Dirichlet Process Gaussian Mixture Model for Activity Discovery in Smart Homes with Ambient Sensors

Thuong Nguyen
The Australian e-Health Research Centre, CSIRO
Herston, Queensland 4029, Australia
thuong.nguyen@csiro.au

Qing Zhang
The Australian e-Health Research Centre, CSIRO
Herston, Queensland 4029, Australia
qing.zhang@csiro.au

Duc V. Le
Pervasive Systems Group, University of Twente
Enschede, The Netherlands
v.d.le@utwente.nl

Mohan Karunanithi
The Australian e-Health Research Centre, CSIRO
Herston, Queensland 4029, Australia
mohan.karunanithi@csiro.au

ABSTRACT

Existing approaches to activity recognition in smart homes mostly rely on supervised learning from well-annotated sensor data, acquired in a controled lab environment. However obtaining such labeled data in real home scenarios could be prohibitive due to either the privacy concerns of using cameras, or the low adherence of self reports done by home residents. Unsupervised learning, on the other hand, aims at discovering activities through applying fixed complexity models, yet assuming apriori knowledge of the number of activities. Again this is also a non-practical approach because the number of activities could vary drastically, even within a home. In this paper, we propose a novel practical unsupervised Bayesian nonparametric model to discover activities in smart homes, without prior assumption on the number of activities. Instead, our model can automatically infer such number only from sensor readings, thus it can be easily applied to any new home. We test our method on a public dataset and a dataset collected in our project. On the CASAS dataset, which has activity labels, our approach can achieve the performance close to the best of GMM and outperforms K-means. On our smart home dataset, the discovered activities are highly correlated with the typical daily routine of the resident. Such experimental results demonstrate the efficiency of our method for activity discovery in smart homes.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; **Ubiquitous and mobile computing design and evaluation methods**;

KEYWORDS

Activity discovery; smart home; ambient sensing; unsupervised learning; Bayesian nonparametric; Dirichlet process

MobiQuitous 2017, Melbourne, VIC, Australia

978-1-4503-5368-7/17/11…$15.00
DOI: 10.1145/3144457.3144464

1 INTRODUCTION

Population ageing is a big challenge for the healthcare system. In 2014, the older population – persons of 65 and older – accounts for about 15% of the whole Australian population, and is projected to 21% by 2054 [1]. In the US, this age group is predicted to be 83.7 million in 2050, almost double its population in 2012 [22]. This trend is also observed worldwide – the percentage of the older group in 2050 will be double today's figure [13]. Moreover, while the number of people aged 85 and over is small compared to the entire population, it is increasing rapidly compared to younger groups [1]. Although a high prevalence of people in this age group need assistance in their daily life, they prefer to continue living in their home instead of moving to an aged care facility [25]. Thus there is a need for smart systems to support older people maintaining their independent living in their own home.

Developing smart sensing environment or smart home to support independent living has attracted much research interest in recent decades [28]. Early studies in this area employed camera to recognize activities [11]. However, using camera incurs concerns of privacy violation. Thus recent studies have shifted to using ambient sensors, such as motion, reed, and temperature [4, 34]. Moreover, recent advances in sensor technology, which make the sensor smaller, easier to install, and its battery lasts longer, has provided great potential for building smart homes. Following this trend, we also use such ambient sensors in our smart homes.

A crucial step in developing smart home is the activity recognition or discovery step, which infers the activities performed by home residents from the data recorded by sensors deployed around the home. Generally, activity recognition or discovery can be categorized into supervised learning (classification) and unsupervised learning (clustering).

Classification techniques have been widely used to recognize activities from smart home sensor data [32, 34, 35]. However, they typically require a large amount of labeled data to train the classifiers. Obtaining such labeled data would be an infeasible task, especially when the home residents are seniors who may suffer serious mental and physical illnesses. Moreover, the activity set may change over time – a new data observation may not belong to pre-defined activities. These challenges make supervised learning methods unsuitable for activity recognition in smart homes.

There have been some efforts in discovering activities in smart homes using clustering methods [16, 26, 27]. However, their feature

extraction methods are naturally suitable for event-based sensors and may not be directly applicable for analogue sensors[1] such as power or humidity sensors, which can provide meaningful information about home residents' activities. Moreover, the features used in [16, 26, 27] are encoded with sensor deployment information such as location or attached object. Thus, they may be difficult to be applied in new homes with different sensor deployment.

To address these challenges, we develop a smart home system using ambient sensors and an unsupervised approach to discover activities of home residents. Our approach does not rely on any labeled data and can be applied to both event-based sensors and analogue sensors. The missing data of analogue sensors are imputed by interpolation. Furthermore, the features are extracted automatically without using sensor deployment information. Therefore, it can be easily applied to new homes with new sensor deployment. From the raw sensor data, we divide the time dimension into short intervals and extract statistical feature vectors for each interval. Given such data representation, the activities can be considered as clusters of data vectors to be discovered by clustering algorithm such as K-means or mixture models such as Gaussian mixture model (GMM). However, these methods require specifying in advance the number of clusters, which is not always available. More importantly in smart home environments, this number may change as the sensor data grows over time.

To solve this problem, we employ the Dirichlet process Gaussian mixture model (DPGMM) to infer the activities as multivariate Gaussian distributions from the data. As a Bayesian nonparametric model, DPGMM can automatically discover the number of clusters through the inference on data only. For model inference, we use the collapsed Gibbs sampling to infer the latent variables as the exact inference of DPGMM is intractable. Although some other clustering approaches such as DBSCAN also do not require the number of clusters as an input, they are suitable for some specific types of data such as geographical location data [20] where the distance metric can be clearly defined. Our approach, on the other hand, is generic to any type of data, so long as it can be represented by a probabilistic distribution.

We first test our approach on the public CASAS dataset [3, 4] that includes activity labels to evaluate the clustering performance metrics (F-score and Rand-index), which are popularly used to evaluate clustering performance [19]. Without the need to specify the number of clusters in advance, the DPGMM model can automatically infer it and at the same time achieve a performance close to the best of GMM. We then demonstrate the DPGMM on our smart home dataset. The extracted clusters are highly correlated to the daily routine provided by the home resident.

Our main contributions in this paper are as follows. (1) A smart home system comprised of ambient sensors to collect data in a real-life scenario. (2) An automatic feature extraction combining both event-based and analogue sensors. The issue of missing data in analogue sensors is addressed by a linear interpolation. (3) The discovery of activities using DPGMM, which can automatically infer the number of activities. (4) The substantial experiments on two different datasets collected in real-life environments.

[1]In this paper, we use the term 'analogue sensor' to refer to sensors that generate float values, e.g. power or humidity sensors, to distinguish them with the sensors that generate the event logs such as motion or door sensors.

The rest of this paper is organized as follows. Section 2 reviews the literature related to smart homes and activity recognition/discovery in smart homes, and provides some background of mixture models and Dirichlet process, which is the foundation of our approach. Section 3 describes our smart home system and data representation. Section 4 presents the DPGMM for activity discovery and its inference algorithm. Section 5 demonstrates the performance of our approach on the CASAS data and our smart home data. Finally, we conclude this paper in Section 6.

