

Human Activity Recognition in Smart Homes Based on a Difference of Convex Programming Problem

Vahid Ghasemi¹, Ali A. Pouyan¹, and Mohsen Sharifi²

¹School of Computer and IT Engineering, Shahrood University of Technology
Shahrood, Semnan - Iran

[e-mail: vghasemi@shahroodut.ac.ir, apouyan@shahroodut.ac.ir]

²School of Computer Engineering, Iran University of Science and Technology
Tehran - Iran

[e-mail: msharifi@iust.ac.ir]

*Corresponding author: Vahid Ghasemi

*Received March 28, 2016; revised October 14, 2016; accepted November 3, 2016;
published January 31, 2017*

Abstract

Smart homes are the new generation of homes where pervasive computing is employed to make the lives of the residents more convenient. Human activity recognition (HAR) is a fundamental task in these environments. Since critical decisions will be made based on HAR results, accurate recognition of human activities with low uncertainty is of crucial importance. In this paper, a novel HAR method based on a difference of convex programming (DCP) problem is represented, which manages to handle uncertainty. For this purpose, given an input sensor data stream, a primary belief in each activity is calculated for the sensor events. Since the primary beliefs are calculated based on some abstractions, they naturally bear an amount of uncertainty. To mitigate the effect of the uncertainty, a DCP problem is defined and solved to yield secondary beliefs. In this procedure, the uncertainty stemming from a sensor event is alleviated by its neighboring sensor events in the input stream. The final activity inference is based on the secondary beliefs. The proposed method is evaluated using a well-known and publicly available dataset. It is compared to four HAR schemes, which are based on temporal probabilistic graphical models, and a convex optimization-based HAR procedure, as benchmarks. The proposed method outperforms the benchmarks, having an acceptable accuracy of 82.61%, and an average F-measure of 82.3%.

Keywords: Human activity recognition, pervasive computing, smart homes, difference of convex programming problems, uncertainty.

1. Introduction

With the rapid population aging in communities, the smart homes, as ambient assisted living (AAL) environments, have attracted lots of attention. Smart homes can be regarded as the new generations of homes in which *pervasive (or ubiquitous) computing* is incorporated to make the lives of people more comfortable. The general goal of AAL is to hire *ambient intelligence (AmI)* solutions and pervasive computing technologies to assist individuals with specific needs (e.g. elderly people). However, *human activity recognition (HAR)* is of crucial importance in AAL environments; because, the health and behavioral status of the occupants are monitored by means of HAR, and accordingly, essential services are provided proactively.

Occupants' data can be collected by various technologies in smart homes. Wireless sensing technologies provide efficient tools for capturing users' physical data, particularly in indoor environments like smart homes [1-4]. They can provide non-intrusive, privacy friendly, and easy to install solutions, to in-home monitoring. The most common sensors in smart homes include: *environmental sensors* (e.g. luminosity, temperature, etc.), *home usage sensors* (e.g. electric current, RFID tags, switch sensors, etc.), *user location sensors* (e.g. indoor positioning systems), and *medical sensors* (e.g. body temperature, blood pressure, etc.) [2]. The collected data will be further interpreted as the residents' activities, and accordingly, essential aids will be provided.

Interpreting sensors' data streams as activities, that could have been carried out in various scenarios (e.g. sequential, interleaved, concurrent), is still a key research challenge [3, 4]. This has long been investigated in the literature. However, many research works are based on restrictive assumptions that achieve activity inference using local pieces of evidence (e.g. using the data of a sliding window to infer activities) [5-9]. Despite being real-time, they are inaccurate when applied to practical real-world scenarios. Such inaccuracies can have serious unsafe or detrimental effects on the residents. For instance, if the activity "take pill" is recognized mistakenly based on local pieces of evidence, the system will not warn the resident to take his (or her) pills, and an irrecoverable incident may take place. Therefore, an accurate inference is of predominant importance in such environments.

As another issue, managing uncertainty is of crucial importance for HAR. The uncertainties stemming from multiple sources are the main reasons of insufficient performance in lots of HAR mechanisms [10]. Hence, one of the principal concerns of HAR research is to manage or minimize the uncertainties arisen throughout the HAR process [10]. There are two types of uncertainties: *type-A* and *type-B* [10, 11]. The source of *type-A* uncertainty is the random nature of measurements, which is usually modeled by probability distributions. In contrast, the source of *type-B* uncertainty is generally, the lack of knowledge in the measurements and is usually modeled experimentally. In the context of HAR, the random nature of human activities is considered as the source of *type-A* uncertainty, while sensor technologies and incomplete HAR techniques can be the sources of *type-B* uncertainty [10]. For example, in [12], *type-B* uncertainty is considered to stem from sensor hardware errors and is indicated by fixed real values as discounting factors in the Dempster-Shafer theory for sensor data fusion. Whilst many HAR methods, such as temporal probabilistic graphical models, take *type-A* uncertainty into account, they don't consider *type-B* uncertainty. Henceforth, we use the term uncertainty to refer to *type-B* uncertainty unless *type-A* is explicitly pointed out.

The main contribution of this paper is to represent a novel solution for HAR, which can achieve two goals: 1- performing a global inference, and 2- managing uncertainty. The proposed method is based on minimizing a novel objective function in the form of a *difference*

of *convex programming (DCP)* problem. The *DCP* problem is solved by taking all pieces of evidence from the input stream into account. Therefore, the inference is no more local. Furthermore, the proposed method is capable of handling uncertainty by defining appropriate constraints for the *DCP* problem. To achieve these goals, given a sensor data stream, a *primary belief* in each activity is calculated for the sensor events. A primary belief in an activity is defined as the weighted posterior probability of that activity. The primary beliefs compose primary belief vectors (PBVs) for the sensor events. Since the PBVs are calculated based on some abstractions, they bear an amount of noise that would induce uncertainty. We consider the uncertainty of the belief vectors as their entropy. To mitigate the effect of uncertainty, *secondary belief vectors (SBVs)* are calculated out of PBVs by solving the *DCP* problem. The *DCP* problem takes the PBVs as input, and yields the SBVs as its solution by satisfying the following constraints (to be elaborated in section 5.2):

- 1- The secondary belief vectors are not far from the confident primary belief vectors.
- 2- The number of segments of activities in the input sensor data stream is limited.
- 3- The uncertainty of the secondary belief vectors should be minimized.

The *DCP* problem is solved by the standard *convex-concave procedure (CCP)* algorithm [13]. Afterwards, the activities are inferred based upon the values of the secondary beliefs. A well-known and publicly available dataset is used to evaluate the proposed method. Two experiments are carried out for this purpose. In the first experiment, the proposed method is compared to four widely adopted HAR schemes, which are based on temporal probabilistic graphical models. In the second experiment, a convex optimization-based sequence classification scheme is implemented in which the concept of uncertainty is not considered. The results show that the proposed method has an acceptable performance in comparison to the benchmarks.

The remainder of the paper is organized as follows. In the following subsection, we discuss the motivations and the concept of uncertainty; in section 2, the related works are reviewed; section 3 represents some notations and the format of sensor data streams; section 4 describes the activity recognition problem and its characteristics; the proposed method is elaborated in section 5 and its subsections; the experiments and simulation results are taken in section 6; finally, section 7 concludes the paper.

1.1. Motivations and the Concept of Uncertainty

Let s denote a sensor event which is recorded in the smart environment. This event can be the triggering of a motion sensor, switch sensor, etc. Naturally, the observation of s will induce an amount of *belief* in each possible activity. We call this, the *primary belief* of s in the activities. Given s , we define the primary belief in an activity as its posterior probability divided by the *flatness* of the activity's posterior distribution. In other words, the primary beliefs can be thought as the weighted posterior probabilities of the activities. With n possible activities, the primary beliefs of s yield an n -ary vector, which is called the *primary belief vector (PBV)* of s . The i 'th element of this vector denotes the primary belief of s in the i 'th activity.

Having observed a sensor event s , one paradigm for inference is to select the activity in which s has the highest primary belief. But there are situations where there exist more than one activity in which s has significant primary beliefs. The small differences between the significant primary beliefs can be due to some sources of *uncertainty* such as modeling inefficiencies, sensor hardware errors, misplacement of sensors, etc. [10]. In such cases the PBV

of s is said to be *uncertain*. For instance, from the dataset of [1], with 8 possible activities, it turns out that the PBV of the sensor event $M070N$ is $b_{M070N} \approx (0.02, 0.18, 0.17, 0.11, 0.11, 0.07, 0.17, 0.18)$. However, if sensor event $M070N$ is observed, it can not be certainly inferred that which one of activities 2, 3, 7, or 8 is taking place; because their corresponding beliefs are very similar. Therefore, the belief vector b_{M070N} is uncertain. In these situations the *entropy* of the belief vector will take a high value. Therefore, we consider entropy as a measure of uncertainty for a belief vector.

To cope with the problem of uncertain belief vectors, we propose to take into account the effect of the adjacent sensor events. That is, the high uncertainty of a PBV should be frustrated by the belief vectors of the neighboring sensor events in the input stream which have low uncertainties. The resulting beliefs are called *secondary beliefs*. Similar to PBVs, the secondary beliefs compose secondary belief vectors (SBVs) for each sensor event. The final inference is based on the SBVs. In section 5.2, we first define the uncertainty of a belief vector as its entropy formally, and then, represent a solution to calculate the SBVs based on a DCP problem.

