

Machine Learning for Activity Recognition in Smart Buildings: A Survey



Manar Amayri, Samer Ali, Nizar Bouguila, and Stephane Ploix

1 Introduction

New buildings (e.g., commercial, residential, public) are now generally equipped with a variety of smart sensors and smart meters. This makes these buildings smart with improved capacities, opportunities, and applications related, for instance, to energy management systems which main goal is to decrease waste mainly due to irresponsible human behaviors [1]. Indeed, energy deficiency represents a global problem. Hence, energy generation increase and consumption efficiency are two active areas of research [2]. Nonetheless, energy usage is ever in demand particularly given the various technological advances that rely on electrical power for operating.

The automatic reduction of energy requirements in buildings has received a lot of attention recently [3] and early attempts included automatic regulation of light or heating in home automation. However, these approaches were deemed inappropriate due to improper reaction to the expectations of the users; i.e., the occupants [4]. The majority of recent studies has shown the importance of putting the users in the energy-saving loop while ensuring their comfort [5]. Occupants' behavior has a major influence on building energy consumption [6–9]. Hence, [10] introduced methods for modeling occupant behavior and quantifying its impact on building

M. Amayri (✉)
CNRS, Grenoble, France
e-mail: manar.amayri@grenoble-inp.fr

S. Ploix
G-SCOP lab, Grenoble Institute of Technology, University of Grenoble Alps, CNRS UMR5272,
Grenoble, France
e-mail: stephane.ploix@grenoble-inp.fr

S. Ali · N. Bouguila
CIISE, Concordia University, Montreal, QC, Canada
e-mail: nizar.bouguila@concordia.ca

energy use. The major themes include advancements in analytic data collection techniques, modeling methods, and applications that provide insights into behavior-related energy saving's potential and impact [11]. There is a large gap between the predicted energy demand and the actual consumption, once the building is in use [12]. According to [13], occupants' behaviors account for significant uncertainty in building energy use. One cause could be that occupant behavior might not fit with the energy concept and thus cause counterproductive effects [14]. Occupants have influence due to their presence and activities in the building and due to their control actions, which aim to improve indoor environmental conditions (thermal, air quality, light, noise). Consequently, the weight of the user behavior on the energy balance of a building increases [15]. Indeed, several studies suggest huge energy savings in buildings just by detecting occupancy (presence/absence) as shown, for instance, in [10] where motion sensors and magnetic door switches are used to detect occupancy in offices and HVAC (Heating, Ventilation, and air-conditioning) control, thereby estimating potential energy savings from 10–15%. Similarly, [16] focus on how to estimate the number of occupants in a room by processing CO₂ concentration, temperature, and HVAC actuation levels in order to identify a dynamic model. Additionally, there is a lot of potential for energy savings and increasing occupants' comfort by detecting activities and this motivates to carry forward the activity recognition task [17]. Methods investigated for finding occupancy using common sensors vary from basic single feature classifiers that distinguish between two classes (presence and absence) [17, 18] to multi-sensor multi-feature models [16, 19–22]. A primary approach, which is prevalent in many commercial buildings, is to use passive infrared (PIR) sensors for occupancy estimation. However, motion detectors fail to detect presence when occupants remain relatively still. This is quite common during activities like regular deskwork. Furthermore, drifts of warm or cold air on objects can be interpreted as motion leading to false positive detection. This makes the use of PIR sensors alone, less attractive for occupancy counting purposes. Fusion of PIR sensor data with other sensors can be useful as discussed in [10]. As such, motion sensors are usually paired with magnetic reed switches for occupancy detection in order to increase the efficiency of HVAC systems in smart buildings in a simple and non-intrusive manner. Acoustic sensors may also be used [23]. However, environment audio signals may cause many false positives when no support from other sensors is available. The use of pressure, PIR, and acoustic sensors to detect occupancy in single desk offices has been discussed in [24]. Further tagging of activities is based on this knowledge, where a pressure sensor detects chairs occupancy with the offices filmed and then the footage is used to manually classify the activities of people over time.

Smart buildings related tasks in general and activity recognition in particular have been widely approached using classic optimization models (e.g., meta-heuristics, linear programming, dynamic programming, etc.). Unfortunately, these approaches do not take full advantage of the large-scale data generated by smart buildings settings. In order to extract and exploit the knowledge hidden in these data, recent trend and efforts in smart building applications have been based on data mining and machine learning techniques [25]. The main goal is to build specific models from the

available data with respect to the task to tackle. A typical data mining and machine learning framework is generally based on the following steps. The first step is data collection from the available sensors. The second step concerns preprocessing (e.g., data cleaning, data enrichment, normalization, feature selection and/or extraction, outliers rejection, etc.) the collected data. Finally, a model learning step is performed in which a machine learning technique is applied.

Different learning approaches have been deployed in the past in smart building applications. In [26], for instance, hidden Markov models (HMMs) [27] have been used for estimating occupancy using a wireless ambient sensing system, CO₂ sensors, and a wired camera network in order to establish actual occupancy levels. The large variance in the energy consumption was found to be primarily due to the operating mode; occupants that are elected to run their AC for longer durations, at lower set points and/or throughout a larger space, consumed more energy than occupants that did not [28]. Consequently, energy reduction methods must encompass a combination of technological development, building physics, and occupants' behavior to achieve the desired performance [29]. As such, numerous studies have developed control systems and modeling methodologies to better assist occupants to play active roles in buildings. In [30], a supervised learning approach is investigated. It initially determines the common sensors to be used to estimate and classify the approximate number of people (within a range) in a room and their activities. Means to estimate occupancy include motion detection, power consumption, CO₂ concentration sensors, microphone, or door/window positions. The most useful measurements in calculating the information gains when added to the classification algorithm are then determined. Next, estimation that relies on decision tree and random forest learning algorithms is performed. The reason behind the choice of the algorithms is that they yield decision rules readable by humans, which correspond to nested if-then-else rules, where thresholds can be adjusted depending on the considered living areas. One office has been used for testing and two video cameras have been used in this approach. This highly limits the implementation of the application because of the privacy issues.

