Computing
https://doi.org/10.1007/s00607-018-0603-z

WITS: an IoT-endowed computational framework for activity recognition in personalized smart homes

Lina Yao[1] · Quan Z. Sheng[2] · Boualem Benatallah[1] · Schahram Dustdar[3] · Xianzhi Wang[1] · Ali Shemshadi[4] · Salil S. Kanhere[1]

Received: 22 February 2018 / Accepted: 28 February 2018

Abstract Over the past few years, activity recognition techniques have attracted unprecedented attentions. Along with the recent prevalence of pervasive e-Health in various applications such as smart homes, automatic activity recognition is being implemented increasingly for rehabilitation systems, chronic disease management, and monitoring the elderly for their personal well-being. In this paper, we present WITS, an end-to-end web-based in-home monitoring system for convenient and efficient care delivery. The system unifies the data- and knowledge-driven techniques to enable a real-time multi-level activity monitoring in a personalized smart home. The core components consist of a novel shared-structure dictionary learning approach combined with rule-based reasoning for continuous daily activity tracking and abnormal activities detection. WITS also exploits an Internet of Things middleware for the scalable and seamless management and learning of the information produced by ambient sensors. We further develop a user-friendly interface, which runs on both iOS and Andriod, as well as in Chrome, for the efficient customization of WITS monitoring services without programming efforts. This paper presents the architectural design of WITS, the core algorithms, along with our solutions to the technical challenges in the system implementation.

Keywords Activity recognition · Localization · Shared dictionary learning · Internet of Things · Smart homes

✉ Lina Yao
lina.yao@unsw.edu.au

[1] School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

[2] Department of Computing, Macquarie University, Sydney, Australia

[3] TU Wien, Vienna, Austria

[4] School of Computer Science, University of Adelaide, Adelaide, Australia

Published online: 06 March 2018

Mathematics Subject Classification 68U35

1 Introduction

The world population is aging rapidly due to the increasing life expectancy and declining birth rate. The global share of elder people (aged 60 years or over) has increased from 9.2 percent in 1990 to 11.7 percent in 2013. It is expected that this trend will continue, reportedly reaching 21.1 per cent by 2050.[1] In the meantime, the human lifespan has increased as a result of increased health awareness and improved quality of food and medicine. Yet, the elderly are susceptible to various types of injuries and accidents and consequently require more medical care facilities. The increasing expenditures on health care for the elderly result in great financial impacts, urging healthcare providers to develop novel technologies with good usability and cost-effectiveness, such as smart homes, to address issues like monitoring the status of the elderly or chronically ill patients in their own homes. Such technologies are essential to the elderly for them to live independently and safely [7,16,23].

A typical smart home is equipped with sensors, wherein cameras, IR sensors, ambient sound, heat, as well as contact sensors are mounted on furnitures and used in the home environment in a non-intrusive manner. The sensors continuously collect data about the location and activity of the subjects in the environment without interfering with residents' daily activities. Such data would be a valuable asset to understand people's behaviors and their well-beings [2,6,17]. For example, in the following scenario, by monitoring the daily behavior routine of an old person, an assistant service can track how completely and consistently his daily routine is performed and on this basis to determine whether and when intervention or assistance is needed, e.g., whether the person is falling in bathroom, whether the person frequently forgets to take medicine or turn off the oven (which are early signs of dementia that harms his cognitive health), or whether the person has spent too much watching TV instead of doing exercises.

Over the years, significant research efforts have been contributed to developing smart home systems for monitoring the activities of daily living (ADLs). These efforts can be used to infer broader patterns such as common daily routines. However, several challenges remain concerning the interoperability among sensors and heterogeneous information, affordability, and accuracy. The rising of the Internet of Things (IoT) pushes the ambient intelligence forward by expanding the scale and scope of the healthcare domain significantly. IoT also bring new opportunities by facilitating the collection and reflection of diverse information in the physical world.

We propose a novel Web-based personal-wellness monitoring system to observe and quantify the residents' activities and abnormalities at homes. The proposed project is based upon our previous work on Internet of Things-aware smart homes [28,31], one of the stepping stones along a pathway of innovations that enable off-site specialists to observe and diagnose patients or residents' wellness reliably and accurately in real-time. It is especially important for special groups of people, such as old adults

[1] http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf.

with physical and cognitive limitations or postpartum women, for whom it is often inconvenient or impossible to take outdoor activities.

In this paper, we present WITS, Web-based Internet of Things Smart home, as an end-to-end solution to facilitate the development of smart home applications. We propose to develop a hybrid recognition framework, which leverages multi-task learning, dictionary learning, and rule-based reasoning to observe and quantify the changes in the readings of sensors deployed in home environments, so as to continuously track residents' daily behaviors and detect any abnormal events for early and timely medical assistance. We develop a scalable IoT middleware that connects the physical world and the cyber world to overcome the home interactivity and interoperability issues.

WITS makes contributions by addressing three major challenges.

How to seamlessly integrate the heterogeneous contexts of a smart home with better interoperability? There have been many insights proposed to boost smart home systems [16]. However, most of current systems are still hampered by interoperability issues [8]. We propose to address this challenge by developing an IoT middleware, which provides the necessary infrastructure to transparently and seamlessly glue heterogeneous resources and services together, by accessing sensors and actuators across different protocols, platforms, and locations over the Internet stack. Our IoT middleware makes smart home interactivity a reality, and enables smart homes to become more affordable and accessible with better interoperability between IoT-enabled objects, user devices, and cloud services by leveraging the Web architecture. Compared to existing IoT middlewares [19,22,26], our work not only deals with the interoperability issue as many smart home systems but also offers an effective way to describe diverse objects for provisioning disparate information flow in smart home systems.