2. Related Works

HAR methods can be classified into three main categories based on their activity modeling schemes: *data-driven*, *knowledge-driven* and *hybrid methods* [3]. Data-driven HAR is based on learning activity models from pre-existent data, using data mining and machine learning techniques. In knowledge-driven schemes, activity models are directly acquired by exploiting rich prior knowledge from the domain of interest, using knowledge engineering and management technologies. In hybrid methods, a combination of the previous techniques is used. All of these approaches have pros and cons. For instance, data-driven approaches can model the probabilistic nature and temporal information of activities, while they suffer from the problems of scalability and reusability. Knowledge-driven methods have the advantages of being semantically clear and highly reusable, while they suffer from static activity models and possible incomplete or inaccurate knowledge. Emerging hybrid HAR schemes, such as the approaches of [14, 15], aim at taking the advantages of the previous two cases. However, these methods and also the knowledge driven approaches are generally weak at managing uncertainty [3, 15].

Based on the types of the sensors, HAR methods can also be divided into vision-based and sensor-based techniques. Sensor-based techniques can in turn be further divided into two categories of *wearable sensor-based*, and *dense sensing-based* methods [3, 4]. In dense sensing-based methods, sensors are attached to the environmental objects, while in the wearable sensor-based schemes, sensors are positioned on the human body. For the purpose of this paper, *data-driven dense sensing-based* HAR is considered.

Some HAR schemes try to infer the ongoing activities based on local pieces of evidence (e.g. the evidence from a limited sliding window). However, they usually suffer from not being general enough to cover some common human activity scenarios in realistic environments. In [5], multiple sliding window-based HAR schemes (*time-based* and *sensor-based* windowing) are implemented. In these schemes, only the features of the current window along with the probability of activities from the previous window are considered to infer an activity for a sensor event. In [6], a dynamic sensor stream segmentation method is presented. In this method, an arriving sensor event is added to the past segments, if it has a time correlation with their first and last sensor events, and value correlation with their last sensor event. Otherwise, a new segment will be initiated by the new sensor event. Finally,

some typical classifiers (e.g. naïve Bayes classifier, hidden Markov models, etc.) are hired to classify the segmented sequences. In [7], a method for sensor event segmentation and classification is introduced. In this method, a sensor event is added to a previously initiated segment based on the results of an activity inference engine, and the Jaro-Winkler distance between the sequence of the current segment and the training set. Otherwise, a new segment is initiated by the incoming sensor event. In [8] (and similarly in [9]), a *support vector machine (SVM)* is used to classify the features that have been computed for fixed-length time frames, into daily activities. All the mentioned methods suffer from common shortcomings that stem from local inference. For example, even though the subsequent pieces of evidence would be available, the dependency of the current sensor events with them is not considered.

In contrast to the previous methods, there are other HAR schemes, which try to achieve a global inference by considering the shreds of evidence from all over the input stream. Among these methods, *temporal probabilistic graphical models* provide formal mechanisms for learning activity models and inference. These models use a template graph structure to represent the dependencies among observed random variables (sensor readings) and unknown variables (activity labels). Given a sequence of observations, the template graph is unrolled and the most likely sequence of unknown variables is estimated, using a formal inference method (e.g. *Viterbi algorithm*) [16, 17]. Various types of graphical models exist based on the characteristics of the template graph and the unrolling rules. *Dynamic Bayesian networks (DBNs)* [18, 19], *hidden Markov models (HMMs)* [1, 20, 21], *hidden semi-Markov models (HSMMs)* [22], *coupled HMMs (CHMMs)* [23, 24], *conditional random fields (CRFs)* [20, 25, 26], *skip-chain CRFs (SCCRFs)* [20, 27], *factorial CRFs (FCRFs)* [24], *latent dynamic CRFs (LDCRFs)* [28], and some other variants of such models, are instances that have been used for HAR.

There are also some HAR approaches that achieve a global inference based on frequent sequential pattern mining. Generally, these methods try to find frequent patterns throughout the sensor data streams, for further interpretation as activities by their subsequent processes. The approaches of [29-31] are instances of such works. In [29], the concept of emerging patterns is introduced for HAR, which is based on the item-set mining methods. In [30], a greedy search for graph-based pattern discovery is performed in an unsupervised manner to find patterns that best compress the input dataset. Then the patterns are clustered into more abstract sets. Afterwards, a sensor-based windowing scheme is used to classify the cluster members when they appear in the input stream. The idea of compression is somehow similarly used in the approach of [31], to cluster the unlabeled sequential sensor data into different activity patterns, based on the LZW (Lempel–Ziv–Welch) compression algorithm. Some other instances include the works of [32, 33].

In spite of the huge body of works for HAR, most of the state-of-the-art HAR technologies suffer from neglecting uncertainty management as their main concern [10]. The HAR methods discussed above are instances of such works. Among data-driven HAR schemes, the approach of [34] manages the uncertainty of observations at the decision level. This is achieved through the verification of assigned labels, while utilizing the environment interactive sensors for the recognition of general activities. However, this approach is designed for the classification of pre-segmented activities. There are also some knowledge-driven methods that deal with uncertainty. The works of [12, 35-37] are instances of such methods.

In this paper, a novel data-driven HAR approach is represented, in which the pieces of evidence from all over the input stream are considered for inference. Therefore, the inference is no more local. Furthermore, the proposed method is capable of mitigating the effect of uncertainty in inference. These goals have been achieved by a novel formulation of HAR as a

DCP problem.

3. Sensor Data Stream and Some Notations

The data of users' activities are gathered by various types of sensors in the environment. Each sensor is assigned a unique identifier (ID). A *sensor event* is considered as the combination of the triggered sensor ID, followed by its value. The domain of sensor values can be either continuous (e.g. temperature or humidity sensors), or discrete (e.g. switch sensors or RFID tags). In the case of continuous sensors, a partitioning method [38] can be used for discretization. Therefore, we can assume that all the sensors are discrete-valued. The domain of sensor i is denoted by D_i . **Table 1** shows an example of the sensors' IDs, types, domains, and their possible events. Note that sensor "T03" is continuous and its domain is divided into three intervals, numbered as 1, 2, and 3.

Table 1. An example of sensor IDs, types, domains, and sensor events from the dataset of [1].

ID	Sensor Type	Sensor Domain	Sensor Events
M15	Motion (Discrete)	$D_{M01} = \{"ON", "OFF"\}$	M01_ON, M01_OFF
T03	Temperature(Continuous)	$D_{T01} = \{1,2,3\}$	T03_1, T03_2, T03_3
D07	Switch Sensor (Discrete)	$D_{D07} = \{"CLOSE", "OPEN"\}$	D07_CLOSE, D07_OPEN

A *sensor data stream (or sensor event stream)* is the sequence of sensor events, ordered non-decreasingly by their occurrence time. A sensor data stream of length T is denoted by $x_{1:T} = x_1, \dots, x_t, \dots, x_T$, where x_t is an individual sensor event. **Fig. 1** shows a sensor data stream, extracted from the dataset of [1]. In a labeled sensor data stream, sensor events are labeled by the IDs of activities too.

In the following sections, some common notations are used as follows. The number of activities and the set of all possible sensor events are denoted by n and S , respectively. $x_{1:T} = x_1, \dots, x_t, \dots, x_T$ stands for a sensor data stream of length T , $x_i \in x_{1:T}$ is the i 'th sensor event, and $l_i \in \{1, \dots, n\}$ is the ID of the inferred activity for x_i . The set of *activities of daily life (ADLs)* is denoted by $\mathcal{A} = \{a_1, \dots, a_n\}$, and $a_j \in \mathcal{A}$ is the label of the j 'th activity. Also, the vectors and matrixes are depicted by bold letters.

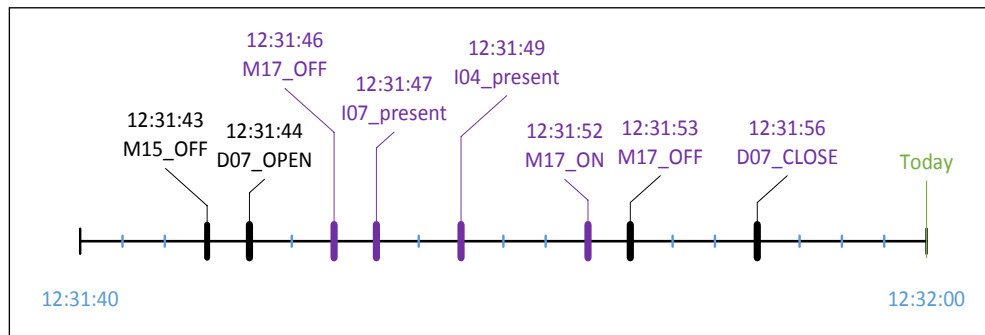


Fig. 1. An example of a sensor data stream from 2008-07-29 12:31:40 to 2008-07-29 12:32:00 in the dataset of [1]. Vertical bars are sensor events, and horizontal axis is the timeline.

4. Activity Recognition Problem

The main focus of this work is the task of supervised data-driven human activity recognition (HAR). This can be defined as follows:

Definition 1. Let $\mathcal{A} = \{a_1, \dots, a_n\}$ denote a known set of n ADLs, $\mathcal{D} = \{x_{1:T_i}^i | x_{1:T_i}^i = x_1^i, \dots, x_{T_i}^i, T_i \in \mathbb{N}, i = 1, \dots, L\}$ denote a training dataset of L labeled sensor data streams (i and T_i are the index, and length of the i 'th sensor data stream in \mathcal{D} , respectively), and $x_{1:T} = x_1, \dots, x_T$ stand for a novel observed sensor data stream of length T . The objective of HAR is to train a model based on \mathcal{D} , that can infer a subset of \mathcal{A} , as the label of each sensor event $x_t \in x_{1:T}, t = 1, \dots, T$.

