进行协商,在最优度量下,使用分布式 DBSCAN 方法,对设备进行聚类;(5)依据类中的已有标签,为同一类设备给定标签.

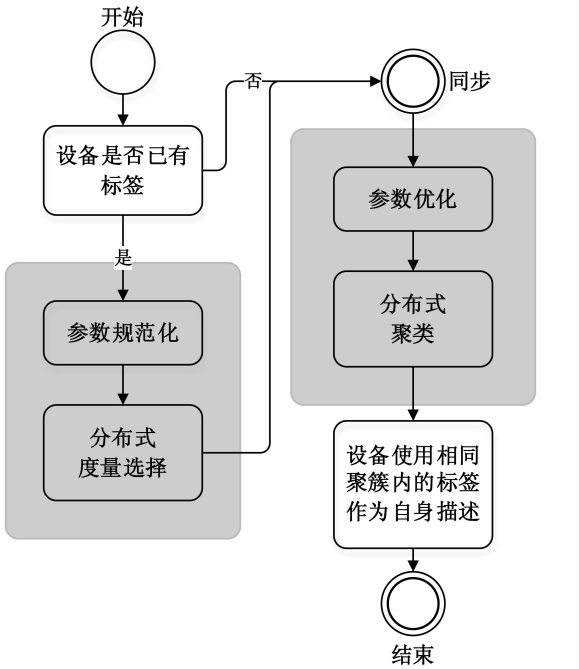


图2　自动描述方法执行框架

本文采用了一种简化的度量学习策略,这种度量学习策略主要用于优化原始数据特征之间的距离函数;同时,该优化过程中获得了设备数据特征的整体分

布情况,可以用以获得后半部分的推断方法的输入参数.

在进行设备间关系推断时,使用 DBSCAN 聚类方法作为基础方法主要是出于以下考虑. 首先,感知设备自动描述任务可以使用分类方法或聚类方法实现,但分类方法可能会遗漏部分类. 具体而言,如果训练集本身不包含某类数据点,该类数据样点无法参与训练过程,所获得的分类器无法区分该类数据. 在海量设备的场景下,训练集样点的数目将远小于测试集,这种情况将大大影响算法的准确率. 其次,其相对于其他聚类方法,DBSCAN 本身特性决定可以更好地被改写为分布式的方法. DBSCAN 在执行过程中每个节点不需要实时了解全局信息,同时节点自身状态较为简单. 已知聚类参数的情况下,单个设备仅需知道与其他设备之间的特征距离,判定自身是否属于核心点. 此外,在节点数目较多的情况,这种方法下具有较小的计算量,整体迭代次数易于估计.

4 自动描述方法的实现

本节对分布式度量优化方法和分布式聚类方法两个关键步骤进行描述. 为方便表述,下文定义参与自动描述的感知设备数目为 N,N 中包括已包含描述信息的设备 N_L 数目,其中 $N_L \ll N$. 每个设备产生的数据序列有 M 种候选特征可以选择,使用不同的特征参与相似度计算会得到不同的结果. 设备 x_i 与 x_j 之间的、度量 f 下的距离记为 $f(x_i,x_j)$.

4.1 分布式度量优化方法

度量优化的目的是根据少量的已有标签设备的数值特征,获得一个具有较好特性的相似性度量方式. 本小节首先将用于设备描述的度量优化问题建模成为一个线性规划问题,之后利用线性规划可行域为凸集这一特性,设计一种基于迭代的分布式优化方法,可计算指定精度的数值解.

物联网感知设备的原始数据较为多样,但仍可以根据其数值分布情况、频域、变化速率等数值特征进行区分. 可以使用的特征包括不同窗口大小下的直方图、某个频段的 FFT 系数、小波系数等. 但不同的特征组合对数据的区分能力不尽相同,一般情况下需要人工对数据集的物理含义进行研究,以确定使用一个或一组特征作为度量,否则,需要学习算法本身对数据集的特性进行分析. 本文考虑使用一种自动的方式对度量进行优化. 人工选择多组度量作为候选后,算法可以从这些度量中找到若干最为有效的度量.

现有研究中,度量学习(Metric learning)[5,13]提供了一类针对度量函数的优化方法. 度量学习通常指定一个优化目标,根据已有分类标签优化距离的相关度矩阵,通过对度量的优化,使节点变得更容易区分. 但度量学习需要计算一个完整的相关性矩阵,在分布式感知设备的环境下,设备需要多次通讯与迭代才能完成. 为了避免过大的计算量,本文使用了一种简化的度量学习方法,仅计算对角线上的元素权重以替代相关性矩阵. 以下给出具体计算方法.