How to effectively recognize user activities and detect abnormalities by utilizing easily-accessible sensory data streams? The proposed system employs a series of core algorithms to recognize and track people's daily routines of activities and to further identify abnormalities. The objective is to discern activity patterns from the home sensory data by considering multidimensional contexts. However, it has not been fully investigated how to interpret human behaviors from the massive, user-generated, heterogeneous, multi-modal, and context-relevant sensory data [3,5,17,29,30], as a result of several challenging issues. The first issue is how to deal with intra-class activity variability. Since every people may have their distinct behavior patterns, the same activity may be performed differently by different people. In many cases, the same person may perform the same activity differently, due to various factors such as the stress, fatigue, or emotion of the person, or even the surrounding environment. The second issue is how to handle unseen samples effectively. It is usually hard to obtain a large set of activity behaviors owing to the difficulty of gathering sufficient training samples from multi-domain contexts (time-consuming and tedious annotating). The third challenge is that most approaches require intensive training and assume all of the training samples are annotated and available in advance. Yet such assumptions are often impractical for real-world applications, as in many cases, unlabeled new data keeps coming continuously.

To tackle above challenges, we propose a novel method that leverages the joint advantages of multi-task learning [9] and dictionary learning [1] for automatic activity recognition. Both of the above techniques have proven successful in a wide range of areas [10,25,27]. Dictionary learning can generate a compact and discriminative manifesto of features, and multi-task learning can achieve good performance by learning multiple tasks that share commonalities simultaneously. Since such learning methods generally require more ground-truth data than we can obtain for complex activity inference, we introduce rule-based reasoning to help handle the unseen data. We specifically propose a joint framework to derive both activity-specific dictionaries (i.e., people-independent knowledge) and person-specific dictionaries, where intrinsic relationships have invariant properties and are less sensitive and variant with different subjects. Such properties can be used as signatures to profile activities and to support more accurate activity recognition via rule-based reasoning based on multi-type contextual information (e.g., symbolic location and object use) .

How to manage the personalized smart home in a more friendly way? Another key challenge to smart homes is how to create customizable services in an effective and user-friendly manner. This involves designing an easy use and powerful interface for accessing, exchanging, and manipulating information from the smart homes, which is what most of the existing smart home systems lack, to pursue the user satisfactions.

To address this challenge, we develop a user-friendly graphical interface using the Trigger-Action Programming (TAP) model to achieve value-added service customization. This model allows users to configure higher-level rules via a Web browser by integrating the inferred contexts, e.g., user locations, activities and objects in use, without programming effort. Various services and their semantic information are abstracted as universal services in WITS and represented as icons on the graphical interface. Through this easy-to-use Web interface, users can specify and manage complex rules and build their personalized smart home applications in a Drag-and-Drop manner. Compared to the existing efforts that adopt TAP [21,24], WITS can support advanced rule customization with better scalability and flexibility.

The remainder of this paper is organized as follows. We first review the related work. Next, we overview the WITS system, followed by the core algorithms for activity recognition. We then describe our system implementation including software, hardware, and interface design, and finally, we conclude the paper by pointing out possible future directions.

2 Related work

Over the years, significant research efforts have been contributed to smart home systems such as the Aware Home [13], House_n in MIT [12], CASAS [20], and Adaptive House [18]. All these efforts focus on people's direct interactions with the technologies and help to infer broad patterns such as common daily routines such as facilitating the study of different smart home technologies in more depth and in contexts that closely resemble the real-world domestic spaces. However, several challenges remain: (1) heavily reliance on people's involvement such as the wearing of battery-powered sensors, which may not be practical in real-world situations; (2) lacking a synthetic way of deploying ubiquitous available sensors and taking advantage of disparate services offered by smart homes; and (3) lacking a user-friendly interface for end users to access and create personal rules for smart home automation.

In addition, most existing IoT middlewares allow users to add sensors as they desire and offer users tools (simple App or Web browser) to view the sensor-collected raw data. However, Neither of them is able to support smart home applications very well because they usually provided limited functionalities in terms of helping user develop complex functionality, interacting with other applications, or interpreting the data into reusable high level services. Some systems even limit end-users on the types and the numbers of sensors that they can use. In order to achieve scalability and usability, our proposed system not only monitors the data collected from sensor in real-time but also automatically converts the data into actionable information using intelligent event generator or according to users predefined conditions. This enables IoT applications to be developed based on high-level contexts that are independent of low-level physical properties of the sensors or devices. A thorough comparison of 22 existing IoT middleware systems and research prototypes including our proposed WITS is shown in Table 1. bidding adieu to

3 WITS overview

WITS consists of three main components, namely *Sensing Management*, *Context Management*, and *Rule Management*, as shown in Fig. 1.

The *Sensing Management* component manages all types of sensors (including RF tags and ambient sensors), collects and processes the raw data, and provides a universal API for higher-level programs to retrieve the status of physical entities. This component works in a scalable, plug-and-play fashion, where new sensors can be easily plugged-in and old sensors can be easily removed. Several software components are included in this part for filtering and cleaning raw sensor data, and adapting such data for high-level applications. Since some devices work with more than one sensor and the sensor readings may come asynchronously, a data access processor is used to allow the system to provide data synchronously, which lays the foundation of interoperability for smart home applications [8]. WITS allows physical entities to be mapped to virtual resources to achieve a seamless integration of physical environment with organization business processes. In WITS, we exploit the schema.org, a recent initiative (launched in 2011 by Bing, Google, and Yahoo!) for different data semantics to be understood by both human users and machines. The universal RESTful API provided in our design enables higher level programs to retrieve the status of physical things with specified addresses, without needing to know where and how to find the physical sensors associated with the objects.