Studying occupants' activity and behavior is a key for building adaptation and energy saving, thus not limited to occupancy detection and estimation only [31–33]. The primary motivation behind studies of activity recognition is to contribute to buildings, while a comprehensive model can improve the energy performance of a building. This has been studied by previous research in this area, and large savings can be obtained with activity aware building energy management system. Such building energy management system can also warn users about activities or behaviors that adversely affect energy savings of the building. This induces an energy aware behavior that can add one-third to a building's designed energy performance [17]. Thus, the goal of this chapter is to provide a review on machine learning approaches related to activities recognition in smart buildings. Furthermore, it serves to facilitate the definitions and introduction of machine learning techniques to domain beginners and practitioners alike. Moreover, the chapter also sheds light on the various advancements made in activity recognition in smart homes using machine learning, presenting the first survey of such methods, to

the best of our knowledge. Several machine learning models have been deployed over the years for activity recognition [34–36]. The process generally involves learning activity models from training data. The model learns to recognize patterns that differentiate various classes in the training data and apply this knowledge for the prediction and classification over the test data. This allows the actualization of a solution without necessarily providing domain specific knowledge. Since the problem emanates from pattern recognition or data analysis, such methods are termed data-driven. [37] identify such data-driven approaches and categorize them into generative, discriminative, and heuristic-based modeling:

1. Generative modeling: uses training data samples to form a description of the complete input space. Probabilistic models like Bayesian networks, Gaussian mixture models, and HMMs fall under this category. The underlying assumption of this model is that the training samples are representative of the entire input space/distribution and thus enough data must be available to learn the complete probabilistic representation.
2. Discriminative modeling: has the primary objective of finding a decision boundary or boundaries, rather than representing the entire input space. A basic example of this model is K-nearest neighbor (KNN) classifier, where a test point is assigned to a cluster that is at a minimum distance (the notion of distance may vary accordingly) to it. Similarly, but better performing algorithms in the same category, are decision trees and SVMs [38].
3. Heuristic-based modeling: uses a combination of both generative and discriminative models along with some heuristic information [39].

It is noteworthy that other approaches that take advantage of both generative and discriminative learning simultaneously, called hybrid generative discriminative approaches, have been proposed recently in the literature [40–43]. When training data (i.e., labeled data where the output's correct value for each instance is known) are considered, the learning approach is called supervised. Classification and regression are typical examples of supervised learning tasks. Using a set of training data grouped into classes, the goal of classification is to build a classifier to predict to which class a new observation should be assigned. Examples of classification approaches include support vector machine (SVM), decision trees, random forests, artificial neural networks, and K-nearest neighbors. Regression, on the other hand, is related to predicting a numerical value using a function built by relating outputs to inputs. Examples of regression approaches include linear regression and support vector regression. In many cases, however, the data are unlabeled and need the deployment of unsupervised learning technique to infer possible regularities (e.g., clusters) in the input space. Clustering (partitional or hierarchical) is the main example of unsupervised learning and consists of grouping observations such that intraclass and interclass similarities are maximized and minimized, respectively [44]. Partitional approaches include both centroid-based (ex. K-Means) and density-based (ex. DBSCAN) clustering. Hierarchical approaches include both divisive (i.e., top down) and agglomerative (i.e., bottom up) approaches. A compromise between supervised and unsupervised learning, called semi-supervised learning, allows to consider labeled data jointly with unlabeled data. An example of semi-supervised

learning techniques is active learning which necessitates an interaction with the user to get the desired outputs for new test data. In order to avoid collecting data from scratch and disturbing the daily life of users some activity recognition approaches have been based on transfer learning. The main idea consists of transferring learned knowledge as much as possible from an existing environment, the so-called source domain, to a new target one (i.e., the environment where knowledge is applied) to reduce data collection effort. It is noteworthy that in transfer learning, feature sets, label sets as well as learning tasks in both source and target domains datasets can be different. Transfer learning approaches can be roughly classified into three groups of approaches: instance-, feature-, and parameter-based transfer techniques.

The rest of this chapter is organized as follows: Sect. 2 describes the machine learning algorithms and reviews the relevant papers in the literature pertinent to the topic at hand, Sect. 3 presents an extensive case study, and finally Sect. 4 concludes the chapter.

2 Activity Recognition in Smart Buildings

In this section, we overview the main families of approaches that have been deployed for activity (e.g., cooking, sleeping, eating, etc.) recognition in smart buildings: classification, regression, and clustering. The first two are often referred to under the umbrella of supervised learning while the latter is an unsupervised learning method. These form the two main branches of machine learning. Other derivatives and hybrid categories such as semi-supervised learning [45] and the popular deep learning methods [46] have been researched extensively recently. However, they are usually founded on one of the two main categories or even combines both of the approaches. It is noteworthy to mention that when deep learning techniques studies arise in the literature, we list them as part of the supervised learning approach.