首先定义距离函数. M 种不同的度量记为 f_k,每种度量对应权重参数 $\boldsymbol{\lambda}_k$,定义两个感知设备 x_i 与 x_j 之间的距离 F:

$$F(x_i,x_j \mid \lambda_1,\cdots,\lambda_n) = \sum_k^M \boldsymbol{\lambda}_k f_k(x_i,x_j) \tag{1}$$

其次,需选取合适的优化目标. 本文选择不同类之间"最小距离最大化"作为优化准则. 这种选择目的在于可以较好地配合 DBSCAN 聚类算法:(1)这种方式可以最大程度地避免误判,将不同类别的设备错归为一类. DBSCAN 的同类判定依据是数据点之间双向密度连通,根据最小距离最大化设定 DBSCAN 核心点判定半径,可以避免不同类之间最近两点直接密度相连. 如果两类之间的间隙不出现测试点,理论上可以保证不出现错判,文本在 4.2 节参数推断部分给出了详细说明;(2)这种方式便于计算. 不同于度量学习中常用的类内距离和最小化等方法,最小距离最大化可以方便地转化为线性规划问题,可以进行快速的求解,且可以方便地改写为分布式方法. 原问题可以表示为:

$$\max \min_{(x_i,x_j)\in\mathbb{D}} \sum_k^M \lambda_k f_k(x_i,x_j) \tag{2}$$

其中 $(x_i,x_j)\in\mathbb{D}$ 指 x_i,x_j 在不同的类中,满足 $\sum_{k=1}^M \boldsymbol{\lambda}_k = 1$. 这里令 $v = \min_{(x_i,x_j)\in\mathbb{D}} \sum_k^M \boldsymbol{\lambda}_k f_k(x_i,x_j)$,原问题可以改写为线性规划问题的标准型.

$$\begin{aligned} &\max \quad v, \\ &\text{s.t.} \quad \sum_k^M \boldsymbol{\lambda}_k f_k(x_i,x_j) - v \geq 0, (x_i,x_j)\in\mathbb{D} \\ &\qquad \sum_{k=1}^M \boldsymbol{\lambda}_k = 1, \\ &\qquad 0 \leqslant \boldsymbol{\lambda}_k \leqslant 1 \ k = 1,\cdots,M \\ &\qquad v \geq 0 \end{aligned} \tag{3}$$

经典的线性规划问题可使用单纯形等集中式算法快速求解,但不适合在分布式且资源受限的设备上运行. 注意到上述的线性规划问题(3)有 $O(M^2)$ 个不等式约束,而设备本身的计算能力非常有限,单个设备不适合处理这种规模的计算任务.

针对这种情况,本文使用一种基于迭代的分布式求解方法. 由于线性规划的可行域是一个凸集,使用迭代方法可以保证每次以一个固定步长改善解向量都会改进目标函数[14]. 由于 $v \leqslant \sum_k^M \boldsymbol{\lambda}_k f_k(x_i,x_j)$,且每次迭

代目标函数 v 递增,由单调有界定理可知目标函数是收敛的,故这种方法最终会得到最优解. 所以,该方法可以每次将解向量修改一个步长,然后验证该变动是否会导致一个更好的结果,如果出现更好的结果则更新当前解向量,否则就尝试其他可能,如此可以保证在有限的步数内获得最优结果. 步长的选择会影响最终结果的精度和迭代的速度.

在优化进行之前,假设每个设备已知不同度量下自身与其他所有设备的距离,同时存储一个初始参数向量 $\boldsymbol{\lambda}$ 与优化目标 v. 在每次迭代完成之后,设备均可以通过广播,同步当前时刻的解向量$(\lambda_1,\cdots,\lambda_M,v)$. 迭代的步长记为 Δ,有标签感知设备数据序列之间的度量记为 d,整个迭代过程分三个主要步骤:

(1)进行"瓶颈节点"选举. "瓶颈节点"指持有最小类间距离的设备节点. 每个设备使用当前参数 $\boldsymbol{\lambda}$,计算自身与其他非同类设备之间的最小距离 $d_x=\min_{x'}\boldsymbol{\lambda}f_k(x,x')$,之后定期自发广播 d_x;每个设备如果收到来自 y 的大于自身最小距离的值 d_y,则向设备 y 发送一个抑制信息. 没有收到任何抑制信息且一段时间内网络中没有广播的节点 x_0 可认为自身是"瓶颈节点",且持有优化所需的最短距离 d_x,进入步骤 2;如果设备收到抑制信息,则等待其他设备的通知信息.