Basically, the activities can be carried out in three main forms: *sequential*, *interleaved*, and *concurrent* [27]. In sequential style, an activity starts after the previous one is finished. For example, one starts “washing the dishes”, finishes it, and then goes to “watch TV” (no overlap between sensor events happens). In interleaved fashion, two or more activities proceed simultaneously but each sensor event pertains to a single activity (has a single label), and no overlapping sensor event exists. For example, one starts “washing the dishes”, stops it in the middle, goes to “answer the phone”, and again returns to “washing the dishes”. In concurrent mode, two or more activities progress simultaneously and there are overlapping sensor events too. For instance, one enters the living room and turns on the light, then starts “listening to music” and “writing a postcard” simultaneously. In this scenario, the two activities are accomplished simultaneously, and the sensor event of “turning on the light” pertains to both activities (it will have two labels). Fig. 2 shows these scenarios for two activities. The recognition of sequential and interleaved activities (i.e. single labeled sensor events) is the main focus of this paper. The proposed method is capable of recognizing such patterns effectively.

5. The Proposed Method

The proposed method comprises train and test phases. An overall view of these phases is represented in the flowchart of Fig. 3, and also, the pseudo codes of Table 2 and Table 3. The flowchart along with the corresponding pseudo codes, are illustrated in the following. Afterwards, the main components of the proposed method, i.e. the calculation of PBVs, and the formulation of the DCP problem, are illustrated in detail in the following two subsections.

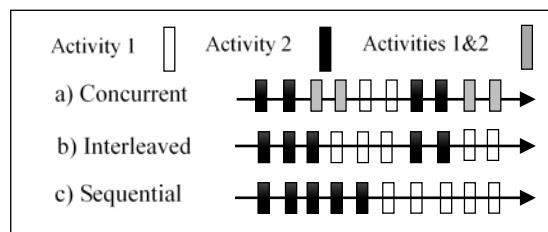


Fig. 2. Three types of activity executions. The vertical bars are sensor events, and the horizontal axis is the timeline.

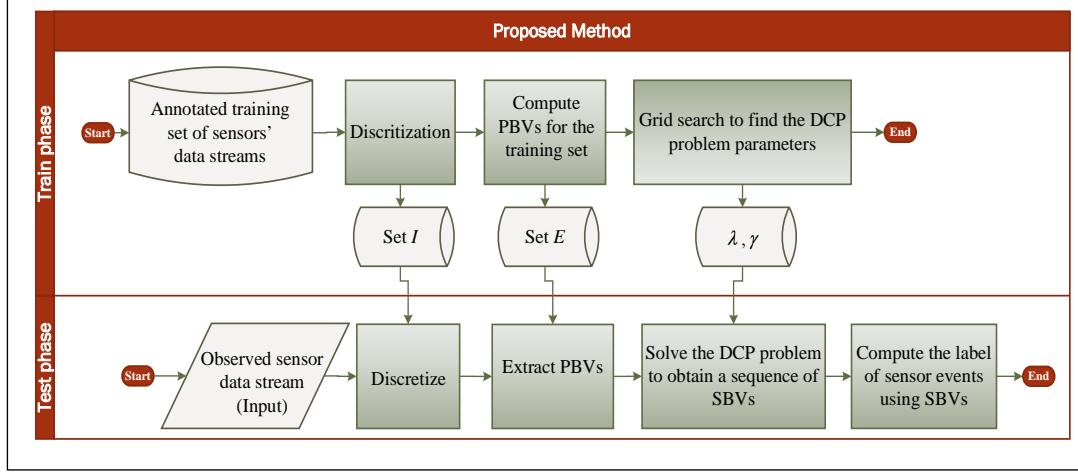


Fig. 3. The flowchart of the proposed method. (Only permanent data, processes, and input data are depicted for simplicity).

Train phase: The pseudo code of this phase is taken in [Table 2](#). In this phase, at first, the domain of every continuous sensor is discretized into a number of intervals, using the entropy-based approach of [\[38\]](#) (function *Discretization*). Each interval is assigned a unique ID. Every time that a continuous sensor is triggered, it takes the ID of the discretized interval that encompasses its sensed value. The set of discretized intervals of all sensors is shown by I for simplicity. After this preprocessing step, an n -ary PBV is calculated for each possible sensor event (function *Compute_PBVs*). The computations of the PBVs are elaborated in section 5.1. The set of PBVs is shown by E . It should be noted that the calculation of E is feasible, since the number of installed sensors and their maximum number of discrete values are limited. Afterwards, the parameters of the DCP problem which was described in section 1, are calculated (function *Grid_Search*). The DCP problem takes a sequence of PBVs as input, and calculates a sequence of SBVs, which can be used to classify the input stream (to be described in section 5.2). To calculate the regularization parameters of the DCP problem, namely λ and γ , a grid search is performed to obtain the values that maximize the classification performance on the training set. Totally, the outputs of the training phase can be summarized as sets I , E , and parameters λ , and γ .

Test phase: The pseudo code of this phase is depicted in [Table 3](#). In this phase, firstly the novel input stream of sensor events $x_{1:T} = x_1, \dots, x_T$, is discretized using set I , as described in the training phase (function *Discretize*). Then, a sequence of PBVs, namely $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_T]^T$, is extracted for $x_{1:T}$ using set E (function *Extract_PBV*). Afterwards, the DCP problem is solved for the input sequence \mathbf{B} , by means of the CCP algorithm [\[13\]](#) (function *Solve_With_CCP*), and a sequence of SBVs, namely $\mathbf{M} = [\mathbf{M}_1, \dots, \mathbf{M}_T]^T$, is obtained (this process is described in section 5.2). Finally, the label of each sensor event $x_i \in x_{1:T}$ is inferred by equation $l_i = \text{argmax}_j(M_{i,j})$, where $M_{i,j}$ is the j 'th element of \mathbf{M}_i .

Table 2. The pseudo codes of the train phase.

1.	Input
2.	Annotated training set: $\mathcal{D} = \{x_{1:T_i}^i i = 1, \dots, L\}$
3.	Formulated DCP problem (equation (7)): P
4.	Maximum and minimum of λ , and γ : $Max_\lambda, Min_\lambda, Max_\gamma, Min_\gamma$
5.	Output
6.	Set of possible PBVs: E
7.	Set of discrete intervals: I
8.	Regularization parameters: λ, γ
9.	Variables
10.	Discretized training set: \mathcal{D}'
11.	Body
12.	$[I, \mathcal{D}'] = Discretization(\mathcal{D})$
13.	$E = Compute_PBVs(\mathcal{D}')$
14.	$[\lambda, \gamma] = Grid_Search(P, E, \mathcal{D}', Max_\lambda, Min_\lambda, Max_\gamma, Min_\gamma)$
15.	End

Table 3. The pseudo codes of the test phase.

1.	Input
2.	Input sensor data stream: $x_{1:T} = x_1, \dots, x_T$
3.	Set of discretized Intervals: I
4.	Set of PBVs: E
5.	Formulated DCP problem (equation (7)): P
6.	DCP regularization parameters: λ, γ
7.	Output
8.	Sequence of inferred activity IDs: $l_{1:T} = l_1, \dots, l_T$
9.	Variables
10.	Matrix of PBVs: $B \in \mathbb{R}^{T \times n}$ // i 'th row (i.e. B_i) is the i 'th PBV
11.	Matrix of SBVs: $M \in \mathbb{R}^{T \times n}$ // i 'th row (i.e. M_i) is the i 'th SBV
12.	Body
13.	For $i = 1$ to T
14.	If x_i is continuous
15.	$x_i = Discretize(x_i, I)$
16.	End
17.	$B_i = Extract_PBV(x_i, E)$
18.	End
19.	$M = Solve_With_CCP(P, B, \lambda, \gamma)$
20.	For $i = 1$ to T
21.	$l_i = argmax_{j \in \{1, \dots, n\}}(M_{i,j})$
22.	End
23.	End

5.1. Primary Belief Vectors (PBVs)

Having n activities, and a set of total sensor events S , an n -ary feature vector $\mathbf{b}_i = (b_{i,1}, \dots, b_{i,j}, \dots, b_{i,n})$ is calculated for each sensor event $x_i \in S$ ($i = 1, \dots, |S|$), where $b_{i,j} \geq 0$ is the *belief* of sensor event x_i in activity a_j , and $\sum_{j=1}^n b_{i,j} = 1$. \mathbf{b}_i is the *PBV* of sensor event x_i . We define the primary beliefs, as functions of *posterior probability* of the underlying activity and its *flatness*, as follows.

Intuitively the belief $b_{i,j}$, is considered directly proportional to the posterior probability of activity a_j , given sensor event x_i . This is shown as

$$b_{i,j} \propto p(A = a_j | X = x_i) \quad (1)$$

, where A and X are the variables that take the IDs of an activity and a sensor event, respectively.

In a smart environment, the values of $p(A = a_j | X)$ are expected to be high for a limited number of sensor events that are related to activity a_j , and low for the others; because, each activity often deals with a limited number of repetitive appliances, and in each occurrence, triggers approximately the same subset of sensor events. Therefore, $p(A = a_j | X)$ is expected not to be highly *flat*. Hence, we make the following assumption.

Assumption 1. Having the above-mentioned notations, it is expected that $f(X) = p(A = a_j | X)$, should not be highly flat in a smart environment.

Similar to the flatness of a probability distribution function (PDF), which can be calculated by its entropy [39], we calculate the flatness of $p(A = a_j | X)$ after normalizing it, as

$$R_j = - \sum_{k=1}^{|S|} (\bar{p}(A = a_j | X = x_k) \times \log(\bar{p}(A = a_j | X = x_k))) \quad (2)$$

, where $R_j \geq 0$ is the flatness of $p(A = a_j | X)$, and $\bar{p}(A = a_j | X = x_k) = \frac{p(A=a_j|X=x_k)}{\sum_{k'=1}^{|S|} p(A=a_j|X=x_{k'})}$.