Supervised learning refers to methodologies whereby input data has explicit labels for each of its entries or objects, depending on the nature of the pertaining data. Such data is then split into training and testing sets that are used for the learning of the parameters of the desired algorithm. Specifically, classification is presented in Sect. 2.1 and regression in Sect. 2.2. On the other hand, unsupervised learning has to be carried out without the availability of labels for the data at hand. We also review the relevant literature of the latter method applied for activity recognition in smart buildings, as appropriate. Clustering is detailed in Sect. 2.3. A complete list of the papers with the respective algorithm(s) used as well as other miscellaneous details is described in Sect. 2.4.

2.1 Classification

This section is split into two subsections whereby Sect. 2.1.1 presents general classification approaches for activity recognition and Sect. 2.1.2 expands on HMMs and their utilization in the field.

2.1.1 General Classification Approaches

Given a set of data with discrete labels or classes that may be used for training, *Classification* then refers to the correct identification of the label or class that testing data falls under. Mathematically, classification is a mapping between input data x and output label y such that:

$$y = g(x|\theta) \quad (1)$$

where $g()$ represents the classification function or algorithm, and θ is its respective parameters. The function uses the training data to approximate the parameters. The closer the approximation to the true parameters, the better the fit and hence the performance of the classification algorithm. Thus, $g()$ can also be viewed as a separator between the data points of the various classes or labels in a problem.

This approach has been critical in developing various activity recognition approaches in smart buildings. For example, [47] use ontological modeling and semantic reasoning for a real-time multisource sensor data based activity recognition system in smart homes. The algorithm first converts detected sensor activation corresponding properties into context ontologies. This constructs an activity description and then equivalency and subsumption reasoning are performed for activity recognition. Finally, semantic retrieval is used for obtaining the set of atomic activity concepts.

Hu et al. [48] present a new classification algorithm based on feature incremental random forests. Random forests are another classification algorithm that may be utilized for activity recognition. They are based on decision trees whereby the overfitting is addressed by reporting the final classification result as the mode of the various individual trees. Indeed, a decision tree approach is used for the real-time smart watch system presented in [49] for activity recognition. Incremental learning, on the other hand, refers to updating the existing model dynamically with new data or sensors instead of retraining the model from scratch and disposing of the existing one.

It is sometimes referred to by online learning in the machine learning community [50, 51]. This proposed method [48] has been tested on three different datasets and reportedly consistently outperformed other incremental learning methods. Similarly, [52] also investigate a new methodology to incorporate incremental learning for dynamic activity recognition using random forests. However, the latter is comparable to the performance of batched random forests and extremely randomized trees.

Batch learning is the opposing concept to online learning and refers to retraining the entire model as new data or data sources, such as new sensors in the case of activity recognition, become available. Online or incremental learning is usually used because it saves time and resources as well as enables real time processing for real world applications.

Gu et al. [53] introduce a classification approach based on emerging patterns that defines significant changes between different activity classes. Hence, it has the advantage of independence from the dataset used for training given that it identifies the underlying sequential patterns of an activity regardless of whether it is interleaved or concurrent. This brings us to two other important definitions in action recognition: *Concurrent* activities refer to ones in which each of the activity can be broken down into multiple ones that are carried out at the same time or simultaneously such as eating or walking. *Interleaved* activities refer to simple activities such as a wave of the hand or sleeping.

A three phased activity recognition method is proposed in [54]. Classification of the activities is carried out by four different machine learning models: random forests, K-nearest neighbors, support vector machines, and decision trees. In normal activity detection, the four models perform comparably, while the random forest approach outperforms all others in interleaved activity recognition. Support vector machine are also used for activity recognition in smart homes in [55].

Multiple classification algorithms are studied in [56] for activity recognition. These include decision tables, decision trees (C4.5), K-nearest neighbors, support vector machines, and naive Bayes. Interestingly, meta classifiers are also compared for designing the optimum classifier for the problem. These include boosting, bagging, and plurality voting. This represents the first investigation carried out to find whether combining classifiers trained on accelerometer features would result in an improved result, as claimed by the authors. Data was collected for eight different activities carried out by two individuals over different days in multiple setups and with no noise filtering. The activities were standing, walking, running, climbing up stairs, climbing down stairs, sit-ups, vacuuming, and brushing teeth. Gradient boosting, K-nearest neighbor, linear discriminant analysis, and random forests are also researched for activity recognition in smart homes in [57] as well as kernel Fisher discriminant analysis and extreme learning machine in [58].

All in all, plurality voting was found to be the optimum implementation with consistent performance across different setups [56]. It is noteworthy to mention that an accelerometer is a well-researched sensor for activity recognition. Indeed, the utility of even simple sensors has proven effectiveness such as in [59], even with an elementary classifier such as the naive Bayes [60] or with deep learning techniques such as convolutional neural networks and long short-term memory [61]. For instance, [62] use accelerometer data from 20 individuals each with five different accelerometers and a decision tree classifier setup. The results suggest that the use of multiple accelerometers improves recognition.

Long short-term memory and convolutional neural networks are also used in [63], while only the latter is deployed in [64] and compared to the K-nearest neighbor and support vector machine methods. Other deep learning techniques such

as recurrent neural networks are also studied in [65] (with support vector regression) and in [66] (with support vector machine, naive Bayes, and logistic recognition).

2.1.2 Hidden Markov Models

HMMs are one of the most popular methods used in the field due to the sequential nature of the problem [67, 68]. An HMM is a well-received double stochastic model that uses a compact set of features to extract underlying statistics [69]. Its structure is formed primarily from a Markov chain of latent variables with each corresponding to the conditioned observation. A Markov chain is one of the least complicated ways to model sequential patterns in time series data. It allows us to maintain generality while relaxing the independent identically distributed assumption [70].