(2)"瓶颈节点" x_0 尝试对自身的解向量 $\boldsymbol{\lambda}$ 进行优化,得到 $\boldsymbol{\lambda}'$. 由于存在等式约束,使用类似 SVM 中的 SMO 技术[15],同时尝试以步长 Δ 调大一个值 λ_i 并调小另一个值 λ_j,原有的参数向量会被改写为$(\lambda_1,\cdots,\lambda_i+\Delta,\cdots,\lambda_j-\Delta,\cdots,\lambda_M)$. 如果已经尝试过所有组合,则已经获得了最优的解向量,迭代结束;否则进入步骤 3.

(3)"瓶颈节点" x_0 广播调整后的 $\boldsymbol{\lambda}'$,通知其他设备对 $\boldsymbol{\lambda}'$进行测试. 其他设备获得新的参数后,重新计算自身的最小距离 d_y,之后全局再次执行步骤 1 中的最短距离搜索过程,发起设备 x_0 可以获得一个新的距离,如果新的距离大于原有距离,说明步骤 2 的优化是有效的,本轮迭代结束,回到步骤 1. 如果新的距离小于原有距离,说明 2 中尝试的优化无效,回退到步骤 2 重新选择.

该分布式算法是一致的,即所有设备均具有相同的执行逻辑. 节点 i 使用 x_i 表示;其中 $x_i.s$ 表示节点当前状态,包括"瓶颈节点"状态和正常节点状态;v 表示当前的目标函数值,$\boldsymbol{\lambda}$ 表示节点当前的参数值. 算法输入为每次迭代的步长 Δ,算法的输出为迭代最终的目标函数值 v_{opt}以及参数值 $\boldsymbol{\lambda}_{\text{opt}}$. 整体过程可由算法 1 表示.

这里分析算法 1 的精度、时间复杂度以及通讯量. 该方法的精度与性能与步长 Δ 的选择有很大关系. 步长 Δ 直接决定了最终的精度,最坏情况下会为解向量 $\boldsymbol{\lambda}$ 带来 $\Delta/2$ 的偏差. 每次迭代中,单个设备在步骤 2 时需要进行 $O(N_L^2)$次尝试;每次尝试的验证过程中,每个节点需要进行 $O(N_L)$次通讯,以询问所有的其他分类中的设备. 由于可行域为凸集,每次进行优化时,参数在数值上只可能向单方向移动,此外参数 $\boldsymbol{\lambda}$ 的和为 1,故迭代造成的参数最大变化量为 1,最坏情况下的迭代次数为 $1/\Delta$. 这里由于通讯的时间 C 远大于设备计算时间,故整个迭代过程的时间可有通讯时间进行估计,复杂度为 $O(CN_L^2/\Delta)$. 去除"瓶颈节点"进行的 1 次额外广播,由于最坏情况下每次迭代可能尝试 N_L^2 次,将产生 N_L^2 次广播事件,故整个过程中的消息传递次数为 $O(N_L^3)$. 虽然执行时间与消息传递次数据有较快的增长速度,但由于 $N_L \ll N$,分布式度量优化并不存在性能瓶颈.

算法 1　分布式度量优化算法

```
Input: Δ, x, v, λ
Output: v_opt, λ_opt
1  v_min ← ∅, x_min ← ∅
2  WHILE exist f_k(x,y) that is not computed
3     Send the features of x
4     IF heard features of x'
5        f_k(x,x') ← calculate the metric of (x,x') under λ
6        v_min ← min v_min, f_k(x,x'), x_min ← x', IF (x,x') ∈ 𝔻
7  Send SYN signal, wait SYN from all other node
8  WHILE on receiving v' from other node
9     Send v_min
10    IF v' ≤ v_min OR received MUTE, x.s ← NORMAL, BREAK
11    ELSE Send MUTE signal
12 IF x.s = NORMAL
13    WHILE on receiving MSG
14       IF MSG = TEST λ'
15          v'' ← min f_k(x, y | λ')
16          IF v'' > v_min, Send NO_IMPROVEMENT, ELSE Send OK
17       ELSE IF MSG = UPDATE_MIN v', v_min ← v'
18 ELSE
19    WHILE true
20       FOREACH (λ_i, λ_j) ∈ λ
21          λ' ← (λ_1, ..., λ_i + Δ, ..., λ_j − Δ, ...), Send λ'
22          WHILE on receiving MSG
23             IF MSG = NO_IMPROVEMENT, BREAK
24             ELSE IF all the other nodes send OK, GOTO step 8
25       RETURN v_min, λ, IF Cannot improve, optimization done
```