The more flat $p(A = a_j | X)$ is, the higher value R_j will take. R_j takes the maximum value, if the values of $p(A = a_j | X)$ are the same for all possible sensor readings (*fully flat*). On the other extent, it will be minimum (zero) if the value of $p(A = a_j | X)$ equals to one for a single sensor reading, and zero for the others (*fully peaked*). Regarding *Assumption 1*, the belief $b_{i,j}$ is defined inversely proportional to the flatness R_j , to give more weight to the posterior probabilities calculated by a non-flat function $f(X) = p(A = a_j | X)$. This is shown as

$$b_{i,j} \propto \frac{1}{R_j + \mathcal{E}} \quad (3)$$

, where $\mathcal{E} \in \mathbb{R}^+$ is a small value ($\mathcal{E} \rightarrow 0$) to avoid division by zero. Plugging (1) and (3) into a single equation yields the final definition of the belief $b_{i,j}$, as

$$b_{i,j} = \frac{p(A = a_j | X = x_i)}{R_j + \mathcal{E}} \cdot Z_i \quad (4)$$

, where Z_i is a normalization factor that makes $\sum_{k=1}^n b_{i,k} = 1$, and is calculated as

$$Z_i = \left(\sum_{k=1}^n \frac{p(A=a_k|X=x_i)}{R_k+\mathcal{E}} \right)^{-1}. \quad (5)$$

Having the above formulations, the PBV is defined formally as follows:

Definition 2. The *primary belief vector (PBV)* of sensor event $x_i \in S$ is a vector $\mathbf{b}_i \in \mathbb{R}^n$, where $b_{i,j} \geq 0$ is the belief of sensor event x_i in activity a_j , such that $\sum_{k=1}^n b_{i,k} = 1$, and $b_{i,j}$ is calculated through (4).

The posterior probabilities of $p(A = a_k | X = x_i)$ can be learned from the training set using the standard statistical methods of *maximum likelihood estimation (MLE)* or *maximum a posteriori probability (MAP)* estimation. We use MLE in its standard form to calculate this quantity from the annotated training set.

5.2. Activity Recognition via a Difference of Convex Programming Problem

In this section, a method for mitigating the effect of uncertainty and HAR is proposed. This scheme is based on solving a DCP problem.

5.2.1. The Definition of Uncertainty

The uncertainty of a random variable can be considered as its *entropy* in information theory. Similarly, we define the uncertainty, associated with a belief vector as follows:

Definition 3. Let $\mathbf{x} \in \mathbb{R}^n$ denote a belief vector. The uncertainty of \mathbf{x} , namely $u(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$, is considered as its *entropy*, and is defined as

$$u(\mathbf{x}) = - \sum_{j=1}^n x_j \log(x_j). \quad (6)$$

It should be noted that $u(\mathbf{x}) \geq 0$ will be maximized if the elements of \mathbf{x} are all equal, and minimized (zero) if a single element of \mathbf{x} equals to 1, and thus, the others equal to zero. Function $u(\mathbf{x})$, and its maximum and minimum points for $\mathbf{x} \in \mathbb{R}^3$, are depicted in Fig. 4. As it can be seen, $u(\mathbf{x})$ is maximized for $\mathbf{x} \approx (0.33, 0.33, 0.33)$, and minimized for $\mathbf{x} = (1, 0, 0)$, $\mathbf{x} = (0, 1, 0)$, and $\mathbf{x} = (0, 0, 1)$.

It should be noted that in (6), the term $-x_j \log(x_j)$ is *concave* because its second derivative with respect to x_j is negative. Therefore, $u(\mathbf{x})$, which is the summation of concave terms, is also concave.

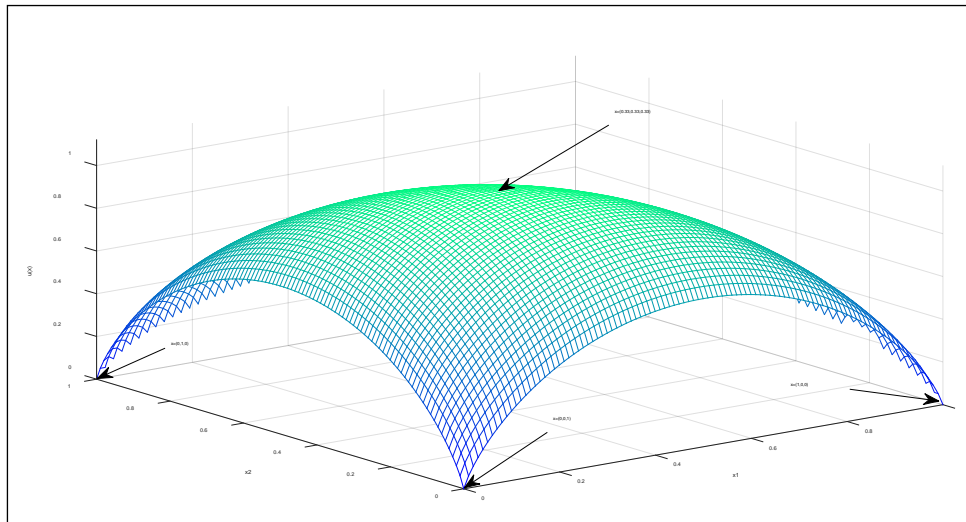


Fig. 4. The uncertainty function $u(\mathbf{x})$ for belief vector $\mathbf{x} = (x_1, x_2, x_3)$, such that $x_1 + x_2 + x_3 = 1$ and $x_i \geq 0, i \in \{1, 2, 3\}$. The axis x_3 is not shown for simplicity, because $x_3 = 1 - x_1 - x_2$.

5.2.2. Human Activity Recognition by Secondary Beliefs

To resolve the problem of uncertain PBVs, a *secondary belief vector (SBV)*, namely $\mathbf{M}_i \in \mathbb{R}^n$ ($i = 1, \dots, T$), is calculated for each sensor event $x_i \in x_{1:T}$. The SBVs are required to

satisfy the triple constraints of **Table 4** simultaneously. The SBVs have the same properties as PBVs, i.e. $M_{i,j} \geq 0$, and $\sum_{k=1}^n M_{i,k} = 1$.

Satisfying the constraints of **Table 4** results in belief vectors which their uncertainties are trammled by the neighboring sensor events. The first constraint of **Table 4** makes SBVs to be close to their associated PBVs, which have low uncertainties. The second constraint enforces the neighboring SBVs to close to each other. As a result, the simultaneous aggregation of constraints 1, and 2, i.e. $\sum_{i=1}^{T-1} \|\mathbf{M}_i - \mathbf{M}_{i+1}\|_2 < K$, makes the highly uncertain belief vectors approach to the neighboring belief vectors, which have low uncertainties; therefore, their uncertainties are reduced implicitly. Also, the third constraint lowers the uncertainty of SBVs explicitly. It makes confident decisions, based on \mathbf{M}_i s, possible in the future.

Table 4. The constraints for calculating secondary belief vectors $\mathbf{M}_i, i = 1, \dots, T$.

1. The less the *uncertainty* of \mathbf{b}_i is, the more close \mathbf{M}_i to \mathbf{b}_i should be.
2. There are a limited number of segments of activities in $x_{1:T}$. Accordingly, the number of unequal consecutive SBVs should be limited. That is:

$$\exists K \in \mathbb{R}^+, \quad \sum_{i=1}^{T-1} \|\mathbf{M}_i - \mathbf{M}_{i+1}\|_2 < K$$

3. The *uncertainty* of $\mathbf{M}_i, i = 1, \dots, T$ should be minimized.

It should be noted that the second constraint of **Table 4** holds, because SBVs of sequence $\mathbf{M}_j, \dots, \mathbf{M}_{j+t}$ will be the same, if sensor events x_j, \dots, x_{j+t} belong to the same segment of an activity (as discussed in [40, 41] too); and naturally, since the number of segments of activities is limited, the number of unequal consecutive SBVs should be limited too.

The constraints of **Table 4** can be termed mathematically, and then minimized together, to obtain the SBVs. To do this, we propose to solve the non-convex optimization problem of (7).

$$\mathbf{M} = \underset{\hat{\mathbf{M}} \in \mathbb{R}^{T \times n}}{\operatorname{argmin}} \left(\sum_{i=1}^T \frac{1}{u(\mathbf{b}_i)} \|\mathbf{b}_i - \hat{\mathbf{M}}_i\|_2 + \lambda \sum_{i=1}^{T-1} \|\hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i\|_2 + \gamma \sum_{i=1}^T u(\hat{\mathbf{M}}_i) \right) \quad (7)$$

$$\text{s.t.} \quad \begin{cases} \sum_{j=1}^n \hat{M}_{i,j} = 1, & \forall i = 1, \dots, T \\ \hat{M}_{i,j} \geq 0, & \forall i = 1, \dots, T, \text{ and } \forall j = 1, \dots, n \end{cases}$$

, where $\hat{\mathbf{M}}_i \in \mathbb{R}^n$ and $\mathbf{M}_i \in \mathbb{R}^n$ are the i 'th rows of matrixes $\hat{\mathbf{M}} \in \mathbb{R}^{T \times n}$, and $\mathbf{M} \in \mathbb{R}^{T \times n}$, respectively. Also, $\lambda > 0$, and $\gamma > 0$ are the regularization parameters (they can be thought as the Lagrange coefficients) that will be adjusted using the training set.

In (7), the first term implies the first constraint of **Table 4**. Because, if the uncertainty of \mathbf{b}_i is decreased, then the coefficient $\frac{1}{u(\mathbf{b}_i)}$ will increase. Therefore, to minimize (7), vector \mathbf{M}_i will be determined close to \mathbf{b}_i , to moderate the value of the first term.