This module is also in charge of data preprocessing. The physical activity and movement sensory data may contain noise, which deteriorates the clustering quality. To alleviate the disturbance from noise, we adopt the Hodrick–Prescott filter [11], a well-known trend analysis method in economics, to smooth the sensory data stream. This filter separates the time-series RSSI data into growth component g_t and cyclical components c_t. The objective function is

$$\sum_{t=1}^{T} c_t^2 + \lambda \sum_{t=2}^{T-1} ((g_{t+1} - g_t) - (g_t - g_{t-1}))^2 \tag{1}$$

Table 1 IoT middleware comparison

Categories	Prototypes	Functionality							Interface		Implementation		
		S	P	RM	RE	CAR	SN	MSS	M	W	RDL	API	PL
Smart energy	Webenergy	–	–	–	–	–	–	–	●	●	JSON XML	REST	Java
Tagging objects	EPCIS Web Adapter	–	–	–	–	–	–	–	–	●	JSON XML	REST	–
IoT middleware	Zetta	–	–	●	–	–	–	●	–	●	–	REST	–
	Kaa IoT Platform	–	–	●	–	–	–	●	–	–	–	–	–
	w3c/web of things	●	●	–	●	–	–	–	●	●	JSON TDL LD	REST	–
	ThngSpeak	–	–	●	–	●	–	–	–	●	–	REST	–
	EVRYTHNG	●	●	●	–	–	●	●	–	●	JSON	REST	–
	OpenIoT	●	●	●	–	–	–	●	–	●	SSN Specification	–	–
Smart Homes	openHAB	–	–	–	●	–	–	●	–	–	–	–	–
	Eclipse Smart Home	–	–	●	–	–	–	●	●	●	–	–	Java
	OpenDomo	●	●	●	●	–	–	●	–	–	–	–	–
	FreeDomotic	●	●	●	●	–	–	–	–	–	–	–	–
	Calaos	–	–	●	–	–	–	–	●	●	–	–	–
	MisterHouse	–	–	–	●	–	–	–	–	●	–	–	Perl
	Wosh	–	●	●	–	–	–	●	–	●	XML	SOA	C++
	IoTivity	●	●	–	–	–	–	●	●	●	JSON	REST	–
	RaZberry	–	–	–	–	–	–	–	●	●	JSON	–	JavaScript
	The Things System	●	●	●	●	–	–	●	–	●	JSON	–	node.js
	PrivateEyePi	●	●	●	●	–	–	●	●	●	–	–	Python
	IoTSyS	–	–	●	–	–	–	●	–	●	–	–	Java
	House_n	–	–	–	●	●	–	●	–	–	–	–	–
	CASAS	–	–	–	●	●	–	●	–	●	–	–	–
Hybrid project	WITS	●	●	●	●	●	●	●	●	●	JSON	REST	C#/Java

S, security (authentication, https etc); P, privacy (access control etc.); RM, real-time monitoring; RE, Rule Engine; CAR, complex activity reasoning; MSS multi-source support; SN, social network (Twitter etc.); M, mobile interface; W, web interface; RDL, Resource Description Language; PL, programming language; API, application programming interface; ●, function enabled; –, not available/not applicable

where λ is the smoothing parameter. The programming problem is to minimize the objective over all $g_t|_1^T$. The incoming sensor readings are divided into data segments of length δt= 10 seconds. This time slice duration is long enough to be discriminative yet

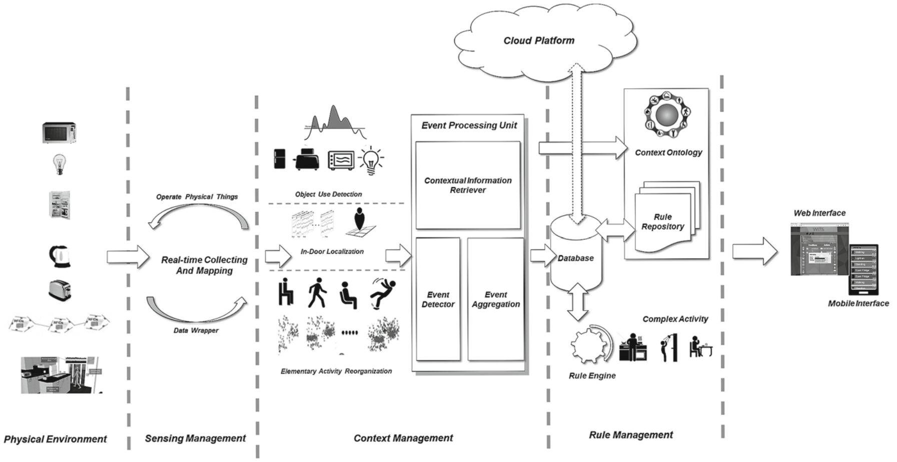


Fig. 1 Overview of the WITS system workflow

still short enough to provide accurate labeling results. The information of each segment is then transformed into 12 types of statistical features such as Min, Max, Mean, Root Mean Square, Variance, Standard Deviation, Kurtosis, Skewness, Entropy, Median, Zero Crossing Rate, and Mean Cross Rate. The *Context Management* component aims at capturing contextual events. A typical smart home scenario involves three *atomic* contexts, namely basic human activities (e.g., getting up, walking, or lying on the bed), indoor localization (e.g., in bedroom or kitchen), and human–object interactions (e.g., using kettle or turning on light). Accordingly, a Event Detector employs a set of machine learning algorithms to analyze the signal fluctuations, in order to detect three contextual events, namely user activities, symbolic location information (room-level position), and object use events.

WITS focuses on extracting and aggregating contextual events in a pipeline manner automatically. It extracts contextual information (e.g., temporal and spatial information), and then indexes and stores all the events with their related information in a database for data mining purposes. In WITS, users can focus on the functionalities of applications without worrying about issues such as connecting to the database, opening connections, querying with specified languages, or handling the results. For example, in the context of aged care, doctors can access various personal data like pulse rate, blood pressure of a old person, whether the old person is taking medicine on time, or whether the person stays in a certain place for too long to make better decisions

The *Rule Management* component supports complex activity reasoning with the *Rule Engine* and high-level customizable composition to realize context-aware home automation. It derives underlying knowledge from the heterogeneous and uncertain historical context to supervise the service customization and enables the system to generate new context-aware automation rules (stored in the *Rule Repository*) and to adapt to dynamic context environments. In particular, an ontology-based knowledge base, *Context Ontology*, is used to represent the key contexts and their interrelations in smart home environments. It provides the formal semantics to describe context knowledge about objects, relationships and domain constraints, and is assisted with rule matching to support the sharing and integration of structured context information.

3.1 Activity recognition

This section details the core techniques and algorithms for activity recognition in WITS.