4.2　分布式聚类方法

如算法 2 所示,分布式聚类方法利用上节求得的最优度量,可以快速计算所有设备(有标签设备与无标签设备)的归属信息. 相较于集中式方法,分布式 DBSCAN 可以充分利用设备的并行特性,使用一种异步非阻塞方式完成聚类. 每个节点首先通过邻域内节点的密度,判定自身是否是核心节点,之后核心节点自发地扫描

邻域内的节点,如果这类节点与自身距离足够接近,则可以判定自身与邻居节点是否属于同一类,并给定相同的类标签,这里的距离是指 4.1 中得到的度量上的距离. 本小节主要讨论分布式 DBSCAN 方法的设计以及参数的选取.

算法 2 分布式聚类算法

Input: $x, v, f_{\text{opt}}(x_i, x_j), M, N$
Output: x
1 $x.L \leftarrow$ unique identity of x
2 $N_\epsilon(x) \leftarrow \varnothing$
3 $\epsilon \leftarrow v/2, N_\epsilon \leftarrow M(2\epsilon)^N$
4 FOR any x' ! $= x$
5 IF $f_{\text{opt}}(x, x') < \epsilon$
6 $N_\epsilon(x) \leftarrow x'$
7 cnt$\leftarrow M$
8 IF $|N_\epsilon(x)| \geq N_\epsilon$
9 WHILE on receiving MSG MERGE x'
10 Send MERGE to next $x'' \in N_\epsilon(x)$
11 IF $x'.L < x.L$
12 $x.L \leftarrow x'.L$, set next $x'' \in N_\epsilon(x)$ to the start, GOTO 9
13 IF all of $N_\epsilon(x)$ is visited, stop
14 $cnt \leftarrow cnt - 1$, IF $cnt = 0$, RETURN
15 ELSE
16 WHILE on receiving MSG MERGE x'
17 $x.L \leftarrow x'.L$
18 $cnt \leftarrow cnt - 1$, IF $cnt = 0$, RETURN

设候选特征数目为 M,设备节点数目为 N,任意两设备间最长通讯时间为 t_m. 聚类参数包括核心点判定半径为 ϵ 与核心点密度阈值为 N_ϵ. 参数 M 已知,N 与 t_m 可通过全网广播获得,ϵ 与 N_ϵ 的选择方法将在本小节后半段给出. 聚类具体步骤如下:

(1)设备初始化. 所有设备 x 在启动后,使用唯一的序号作为初始的类标签,该标签可以使用设备的 IP 地址等固有信息生成. 同时,计算一段时间内的数据序列的 M 种特征并进行广播,收取其他设备 x' 广播的特征.

(2)设备对自身是否是核心点进行判断. 每个设备 x 使用 4.1 中得到的距离函数 $F(x, x'|\boldsymbol{\lambda})$ 对距离进行计算,获得距离小于 ϵ 的邻域内的节点 $N_\epsilon(x)$. 如果节点个数 $|N_\epsilon(x)| > N_\epsilon$ 则标记自身为核心点,否则自身停止操作,等待其他设备的通知信息.

(3)设备自发扩散自身类标签. 每个核心点设备 x 对相邻节点 $N_\epsilon(x)$ 进行扫描,验证每个节点 x' 是否是核心点,如果是核心点则协商两者公用的类标签号,这里始终选择字典序较小的标签号作为两者的公用标签. 每个核心节点对于所有相邻节点的扫描完成之后进入等待状态.

(4)设备对结束时机进行判定. 任何等待时间超过 $N\,t_m$ 的节点均认为自身已经获得了正确的类标签.

步骤 4 的判定依据如下. 由于节点的标签可能被多次改写,所以节点仅根据自身的分类信息并不能判定算法的结束. 本文采用的方案是为每个节点设置一个超时时间 $N\,t_m$. 原因在于,最坏情况下,网络所有测试点均密度连通,执行步骤 2、3 进行扩散时,每次只能加入一个测试点,此时需要 N 个周期的时间才能严格判定聚类过程的结束.