The second term of (7) imposes the second constraint of **Table 4**. In fact, this term is the sum of ℓ_2 -norms, and therefore induces row-sparse solutions [40, 41] (i.e. many successive \mathbf{M}_i s will be the same). Minimizing this term limits the number of segments of activities as addressed in the second condition. It should be noted that there is a tradeoff between the first and second terms of (7). The first term is minimized with T segments (with $\hat{\mathbf{M}}_i = \mathbf{b}_i, i = 1, \dots, T$), while the second term is minimized with a single segment (with $\hat{\mathbf{M}}_{i+1} = \hat{\mathbf{M}}_i, i = 1, \dots, T$). This tradeoff can be controlled via λ . If λ is increased, the number of segments will decrease to moderate the value of the second term, and if it is decreased the number of

segments will increase, to lower the value of the first term.

The third term of (7) results in the third constraint of **Table 4** directly, because it is the summation of uncertainties of SBVs. The parameter $\gamma > 0$ controls the total amount of allowed uncertainty associated with the SBVs, such that the increase of γ results in the decrease of uncertainty of the SBVs. For instance, $\gamma = \infty$, enforces $u(\mathbf{M}_i) = 0, i = 1, \dots, T$.

Since \mathbf{M}_i s are belief vectors, the constraints of (7) guarantee that the elements of \mathbf{M}_i are greater than or equal to zero, and sum up to 1. As it will be discussed, (7) shows a DCP problem, and can be solved by the standard CCP algorithm.

Having calculated the secondary belief vectors $\mathbf{M}_i \in \mathbb{R}^n, i = 1, \dots, T$, the label of the activity for sensor event $x_i \in X_{1:T}$, namely $l_i \in \{1, \dots, n\}$, is calculated as

$$l_i = \operatorname{argmax}_j (M_{i,j}). \quad (8)$$

Similar to our proposed procedure, [40] proposes to solve the following convex optimization problem to segment a sequential data stream when no outlier is considered.

$$\mathbf{M} = \operatorname{argmin}_{\hat{\mathbf{M}} \in \mathbb{R}^{T \times n}} \left(\sum_{i=1}^T \|\mathbf{b}_i - \hat{\mathbf{M}}_i\|^2 + \lambda \sum_{i=1}^{T-1} w_i \|\hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i\|_2 \right). \quad (9)$$

Here, $w_i > 0$ is a weight that shows how likely two adjacent sensor events would be in the same segment, and the other notations and constraints are the same as (7). w_i s are determined based on the prior knowledge of the adjacent data points. In the absence of prior knowledge, $w_i = 1$ is used [41].

The formulation of (9) can be interpreted similar to (7), with the difference that the concept of uncertainty is omitted in (9), and $w_i = 1$ for $i = 1, \dots, T - 1$. Therefore, if (7) is substituted by (9) with the same constraints, then the proposed method becomes a HAR scheme which is based on *convex optimization* and does not take uncertainty into account. We call this method *convex optimization-based activity recognition (CPAR)* throughout the paper. Our simulations show that the proposed method outperforms CPAR.

5.2.3. Difference of Convex Programming Problems

A *difference of convex programming (DCP)* problem is an optimization problem that can have the form

$$\begin{aligned} \mathbf{x} &= \operatorname{argmin}_{\hat{\mathbf{x}}} (f_0(\hat{\mathbf{x}}) - g_0(\hat{\mathbf{x}})), \\ \text{s.t.} \quad & f_i(\hat{\mathbf{x}}) - g_i(\hat{\mathbf{x}}) \leq 0, \quad i = 1, \dots, h \end{aligned} \quad (10)$$

, where $\hat{\mathbf{x}} \in \mathbb{R}^n$ is the optimization variable, $\mathbf{x} \in \mathbb{R}^n$ is the solution, and $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ for $i = 1, \dots, h$ are convex. This problem can be solved by the *convex-concave procedure (CCP)* algorithm effectively [13]. **Table 5** summarizes the CCP algorithm [13].

According to (10), (7) can be written as a DCP problem. That is, because the third term in (7) is concave, and its first two terms are convex. Also, the left sides of the constraints of (7) are affine expressions. Thus, (7) denotes a DCP problem and can be solved using the standard CCP algorithm. We set $\hat{\mathbf{M}}_0 = [\mathbf{b}_i]_{T \times n}, i = 1, \dots, T$ as the initial feasible points in the CCP algorithm to solve the DCP problem of (7).

6. Simulations and Results

This section represents the results of the empirical studies. Two experiments are carried out to assess the performance of the proposed method. In the first experiment, the proposed

Table 5. The standard CCP algorithm.

1.	Input
2.	Initial feasible point: $\hat{\mathbf{x}}_0$
3.	Return threshold: δ
4.	Output
5.	The solution of the DCP problem of (10) : \mathbf{x}
6.	Variables
7.	Step counter: k
8.	Body
9.	$k = 0$
10.	Do
11.	$\forall i = 0, \dots, h, \hat{g}_i(\mathbf{x}) = g_i(\hat{\mathbf{x}}_k) + \nabla g_i(\hat{\mathbf{x}}_k)^T (\mathbf{x} - \hat{\mathbf{x}}_k)$
12.	$\hat{\mathbf{x}}_{k+1} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} (f_0(\hat{\mathbf{x}}) - \hat{g}_0(\hat{\mathbf{x}})), \text{ s.t. } \forall i = 1, \dots, h, f_i(\hat{\mathbf{x}}) - \hat{g}_i(\hat{\mathbf{x}}) \leq 0$
13.	$k = k + 1$
14.	While $(f_0(\hat{\mathbf{x}}_{k+1}) - \hat{g}_0(\hat{\mathbf{x}}_{k+1})) - (f_0(\hat{\mathbf{x}}_k) - \hat{g}_0(\hat{\mathbf{x}}_k)) > \delta$
15.	$\mathbf{x} = \hat{\mathbf{x}}_k$
16.	End

method is compared with four benchmark schemes, which are based on *hidden Markov models (HMMs)*, *conditional random fields (CRFs)* and *skip chain conditional random fields (SCCRFs)*. In the second experiment, the recent approach of [40], which is based on (9) and does not take uncertainty into account, is incorporated in the proposed method. The results reveal the role of handling uncertainty.

We use precision, recall, F-measure, and accuracy to evaluate the performance of the algorithms as follows.

$$\text{Precision(Specificity)} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{Recall(Sensetivity)} = \frac{TP}{TP + FN} \quad (12)$$

$$F_{\text{measure}} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

, where TP, TN, FP, and FN denote *true positive*, *true negative*, *false positive*, and *false negative*, respectively, for the labeling of a sensor event.

6.1. The Dataset

A publicly available dataset, named *Kyoto*¹ [1], from the Center for Advanced Studies in Adaptive Systems (CASAS) smart house project, is used for the simulations. The dataset represents 20 participants performing eight ADLs in an apartment. This dataset consists of two parts. The first and the second parts of the dataset include pre-segmented and interleaved sensor data streams of activities, respectively. Each sensor data stream of the second part pertains to a single person. The activities, their assigned IDs, and the number of their sensor events are listed in Table 6. To compose the first part of the dataset, the participants have

¹ Available online from <http://casas.wsu.edu/datasets>.

Table 6. The activities and the number of their sensor events.

ID	Activity label	Number of sensor events	ID	Activity label	Number of sensor events
1	Fill medication dispenser	624	5	Prepare birthday card	854
2	Watch DVD	1100	6	Prepare soup	1251
3	Water plants	1090	7	Cleaning	1765
4	Answer the phone	412	8	Choose outfit	564

performed the tasks individually, and the corresponding data have been stored separately for each activity. For the second part of the dataset, the participants have been asked to perform the entire set of activities again in any order, interleaved or parallel style if needed (these data have also been stored separately). This dataset well suits our study, because the sensor events of activities are highly interleaved.

We use the first part of the dataset only as a training set for discretization process and to compute PBVs for sensor events as described in section 5.2. The performance of the proposed method is measured for labeling the interleaved sensor data streams of the second part. In the following sections, the IDs of activities from Table 6 are used instead of their labels.

The data of users' activities have been collected by various types of sensors. They include 72 discrete sensors with binary domains and 6 continuous sensors. We discretize each continuous sensor domain into three intervals. Therefore, there will be 162 possible sensor events totally i.e. $|S| = 162$. The map of the smart home is depicted in Fig. 5.

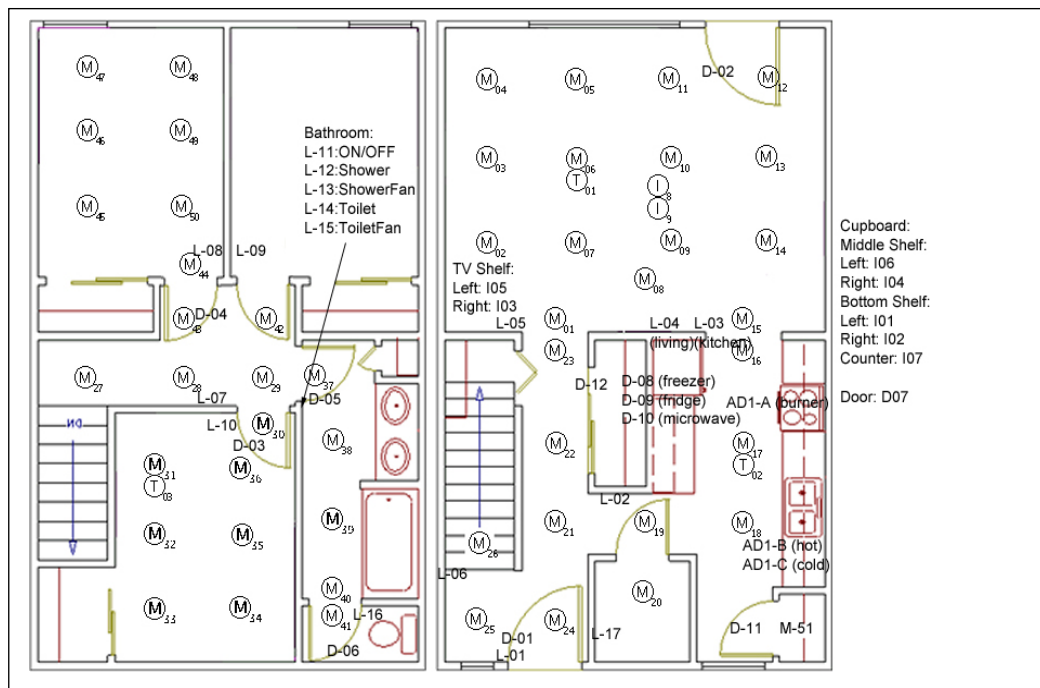


Fig. 5. The smart home design (M= motion sensors, T= temperature sensors, I=item sensors, AD= water and burner sensors, D=cabinet sensors) [1].