2 BACKGROUND

In this section, we first review the literature related activity recognition or discovery in smart homes. We then provide some background on mixture models and the Dirichlet process.

2.1 Smart Home Systems and Activity Recognition or Discovery in Smart Homes

According to De Silva et al. [6], smart homes can be categorized into three types: (1) providing services to the residents by recognizing their activities or detecting health conditions, (2) storing and retrieving multi-media captured within the home to enhance living experience, and (3) surveillance in order to protect the home and its residents from hazards such as burglaries, theft, or natural disaster. In this paper, we focus on the first category of smart homes, which can be used to monitor daily activities of home residents.

Early studies of smart homes employed cameras to collect data for activity recognition [11]. However, privacy is always a big concern when using cameras. Moreover, while cameras are widely accepted for security surveillance, they are still not welcome for in-home monitoring. Thus most recent smart homes use ambient sensors such as motion (a.k.a occupancy) sensor, reed switch, and temperature/humidity sensor. This is also the choice of our project. In fact, all sensors and devices used in our project are non-invasive.

The architecture for smart homes using ambient sensors was proposed in the early 2000s. For example, Cook et al. [5] proposed the MavHome architecture that includes sensors to recognize user activity and actuators to control objects. Another example is the Gator Tech Smart House [14], which includes a sensor layer, an information processing layer, and a control layer.

Existing smart home systems were typically deployed as test-beds to collect data for a short period of time. For example, Tapia et al. [34] used state-change sensors attached to everyday objects to detect participants' interaction with them. The iDorm project [9] was set up as a dormitory room inhabited by a participant and the data was collected for 5 days. The biggest and longest project in this area is the CASAS project [3, 4], which includes multiple test-beds set up at different locations to collect data in several months. We will use data from one of the test-beds in the CASAS project in our experiment to validate our approach. The CASAS project focused on using ambient sensors such as motion or reed sensors and environmental sensors such as temperature sensor. The Necesity system [2] also focused on setting up a smart home for older people. The data was collected for a five-month period. Suryadevara & Mukhopadhyay [33] proposed a smart home environment to assess home resident's wellness using sensors attached to domestic objects to detect usage and interaction. Riboni et al. [29] implemented

the FABER system to recognize abnormal behavior. However the participant needs to actively interact with the system (for example, he or she needs to scan the medicine bar code before taking it). This requirement would be a challenge, especially for older adults.

Overall, typical ambient sensors used in smart homes include motion (or occupancy), reed, light, and temperature sensors. In our smart homes, besides motion sensors and reed switches, we also use power sensors and circuit meters to measure the energy consumed by appliances, and humidity sensors to record humidity changes.

Activity recognition or discovery is a fundamental component in any smart home system, which can be categorized into supervised learning (classification) and unsupervised learning (clustering).

Supervised learning methods, specifically classifiers, have been popularly employed for activity recognition in smart homes. For example, Tapia et al. [34] used Naive Bayes to classify activities from the data collected by state-change sensors. Kasteren et al. [35] employed Hidden Markov Model and Conditional Random Field to recognize activities, also from state-change sensor data. Storf et al. [32] and Krishnan & Cook [17] used Support Vector Machine to recognize activities from event-based sensor data.

Although achieving acceptable classification performance, these approaches require a set of labeled data to train the classifiers. These labels are typically obtained by annotations from camera videos or annotated by the participants themselves. Using camera would violate residents' privacy while self-annotating is a time-consuming process and may not be applicable to older adults. Therefore recent studies have shifted to using unsupervised approaches.

Unsupervised learning has also been used for activity discovery from data observations without using any labeled data [16, 26, 27]. In particular, the event-based sensor data is converted into symbolic sequences, then sequence mining techniques are applied to discover typical patterns in such symbolic sequences. For example, Rashidi et al. [26, 27] discovered frequent patterns in the symbolic sequences and then apply clustering methods on these patterns to find typical sequences. Another example is the AALO method [16], which represents the occupancy data as item-sets and performs clustering to discover activities.

However, as the features in these methods are discrete, they are naturally suitable only for event-based sensors. Besides event-based sensors, analogue sensors such as power or humidity sensor can also provide valuable information about user's activities, such as appliance usage or taking shower (indicated by an abrupt change of humidity). Moreover, in these methods, the semantic annotations of the sensors, e.g. location or attached objects, are encoded in the features. This would be a challenge as each home may have a different and sensor deployment. Thus, this feature extraction needs to be performed or monitored by a researcher for each new home. Therefore, in our approach, the features are extracted automatically from sensor data, without using semantic annotations of the sensors. Our approach can also combine event-based sensors and analogue sensors. The data extracted by our method can be clustered by any standard clustering method or mixture model.

Some recent work [30, 36] employed ontology methods to recognize activities by reasoning the relationship between the sensor events and activities. However, these methods require a substantial effort in defining the ontology relationships.

2.2 Mixture Models for Data Clustering

As a repeated activity may fire the same set of sensors resulting in similar observations, we consider activities as the clusters of data vectors. Such clusters can be discovered by mixture models. For this, data vectors are assumed to be generated from a mixture of probabilistic distributions. Formally, let $f(\cdot \mid \phi_k)$ denote the density function of the distribution representing the clusters, a data vector is generated from a mixture of K distributions as

$$p(x_i \mid \pi, \phi_{1:K}) = \sum_{k=1}^{K} \pi_k f(x_i \mid \phi_k) \tag{1}$$

where π is the mixture proportion vector in which π_k is the probability of a data vector belonging to the k-th distribution. Note that π is typically normalized so that for all nonnegative π_k, $\sum_{k=1}^{K} \pi_k = 1$. The distribution representing data can selected according to data type. For example, Gaussian distribution can be used to model continuous data while multinomial or categorical distribution can represent discrete data.

For the ease of inference, it is typical to introduce the latent variable z_i associating with data vector x_i to indicate the cluster of x_i (z_i takes an integer number in range $[1, K]$). Given such mixture models, the parameters and hidden variables can be inferred using the Expectation-Maximization (EM) algorithm [7].

To perform Bayesian inference on the mixture models, the parameters are assumed to follow some prior distributions. To derive the posteriors, the prior distributions are selected as conjugate priors. For example, a Gaussian-Gamma is the conjugate prior of the univariate Gaussian distribution; a Gaussian Inverse-Wishart is the prior of the multivariate Gaussian distribution. As the mixture proportion π is a discrete or categorical distribution, a Dirichlet distribution is chosen as its conjugate prior. A Dirichlet distribution is a probabilistic distribution over a K-simplex space $\pi = (\pi_1, \ldots, \pi_K)$ where $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \in [0, 1]$. Generally, it is parametrized by a vector of positive real numbers. However, to simplify the model, a symmetric Dirichlet distribution is typically used. As such, its parameters are uniform and can be denoted as α. Given such symmetric Dirichlet distribution, the likelihood of a mixture proportion vector π can be computed as

$$f(\pi_1, \ldots, \pi_K; \alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{k=1}^{K} \pi_k^{\frac{\alpha}{K}-1} \tag{2}$$

The inference of the Bayesian (finite) mixture models described so far is typically developed using Markov chain Monte Carlo (MCMC) methods, such as Gibbs sampling [8].