Mathematically, an HMM is characterized by an underlying stochastic process with K hidden states that form a Markov chain. Each of the states is governed by an initial probability π , and the transition between the states at time t can be visualized with a transition matrix $B = \{b_{i'i'} = P(s_t = i' | s_{t-1} = i)\}$. In each state s_t , an observation is emitted corresponding to its distribution which may be discrete or continuous. This is the observable stochastic process set (Fig. 1).

The emission matrix of the discrete observations can be denoted by $\Xi = \{\xi_{it}(m) = P(X_t = \xi_m | s_t = i)\}$ where $[m, t, i] \in [1, M] \times [1, T] \times [1, K]$, and the set of all possible discrete observations $\xi = \{\xi_1, \dots, \xi_m, \dots, \xi_M\}$. On the other hand, the respective parameters of a probability distribution define the observation emission for a continuous observed symbol sequence. The Gaussian distribution is most commonly used which is defined by its mean and covariance matrix $\kappa = (\mu, \Sigma)$ [71–73]. Consequently, a mixing matrix must be defined $C = \{c_{ij} = P(m_t = j | s_t = i)\}$ in the case of continuous HMM emission probability distribution where $j \in [1, M]$ such that M is the number of mixture components in set $L = \{m_1, \dots, m_M\}$. Hence, a discrete or continuous HMM may be defined with the following respective parameters $\lambda = \{B, \Xi, \pi\}$ or $\{B, C, \kappa, \pi\}$.

For the thorough explanation of the HMM algorithms to follow, we also introduce another visualization that depicts the graphical directed HMM structure as shown

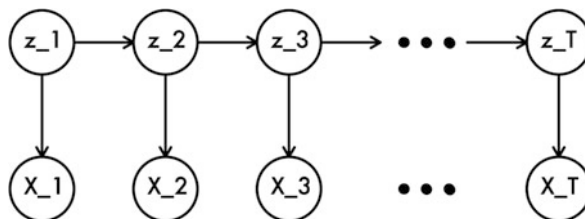


Fig. 1 A typical hidden Markov chain structure representation of a time series where z_1 denotes the first hidden state z_1 and X_1 denotes the corresponding observed state X_1 . This is shown accordingly for a time series of length T

in Fig. 2. Figure 3 shows transitions then when they become trellis or lattice. Indeed, Rabiner first introduces the three classical problems of HMMs in [71] as: (1) evaluation or likelihood, (2) estimation or decoding, and (3) training or learning. These are described as follows:

1. The evaluation problem is mainly concerned with computing the probability that a particular sequential or time series data was generated by the HMM model, given both the observation sequence and the model. Mathematically, the primary objective is computing the probability $P(X|\lambda)$ of the observation sequence $X = X_1, X_2, \dots, X_T$ with length T given an HMM model λ .
2. The decoding problem finds the optimum state sequence path $I = i_1, i_2, \dots, i_T$ for an observation sequence X . This is mathematically $s^* = \operatorname{argmax}_s P(s|X, \lambda)$.
3. The learning problem refers to building an HMM model through finding or “learning” the right parameters to describe a particular set of observations. Formally, this is performed with maximizing the probability $P(X|\lambda)$ of the set of observation sequence X given the set of parameters determined λ . Mathematically, this is $\lambda^* = \operatorname{argmax}_\lambda P(X|\lambda)$.

In the following discussion, we present the respective solutions for each of the HMM problems. We assume discrete emission observations. However, it is straightforward to extend these solutions to the HMM of continuous emission distributions given their parameters and mixing matrix. We also briefly recall the two conditional independence assumptions that allow for the tractability of the HMM algorithms [74]:

1. Given the $(t - 1)$ st hidden variable, the t th hidden variable is independent of all other previous variables such that:

$$P(s_t | s_{t-1}, X_{t-1}, \dots, s_1, X_1) = P(s_t | s_{t-1}) \tag{2}$$

Fig. 2 An HMM transition diagram with three states

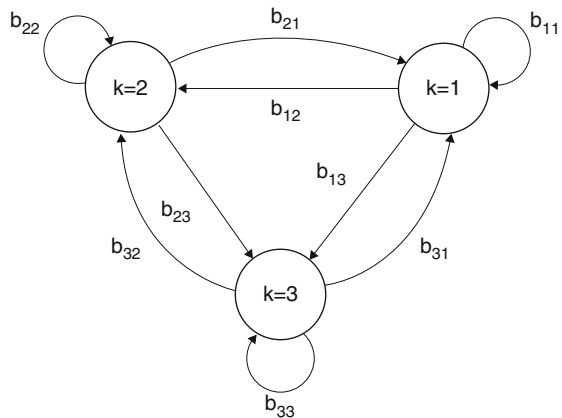
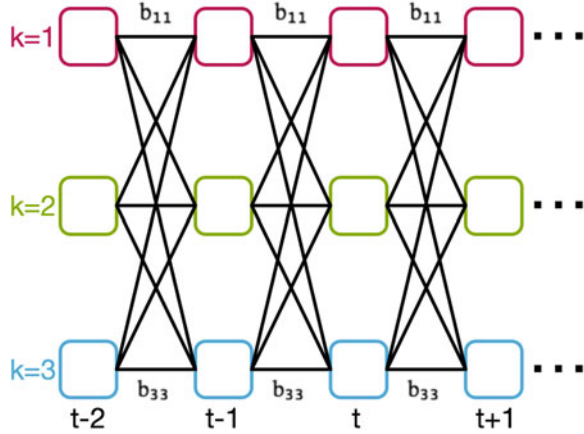


Fig. 3 Lattice or trellis HMM structure which is a representation of the hidden states



2. Given the t th hidden variable, the t th observation is independent of other variables such that:

$$\begin{aligned}
 &P(X_t|s_T, X_T, s_{T-1}, X_{T-1}, \dots, s_{t+1}, X_{t+1}, s_t, s_{t-1}, X_{t-1}, \dots, s_1, X_1) \\
 &= P(X_t|s_t)
 \end{aligned}
 \tag{3}$$

The first problem we address is the evaluation problem.