6.2. Experimental Settings

In the following experiments, a leave-one-out-cross-validation (LOOCV) strategy is conducted to evaluate every HAR method. That is, the dataset is divided into $k = 20$ folds, with each fold containing a single participant's data stream. In each round of LOOCV, one distinct fold is selected for testing, and the remaining $k - 1$ folds are retained for training. This process is repeated k times, and in each round, a distinct fold is selected as the test set. The results are aggregated to yield the final performance measure values. **Table 7** sums up the common parameters in the following experiments. The computations of λ and γ , are illustrated in the following section.

Table 7. Parameter settings.

Parameter	Value	Description
k	20	The number of folds in the cross-validations.
λ	14	Lagrange coefficient in the DCP problem (i.e. equation (7)).
γ	3	Lagrange coefficient in the DCP problem (i.e. equation (7)).
δ	0.01	The return threshold of the CCP algorithm.
$\hat{\mathbf{M}}_0$	$[\mathbf{b}_i]_{T \times n}$	The initial feasible point of the CCP algorithm.

6.3. Implementation Setup for the Proposed Method

Our goal is to assess the performance of the proposed method for labeling the interleaved sensor data streams of the second part of the dataset. We conduct an LOOCV strategy (as described in section 6.2) to assess the performance of the proposed method. However, the parameters γ , and λ of (7) should be calculated beforehand. To do this, in each round of LOOCV, the values of γ , and λ are obtained by a grid search scheme such that they maximize the classification accuracy of the training set. Having calculated these values, (7) is solved to segment and classify the sensor events of the test set as described in section 5.2.

Fig. 6 shows the color map of the grid search strategy for the first round of LOOCV to determine γ and λ . The ranges $[1,10]$, and $[1,20]$ are swept for γ , and λ , respectively. No

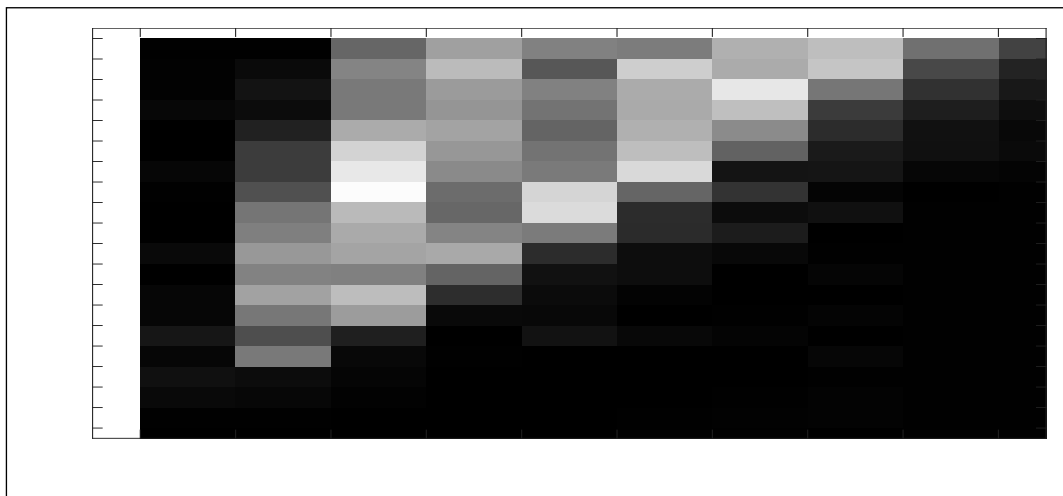


Fig. 6. The grid search results, to determine λ , and γ . The brighter areas show higher values of accuracy. $\lambda = 14$, and $\gamma = 3$ yield the maximum accuracy.

better accuracy achieved for some further extents of values. As it can be seen, $\gamma = 3$, and $\lambda = 14$ (and similarly $\gamma = 3$, and $\lambda = 13$) result in the maximum recognition accuracy. The calculations of γ and λ are carried out the same in the other rounds too. However, it is turned out that, $\gamma = 3$, and $\lambda = 14$ values yield acceptable results (maximum or near to maximum accuracies) in the other rounds too.

6.4. Experiment 1

In this experiment, the proposed method is compared with four widely adopted HAR schemes as benchmarks. At first, a comparison with an HMM-based HAR procedure is performed. Then, the proposed method is compared with three approaches which are based on CRFs and SCCRFs. The LOOCV strategy of section 6.2 is conducted to evaluate all of the approaches.

To implement the HMM, similar to some other works, such as [1, 4, 20], the activities and sensors are treated as the hidden and observable states, respectively. A first-order HMM with fully interconnected hidden states is implemented for this purpose. The well-known *Viterbi* algorithm is used for inference.

The results of this experiment are taken in Table 8. The higher average results are depicted in yellow. As it can be seen, the precisions, recalls, and F-measures of the proposed method are higher than, or comparable to HMM for most of the activities. Totally, the proposed method has a higher average of precision and recall than HMM. Also, the 82.3% average F-measure of the proposed method outperforms the 73.5% average F-measure of HMM. The variances of the performance measures show that the proposed method, with much lower variances than HMM, has a more stable performance. Also, as it is shown in Fig. 7, the total accuracy of the proposed method amounts to 82.61%, which outperforms the 76.83% accuracy of HMM.

Table 8. The performance measures of HMM, and the proposed method (PM).

Activity ID	Precision		Recall		F-Measure	
	HMM	PM	HMM	PM	HMM	PM
1	0.810	0.937	0.827	0.808	0.818	0.867
2	0.729	0.926	0.860	0.750	0.790	0.829
3	0.809	0.742	0.862	0.759	0.834	0.750
4	0.181	0.696	0.244	0.682	0.208	0.689
5	0.841	0.805	0.850	0.904	0.845	0.852
6	0.891	0.874	0.830	0.884	0.860	0.879
7	0.720	0.781	0.645	0.865	0.681	0.821
8	0.842	0.912	0.853	0.862	0.847	0.886
Average	0.728	0.835	0.747	0.814	0.735	0.823
Variance	0.052	0.008	0.046	0.006	0.049	0.005

In the following, the proposed method is compared with three discriminative modeling formalisms for HAR. They are based on *linear-chain conditional random fields* (LCCRF), and *skip chain conditional random fields* (SCCRFs). Since SCCRFs can capture long range dependencies, they are claimed to be suited for inferring interleaved activities.,.

To employ LCCRFs, two approaches were implemented. In the first approach, a separate model with two hidden states (*happen* (Y), *not happen* (N)) is trained for each activity. A novel stream is inputted to the LCCRFs of activities, and the activity with the highest probability is inferred for each sensor event. This approach is called *SMSA* (*single model for a single*

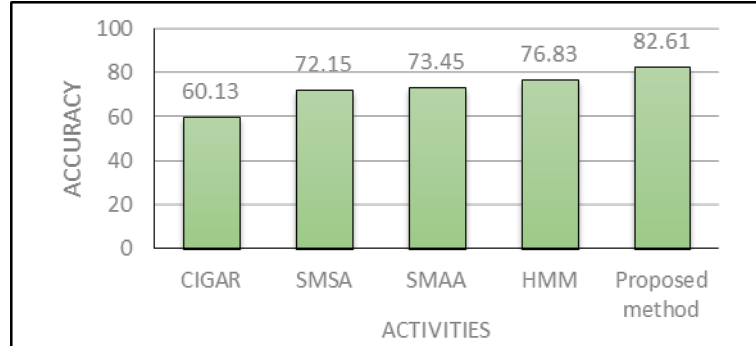


Fig. 7. The total accuracies of HMM, CIGAR, SMSA, SMAA, and the proposed method.

activity). In the second approach, similar to some other works, such as [20, 42], a single CRF model is trained for all activities, such that the activities and sensor events are considered as hidden states and observable states, respectively. Having an observed sensor data stream, it is inputted to the CRF to yield the most probable sequence of activities. This approach is named *SMAA* (single model for all activities). In both approaches, the *loopy belief propagation* (LBP) mechanism is used for inference.

To employ SCCRFs, the main concern is how to add the skip chains to the corresponding unrolled graph. A heuristic for this task is proposed in [27] (named *CIGAR*). That is, for each activity, named a , a separate SCCRF with two hidden states (*happen* (Y), *not happen* (N)) is trained. A skip chain between two unrolled hidden states i , and j ($j > i$) is added if it is observed that $p(s_j | s_i, a) > \theta$, where s_i and s_j are the i 'th and j 'th sensor events in the input stream respectively, $\theta > 0$ is a threshold, and $p(s_j | s_i, a)$ is the probability of s_j occurring after s_i , for activity a . Given a novel sensor data stream, it is inputted to the trained SCCRFs of the activities, and the activity with the maximum probability is inferred as the label of the sensor events. In our experiments, $\theta = 0.95$ yielded the best results. More details on the framework of *CIGAR* can be found in [27]. In this approach the LBP mechanism is used for inference too.