3.1.1 Model formulation

Intuitively, it is reasonable to assume some local commonalities under the intra-class variabilities shared by all the activities. For example, the same activities performed by different people should share some commonality, e.g., walking forward and standing both shares a torso perpendicular-like motion. These intrinsic relationships have invariant properties and are less sensitive to different people. Thus, they can be used as a signature to profile each activity style in activity recognition. Instead of uncovering the shared common space across activities using the original feature space or learning dictionaries from each activity separately, we aim at leveraging both the advantages of dictionary learning and multi-task learning by extracting the underlying K style-specific dictionaries (person-independent), which are the more compact and discriminative representations of activities under a multi-task learning framework. We integrate them along with a collection of activity-specific dictionaries (person-dependent) in a joint framework

Let $\mathbf{X}_k \in \mathbb{R}^{n_k \times m} = [\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{n_k}]$ be the sensor samples, where $\mathbf{x}_k^i \in \mathbb{R}^m$ is a m-dimensional feature vector of sensory data, n_k is the number of samples in the k-th task, and $k \in [1, K]$ tasks corresponding to K activity styles. We assume a shared structure between different activities $\mathbf{Q} \in \mathcal{R}^{m \times sd}$, where $sd < m$ is the dimensionality of the subspace to project the original feature space into a low dimensional subspace. Furthermore, we consider the relationships between these weight vectors imposed by the neighbourhood structure of sensory data points that characterize each type of activity and define the following objective function:

$$\begin{aligned}\mathcal{J}(\mathbf{X}_k, \mathbf{C}_k, \mathbf{D}_k, \mathbf{D}, \mathbf{Q}) = & \underbrace{\sum_{k=1}^{K} ||\mathbf{X}_k - \mathbf{C}_k\mathbf{D}_k||_F^2}_{1} + \underbrace{\lambda_1 \sum_{k=1}^{K} ||\mathbf{C}_k||_1}_{2} \\ & + \underbrace{\lambda_2 \sum_{k=1}^{K} \sum_{a,b} \mathbf{W}_{a,b} ||(\mathbf{C}_k)_{a.} - (\mathbf{C}_k)_{b.}||_F^2}_{3} \\ & + \underbrace{\lambda_3 \sum_{k=1}^{K} ||\mathbf{X}_k\mathbf{Q} - \mathbf{C}_k\mathbf{D}||_F^2}_{4} \qquad (2)\end{aligned}$$

where $\mathbf{D}_k \in \mathbb{R}^{d\times m}$ is the activity dictionary ($d \leq m$), and $\mathbf{C}_k \in \mathbb{R}^{n_k\times l}$ corresponds to the sparse representation coefficients of $\mathbf{X}_k$. $\mathbf{D} \in \mathbb{R}^{d\times sd}$ is the activity style dictionary learned in the shared subspace and $\mathbf{D}_{j.}$ in the constraints denotes the j-th row of $\mathbf{D}$. There are four terms in the right side of the equation:

1. Reconstruction error on individual dictionary The reconstruction error (the first term) for a usual/normal activity should be a small value because the learned dictionary represents knowledge in the previously seen sensory data. A small reconstruction error indicates the information within the newly observed data segment has appeared in early available samples.

2. Sparsity regularization Since the dictionary is learned to maximize the sparsity of reconstruction vectors for normal activities (along with a fairly small reconstruction error), it is necessary to impose sparsity for reconstructing normal events. The sparsity regularization (the second term) ensures that reconstruction vectors are sparse for normal activities while dense for unusual/abnormal activities.

3. Smoothness regularization The smoothness regularization (the third term) is calculated based on the fact that neighboring sensory variations are more likely to be involved in a same categorical behavior, where $\mathbf{W} \in \mathbb{R}^{n_k\times n_k}$ denotes the adjacency matrix of $[\mathbf{x}_k^1, \ldots, \mathbf{x}_k^{n_k}]$. We use the cosine similarity to measure the affinity.

$$\mathbf{W}_{a,b} = \frac{\mathbf{C}_k(a,:)\cdot\mathbf{C}_k(b,:)}{\sqrt{\sum_i \mathbf{C}_k^2(a,i)}\sqrt{\sum_i \mathbf{C}_k^2(b,i)}} \tag{3}$$

4. Reconstruction error on shared dictionary The reconstruction error of the shared dictionary (the fourth term) manifests the underlying commonalities between the people who perform the same activities with multi-task learning. The term is used to resist the noisy data and reduce the misclassification caused by inter-person variability. It should be small for normal activities, while big for abnormal activities.

3.1.2 Optimization

The objective function (Eq. 2) quantifies the normality of activity in every data segment $\mathbf{X}_k$ with any reconstruction weight vector $\mathbf{C}_k$, any people-specific dictionary $\mathbf{D}_k$, and activity-specific dictionary $\mathbf{D}$. The lower J is, the more likely $\mathbf{X}_i$ is generated by normal behavior. We obtain the optimal weight vectors $\mathbf{C}_k^*$ and dictionaries ($\mathbf{D}_k^*$, $\mathbf{D}^*$, $\mathbf{Q}^*$) by solving the following optimization problem:

$$\begin{aligned}(\mathbf{C}_k^*, \mathbf{D}^*, \mathbf{D}_k^*, \mathbf{Q}) &= \arg\min_{\mathbf{D}_k,\mathbf{C}_k,\mathbf{Q},\mathbf{D}} \sum_{k=1}^{K} J(\mathbf{X}_k, \mathbf{C}_k, \mathbf{D}, \mathbf{D}_k, \mathbf{Q})\\ \text{s.t.}\quad \mathbf{Q}^\top\mathbf{Q} &= \mathbf{I}\\ (\mathbf{D}_k)_{j.}(\mathbf{D}_k)_{j.}^\top &\neq 1, \quad \forall j = 1,\ldots,l\\ \mathbf{D}_{j.}\mathbf{D}_{j.}^\top &\neq 1, \quad \forall j = 1,\ldots,l\end{aligned} \tag{4}$$

Since there exists ℓ_1 minimization, we use alternate optimization algorithm to find a local optimum. In particular, we alternatively minimize one variable while fixing the other three parameters. For example, with a learned dictionary $\mathbf{D}^*$, given a newly observed activity $\mathbf{X}'$, the algorithm learns the optimal reconstruction weight $\mathbf{C}'_k$, $\mathbf{Q}'$, and $\mathbf{D}'$. Consequently, $J(\mathbf{X}', \mathbf{C}'_k, \mathbf{Q}', \mathbf{D}^*)$ measures the normality of $\mathbf{X}'$. $\mathbf{X}'$ is detected as unusual if its corresponding $J(\mathbf{X}', \mathbf{C}'_k, \mathbf{Q}', \mathbf{D}*)$ is larger than a certain threshold.