2.3 Dirichlet Process

The key drawback of the Bayesian finite mixture models described in section 2.2 is that they require the number of clusters or mixture components to be specified in advance. This requirement limits their usability in real-world problems as the number of clusters is not always available and may even change when more data arrive. For example, in the smart home scenario, the set of activities performed by the resident may change over time.

This limitation can be overcome by replacing the Dirichlet distribution in Eq. (2) by a Dirichlet process (DP) [10]. A DP can be

considered as a generalization of Dirichlet distribution when allowing the dimensionality K to be infinite. There are several ways to illustrate a DP. One of such ways is the stick-breaking construction [31], which realizes the DP as follows. Let us begin with a stick of length 1. First we sample $\beta_1 = Beta\,(1, \alpha)$ and break β_1 from the stick. For $k = 2, 3, \ldots$, we sample $\beta_k = Beta\,(1, \alpha)$, and break β_k from the remaining stick. Formally, π can be obtained from this breaking construction as

$$\pi_1 = \beta_1; \quad \pi_k = \beta_k \prod_{j=1}^{k-1} \left(1 - \beta_j\right) \text{ for } k = 2, 3, \ldots \tag{3}$$

It is obvious that the probability of $\sum_{k=1}^{\infty} \pi_k = 1$ is equal to 1. Thus this construction can result in a mixture proportion vector of infinite limit.

Another way to explain a DP is the Chinese restaurant process (CRP) metaphor [24], which describes the process to obtain the indices z_i. Note that in clustering, the actual value of cluster indices are unimportant – two set of indices are equivalent so long as the partitions of data vectors are the same. Thus in CRP, assigning a data vector to a cluster is equivalent to a customer choosing a table to sit. Assume we have a restaurant with an infinite number of tables, each table has infinite number of seats. The first customer chooses any random table. The following $(n + 1)$-th customer chooses an existing table k with the probability proportional to the number of customers that already sit there, or a random new table with the probability proportional to α. The final mixture proportion π can be calculated based on the number of data vectors in each clusters. Overall, the CRP can be summarized as

$$p\left(z_{n+1} = k \mid z_{1:n}\right) = \begin{cases} \frac{n}{n+\alpha}, \text{for existing table} \\ \frac{\alpha}{n+\alpha}, \text{for new table} \end{cases} \tag{4}$$

The main characteristic of this process is that the rich gets richer. As can be seen in Eq. (4), a table (or cluster) with the higher number of customers (or data vectors) has a higher chance to attract new customers. It can also be observed that α, which is called the *concentration* parameter, can control the behavior of the process. The larger α may produce a larger number of clusters.

The DP is also denoted as GEM (α), where the letters stand for Griffiths, Engen and McCloskey [24]. This process, however, only shows the prior knowledge that we apply. Further details on the theory and applications of DP can be found in [15]. In the context of activity discovery for smart homes, we will present our approach using DP in section 4.

3 OUR SMART HOME SYSTEM AND DATA REPRESENTATION

In this section, we present the smart home system and data representation used in our project.

3.1 Our Smart Home System

In this project, we only use ambient sensors to preserve residents' privacy. In particular, we use the following sensors. Motion (a.k.a. occupancy) sensors are placed in each room to collect participant's appearance. Reed switches or accelerometer-based door sensors are attached to the doors to record door interaction. Power sensors or circuit meters are used to collect power consumption of the appliances. Humidity sensors are placed in the bathroom to record changes in humidity. All sensors (except power sensors and circuit meters) are powered by batteries and communicate with a network hub via ZigBee wireless connection to transfer recorded data to our centralized server. Thus, they can be placed at any desired location without depending on a wired connection. We optimize the sampling rate and data communication to lengthen the battery life. On average, a battery lasts for 6 to 8 months. This long battery life reduces the maintenance cost of the system.

We started a smart home clinical trial together with a aged care service provider in its independent living homes. The ethical application of our project was approved by the Commonwealth Scientific and Industrial Research Organization (CSIRO) Health and Medical Research Human Research Ethics Committee (#12/17). We recruited the participants in a voluntary basis. We then explained the protocol used in the project to the participants and obtained their signed consent form granting us the privileges to use their data for our research. To preserve the participants' privacy, we anonymized their identifiable information. Totally, we deployed the system to about 20 houses. Most of them are single resident houses occupied by senior adults. The collection period varies among the houses, the shortest one is about 2 months. Table 1 presents a period of 5 min data collected from one of our smart homes.

Table 1: An example period of sensor readings from our smart home project. Note that the original timestamps include date and time, however we only print the time part in this table for a short presentation.

Timestamp	Sensor	Sensor Type	Location	Readings
1:23:00 PM	P3	Power	Kettle	0.014
1:23:00 PM	C1	Circuit	Stove	0.005
1:24:00 PM	P1	Power	TV	0.001
1:24:00 PM	C1	Circuit	Stove	0.009
1:24:40 PM	R3	Reed	Wardrobe	Open
1:25:00 PM	P1	Power	TV	0.001
1:25:00 PM	C1	Circuit	Stove	0.015
1:26:00 PM	C1	Circuit	Stove	0.006
1:26:16 PM	H1	Humidity	Bathroom	37.91
1:27:00 PM	P1	Power	TV	0.001
1:27:00 PM	C1	Circuit	Stove	0.008
1:27:38 PM	M4	Motion	Kitchen	1
1:28:00 PM	C1	Circuit	Stove	0.006
1:28:11 PM	M2	Motion	Livingroom	1

3.2 Data Representation

In a smart home, multiple sensors may be activated concurrently (or in a short period of time) as can be seen in Table 1. A repeated activity may activate the same set of sensors. For example, preparing meal may activate motion sensor in the kitchen and the circuit meter attached to the stove. Therefore, we need a data representation that expresses this concurrency to help us discover the activities. There are two challenges in dealing with the data in this scenario: (1) missing of analogue sensor data, and (2) how to combine event-based and analogue sensor data.

The first challenge is because of the sparse sampling rate of the analogue sensors, e.g. every 5 min for humidity sensors, while the real values are actually continuous. To address this challenge, we use a linear interpolation to fill in the missing values between the samples. For appliance power consumption, if the missing period is shorter than the sampling interval, the interpolated values are filled in; otherwise we assume that the appliance is not in used (the power consumption equals to 0).

To address the challenge of combining event-based and analogue sensor data, we split the data into 1 min intervals and extract a feature vector for each interval. For each sensor, we compute statistical values using its recorded data in each interval. As we use both event-based sensors (motion and door) and analogue sensors (power sensors, circuit meters, and humidity sensors), we derive different formulation for each sensor type. For event-based sensors (motion and door), we count the number of times each sensor is activated during each time interval. For analogue sensors (e.g., power sensors and circuit meters), we compute the average value of each sensor during each time interval. This average value indicates the power consumption of each appliance during such time interval.