The precisions, recalls, and F-measures, resulted from the above-mentioned methods are depicted in **Table 9**. Higher average results are depicted in yellow. As it is shown, the F-measures resulted from the proposed method are higher than or comparable to those of CIGAR, SMSA, and SMAA for most of the activities. Totally, the proposed method with an

Table 9. The performance measures of CIGAR, SMSA, SMAA, and the proposed method (PM).

Activity ID	Precision				Recall				F-Measure			
	CIGAR	SMSA	SMAA	PM	CIGAR	SMSA	SMAA	PM	CIGAR	SMSA	SMAA	PM
1	0.654	0.117	0.923	0.937	0.027	0.973	0.575	0.808	0.052	0.209	0.709	0.867
2	0.870	0.681	0.752	0.926	0.576	0.914	0.816	0.750	0.693	0.780	0.782	0.829
3	0.824	0.739	0.947	0.742	0.395	0.817	0.343	0.759	0.534	0.776	0.504	0.750
4	0.298	0.117	0.526	0.696	0.087	0.676	0.097	0.682	0.135	0.199	0.164	0.689
5	0.803	0.831	0.855	0.805	0.754	0.873	0.855	0.904	0.778	0.852	0.855	0.852
6	0.589	0.847	0.805	0.874	0.810	0.626	0.846	0.884	0.682	0.720	0.825	0.879
7	0.477	0.958	0.595	0.781	0.955	0.624	0.948	0.865	0.637	0.756	0.731	0.821
8	0.694	0.693	0.789	0.912	0.257	0.905	0.876	0.862	0.375	0.785	0.830	0.886
Average	0.650	0.624	0.774	0.835	0.485	0.801	0.671	0.814	0.486	0.636	0.673	0.823
Variance	0.037	0.105	0.021	0.008	0.119	0.019	0.093	0.006	0.073	0.072	0.055	0.005

average F-measure of 82.16%, outperforms CIGAR, SMSA, and SMAA with average F-measures of 48.59%, 63.46%, and 67.50%, respectively. The 81.4% average recall and 83.5% average precision resulted from the proposed method outperform the other methods too. Also, as it is shown in Fig. 7, the 82.61% accuracy of the proposed method is better than the 73.45%, 72.15%, and 60.13% accuracy of SMAA, SMSA, and CIGAR, respectively. Moreover, the proposed method has a more stable performance than others, because of its low variances of the performance measures.

It should be noted that the lower performance of CIGAR is due to the possibility of adding inefficient skip chains. This problem is inevitable in this framework.

It is worth mentioning that despite the higher average results of the proposed method, it is not always the case for each individual activity; such as activity number 3, i.e. “water plants”, in Table 8 and Table 9. The confusion matrix of the proposed method is shown in Table 10. It testifies that the lower performance of recognizing activity 3, is mainly because of its misclassification as activity 7, i.e. “cleaning”, and vice versa. As it is highlighted in Table 10, 218 sensor events of activity 3, and 198 sensor events of activity 7 have been misclassified as activities 7 and 3, respectively. The reason is that both activities generate many common sensor events. In both activities 7 and 3, the users highly interact with the appliances in the kitchen closet and kitchen environment, according to the scripts of activities in the dataset. Therefore, both activities trigger many common sensors. This makes the discrimination between activities 3 and 7 very difficult. Now, consider HMM results in Table 8. Although the F-measure of HMM is higher than the proposed method for activity 3, but its F-measure for activity 7 is much lower than the proposed method. This is also the case for SMSA and SMAA algorithms as shown in Table 9.

Table 10. The confusion matrix of the proposed method.

		Predicted Classes							
		1	2	3	4	5	6	7	8
Correct Classes	1	504	13	5	17	33	48	4	0
	2	6	825	3	27	127	52	53	7
	3	2	10	827	6	0	24	218	3
	4	11	5	13	281	0	24	72	6
	5	0	7	1	37	772	9	6	22
	6	15	7	48	21	0	1106	46	8
	7	0	22	198	6	8	3	1527	1
	8	0	2	19	9	19	0	29	486

One reason for generating high ratios of common sensor events for different activities, can be the characteristics of the underlying sensor network, such as the placements of the sensor nodes [43]. If the sensor network has not been deployed efficiently, it may not capture discriminative data patterns, which are effective for distinguishing between activities. Efficient deployment of sensor networks in smart homes can be the topic of further research [43].

6.5. Experiment 2

In this experiment, the DCP problem of the proposed method (i.e. Equation (7)) is substituted by the convex optimization problem of [40], i.e. Equation (9), as described in section 5.2. This procedure does not consider uncertainty. The other steps are the same as the

proposed method. This method is named *CPAR* (*convex programming activity recognition*). The results of this experiment can reveal the role of managing uncertainty in HAR.

Since we don't consider any prior knowledge of whether two adjacent sensor events belong to the same segment or not, the coefficients w_i ($i = 1, \dots, T$) in (9) are considered equal to 1. Similar to the proposed method, the LOOCV scheme along with a grid search strategy is performed to calculate λ in each round. **Fig. 8** shows the grid search results in the first round of LOOCV. A range of $[0.1, 10]$ is swept to find the best value of λ . $\lambda = 1.5$ yields the maximum accuracy in this round. The process is the same for the other rounds too.

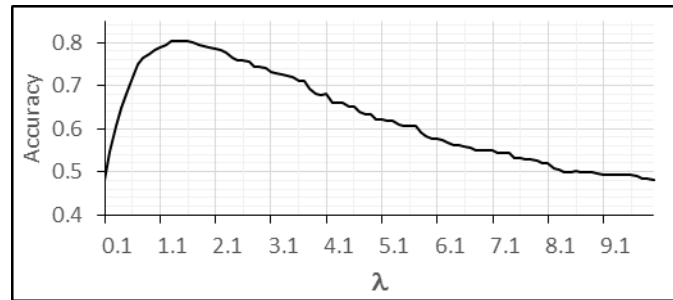


Fig. 8. The grid search results, to find the best value of λ . $\lambda = 1.5$ maximizes the accuracy.

The performance measures resulted from CPAR, and the proposed method are summarized in **Table 11**. Higher average results are depicted in yellow. As it can be seen, the proposed method has a higher F-measure in comparison to CPAR for all activities. Totally, CPAR with average precision, recall, and F-measure of 82.3%, 76.9%, and 78.2%, respectively, has a lower performance in comparison to the proposed method. Moreover, the accuracy of CPAR amounted to 79.97%, which is lower than the 82.61% accuracy of the proposed method. Therefore, our method outperforms CPAR, too. As it is depicted in **Table 11**, the variances of performance measures in both methods are very low.

The results of this experiment show the key role of managing uncertainty in our approach. That is, if uncertainty minimization is omitted from the proposed method (i.e. using equation (9) instead of (7)), then the HAR performance will be deteriorated.

Table 11. The performance measures of CPAR and the proposed method (PM).

Activity ID	Precision		Recall		F-Measure	
	CPAR	PM	CPAR	PM	CPAR	PM
1	0.919	0.937	0.747	0.808	0.824	0.867
2	0.896	0.926	0.675	0.750	0.770	0.829
3	0.890	0.742	0.648	0.759	0.750	0.750
4	0.797	0.696	0.476	0.682	0.596	0.689
5	0.753	0.805	0.936	0.904	0.834	0.852
6	0.797	0.874	0.890	0.884	0.841	0.879
7	0.728	0.781	0.916	0.865	0.811	0.821
8	0.802	0.912	0.863	0.862	0.832	0.886
Average	0.823	0.835	0.769	0.814	0.782	0.823
Variance	0.005	0.008	0.026	0.006	0.007	0.005

7. Conclusion

This paper introduced a novel mechanism to perform HAR in smart homes. It is based on minimizing an objective function in the form of a DCP problem. The proposed method performs a global inference by solving the DCP problem on the entire input sensor data stream. Furthermore, the effect of uncertainty is mitigated by considering appropriate constraints in order to formulate the DCP problem. To tackle these issues, given an input sensor data stream, a primary belief vector (PBV) is calculated for each sensor event based on the posterior PDFs of activities, and their flatness. Then, the uncertainties of the PBVs are alleviated by solving the DCP problem. The solution of the DCP problem yields a sequence of secondary belief vectors (SBVs). The final activity inference is based on the resulting SBVs.

Two experiments were carried out to assess the performance of the proposed method on a well-known and publicly available dataset from the CASAS smart home project. In the first experiment, four HAR approaches based on HMMs, LCCRFs, and SCCRFs were implemented as benchmarks. The proposed method outperformed these schemes with an acceptable accuracy of 82.61% and an average F-measure of 82.3%. In the second experiment, a similar approach, based on convex optimization, was implemented. This approach did not take uncertainty into account. In this case, the performance of the HAR method deteriorated to 79.97% accuracy, and an average F-measure of 78.2%. This experiment testified that the handling of uncertainty in the proposed HAR method had been a promising paradigm.