DBSCAN 所需要确定核心点半径参数 ϵ 和密度阈值参数 N_ϵ,而这两种阈值可以使用 4.1 中的优化结果进行推断. 在设备描述场景下误判会造成严重的后果,所以在参数选择方式上,尽可能使用保守的策略,以减少误判. 本文利用 4.1 节中的距离优化的目标函数值 v 作为以上两个参数选择的依据. v 的实际意义是两个不同类别的聚簇之间的间隙,距离大于 v 的聚簇将无法进行合并. 但由于 v 本身是使用有标签集合计算得到的,而该数据集只是全部数据的一个子集,所以在全部设备数据集上运行时,v 产生的间隙很有可能无法将两个类区分.

本文选择 $\epsilon = v/2$ 作为 DBSCAN 的半径参数. 原因在于这种参数选择方案可以避免以下极端情况:假设距离最小的不同类数据点为 $(x_i, x_j) \in \mathbb{D}$,且 x_i 与 x_j 均为核心点;在完整数据集中,x_i 与 x_j 的中心点上出现了一个核心点 T,如果中心线上的 T 之外的测试点均不是核心点,那么根据核心点的定义,T 也不可能是核心点. 如果 $\epsilon \leqslant v/2$,对于两个不同的类,其半径均不可能覆盖到中心的核心点,两类之间不会是密度可达的. 故选择 $\epsilon = v/2$ 会有效地减少错误的连通关系,从而降低误判率.

方法中的密度阈值选择 $N_\epsilon = N(2\epsilon)^M$. 确定半径参数 ϵ 后,在理想情况下,如果类的中间区域没有数据样点,则选择一个尽可能小的 N_ϵ 即可. 但在实际情况中,由于有标签设备个数 N_L 远小于全部设备个数 N,4.1 节得到的间隙 v 中很有可能存在大量数据点,这样以来使用一个较小的 N_ϵ 可能会导致错误的类合并. 当所有的测试点在特征空间均匀分布时,N_ϵ 参数选择最有可能导致错误的合并. 在数据样点均匀分布的情况下,可以估计出聚簇间隙中的数据点的上限,以此推断 N_ϵ 的取值下界. 令全部数据的宽度为 1,全部测试点的数目为 N,那么每个间隙的宽度为 v,间隙中的数据样点出现的概率为 $s^M/1$,s 为数据样点可能出现的区域,故间隙中样点出现次数的期望为 Ns^M. 由于之前已经获得了半径阈值 ϵ,可以将空间大小近似估计为 $s = (2\epsilon)^M$,所以本文选择 $N(2\epsilon)^M$ 作为密度阈值. 图 3 显示了在 Intel Lab 数据集上使用不同参数能达到的最佳的 F-measure

值,图中标注的点为使用以上参数得到的 F-measure 值. 以上方法较为保守地进行了参数选择,且获得了较好的 F-measure 值.

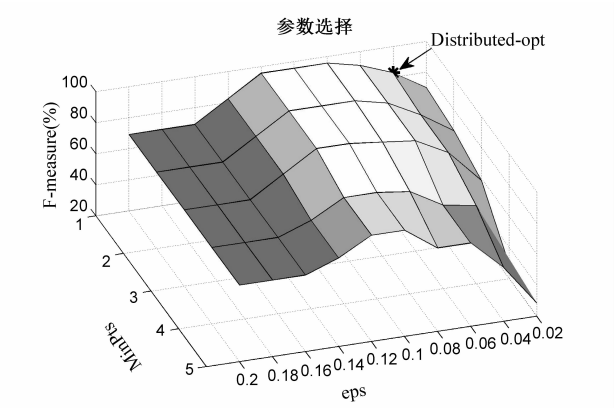


图3　基于间隔最大化的聚类参数选择效果示意

定义 $f_{\text{opt}}(x_i, x_j)$ 为由 4.1 中的算法求得的最佳距离函数,$x.L$ 为 x 的聚类标签,分布式聚类过程如算法 2 所示. 该分布式聚类算法的最坏时间复杂度等价于上文所讨论的超时时间,为 $O(N t_m)$. 平均情况下,设原始设备最终可被归为 K 类,那么整体的时间消耗等价于所有类内部遍历过程的最大时间消耗,均摊时间复杂度可以估计为 $O(t_m \log N/K)$.

5　仿真实验

实验包含两部分,首先对参数选择方法的有效性进行评估,之后评估算法整体性能,重点测试算法的准确性. 本实验需要使用较多的节点设备,故采用了仿真实验的方式进行验证. 仿真采用以下方式进行:每次循环分别处理每个节点的消息和事件,之后将状态进行存储,以模拟节点状态的变化;每次循环使用随机的顺序执行不通过节点的程序片段,以模拟真实情况的并发特性.