The humidity sensor is slightly different, as the change of humidity is more important than the raw sensor reading. For example, when the shower is in used, the humidity in the bathroom increases abruptly. It also reduces quickly afterwards. Thus we compute the absolute difference of humidity per minute.

As each feature has a different value range, we standardize them so that each feature has zero mean and unit variance. This standardization eliminates the effect of feature scales on clustering results. For a feature d, the standardized value is computed as:

$$x_i'^d = \frac{x_i^d - \overline{x}^d}{\sigma^d} \tag{5}$$

where $\overline{x}^d$ and σ^d are the mean and standard deviation of d.

Our automatic feature extraction has several advantages. First, it can be applied to any type of ambient sensor such as event-based sensor and analogue sensor. Second, it does not need information about sensor deployment. And finally, it is totally automatic and thus it can be easily applied to new homes with new sensor deployment and settings.

4 DIRICHLET PROCESS GAUSSIAN MIXTURE MODEL FOR ACTIVITY DISCOVERY

This section describes the Dirichlet process Gaussian mixture model for activity discovery and its inference algorithm using collapsed Gibbs sampling.

4.1 Model Description

To discover activities from the aforementioned data, we consider the activities as the clusters of data vectors. As the data is represented as continuous vectors, we use multivariate Gaussian distributions to model the clusters. To infer the number of clusters automatically, we employ the Dirichlet process Gaussian mixture model (DPGMM) of which graphical representation is illustrated in Figure 1. In this figure, the observed data is presented as shaded nodes x_i, while the hidden variables are presented as white nodes. The model assumes that data is generated from a mixture of multivariate Gaussian

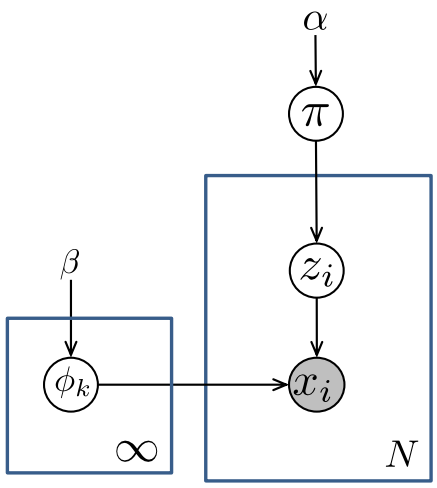


Figure 1: Graphical representation of DPGMM. Note that α is the concentration parameter of the Dirichlet process, $\phi_k = (\mu_k, \Sigma_k)$ is the parameters of the multivariate Gaussian distribution, and $\beta = (\mu_0, \lambda_0, \upsilon_0, \mathbf{W}_0)$ is the hyper-parameter of the Gausian Inverse-Wishart distribution.

distributions parametrized by (μ_k, Σ_k), which are denoted as ϕ_k in Figure 1. For the ease of model interpretation, we introduce a latent variable z_i indicating the cluster that x_i belongs to. Given z_i, the probability likelihood of x_i is computed as

$$p(x \mid z_i, \phi) = f(x \mid \mu_{z_i}, \Sigma_{z_i}) \tag{6}$$

where $f(x \mid \mu_k, \Sigma_k)$ is the density function of the k-th Gaussian distribution. As the indicators z_i's are discrete, they are assumed to be generated from a multinomial distribution parametrized by π.

To perform Bayesian inference, we further endow the parameters with prior distributions. In particular, the multivariate Gaussian distributions are assumed to follow a Gaussian Inverse-Wishart distribution with the parameters (μ_k, Σ_k) drawn from

$$\begin{aligned} \mathbf{\Lambda} \mid \mathbf{W}_0, \upsilon_0 &\sim \mathcal{W}^{-1}(\mathbf{\Lambda} \mid \mathbf{W}_0, \upsilon_0) \\ \mu \mid \mu_0, \lambda_0, \mathbf{\Lambda} &\sim \mathcal{N}\left(\mu \mid \mu_0, (\lambda_0 \mathbf{\Lambda})^{-1}\right) \end{aligned} \tag{7}$$

where $(\mu_0, \lambda_0, \upsilon_0, \mathbf{W}_0)$ are the hyper-parameters of the Gaussian Inverse-Wishart distribution. Note that in Figure 1, $(\mu_0, \lambda_0, \upsilon_0, \mathbf{W}_0)$ are denoted as β.

Traditional mixture models assume that the mixture proportion π follows a Dirichlet distribution that only supports a fixed number of clusters K. However, this fixed K limits the model usability in practice as it is difficult to define K in advance. In our problem, this number may even change when data grows. In contrast, DPGMM assumes that π is drawn from a Dirichlet process (DP) [10]. As discussed earlier in Section 2.3, a DP can generate the indices from an infinite pool of integers. It means that DPGMM can generate as many clusters as it needs. The process of obtaining cluster indices from the observed data will be discussed in Section 4.2.

The *generative process* of DPGMM is summarized as follows.

(1) Sampling the mixture proportion $\pi \sim \mathrm{DP}(\alpha)$
(2) Sampling the Gaussian parameters μ_k and $\Sigma_{\mathbf{k}}$ using Eq. (7).
(3) For each data vector x_i:
    (a) Sampling the cluster indicator z_i using Eq. (4).
    (b) Sampling the data vector x_i using Eq. (6).

4.2 Model Inference with Collapsed Gibbs Sampling

Similar to other Bayesian nonparametric models, the exact inference of DPGMM is intractable. Instead, we use a Markov chain Monte Carlo (MCMC) sampling method [12]. The intuition of MCMC methods is that it iteratively samples a hidden variable using the current status of the remaining ones. When this process is repeated for a large enough number of iterations, the set of sampled variables converges. As we have chosen the conjugate prior distributions for the parameters, we can integrate out the parameters and sample only the cluster indicators z_i. This technique is called collapsed Gibbs sampling [18]. In particular, we iteratively take each data vector i out to consider it as the last arrival one and sample its indicator z_i using the following equation

$$\begin{aligned} p(z_i = k \mid z_{-i}, x, \alpha, \beta) \propto & p(z_i = k \mid z_{-i}, \alpha) \\ & \times p(x_i \mid z_{-i}, z_i = k, x_{-i}, \beta) \end{aligned} \tag{8}$$

where z_{-i} denotes all indicators excluding z_i and x_{-i} denotes all data vectors excluding x_i. The first component in the right hand side of Eq. (8) is the prior distribution of z_i, which can be replaced by the CRP in Eq. (4).