Learning $\mathbf{D}$ *with* $\mathbf{D}_k$, $\mathbf{C}_k$ *and* $\mathbf{Qs}$ *fixed* With the dictionary $\mathbf{D}$ fixed, the optimization problem turns into:

$$\min_{\mathbf{D}} \sum_{k-1}^{K} ||\mathbf{X}_k\mathbf{Q} - \mathbf{C}\mathbf{D}|| \text{ s.t. } \quad \mathbf{D}_{j.}\mathbf{D}_{j.}^{\top} \leq 1, \quad \forall j = 1, \ldots, l \tag{5}$$

The constraint in Eq. (5) is to prevent the terms in $\mathbf{D}$ from being arbitrarily large, which would lead to small values of coefficients. The above optimization problem is a least square problem with quadratic constraints, which can be solved using Lagrange dual. We adopt the efficient Algorithm 2 in [15].

Leaning $\mathbf{D}_k$ *with* $\mathbf{D}$, $\mathbf{C}_k$ *and* $\mathbf{Q}$ *fixed*. To compute $\mathbf{D}_k$, the optimization problem is as follows.

$$\min_{\mathbf{D}_k} ||\mathbf{X}_k - \mathbf{C}_k\mathbf{D}_k||_F^2 \text{ s.t. } \quad (\mathbf{D}_k)_{j.}(\mathbf{D}_k^{\top})_{j.} \leq 1, \quad \forall j = 1, \ldots, l \tag{6}$$

This problem can also be solved in closed-form using the same algorithm [15].

Learning $\mathbf{Q}$ *with* $\mathbf{D}_k$, $\mathbf{C}_k$ *and* $\mathbf{D}$ *fixed* The optimization problem is as follows:

$$\min_{\mathbf{P}} \sum_{k=1}^{K} ||\mathbf{X}_k\mathbf{Q} - \mathbf{C}_k\mathbf{D}||_F^2 \text{ s.t. } \quad \mathbf{Q}^{\top}\mathbf{Q} = \mathbf{I} \tag{7}$$

We derive $\mathbf{D} = (\mathbf{C}^{\top}\mathbf{C})^{-1}\mathbf{C}^{\top}\mathbf{X}\mathbf{Q}$ according to the Lagrange dual and replace $\mathbf{D}$ in above Eq. (7), which can be rewritten as:

$$\min_{\mathbf{P}} \sum_{k=1}^{K} tr(\mathbf{Q}^{\top}\mathbf{X}^{\top}(\mathbf{I} - \mathbf{C}(\mathbf{C}^{\top}\mathbf{C})^{-1}\mathbf{C}^{\top})\mathbf{X}_k\mathbf{Q}) \tag{8}$$

The optimal $\mathbf{P}$ is composed of eigenvectors of the matrix $\mathbf{X}^{\top}(\mathbf{I} - \mathbf{C}(\mathbf{C}^{\top}\mathbf{C})^{-1}\mathbf{C}^{\top})\mathbf{X}$ corresponding to the s smallest eigenvectors.

Learning $\mathbf{C}_k$ *with* $\mathbf{D}_k$, $\mathbf{Q}$ *and* $\mathbf{D}$ *fixed* Equation 2 is equivalent to the following optimization problem:

$$\min_{\mathbf{C}_k} \sum_{k=1}^{K} ||\mathbf{X}_k - \mathbf{C}_k\mathbf{D}_k||_F^2 + \lambda_1 \sum_{k=1}^{K} ||\mathbf{C}_k||_1 + \lambda_2 \sum_{k=1}^{K} \sum_{a,b} \mathbf{W}_{a,b}||(\mathbf{C}_k)_{a.} - (\mathbf{C}_k)_{b.}||_F^2$$
$$+ \lambda_3 \sum_{k=1}^{K} ||\mathbf{X}_k\mathbf{Q} - \mathbf{C}_k\mathbf{D}||_F^2 \tag{9}$$

As the above equation contains ℓ_1 minimization, it is a non-convex problem. We adopt the feature-sign search algorithm [14], which considers the non-zero elements of coefficients to convert this objective function into a standard, unconstrained quadratic optimization problem, to solve the ℓ_1 regularization problem. After obtaining the optimal dictionaries $\mathbf{D}^*$ and $\mathbf{D}_k^*$, given a test sample $\mathbf{X}'$, the activity label of $\mathbf{X}'$ can be obtained by computing its sparse coefficient $\hat{\mathbf{C}}_k$ and the minimal reconstruction error.

In addition, the proposed method can detect unusual activities (e.g., falls) after learning the optimal behavior-specific dictionary and person-specific dictionary. Assume that we have instances of all the normal activities, then we can detect a test instance whose patterns deviate from the learned model as an anomaly as long as a activity label is assigned to the test instance. For example, given a newly observed sensor data segment $\mathbf{X}'$ and current dictionaries $\mathbf{D}*$ and $\mathbf{D}_k^*$ and computed sparse coefficients $\mathbf{C}_k'$, a test instance is detected as unusual activity segment if the following criterion is satisfied: $J(\mathbf{X}', \mathbf{C}', \mathbf{Q}', \mathbf{D}^*) > \epsilon$, where ϵ is a threshold defined by users.

3.2 Coupling with rule-based reasoning

Compared to basic activities like walking, sitting and lying, which purely rely on signal fluctuations and do not consider any contextual information (e.g., location information or object use), we combine such contextual information to derive a series of complex activities. A complex activity is defined as a set of activities, multiple objects, and symbolic locations involved in a complex activity. For example, cooking is a complex activity, including a series of actions like walking, moving arms, standing etc, and a set of objects such as chop-board, knives, and oven. Since cooking usually involves a large number of different objects which are shared across multiple activities, object usage information and symbolic information (e.g., kitchen or living room) can help discriminate between activities such as making a toast or making a coffee. Such distinctions can be important for applications such as health monitoring or memory aids.