实验使用了 Intel Lab 数据集作为验证,使用 Matlab 进行仿真. Intel Lab 数据集包含 54 个节点,每条数据包含温度、湿度、光照、节点电压 4 类数据. 本实验选择上述 4 类数据,同时选择了 20 个节点,截取了其中的一个小时的数据作为原始采样. 当使用全部数据输到本文的方法时,整个输入规模为 80 条原始数据,最终聚为 4 类.

实验验证聚类结果准确率时,使用 F-measure[16] 作为衡量指标. F-measure 方法可以综合聚类方法的查准率与查全率性能,其表达式为:

$$F = (1+\beta^2)PR/(\beta^2 P + R) \tag{4}$$

其中 β 为查准率与查全率的权重参数,实验中使用 $\beta = 1$. 对于聚类操作,查准率为正确分入该类的数据点数目与全部分入该类的数据点数目的比例,P 是每个类查准率的平均值;查全率为正确分入的数据点数目与应该分入该类的数据点数目的比例,R 为每个类查全率的平均值. 实验使用 F、P、R 这三项指标评价方法的准确率.

5.1　参数选择方法效果评估

本小节对 4.2 节所使用的参数选择方法的效果进行了评估. 实验分别测试了个数阈值参数 N_ϵ 与核心点判定半径阈值 ϵ 取不同值时(本文针对 Intel Lab 数据集,取参数 $\epsilon \in [1,5]$,搜索步长为 1,取参数 $N_\epsilon \in [0.02, 0.2]$,搜索步长为 0.02),使用分布式聚类算法得到可以得到在不同网络规模下的查准率与查全率两种指标的最优值的与平均值,以对比 4.2 节的方法的效果. 距离函数均使用已进行优化的函数.

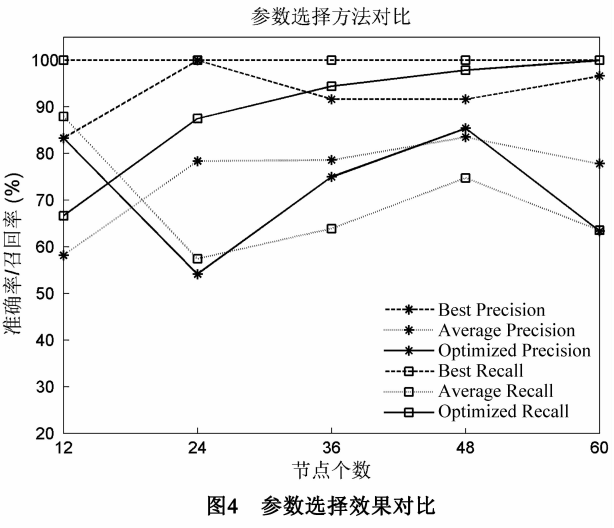


图4　参数选择效果对比

实验结果由图 4 所示. 其中星标线条为聚类方法的查准率,正方形标记为聚类方法的查全率;对于每组数据,点状虚线标记的数据为使用 4.2 节方法推测获得的参数的聚类效果,其他两条数据分别为最佳参数条件下的结果与平均情况. 可以看出,本文的参数获取方法在查准率上有较为明显的优势,在查全率上并没有很好的逼近使用最佳参数的查全率. 但由于平均情况的高查全率建立在较高的错判可能性上,并不宜于在设备描述场景下使用;相较而言,通过优化方法推断得到的参数具有较高的查准率,更为适合设备描述场景.

5.2　整体准确率评估

本文采用以下方式进行整体性能评估实验:分别选择 4、8、12、16、20 个原始设备的 4 类数据作为输入,进行 5 次实验. 每次实验选用 1/4 的数据作为度量优化数据,选用全部数据作为分布式聚类数据. 实验主要用于对比基于分布式度量优化方法的特点,使用 SPITFIRE 中的设备描述方法作为参照. 实现本文度量选择方法时,实验选用了 EMD、FFT、DIFF 三种度量方式.

EMD(Earth Mover's Distance)可以较好地衡量原始数据直方图之间的相似程度;FFT 反映出数据频域内的特性;DIFF 计算两个近邻数据之间的差值,并对差值进行了直方图统计,计算直方图之间的近似程度. SPITFIRE 的实现方法较为简单,仅选用单个特征作为距离函数对 SPITFIRE 进行模拟. 实验中确定聚类参数时,为了确保公平,均使用了度量优化产生的自动参数.