The forward algorithm calculates the probability of being in state s_i at time t after the corresponding partial observation sequence given the HMM model λ . This defines the forward variable $\rho_t(i) = P(X_1, X_2, \dots, X_t, i_t = s_i|\lambda)$ which is solved recursively as follows:

1. initiate the forward probabilities with the joint probability of state s_i and the initial observation X_1 : $\rho_1(i) = \pi_i \Xi_i(X_1), 1 \leq i \leq K$;
2. calculate how state $q_{i'}$ is reached at time $t + 1$ from the K possible states $s_i, i = 1, 2, \dots, K$ at time t and sum the product over all the K possible states:

$$\rho_{t+1}(j) = \left[\sum_{i=1}^K \rho_t(i) b_{ij} \right] \Xi_j(X_{t+1}) \text{ for } t = 1, 2, \dots, T - 1, 1 \leq j \leq K$$

3. Finally, compute $P(X|\lambda) = \sum_{i=1}^K \rho_T(i)$.

The forward algorithm has a computational complexity of K^2T which is considerably less than a naive direct calculation approach. A graphical depiction of the forward algorithm can be observed in Fig. 4.

Next, the Viterbi algorithm aims to find the most likely progression of states that generated a given observation sequence in a certain HMM. Hence, it offers the solution to the decoding problem. This involves choosing the most likely states at each time t individually. Hence, the expected number of correct separate states is maximized. This is illustrated in Fig. 5. To perform this algorithm, we need to define the following:

Fig. 4 Graphical representation of the evaluation of the ρ variable of the forward algorithm in an HMM lattice fragment

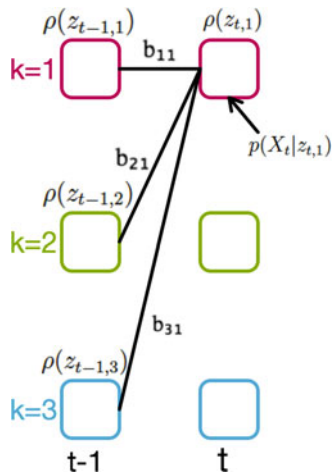
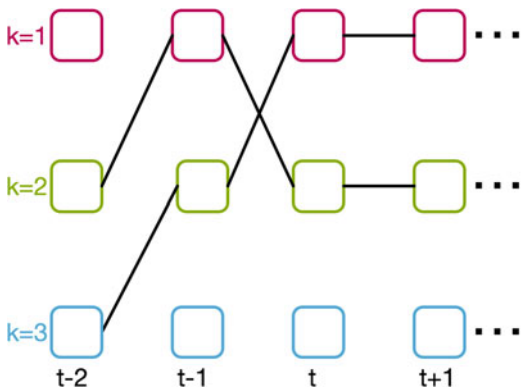


Fig. 5 Graphical representation of two probable pathways in an HMM lattice fragment. The objective of the Viterbi algorithm is to find the most likely one



$$\gamma_t(i) = P(i_t = s_i | X, \lambda) = \frac{\rho_t(i)\theta_t(i)}{p(X|\lambda)} \tag{4}$$

where $\gamma_t(i)$ is the probability of being in state s_i at time t given the observation sequence X and the HMM λ .

The main steps of the Viterbi algorithm can then be summarized as:

1. Initialization

$$\delta_1(i) = \pi_i \Xi_i(X_1), 1 \leq i \leq K \tag{5}$$

$$\psi_1(i) = 0 \tag{6}$$

2. Recursion

$$\text{For } 2 \leq t \leq T, 1 \leq i' \leq K \tag{7}$$

$$\delta_t(i') = \max_{1 \leq i \leq K} [\delta_{t-1}(i) b_{ii'}] \Xi_{i'}(X_t) \tag{8}$$

$$\psi_t(i') = \operatorname{argmax}_{1 \leq i \leq K} [\delta_{t-1}(i) b_{ii'}] \tag{9}$$

3. Termination

$$P^* = \max_{1 \leq i \leq K} [\delta_T(i)] i_T^* = \operatorname{argmax}_{1 \leq i \leq K} [\delta_T(i)] \tag{10}$$

4. State sequence path backtracking

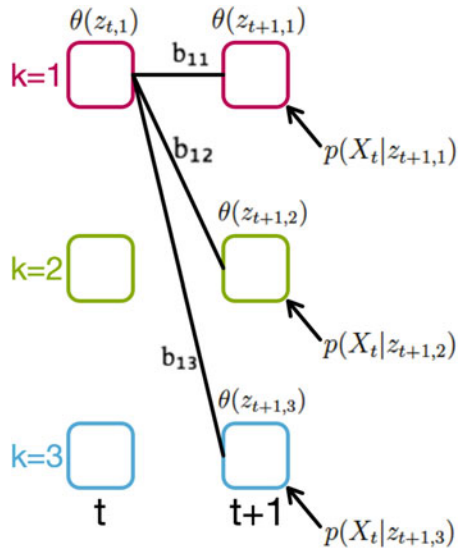
$$i_t^* = \psi_{t+1}(i_{t+1}^*), \text{ for } t = T - 1, T - 2, \dots, 1 \tag{11}$$