Since it is often difficult, even impossible in some cases, to obtain sufficient training data with involving such a broad range of contexts, we further develop a rule-based approach that leverages heterogeneous information such as object uses and location information (e.g., room presents) to recognize complex activities. The rule-based method can also help handle the unexpected cases, such as the unexpected changes in a context or an unseen activity that arrives after training the model, and improve the performance in the activity recognition. We propose to couple shared structure learning with a rule-based method to provide a definitive solution to activity recognition. It has the following advantages: (1) capturing the explicit patterns without training efforts; (2) improving the overall performance by correcting the learning mistakes and handling unseen cases; (3) covering cases that a pure learning method cannot handle, e.g., inferring complex activities based on actions and resident's locations.

To improve the usability of the system, we use the Trigger-Action Programming (TAP) programming paradigm for users to create rules. TAP is a simple programming model applied in smart home applications, in which the user associates a trigger with an action, such that the action can be automatically executed when the trigger event occurs. The most popular TAP interface is an online service called if-this-then-

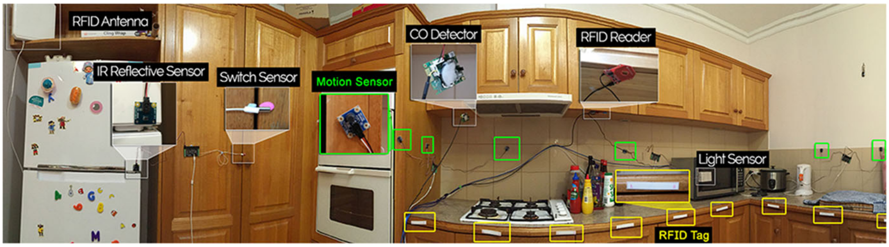


Fig. 2 Deployment of sensing units inside the kitchen area of the house

that (IFTTT). IFTTT allows users to create programs that can automatically perform actions like sending alerts or changing settings of a smart home when certain events or conditions occur.

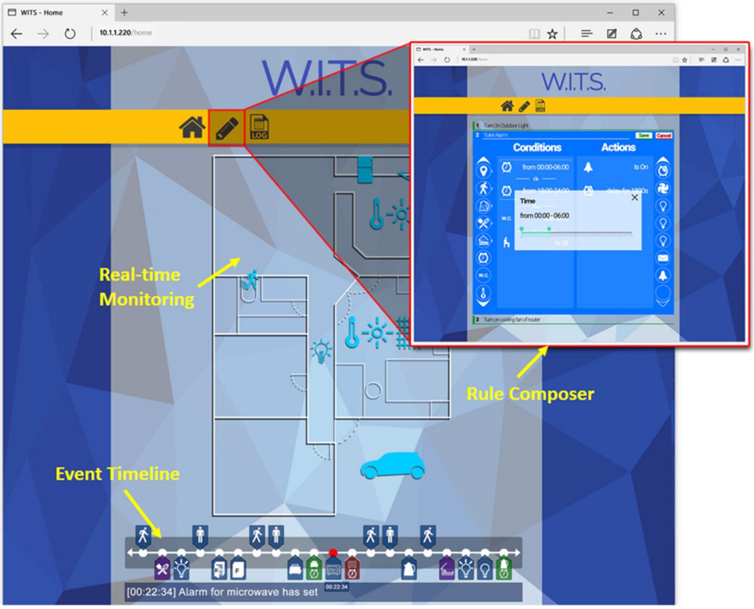
The *Visual Rule Programming Interface* component of WITS not only enables end-users to create rules for smart home automation but also offers a rule-based reasoning for high-level service (or resource) composition under Trigger-Action Programming paradigm through a Web-based rule editing interface. in our system, a rule consists of two parts: a *trigger* and an *action*. A trigger is a composition of a set of Boolean expressions. An action is a set of settings of entities. As shown in the sub figure of Fig. 3, $Toilet.Occupied = true \wedge Duration >= 30mins$, an action, *SendAlert*, will be triggered to send an alert to the corresponding agent (e.g., a caregiver). Compared to previous work that applies TAP in smart home applications, which only support a limited number of logical operators, our system has three advantages: (i) it supports richer logical operators to help end-users create more complex rules [21], (ii) it supports a broader range of triggers such as human activities, changes of object status, and human mobility [24], and (iii) it tolerates ambiguous status and events and allows flexible conjunctions of events and actions. All captured events can be treated as triggers or actions, leading to flexible rule customization.

4 Implementation and evaluation

4.1 Experimental setup

Hardware Since it is time-consuming to deploy a large number of tags and also not practical to tag some objects because of their material (e.g., metal) or usage (e.g., objects used in a microwave oven), we propose to use only some key objects for a specific set of activities by augmenting the object usage with a complementary sensing technique (i.e., accelerometers). Locations of tags include doors, cupboards, refrigerator, dishwasher, etc (see Fig. 2). The detailed configurations of the system like tag density, the shape of the sensor arrangement, sampling rate, have been thoroughly studied in our previous work [29,30].

Data processing The data collection and processing of WITS are fully implemented using the Microsoft .NET framework, and data are analyzed using MATLAB 2014a.



1 Turn On Outdoor Light

2 Toilet Alarm

from 00:00-06:00

from 18:00-24:00

Is On

delay for 1800s

Time

from 00:00 - 06:00

Is Sit

3 Turn on cooling fan of router

[00:22:34] Alarm for microwave has set

Fig. 3 WITS system: *Event Timeline* shows what is going on in the house, e.g., which room is the person in, what activity is performing and which object is used. The sub figure shows the *Rule Composer* interface. For example, for editing a rule like "send an alarm when a person stays in toilet over 30 minutes". A user only needs to drag the person, toilet and clock icons to the Trigger subpanel, the alarm icon to the Action subpanel and then performs some simple adjustments (e.g., adjust the clock slider to set the time period)

The data collection module is developed with deploying based on the developer SDK and open APIs provided by Alien,[2] Arduino,[3] Raspberry Pi[4] and Phidgets.[5]

Graphical UI modules WITS provides a holistic interface for the real-time monitoring, which is a scene-based graphical interface as illustrated in Fig. 3. It works like an extended Harry Potter's Marauder's Map, wherein users can vividly observe what is happening. We also develop a graphical interface for rule customization. The detected events are associated with icons. Therefore, end-users only need to drag and drop the icons to set up their own personal rules (Fig. 3). This Web interface represents a new direction in integrating information from both the physical and virtual worlds, which brings things, locations and activities together over a flexible IoT middleware, helping people be aware of their surrounding environments and thereby making better decisions. The practical experiences gained from this IoT framework provide insights into how IoT can be applied to support critical real-world applications such as assisted living of the elderly.