图 5 显示了不同输入规模下不同方法的查准率 P,其中 Distributed-opt 是度量经过优化后最终聚类的准确结果. 结果表明,使用了分布式度量优化的方法在不同网络规模下的查准率均优于使用单一度量的方法. 这主要得利于 4.2 中较为保守的参数选择策略. 在 Intel Lab 数据集下,这种方式可以达到 93% 的准确率,比平均的单一的特征选择的准确率高出 20.4%,比最坏情况平均高出 31.1%. 实验表明包含分布式度量优化的分布式聚类具有较高的查准率.

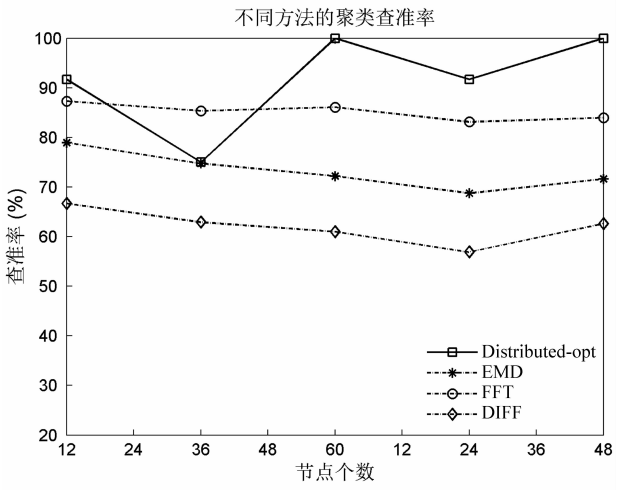


图5 自动描述聚类查准率

类似的,图 6 显示了不同输入规模下算法的查全率 R. 图中的查全率相对于其他方法没有显著的提高,在 Intel Lab 数据集上多次运行发现查全率相对于平均水平仅提高了 2%,没有明显变化. 这种结果的原因有 2:(1)4.1 中提出的优化方法对于离群点过于敏感;(2)真实数据集在不同度量下并不规则. 以上两点导致度量优化本身对相同类内部节点的连通性造成了影响. 可能的解决方案是在 4.1 的优化目标上加入松弛变量作为权衡.

图 7 显示了不同方法的查准率、查全率,以及在 $\beta=1$ 时的 F-measure 值的综合对比. 其中 LP-opt 是本文基于线性规划完成的一个集中式方法,准确性略好于分布式方法 Distributed-opt,原因在于分布式方法进行迭代求解时引入的误差. 总体上,本文的两种基于优化的方式在查准率 P 相对于使用单一特征作为度量的

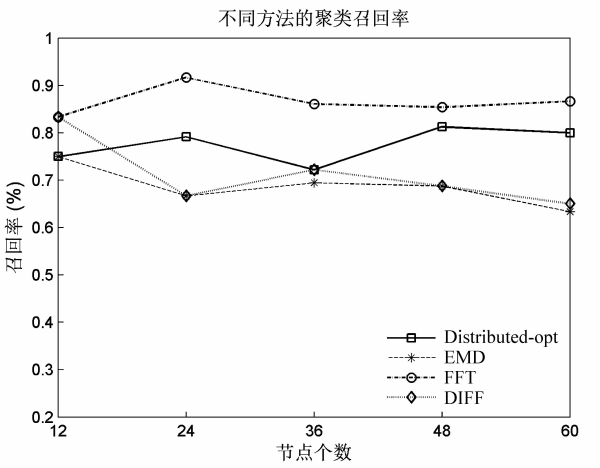


图6 自动描述聚类查全率

SPITFIRE 方法具有明显优势,在查全率上与基本持平.