References

- [1] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Tracking activities in complex settings using smart environment technologies," *International journal of biosciences, psychiatry, and technology (IJBSPT)*, vol. 1, no. 1, pp. 25-28, January, 2009. [Article \(CrossRef Link\)](#).
- [2] K. Rasch, "Smart assistants for smart homes," *Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden*, 2013. [Article \(CrossRef Link\)](#).
- [3] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 790-808, November, 2012. [Article \(CrossRef Link\)](#).
- [4] A. Benmansour, A. Bouchachia, and M. Feham, "Multi-Occupant Activity Recognition in Pervasive Smart Home Environments," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 34, February, 2016. [Article \(CrossRef Link\)](#).
- [5] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138-154, February, 2014. [Article \(CrossRef Link\)](#).
- [6] J. Wan, M. J. O'Grady, and G. M. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 287-301, February, 2015. [Article \(CrossRef Link\)](#).
- [7] H. Cho, J. An, I. Hong, and Y. Lee, "Automatic Sensor Data Stream Segmentation for Real-time Activity Prediction in Smart Spaces," in *Proc. of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, pp. 13-18, May 18'th, 2015. [Article \(CrossRef Link\)](#).
- [8] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no 2, pp. 274-283, March, 2010. [Article \(CrossRef Link\)](#).
- [9] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "A unified framework for activity recognition-based behavior analysis and action prediction in smart homes," *Sensors*, vol. 13, no. 2, pp. 2682-2699, February, 2013. [Article \(CrossRef Link\)](#).

- [10] E. Kim, S. Helal, C. Nugent, and M. Beattie, "Analyzing Activity Recognition Uncertainties in Smart Home Environments," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no 4, pp. 1-28, August, 2015. [Article \(CrossRef Link\)](#).
- [11] B. I. d. P. e. Mesures, C. é. internationale, and O. i. d. normalisation, *Guide to the Expression of Uncertainty in Measurement*, International Organization for Standardization, 1995. [Article \(CrossRef Link\)](#)
- [12] J. Liao, Y. Bi, and C. Nugent, "Using the Dempster–Shafer theory of evidence with a revised lattice structure for activity recognition," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 1, pp. 74-82, January, 2011. [Article \(CrossRef Link\)](#).
- [13] T. Lipp and S. Boyd, "Variations and extension of the convex–concave procedure," *Optimization and Engineering*, vol. 17, no. 2, pp. 263–287, June, 2016. [Article\(CrossRef Link\)](#).
- [14] L. Chen, C. Nugent, and G. Okeyo, "An ontology-based hybrid approach to activity modeling for smart homes," *Human-Machine Systems, IEEE Transactions on*, vol. 44, no. 1, pp. 92-105, February, 2014. [Article \(CrossRef Link\)](#).
- [15] G. Azkune, A. Almeida, D. López-de-Ipiña, and L. Chen, "Extending knowledge-driven activity models through data-driven learning techniques," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3115-3128, April, 2015. [Article \(CrossRef Link\)](#).
- [16] J. Bilmes, "Dynamic graphical models," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 29-42, November, 2010. [Article \(CrossRef Link\)](#).
- [17] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009. [Article \(CrossRef Link\)](#).
- [18] T. v. Kasteren and B. Krose, "Bayesian activity recognition in residence for elders," in *Proc. of 3rd IET International Conference on Intelligent Environments*, pp. 209-212, September 24-25, 2007. [Article \(CrossRef Link\)](#).
- [19] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors," in *Proc of the 3rd international conference on Pervasive Computing*, pp. 62-79, May 8-13 , 2005. [Article \(CrossRef Link\)](#).
- [20] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48-53, January, 2010. [Article \(CrossRef Link\)](#).
- [21] T. van Kasteren, G. Englebienne, and B. Kröse, "Human activity recognition from wireless sensor network data: Benchmark and software," *Activity Recognition in Pervasive Intelligent Environments*, vol. 4, pp. 165-186, May, 2011. [Article \(CrossRef Link\)](#).
- [22] T. van Kasteren, G. Englebienne, and B. J. Kröse, "Activity recognition using semi-Markov models on real-world smart home datasets," *Journal of ambient intelligence and smart environments*, vol. 2, no. 3, pp. 311-325, January, 2010. [Article \(CrossRef Link\)](#).
- [23] P. Natarajan and R. Nevatia, "Coupled hidden semi Markov models for activity recognition," in *Proc. of IEEE Workshop on Motion and Video Computing*, pp. 10-10, February 23-24, 2007. [Article \(CrossRef Link\)](#).
- [24] L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu, "Recognizing multi-user activities using wearable sensors in a smart home," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 287-298, June, 2011. [Article\(CrossRef Link\)](#).
- [25] D. Cook, "Learning Setting-Generalized Activity Models for Smart Spaces," *IEEE Intelligent Systems*, vol. 27, no. 1, pp. 32-38, February, 2012. [Article\(CrossRef Link\)](#).
- [26] V. Ghasemi and A. A. Pouyan, "Activity recognition in smart homes using absolute temporal information in dynamic graphical models," in *Proc of the 10'th Asian Control Conference*, pp. 1-6, May, 2015. [Article\(CrossRef Link\)](#).
- [27] D. H. Hu and Q. Yang, "Cigar: Concurrent and interleaving goal and activity recognition," in *Proc of the 23'rd AAAI Conference on Artificial intelligence*, pp. 1363-1368, July 13-17, 2008. [Article\(CrossRef Link\)](#)
- [28] Y. Tong and R. Chen, "Latent-Dynamic conditional random fields for recognizing activities in smart homes," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 1, pp. 39-55, 2014. [Article\(CrossRef Link\)](#).

- [29] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition," in *Proc. of IEEE Int. Conference on Pervasive Computing and Communications*, pp. 1-9, March 9-13, 2009. [Article\(CrossRef Link\)](#).
- [30] D. J. Cook, N. C. Krishnan, and P. Rashidi, "Activity discovery and activity recognition: A new partnership," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 820-828, June, 2013. [Article\(CrossRef Link\)](#).
- [31] S.-L. Chua, S. Marsland, and H. W. Guesgen, "Unsupervised Learning of Human Behaviours," in *Proc of the 25th AAAI Conference on Artificial Intelligence*, August 7-11, 2011. [Article\(CrossRef Link\)](#).
- [32] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering Activities to Recognize and Track in a Smart Environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no 4, pp. 527-539, April, 2011. [Article\(CrossRef Link\)](#).
- [33] B. Chikhaoui, S. Wang, and H. Pigot, "A frequent pattern mining approach for ADLs recognition in smart environments," in *Proc of IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pp. 248-255, March 22-25, 2011. [Article\(CrossRef Link\)](#).
- [34] L. G. Fahad, A. Khan, and M. Rajarajan, "Activity recognition in smart homes with self-verification of assignments," *Neurocomputing*, vol. 149, pp. 1286-1298, February, 2015. [Article\(CrossRef Link\)](#).
- [35] F. Sebbak, A. Chibani, Y. Amirat, A. Mokhtari, and F. Benhammedi, "An evidential fusion approach for activity recognition in ambient intelligence environments," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1235-1245, November, 2013. [Article\(CrossRef Link\)](#).
- [36] S. Mckeever, J. Ye, L. Coyle, C. Bleakley, and S. Dobson, "Activity recognition using temporal evidence theory," *Journal of Ambient Inteligence and Smart Environments*, vol. 2, no. 3, pp. 253-269, January, 2010. [Article\(CrossRef Link\)](#).
- [37] E. Javadi, B. Moshiri, and H. S. Yazdi, "Activity Recognition In Smart Home Using Weighted Dempster-Shafer Theory," *International Journal of Smart Homes*, vol. 7, no. 6, pp. 23-34, 2013. [Article\(CrossRef Link\)](#).
- [38] U. M. Fayyad and K. B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," in *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022-1027, 1993. [Article\(CrossRef Link\)](#).
- [39] C. R. Rao, *Linear statistical inference and its applications: 2nd Eddition*, John Wiley & Sons Publications, September, 2009. [Article\(CrossRef Link\)](#)
- [40] I. Katz and K. Crammer, "Outlier-Robust Convex Segmentation," *arXiv preprint arXiv:1411.4503*, 2014. [Article\(CrossRef Link\)](#).
- [41] C. Zhu, H. Xu, C. Leng, and S. Yan, "Convex Optimization Procedure for Clustering: Theoretical Revisit," in *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1619-1627, December 8-13, 2014. [Article\(CrossRef Link\)](#).
- [42] E. Nazerfard, B. Das, L. B. Holder, and D. J. Cook, "Conditional random fields for activity recognition in smart environments," in *Proc. of the 1st ACM International Health Informatics Symposium*, pp. 282-286, November 11-12, 2010. [Article\(CrossRef Link\)](#).
- [43] B. L. Thomas, A. S. Crandall, and D. J. Cook, "A Genetic Algorithm approach to motion sensor placement in smart environments," *Journal of Reliable Intelligent Environments*, vol. 2, no. 1, pp. 3-16, 2016. [Article\(CrossRef Link\)](#).



Ali A. Pouyan received his Ph.D. in Computer Engineering from Swinburne University of Technology, Australia in 1997. He is currently with the School of Computer and IT Engineering, and the director of the laboratory of AI and Distributed Systems at Shahrood University of Technology, Iran. His research interests include topics of mobile agents, discrete robotic control, Petri net modeling, multi-agent systems, image processing and DNA computing. Dr. Pouyan is a member of IEEE, IFAC, EASST, IAENG and IACIS.



Mohsen Sharifi received his B.Sc., M.Sc., and Ph.D. in Computer Science from Victoria University of Manchester, United Kingdom, in 1982, 1987, and 1991, respectively. He is currently a Professor at the School of Computer Engineering, Iran University of Science and Technology, Iran. His main research interests include distributed computing and systems, wireless sensor (actor) networks, virtualization technology, computer security, and Web engineering.



Vahid Ghasemi received his M.Sc. in Computer Science from Amirkabir University of Technology, Iran, in 2010, and his B.Sc. in Computer Engineering from Razi University, Iran, in 2006. He is currently a Ph.D. candidate in Computer Engineering and a member of the laboratory of AI and Distributed Systems at the School of Computer and IT Engineering, Shahrood University of Technology, Iran. His research interests include ambient intelligence (AmI), pervasive computing, sequential pattern mining, convex optimization, and wireless sensor networks.