The second term in the right hand side of Eq. (8) is called the posterior predictive likelihood, i.e. the probability likelihood of a data vector x_i over a cluster given its existing data vectors. As we use a conjugate prior (Gaussian Inverse-Wishart) as the prior distribution, we can easily derive this posterior predictive likelihood similar to Eq. (232) of [21]. As a result, this predictive likelihood follows a T-distribution

$$p(x_i \mid x, z_{-i}, \beta) = t_{v_k - D + 1}\left(x_i \mid \mu_k, \frac{\mathbf{W}_k (\lambda_k + 1)}{\lambda_k (v_k - D + 1)}\right) \tag{9}$$

where D is the dimensionality of data, and $(\mu_k, \lambda_k, v_k, \mathbf{W}_k)$ are the parameters of the posterior Gaussian Inverse-Wishart distribution estimated as

$$\begin{aligned} \mu_k &= \frac{\lambda_0 \mu_0 + n_k \bar{\mu}_k}{\lambda_k}; \\ \lambda_k &= \lambda_0 + n_k; \\ v_k &= v_0 + n_k; \\ \mathbf{W}_k &= \mathbf{W}_0 + \bar{\Sigma}_k + \frac{\lambda_0 n_k}{\lambda_k} (\bar{\mu}_k - \mu_0)(\bar{\mu}_k - \mu_0)^T \end{aligned} \tag{10}$$

Note that $\bar{\mu}_k$ and $\bar{\Sigma}_k$ are the sufficient statistics of the k-th Gaussian distribution. This means that we only need to update them when there is a change of the cluster membership and do not have to calculate them again when estimating the posterior. A step-by-step summary of the collapsed Gibbs sampler for DPGMM is presented in Algorithm 1. It can be seen that the algorithm can generate new cluster if required (lines 10 to 12). An existing cluster can also be removed if it does not attract any data vector (line 5). These updates are based on the observed data, thus when converged, the algorithm can automatically find the number of clusters.

Algorithm 1 Collapsed Gibbs sampler for DPGMM

Input: Hyperparameters α, μ_0, λ_0, v_0, $\mathbf{W}_0$, initialized K.
1: Initialize the indicators z_i as random integer numbers in range $[1, K]$.
2: Based on all z_i, calculate the sufficient statistics of the clusters.
3: **repeat**
4: **for** $i = 1$ to N **do**
5: Remove z_i from its current cluster, update the cluster sufficient statistics.
6: If that cluster is empty, remove it, update the indices and decrease K.
7: **for** $k = 1$ to K **do**
8: Calculate $p(z_i = k \mid \cdot)$ according to Eq. (8).
9: **end for**
10: Calculate $p(z_i = k \mid \cdot)$ (for a new cluster) according to Eq. (8).
11: Sample z_i from the normalized $p(z_i = k \mid \cdot)$.
12: Update the sufficient statistics of cluster indexed by z_i.
13: If $z_i = K + 1$, update $K = K + 1$.
14: **end for**
15: **until** Converged.
Output: The cluster indicators z_i.

The convergence condition in line 15 of Algorithm 1 can be flexible in different ways. The algorithm can be considered as converged after a fixed number of iterations, or when the cluster indices do not change after a few iterations. In our experiment, we choose a fixed number of iterations for the ease of implementation. To assure that we obtain the converged results, we assign the number of iterations larger than what we need.

5 EXPERIMENTAL RESULTS

We demonstrate our approach on two datasets: a public smart-home dataset from the CASAS project [3] and the dataset collected in our smart-home project.

5.1 Experiments on Aruba Dataset

We first validate our approach on the Aruba dataset in the CASAS project[2] [3]. This dataset was collected from an apartment occupied by an older lady. There were 39 sensors deployed in the department. The deployment layout can be found in [23]. The data was collected for about 7 months (from November 2010 to June 2011). In this paper, we only use motion and door sensor data collected by 34 sensors as the temperature sensor data may not be useful activity discovery, given the fact that temperature data was also omitted in the CASAS team papers [26, 27]. We note that the CASAS project does not include power consumption sensors. Following the method presented in Section 3.2, we split data into 1 min intervals and extract features for these intervals. We eliminate the feature vectors that do not have any event to obtain 111, 297 active feature vectors.

Besides sensor data, the Aruba dataset also includes some annotated labels of activities, namely enter/leave home, bed to toilet, sleeping, meal preparation, wash dishes, eating, relax, resperate, work, and housekeeping. We use these labels as ground-truth to

[2] http://ailab.wsu.edu/casas/datasets/ (accessed on 6/10/2016)

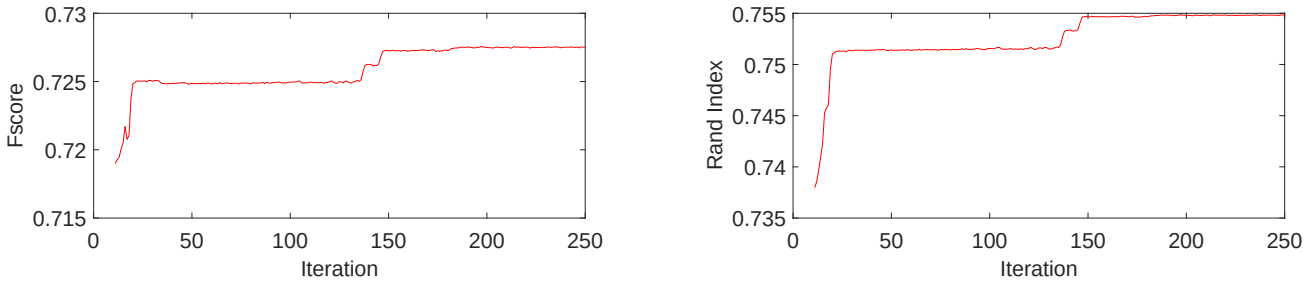


Figure 2: Convergence of DPGMM measured by F-score (left) and Rand-index (right) on the Aruba dataset. We ignore the first 10 iterations as the cluster indices are still unstable. The algorithm converges after less than 200 iterations.

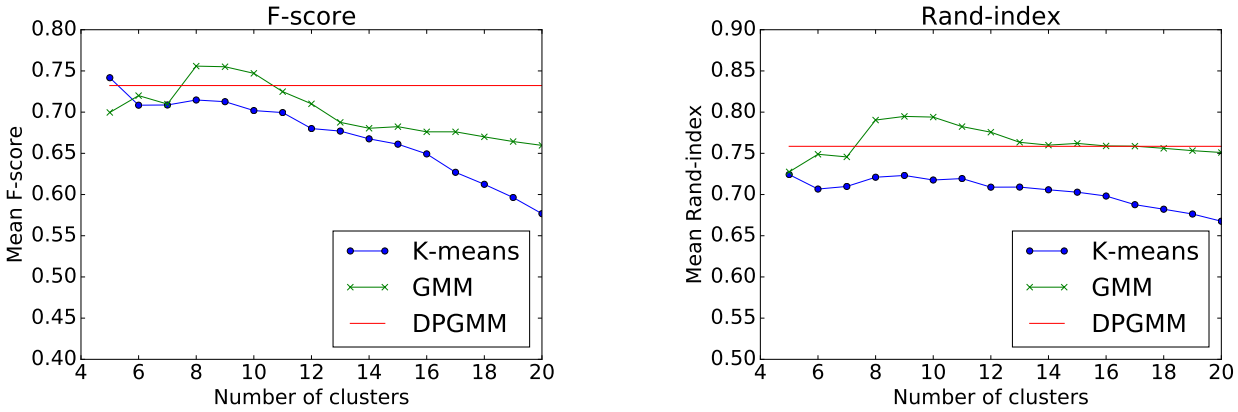


Figure 3: Comparison of F-score and Rand-index obtained by K-means, GMM, and DPGMM on the Aruba dataset. As DPGMM does not require the number of clusters to be specified, we plot its performances as straight lines over the number of clusters.

evaluate our clustering results in standard metrics: F-score (derived from precision and recall) and Rand-index [19]. These metrics have been popularly used to evaluate clustering results.