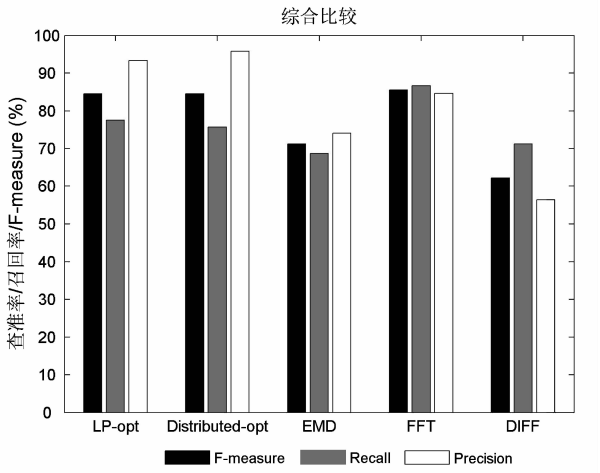


图7 自动描述综合效果对比

6 结束语

本文讨论了海量设备场景下,编写物联网设备描述文件所面对的问题与挑战,提出了一种用于物联网感知设备的自动描述方法. 方法利用不同感知设备采集到的数据数值特性之间关系,通过设备之间的协商,自动获得自身的描述. 相对于现有的其他方法,本文提出的方法具有两个明显优势. 首先,本方法是一种分布式的方法,更有利于设备的快速部署;其次,本方法可以实现"零输入",依靠现有的少量描述信息即可进行推断,可以达到较高的准确率.

一个准确完整的物联网设备描述文件对于物联网设备与资源的发现与共享具有重要意义. 本文提出的方法在查全率与方法鲁棒性上仍有进一步的改进空间;此外本方法只能应用于感知设备,对于更为复杂的中间设备的自动描述任务,需要功能更强的推理机制去完成,这均将是未来的工作方向.

参考文献

[1] Web Services Description Language (WSDL) 1.1[EB/OL]. http://www.w3.org/TR/wsdl.2001-3-15.

[2] Universal Plug and Play (uPnP)[EB/OL]. http://www.upnp.org.2014-4-20.

[3] Sensor Model Language (SensorML)[EB/OL]. http://www.opengeospatial.org/standards/sensorml.2014-2-24.

[4] Dennis Pfisterer, et al. SPITFIRE: toward a semantic web of things[J]. Communications Magazine, 2011, 49(11): 40-48.

[5] Eric P, et al. Distance metric learning with application to clustering with side-information[A]. Advances in neural information processing systems[C]. USA: AAAI Press. 2002. 521-528.

[6] Ester M, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[A]. KDD 96[C]. New York: ACM Press. 1996. 226-231.

[7] Phillip B, Gibbons, et al. Irisnet: An architecture for a worldwide sensor web[A]. Pervasive Computing[C]. USA: IEEE Press. 2003. 22-33.

[8] Nissanka B P, et al. Tiny web services: design and implementation of interoperable and evolvable sensor networks[A]. Proceedings of the 6th ACM conference on Embedded network sensor systems[C]. New York: ACM. 2008. 31-36.

[9] Charith P, et al. Context-aware sensor search, selection and ranking model for internet of things middleware[A]. IEEE 14th International Conference on Mobile Data Management[C]. USA: IEEE Press. 2013. 314-322.

[10] YAN Tingxin, et al. Distributed image search in camera sensor networks[A]. Proceedings of the 6th ACM conference on Embedded network sensor systems[C]. New York: ACM Press. 2008. 155-168.

[11] Cuong Truong, et al. Fuzzy-based sensor search in the web of things[A]. Internet of Things (IOT), 2012 3rd International Conference on the[C]. USA: IEEE Press. 2012. 127-134.

[12] Intel Lab Data, [EB/OL]. http://db.csail.mit.edu/labdata/labdata.html. 2004-6-2.

[13] Brian Kulis. Metric learning: A survey[J]. Foundations & Trends in Machine Learning, 2012, 5(4): 287-364.

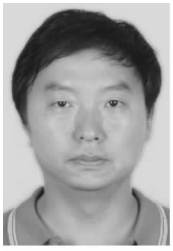
[14] Boyd, S, Vandenberghe, L. Convex Optimization[M]. Cambridge: Cambridge University Press, 2004. 146-152.

[15] Platt, J(1998). Sequential minimal optimization: A fast algorithm for training support vector machines[R]. http://www.msr-waypoint.com/pubs/69644/tr-98-14.pdf. 1-21.

[16] David MW Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation[J]. Journal of Machine Learning Technologies. 2011, 2(1): 37-63.

作者简介

李　勐　男,1989 年出生,现为中科院计算技术研究所博士研究生. 主要研究方向为物联网资源的发现与检索.
E-mail: limeng@ict.ac.cn



王晓峰　男,1978 年出生,博士,目前为中国科学院计算技术研究所助理研究员. 主要研究方向为物联网大数据处理、智能移动计算、数据挖掘、人工智能.
E-mail: wangxiaofeng@ict.ac.cn

崔　莉(通信作者)　女,1962 年出生,博士,中国计算机学会高级会员,中国科学院计算技术研究所研究员、博士生导师、从事无线传感器网络相关研究.
E-mail: lcui@ict.ac.cn