Finally, and in order to address the third HMM problem, we first describe another important algorithm. Similar to the forward algorithm, but now computing the tail probability of the partial observation from $t + 1$ to the end, given that we are starting at state s_i at time t and model λ , is the backward algorithm. This has the variable $\theta_t(i) = P(X_{t+1}, X_{t+2}, \dots, X_T, i_t = s_i | \lambda)$ and is solved as follows:

1. Compute an arbitrary initialization $\theta_T(i) = 1, 1 \leq i \leq K$;
2. $\theta_t(i) = \sum_{i'=1}^K b_{ii'} \Xi_{i'}(X_{t+1})$ for $t = T - 1, T - 2, \dots, 1, 1 \leq i \leq K$

Figure 6 depicts the computation process of the backward algorithm in an HMM lattice structure. Together with the forward algorithm, this forms the forward-backward algorithm through consequent iteration. In the context of HMMs, the forward-backward algorithm is of extreme importance and is also known as the Baum Welch algorithm [71]. The Baum Welch algorithm is traditionally used to solve the estimation problem of HMMs. This iterative algorithm requires an initial

Fig. 6 Graphical representation of the evaluation of the θ variable of the backward algorithm in an HMM lattice fragment



random clustering of the data, is guaranteed to converge to more compact clusters at every step, and stops when the log-likelihood ratios no longer show significant changes [75].

In order to apply the Baum Welch algorithm, we must define

$$\varphi_t(i, i') = P(i_t = s_i, i_{t+1} = s_{i'} | X, \lambda) = \frac{\rho_t(i) b_{ii'} \Xi_{i'}(X_{t+1}) \theta_{t+1}(i')}{p(X|\lambda)} \quad (12)$$

where $\varphi_t(i, i')$ is the probability of path being in state s_i at time t and then transitioning at time $t + 1$ with $b_{ii'}$ to state $s_{i'}$, given λ and X . $\rho_t(i)$ then considers the first observations ending at state s_i at time t , $\theta_{t+1}(i')$ the rest of the observation sequence, and $b_{ii'} \Xi_{i'}(X_{t+1})$ the transition to state $s_{i'}$ with observation X_{t+1} at time $t + 1$. Hence, $\gamma_t(i)$ may also be expressed as:

$$\gamma_t(i) = \sum_{i'=1}^K \varphi_t(i, i') \quad (13)$$

whereby $\sum_{t=1}^{T-1} \varphi_t(i, i')$ is the expected number of transitions made from s_i to $s_{i'}$ and $\sum_{t=1}^{T-1} \gamma_t(i)$ is the expected number of transitions made from s_i .

The general re-estimation formulas for the HMM parameters π , B , and Ξ are then:

1. $\bar{\pi}_i = \gamma_1(i), 1 \leq i \leq K$
2. $\bar{b}_{ii'} = \sum_{t=1}^{T-1} \varphi_t(i, i') / \sum_{t=1}^{T-1} \gamma_t(i)$
3. $\bar{\Xi}_{i'}(k) = \sum_{\substack{t=1 \\ X_t=k}}^T \gamma_t(i') / \sum_{t=1}^T \gamma_t(i')$

Oliver et al. [76] utilize an extension, layered HMMs to detect various activities like deskwork, phone conversations, presence, etc. The layered structure of their model makes it feasible to decouple different levels of analysis for training and inference. Each level in the hierarchy can be trained independently, with different feature vectors and time granularity. Once the system has been trained, inference can be carried out at any level of the hierarchy. One benefit of such a model is that each layer can be trained individually in isolation, and therefore the lowest layer that is most sensitive to environment noises and flickers can be retrained without touching the upper layers. HMMs and conditional random field (CRF) have been used in [77] to recognize seven different activities (leave house, toileting, showering, sleeping, preparing breakfast, preparing dinner, preparing a beverage) in a home setting. An HMM-based approach to recognize independent and joint activities among multiple residents in smart environments has been proposed in [78].

Nonetheless, HMMs suffer from some drawbacks that [79] aimed to overcome by introducing a new variant; namely, Switching Hidden Semi-Markov Model. This model supplements HMMs with a hierarchical structure to benefit from the natural hierarchy depicted by humans in activities. It also incorporates explicit

state duration though the semi-HMM to address the violation of the Markovian assumption when the duration of the state is no longer geometric. The system reportedly outperforms both a traditional HMM as well as a hierarchical one.

2.2 Regression

Regression is often viewed as a variant of classification whereby the data or the variables at hand are, in contrast, of continuous nature. Specifically, *Regression* is the prediction of continuous labels given a set of labeled training data. It is also sometimes referred to as prediction and is closely related to classification. As a matter of fact, Eq. (1) can also be used to represent regression whereby $g()$ represents the regression function that is used to fit the data x to find out y . Notice that while the first assumption of the best function is a linear approximation, it is not always the case. Indeed, higher order approximations are usually used to better estimate the true distribution of the training data.

Given its nature of continuous predictions, it is not often used in the area of activity recognition due to the discrete nature of the data. Nonetheless, regression, in particular linear regression, remains one of the most traditional machine learning methods and the problem may be posed within a continuous framework for its use. For example, linear regression is used for classification of human activities in smart homes and inspires a new regression-tree-based activity forecasting algorithm in [80].