We run DPGMM model on this data with 250 iterations including a burn-in period of 50 iterations. We report the results using the last Gibbs sample as this is the converged sample, which includes 19 clusters. We use the cluster indices (z_i) to compute the performance metrics (F-score and Rand-index) versus the ground-truth activity labels. To illustrate the convergence of DPGMM, we plot the F-score and Rand-index over the iterations in Figure 2. Note that we ignore the first 10 iterations as the cluster indices are not yet stable, thus the performance metrics may not be meaningful. It can be seen in this figure that the metrics are improved quickly during the first 20 iterations. They are then quite stable until another changing period from iterations 140 to 150. During these erupted periods, many clusters are split and formed. After 200 iterations, the performance metrics are not improved significantly. Thus we can assure to achieve the best results with 250 iterations.

To obtain baselines for comparison, we run both K-means and GMM on this data. Note that we do not compare our approach with the symbolic sequence methods in [16, 26, 27] as they are specifically designed for event-based sensors only, while our approach can be applied for both event-based and analogue sensors.

As both K-means and GMM require the number of activities K to be specified in advance, we test them with different values of K, ranging from 5 to 20, and report the performance in Figure 3. As DPGMM does not require the number of clusters to be specified in advance, we plot its performance metrics as straight lines over the number of clusters. We observe from this figure the following results. The performance of DPGMM is always better than K-means over all values of K. Although DPGMM cannot achieve the best performance of GMM, it is only worse than GMM when GMM takes $K = 8$ to $K = 10$. But more importantly, DPGMM can automatically discover the number of clusters. This is especially important when the ground-truth is not available, in which case we cannot evaluate the performance of GMM to select the best value of K.

To have a deeper look at the clustering results, we plot the confusion matrices between ground-truth and clusters obtained by GMM and DPGMM in Figure 4. For GMM, we select $K = 10$, among its best values of K. As can be seen in this figure, the main clusters obtained by GMM and DPGMM are equivalent. We note that a large number of clusters obtained by DPGMM include very few data vectors. These data vectors can be seen as outliers that may not be close to any of the main clusters.

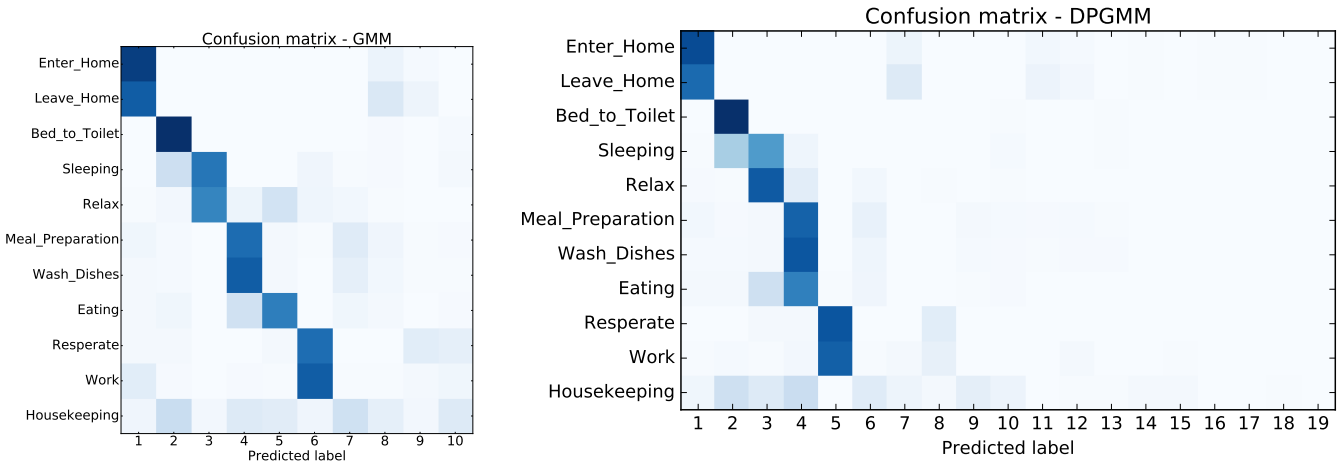


Figure 4: Confusion matrix obtained by GMM with K=10 (left) and DPGMM (right) on Aruba data. As the number of samples for each activity is different, we normalize to make the sum of each row equal to 1.

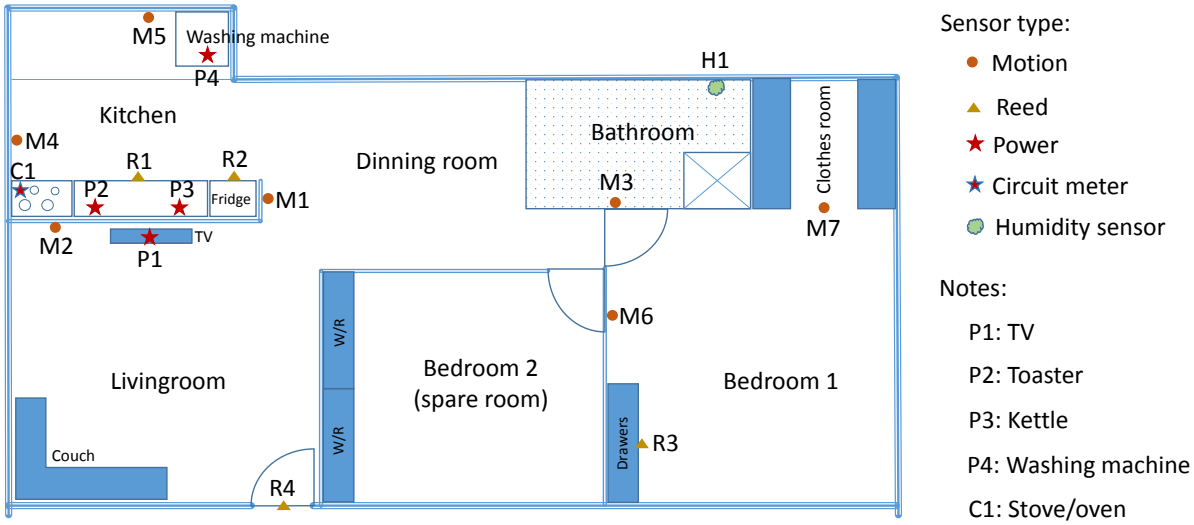


Figure 5: Sensor deployment at H1. Motion sensors are denoted by M, reed sensors denoted by R, power sensors denoted by P, circuit meter denoted by C, and humidity denoted by H.