[2] http://www.alientechnology.com/.

[3] https://www.arduino.cc/.

[4] https://www.raspberrypi.org/.

[5] http://www.phidgets.com/.

Table 2 Activities used in our experiments

Index	Recognizable activities	
	Basic activities	Complex activities
1	Sitting	Taking Medicine
2	Standing	Eating
3	Lying	Watching TV
4	Walking	Reading magazine
5	Arm Movement	Cleaning table
6	Kicking	Vacuuming
7	Crouching	Toileting
8	Falling	Sleeping

4.2 Evaluation

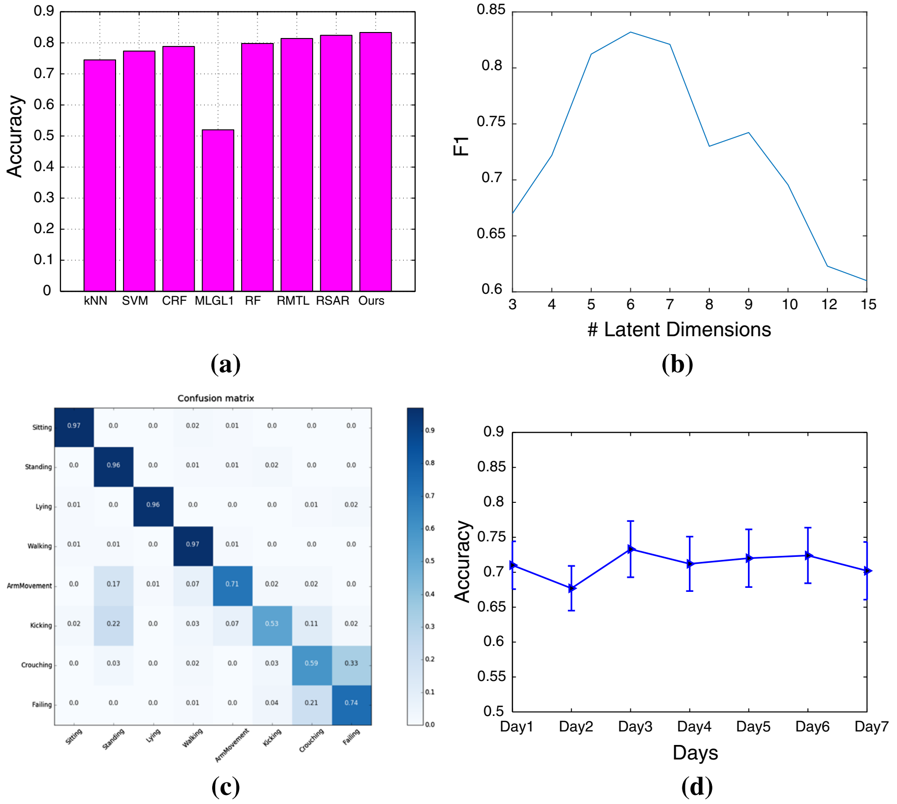
We evaluate the accuracy of recognizing both low-level activities (inferred from signal fluctuations) and high-level activities [inferred from low-level activities, along with object usage and location information (Sect. 3.2)]. The detailed activities evaluated are summarized in Table 2.

We compare our method with a series of existing methods, including the widely-used sensor-based activity recognition methods such as k nearest neighbour (kNN), Support Vector Machine (SVM) with linear kernel, Conditional Random Field (CRF), Multinomial Logistic Regression with ℓ_1 (MLGL1), Random Forest (RF), as well as some closely-related work such as RMTL [4] (which integrates low-rank and group sparsity for multi-task learning) and RSAR [29] (which integrates graph manifold learning and shared structure learning). We obtained the observations from the results as shown in Fig. 4:

- Our method outperforms all the compared methods (Fig. 4a). With larger dimensions of latent space d, the performance of our method degrades, especially when the size becomes bigger than 8 (Fig. 4b).
- The overall performance of the proposed approach is comparatively stable and consistent across all the activities (Fig. 4c), despite the recognition accuracy presents slight ups and downs.
- Combining with symbolic locations and object use events, the average recognition accuracy of high-level activities is generally stable at around 70% (Fig. 4d).

We conclude this section with a brief discussion on system latency, as fast activity detection and notification are critical for many applications, especially aged care applications. For example, we should send an alert to notify the caregivers as quickly as possible for medical assistance when a old person falls. Our system has 4–4.5 s recognition latency, due to three main reasons:

- Our system evaluates subject's postures every 0.5 seconds using the latest 2 seconds of the signal stream. In other words, if the current system time is at the timestamp t, our system will produce the predicted actions in the $[t-2, t-1]$ s, and $[t-1, t]$ s is used to backtrack check if the predicted label complies with predefined rules.



(a) **(b)**

(c) **(d)**

Fig. 4 **a** Comparison with other methods, **b** performance comparison on different dimensions of shared subspace; **c** confusion matrix; **d** high-level activity reasoning accuracy over 7 days

For instance, assume that the label is estimated as: *lying in bed* at $[t-2, t-1]$ interval, if the predicted label in interval $[t-1, t]$ is *walking*, our system will determine the prediction as still *lying*.

- Signal collector is programmed with a timer to poll the signal variations with a predefined order of transmission, which takes around 1 second to complete a new measurement with no workarounds.
- Our system is integrated with a Web-based interface, which sends AJAX requests to services for the latest results and retrieve data fro databases to send back to the Web interface with updated document object model (DOM) elements. Completing such a querying process normally takes 300ms to 500ms.