However, while linear regression is a powerful technique, it is not necessarily the most suitable in all cases. The best approach machine learning approach to be used is always dependent on the nature of the data itself. This is investigated in [81] whereby the authors argue that prior statistical analysis of the problem is imperative for choosing the best machine learning algorithm. They compare the use of random forests and linear regression finding out that the prior outperforms the latter due to the nonlinear nature of the data.

2.3 Clustering

A significant problem when tackling the activity recognition problem using supervised learning approaches is collecting ground truth information. Indeed, the large variety of possible activities makes their recognition in a supervised way challenging.

Since no labels are available in clustering, this presents an added challenge in finding homogeneous groups within the input data. The objective in such algorithms can be straightforwardly defined as: Finding homogeneous groups or clusters in data such that the intra-distance between the data points is minimized and the inter-

distance between the data homogeneous groups or clusters is maximized is known as *Clustering*.

The most famous clustering approach is using mixture models. Consider a set of N observation vectors $\mathcal{X} = \{\vec{\mathcal{X}}_1, \dots, \vec{\mathcal{X}}_N\}$ represented in D -dimensional space where each vector $\vec{\mathcal{X}}_\rho = (\mathcal{X}_{\rho 1}, \dots, \mathcal{X}_{\rho D})$. If we assume that each vector $\vec{\mathcal{X}}_\rho$ is generated from a finite mixture model with ϖ components, then the likelihood of the data is defined as:

$$p(\vec{\mathcal{X}}_\rho | \kappa, \Lambda) = \sum_{\zeta=1}^{\varpi} \kappa_\zeta p(\vec{\mathcal{X}}_\rho | \Lambda_\zeta) \quad (14)$$

where $p(\vec{\mathcal{X}}_\rho | \Lambda_\zeta)$ is the mixture distribution at hand that is used to statistically model the observations or data \mathcal{X} , Λ_ζ is the respective set of component parameters for the distribution, and κ_ζ is the mixing coefficient of the mixture component ζ with $\kappa = (\kappa_1, \dots, \kappa_\varpi)$. The mixing coefficients vector follow constraints of positivity and unit summation resultant on the κ . Each of the data observation vectors $\vec{\mathcal{X}}_\rho$ is assigned to all of the mixture components with a responsibility or posterior probability $p(\zeta | \vec{\mathcal{X}}_\rho) \propto \kappa_\zeta p(\vec{\mathcal{X}}_\rho | \Lambda_\zeta)$.

Clustering represents an attractive solution as it is easy to obtain unlabeled samples from routine experiments; they do not require human effort. This is also applicable for the problem at hand though more research can be invested in this particular area. For example, k-means algorithm is applied in [82] to cluster sensor readings collected from smart homes for activity recognition. Classification of non-separated activities within each cluster is then carried out by K-nearest neighbor classification approach. This also represents a system where a hybrid approach improves the overall classifier performance.

2.4 Miscellaneous

So far, we have presented papers in the literature that address the problem of activity recognition in smart buildings using supervised and unsupervised learning techniques. A summary of these papers can be observed in Table 1. On the other hand, semi-supervised learning techniques applied in [84] and [85] represent another learning approach aiming to address activity recognition issue. It exploits unlabeled data in order to improve model performance. For example, [83] introduce a method for human activity recognition that benefits from the structure and sequential properties of the training and testing data. In the training phase, a fraction of data labels has been obtained and used in a semi-supervised method for recognizing the user's activities. Label propagation has been used on a K-nearest neighbor graph to calculate the probability of the unlabeled data in each class in the training phase. These probabilities have been used to train an HMM

Table 1 A list of the papers detailed in this chapter for activity recognition in smart buildings with the respective machine learning (ML) technique utilized and the algorithm(s) used

Paper	ML technique	Algorithm(s)
[47]	Classification	Ontology modeling and semantic reasoning
[48]	Classification	Feature incremental random forests
[49]	Classification	Decision trees
[50]	Classification	Support vector machines/naive Bayes/hidden Markov models/conditional random fields
[52]	Classification	Incremental learning random forests
[53]	Classification	Emerging pattern technique
[78]	Classification	Hidden Markov models
[76]	Classification	Layered hidden Markov models
[77]	Classification	Hidden Markov models/conditional random field
[79]	Classification	Switching hidden semi-Markov model
[56]	Classification	Decision tables/decision trees (C4.5)/K-nearest neighbors/support vector machines/naive Bayes (with meta classifiers setup)
[60]	Classification	Naive Bayes
[62]	Classification	Decision trees (multiple accelerometers)
[61]	Classification	Convolutional neural networks and long short-term memory
[80]	Regression	Linear regression/regression-tree-based activity forecasting algorithm
[81]	Classification/regression	Random forests/linear regression
[82]	Clustering/classification	k-Means algorithm/K-nearest neighbors
[83]	Semi-supervised learning	K-nearest neighbors/hidden Markov model
[54]	Classification	Random forests/K-nearest neighbors/support vector machines/decision trees
[65]	Regression/classification	Support vector regression/recurrent neural network
[66]	Classification	Support vector machine/naive Bayes/logistic recognition/ recurrent neural network
[55]	Classification	Support vector machine
[58]	Classification	Kernel Fisher discriminant analysis/extreme learning machine
[57]	Classification	Gradient boosting/K-nearest neighbor/linear discriminant analysis/ random forest
[63]	Classification	Convolutional neural networks and long short-term memory
[64]	Classification	Convolutional neural networks/K-nearest neighbor/support vector machine
[59]	Classification	Naive Bayes/K-nearest neighbor/support vector machine