Overall, the key result of DPGMM is that it can automatically determine the number of clusters while still achieves the performance close to the best of GMM and outperforms K-means.

5.2 Experiments on Our Smart Home Data

In this section, we further demonstrate the DPGMM model on the data collected at one of our smart homes (H1). We deploy 7 motion sensors, 4 reed switches, 3 power sensors, 1 circuit meter, and 1 humidity sensor in this home. The sensor deployment layout is presented in Figure 5. The data was collected at this home for more than a year. We extract the features following the process described in Section 3.2 to obtain about 150, 000 data vectors.

We run DPGMM on this data with 250 iterations including a burn-in period of 50 iterations. We report the results using the last Gibbs sample, which includes 22 clusters. We plot the number of occurrences of the clusters in Figure 6. As the cluster indices are randomly assigned in the inference and are not important, we sort the clusters by the descending order of the number of occurrences. Some clusters attract very few data vectors while the majority of the data vectors are distributed into top 10 clusters. The data vectors that belong to the long-tailed clusters may be the outliers.

We further visualize top 6 clusters obtained by DPGMM in Figure 7. The plots include the activated sensors (a) and the accumulative daily active time (b). Note that because of the feature normalization, the scales presented in these plots are the number

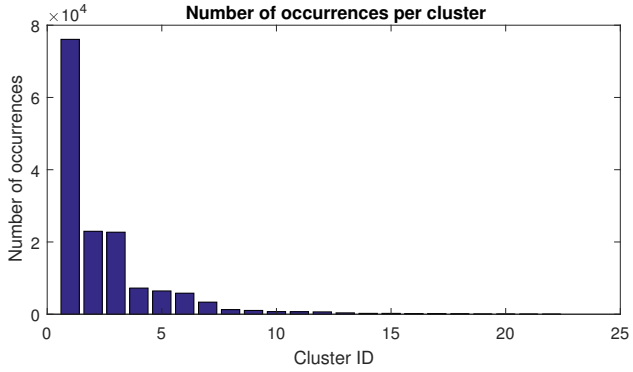


Figure 6: Number of occurrences per cluster obtained by DPGMM on the data of H1 in our project.

variances away from the means. To compute the active time, we divide 24h of a day into 10-min intervals, and count the number of times each cluster is activated in each of these intervals. We aggregate all days into a 24h period to obtain the plot in Figure 7(b). In cluster 1, the motion sensor in the living room (M2) and the power sensor of TV (P1) are activated, and the most active time is from 18:00 to 21:00. In cluster 2, the motion sensor in the dining room (M1) is activated. In cluster 3, the motion sensor in the bathroom (M3) and the bedroom (M6) are fired, mostly around 22:00 (before bedtime). In cluster 4, the cooking appliances (R2 – cupboard, P2 – toaster, and C1 – stove) are on around 12:00 (lunch time) and 18:00 (dinner time). In cluster 5, the motion sensor in the laundry (M5) and the washing machine (P4) are active around 9:00 to 10:00. Note that M4 (kitchen) is also activated in this cluster as the laundry is close to the kitchen. In cluster 6, the motion sensor in the bathroom (M3) and the drawers in the bedroom (R3) are fired in concurrent with the humidity changes (H1) mostly around 20:00 and 22:00.

Although we could not obtain activity labels over time, we interview the participant to ask for her typical routine during a day. We compare this routine with the active time plotted in Figure 7b. Her typical routine is as follows. She got up at 8:00, hygiene, and took some medicines from the fridge. This activity is consistent with the first peak of cluster 2. Note that she also took some medicines before bedtime, thus there is another peak of cluster 2 around 22:00. She had lunch at around 12:30 to 13:00 and dinner at around 19:00, which are coincident with the two peaks of cluster 4. She also watched TV from 18:30 to 20:00, which is the active time of cluster 1. She then took a shower between 21:00 to 22:00, which is the peak of cluster 3 and cluster 6. Then she went to bed at around 23:00. From this time to around 8:00 in the morning, there are not many activities in any cluster. The only cluster that is not relevant to the routine is cluster 5, which includes movement in the laundry and the working washing machine. This cluster is likely to be cloth-washing activity. However, it is not listed in the surveyed routine. That may be because washing is not a daily activity. Note that different clusters may overlap each other in this time plot as we aggregate the counts of different days into 24h. Overall, the active times of the clusters are consistent with the typical routine we obtain from the participant.

The activated sensors and the active time of the clusters presented in Figure 7 show that even without any ground-truth labels, our approach can discover meaningful activities. The activated sensors and the happening time is consistent with participant's routine. Although the clusters are discovered automatically from data, the semantic meaning of the activities that trigger such clusters can be found by looking at the activated sensors and happening time.