5 Conclusion and future work

In this paper, we propose the design and development of WITS, an automatic IoT-based in-home monitoring system. We propose a novel hybrid activity recognition framework that integrates multi-task based dictionary learning and rule-based reasoning for better performance. An IoT middleware has been developed to provide an one-stop offer of interoperability, scalability, and accessibility to the system. We have conducted extensive experiments to validate our proposed system and we anticipate our preliminary study and practical experience gained from this system can provide the foundation and inspiration for subsequent research on developing IoT applications for mainstream usage.

In the future, we will enhance our activity recognition approach by incorporating rich contexts for multi-person activity monitoring. As the evaluation in this paper is limited to a small testing scenario (i.e., the first author's home), we plan to perform a larger scale study of the WITS system by working with an aged care center.

References

1. Aharon M, Elad M, Bruckstein A (2006) SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal process 54(11):4311–4322
2. Bhattacharya S, Nurmi P, Hammerla N, Plötz T (2014) Using unlabeled data in a sparse-coding framework for human activity recognition. Pervasive Mob Comput 15:242–262
3. Bulling A, Blanke U, Schiele B (2014) A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput Surv (CSUR) 46(3):33
4. Chen J, Zhou J, Ye J (2011) Integrating low-rank and group-sparse structures for robust multi-task learning. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 42–50
5. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C Appl Rev 42(6):790–808
6. Cook DJ, Schmitter-Edgecombe M, Dawadi P (2015) Analyzing activity behavior and movement in a naturalistic environment using smart home techniques. IEEE J Biomed Health Inform 19(6):1882–1892
7. Dawadi PN, Cook DJ, Schmitter-Edgecombe M (2013) Automated cognitive health assessment using smart home monitoring of complex tasks. IEEE Trans Syst Man Cybern Syst 43(6):1302–1313
8. Edwards WK, Grinter RE (2001) At home with ubiquitous computing: seven challenges. In: Ubicomp 2001: ubiquitous computing. Springer, pp 256–272
9. Evgeniou A, Pontil M (2007) Multi-task feature learning. Adv Neural Inf Process Syst 19:41
10. Guha T, Ward RK (2012) Learning sparse representations for human action recognition. IEEE Trans Pattern Anal Mach Intell 34(8):1576–1588
11. Hodrick RJ, Prescott EC (1997) Postwar US business cycles: an empirical investigation. J Money Credit Bank 1:1–16
12. Intille S et al (2006) Using a live-in laboratory for ubiquitous computing research. In: Proceedings of international conference on pervasive computing (PERVASIVE)
13. Kidd CD, Orr R, Abowd GD, Atkeson CG, Essa IA, MacIntyre B, Mynatt E, Starner TE, Newstetter W (1999) The aware home: a living laboratory for ubiquitous computing research. In: Streitz NA, Siegel J, Hartkopf V, Konomi S (eds) Cooperative buildings. Integrating information, organizations, and architecture. CoBuild 1999. Lecture Notes in Computer Science, vol 1670. Springer, Berlin, pp 191–198
14. Lee H, Battle A, Raina R, Ng AY (2007) Efficient sparse coding algorithms. In: Advances in neural information processing systems. MIT Press, Cambridge, pp 801–808
15. Mairal J, Bach F, Ponce J, Sapiro G (2009) Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 689–696
16. Mennicken S, Vermeulen J, Huang EM (2014) From today's augmented houses to tomorrow's smart homes: new directions for home automation research. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing. ACM, pp 105–115
17. Minor B, Doppa JR, Cook DJ (2015) Data-driven activity prediction: algorithms, evaluation methodology, and applications. In: KDD, pp 805–814
18. Mozer M (2004) Lessons from an adaptive house. Ph.D. thesis, University of Colorado
19. Pintus A, Carboni D, Piras A (2012) Paraimpu: a platform for a social web of things. In: WWW, pp 401–404

20. Rashidi P, Cook DJ (2011) Activity knowledge transfer in smart environments. Pervasive Mob Comput 7(3):331–343
21. Sohn T, Dey A (2003) iCAP: an informal tool for interactive prototyping of context-aware applications. In: CHI, pp 974–975
22. Taylor K et al (2013) Farming the web of things. IEEE Intell Syst 28(6):12–19
23. Tung J, Snyder H, Hoey J, Mihailidis A, Carrillo M, Favela J (2013) Everyday patient-care technologies for Alzheimer's disease. IEEE Pervasive Comput 12(4):80–83
24. Ur B, McManus E, Pak Yong Ho M, Littman ML (2014) Practical trigger-action programming in the smart home. In: CHI, pp 803–812
25. Wang H, Nie F, Huang H (2013) Multi-view clustering and feature learning via structured sparsity. In: ICML, no 3, pp 352–360
26. Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, Balazinska M, Borriello G (2009) Building the internet of things using RFID: the RFID ecosystem experience. IEEE Internet Comput 13(3):48–55
27. Yan Y, Ricci E, Subramanian R, Liu G, Sebe N (2014) Multitask linear discriminant analysis for view invariant action recognition. IEEE Trans Image Process 23(12):5599–5611
28. Yao L, Benatallah B, Wang X, Tran NK, Lu Q. Context as a service: realizing internet of things-aware processes for the independent living of the elderly. In: international conference on service-oriented computing ICSOC
29. Yao L, Nie F, Sheng QZ, Gu T, Li X, Wang S (2016) Learning from less for better: semi-supervised activity recognition via shared structure discovery. In: Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing (UbiComp). ACM, pp 13–24
30. Yao L, Sheng QZ, Li X, Gu T, Tan M, Wang X, Wang S, Ruan W (2018) Compressive representation for device-free activity recognition with passive RFID signal strength. IEEE Trans Mob Comput 17(2):293–306
31. Yao L, Sheng QZ, Ngu AH, Li X, Benattalah B (2017) Unveiling correlations via mining human-thing interactions in the web of things. ACM Trans Intell Syst Technol (TIST) 8(5):62