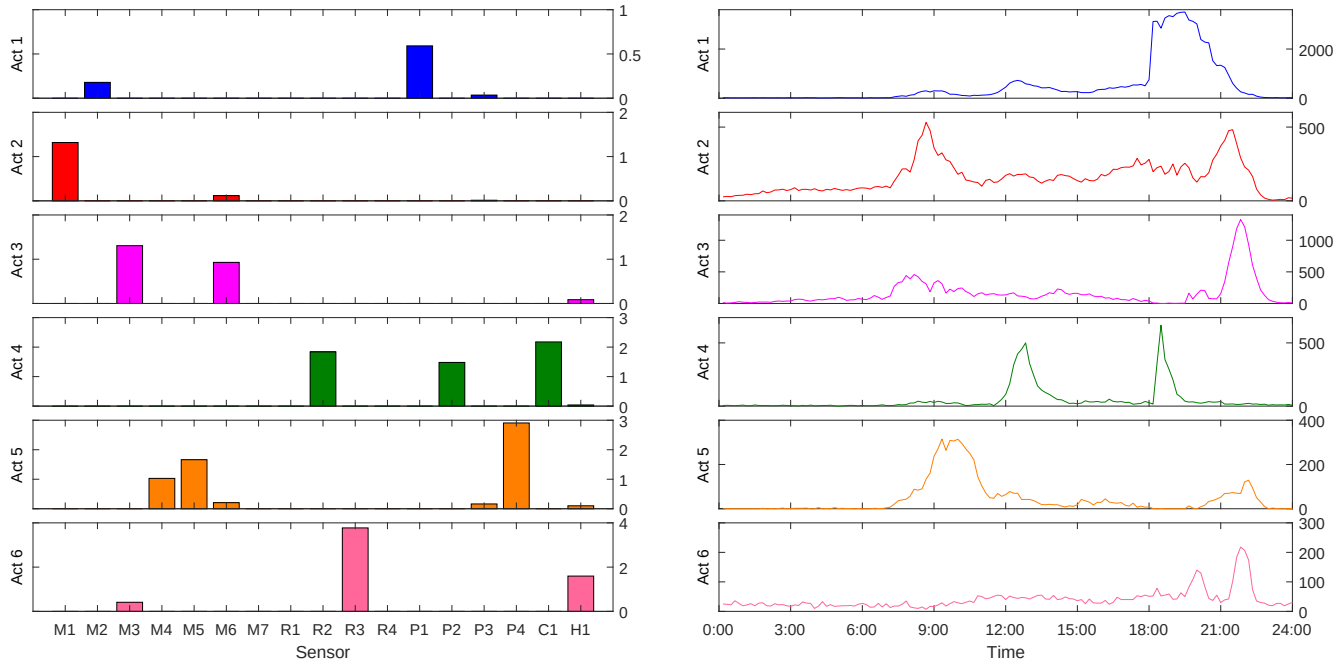
6 CONCLUSION

We have presented the Dirichlet process Gaussian mixture model (DPGMM) for activity discovery in smart homes. Our approach does not require any labeled data. The features are extracted from both event-based sensors (e.g. motion and reed), and analogue sensors (e.g. power and humidity). Missing data of analogue sensors are filled in by interpolation. As we do not use any information related to sensor deployment plan, our smart home system can be easily deployed to new homes with new deployment plan. The DPGMM can automatically infer the number of activities from the data, thus it is especially suitable for smart homes where the activities may change with data. We have demonstrated our approach on a public dataset that includes activity labels to show a good confusion between the discovered clusters and the true activities. The DPGMM can achieve the performance close to the best of GMM without the need to specify the number of activities in advance. We also demonstrated our approach on the data collected in our smart home project to show the correlation between the discovered clusters and participant's daily routine.

Overall, our approach can be applied to different types of ambient sensors in different home settings. More importantly, the features extracted by our approach can be applied to any standard clustering algorithms, which can be investigated further in future studies. Moreover, we are planning to deploy our smart home system to a larger amount of homes to collect more real-world data.

REFERENCES

[1] Australian Institute of Health and Welfare (AIHW). 2015. Australia's welfare 2015 – Growing older. (2015). http://www.aihw.gov.au/australias-welfare/2015/growing-older/
[2] J. A. Botia, A. Villa, and J. Palma. 2012. Ambient assisted living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications* 39, 9 (2012), 8136–8148.
[3] D. J. Cook. 2012. Learning Setting-Generalized Activity Models for Smart Spaces. *IEEE Intelligent Systems* 27, 1 (2012), 32–38.
[4] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. 2013. CASAS: A Smart Home in a Box. *Computer* 46, 7 (2013), 62–69.
[5] D. J. Cook, M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. 2003. MavHome: an agent-based smart home. In *The First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*. IEEE, 521–524.
[6] L. C. De Silva, C. Morikawa, and I. M. Petra. 2012. State of the art of smart homes. *Engineering Applications of Artificial Intelligence* 25, 7 (2012), 1313–1321.
[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.
[8] J. Diebolt and C. P. Robert. 1994. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)* (1994), 363–375.
[9] F. Doctor, H. Hagras, and V. Callaghan. 2005. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 35, 1 (2005), 55–65.
[10] T. S. Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* 1, 2 (1973), 209–230.



(a) Activated sensors. **(b) Active time.**

Figure 7: Top 6 clusters discovered by DPGMM from our smart home data: (a) The activated sensors, and (b) the active time.

[11] S. Fleck and W. Straßer. 2008. Smart camera based monitoring system and its application to assisted living. *Proc. of the IEEE* 96, 10 (2008), 1698–1714.

[12] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. 1996. Introducing Markov chain monte carlo. In *Markov chain Monte Carlo in practice*. Springer, 1–19.

[13] W. He, D. Goodkind, and P. Kowal. 2016. An Aging World: 2015 – International Population Reports. *U.S. Census Bureau* (2016). http://www.census.gov/content/dam/Census/library/publications/2016/demo/p95-16-1.pdf

[14] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. 2005. The Gator Tech Smart House: A Programmable Pervasive Space. *Computer* 38, 3 (2005), 50–60.

[15] N.L. Hjort, C. Holmes, P. Müller, and S.G. Walker. 2010. *Bayesian nonparametrics*. Cambridge University Press.

[16] E. Hoque and J. Stankovic. 2012. AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 139–146.

[17] N. C. Krishnan and D. J. Cook. 2014. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing* 10 (2014), 138–154.

[18] J. S. Liu. 1994. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. Amer. Statist. Assoc.* 89, 427 (1994), 958–966.

[19] C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge University Press.

[20] R. Montoliu and D. Gatica-Perez. 2010. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 12.

[21] K. P. Murphy. 2007. Conjugate Bayesian analysis of the Gaussian distribution. *def* 1, $2\sigma 2$ (2007), 16.

[22] J. M. Ortman, V. A. Velkoff, H. Hogan, and others. 2014. An aging nation: the older population in the United States. *Washington, DC: US Census Bureau* (2014), 25–1140.

[23] H. Park, C. Basaran, T. Park, and S. H. Son. 2014. Energy-Efficient Privacy Protection for Smart Home Environments Using Behavioral Semantics. *Sensors* 14, 9 (2014), 16235–16257.

[24] J. Pitman. 2006. *Combinatorial stochastic processes*. Lecture Notes in Mathematics, Vol. 1875. Springer-Verlag, Berlin. x+256 pages.

[25] Productivity Commission and others. 2015. Housing decisions of older Australians. *Commission Research Paper, Canberra* (2015). http://www.pc.gov.au/research/completed/housing-decisions-older-australians/housing-decisions-older-australians.pdf

[26] P. Rashidi and D. J. Cook. 2013. COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, 4 (2013), 64.

[27] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe. 2011. Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering* 23, 4 (2011), 527–539.

[28] P. Rashidi and A. Mihailidis. 2013. A survey on ambient-assisted living tools for older adults. *IEEE Journal of Biomedical and Health Informatics* 17, 3 (2013), 579–590.

[29] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui. 2015. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 149–154.

[30] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt. 2016. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1–12.

[31] J. Sethuraman. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica* 4, 2 (1994), 639–650. http://www.cs.princeton.edu/courses/archive/fall07/cos597C/readings/Sethuraman1994.pdf

[32] H. Storf, T. Kleinberger, M. Becker, M. Schmitt, F. Bomarius, and S. Prueckner. 2009. An event-driven approach to activity recognition in ambient assisted living. In *European Conference on Ambient Intelligence*. Springer, 123–132.

[33] N. K. Suryadevara and S. C. Mukhopadhyay. 2014. Determining wellness through an ambient assisted living environment. *IEEE Intelligent Systems* 29, 3 (2014), 30–37.

[34] E. M. Tapia, S. S. Intille, and K. Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*. Springer, 158–175.

[35] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate activity recognition in a home setting. In *The 10th international conference on Ubiquitous computing*. ACM, 1–9.

[36] J. Ye, G. Stevenson, and S. Dobson. 2015. USMART: An unsupervised semantic mining activity recognition technique. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 4 (2015), 16.