in a way that each of its hidden states corresponds to one class of activity. Some semi-supervised approaches have been based on active learning, also. For instance, different active learning strategies have been investigated in [86]. In particular, a

dynamic k-means clustering approach has been proposed to discover unseen new activities spontaneously. These unseen activities are detected as outliers which make the clustering algorithm sensitive to the number of clusters that can increase at every iteration. The overall clustering error was recorded using an error function on the set of clusters defined as the sum of the Euclidean distances between the different data instances and the clusters centers. An objective function based on entropy is then defined to fetch the most informative data instances. The activities that were considered are cooking, sweeping, washing, and cleaning which were used for the passive learning. Three other activities, namely eating, sleeping, and talking, on the phone were left to the active learner to discover.

Some recent approaches have been based on transfer learning. In [87], for example, the authors proposed a feature-based approach to reuse learned knowledge from an original environment and tested it successfully to extract and transfer knowledge between two different smart home environments by considering only single-resident scenarios. The problem was formulated as classification task using SVM by matching the different features of the source and target environments. Two cases were considered. In the first one labeled datasets from both environments were supposed to be available. In the second one labeled data are available only in the source environment and the information from the target one is limited to sensor deployment considered as background knowledge.

Another issue refers to the features used. In any of the machine learning techniques, or any algorithm for that matter, the importance of extracted features to be used cannot be overstated. Indeed, some studies were carried out in [88, 89] to analyze the various features and their importance in activity recognition. This falls outside the scope of this chapter, but an interested reader is referred to the paper for further details.

Furthermore, in order to ensure the completeness of the activity recognition survey, it is noteworthy to mention that not all methods are dependent on machine learning techniques. For instance, [90–92] present other algorithms that do not fall under the scope of this survey. An interested reader is referred to [93] for a general reference on human action recognition.

3 Case Study

To evaluate the deployment of machine learning approaches in smart buildings in general and their potential in activity recognition, we present three recent methods for occupancy estimation that have been applied in an office H358 case study (see Chapter “Formalization of the Energy Management Problem and Related Issues”). Extensive work is currently conducted to apply these approaches for activity recognition. The proposed approaches are:

1. Estimating occupancy with a set of sensors, and possible manual labeling by an expert.
2. Estimating occupancy with a set of sensors without manual labeling but using a knowledge-based approach.
3. Estimating occupancy with a set of sensors with self-labeling by occupants (interactive learning).

These approaches depend mainly on collecting and analyzing data from non-intrusive sensors. The use of such sensors is based on the hypothesis that humans interact with their surroundings, i.e., performing some activities. It affects environmental conditions that can be in the form of CO₂ concentration, moisture, temperature rise, or sound. As it is mentioned in Chapter “Formalization of the Energy Management Problem and Related Issues”, different sensors exist in the H358 office, i.e., PIR motion sensors, CO₂ sensors, indoor air temperature, and relative humidity sensor, pressure sensors, acoustic sensors, ultrasonic sensors, power consumption sensors, in order to define the occupancy level.

To perform the task of finding the number of occupants, a link needs to be observed between the office context and the number of occupants in it. The office context can be described as a collection of state variables, $S_t = [s_1, s_2, \dots, s_n]_t$. This group of state variables S must characterize occupancy at each time step t .

A state variable can be presented as a feature, and therefore the features are displayed as a feature vector. Thus, the multidimensional space that includes all potential values of such a feature vector is the feature space. The underlying approach for the experiments is to formulate the classification problem as a mapping from a feature vector into a feature space that comprises several classes of occupancy. Therefore, the success of such an approach depends strongly on how useful (features which give maximum distinction between classes) the chosen features are. In this case, features are attributes from multiple sensors collected over a time interval. The selection of interval duration is highly context-dependent and has to be done according to the required granularity. The results presented here are based on an interval of $T_s = 30$ min (which has been referred here as one quantum).

From the large set of features discussed in [30], some of them may not be worth considering in order to achieve the target of occupancy classification. These features are the ones which, when added to the classification algorithm, make no difference to the overall output. In other words they are not useful enough for our purpose. For example, absolute humidity readings would be useless, as it is not representative of occupancy at all. Defining the most important features (sensors) is considered as a necessary study in a smart building application. It can give an essential conclusion for the required installation of the sensor in the buildings, which leads to minimizing the total cost.

Before any features are extracted for the training data, some basic preprocessing had to be done: application of an *outliers removal algorithm* and *interpolation* for non-existent data. The interpolation part is necessary for filling in missing values from the sensor data. Amayri et al. [30] concludes the most relevant features for the occupancy estimation problem in the office:

1. power consumption.
2. motion counter.
3. acoustic pressure recorded by a microphone.

These three features will be used in the three following experiments of applying machine learning techniques for occupancy estimation.

3.1 Estimating Occupancy with a Set of Sensors, and Possible Manual Labeling by an Expert

Let us start with the first experiment, where supervised learning has been deployed. Collecting the required training data has been done by counting occupancy manually using two video cameras in office H358. The average number of people visiting the office was registered every 30 min during the day.

Different supervised learning methods have been investigated (i.e., support vector machine, decision tree, random forest, linear regression). A *decision tree*-based classification approach has been selected as our prediction model because it provides human-readable results that can be analyzed and easily adapted. Providing decision rules is one crucial aspect from the energy point of view to generalize the model for another similar context.

Power consumption, motion counter, and acoustic pressure are the main features for building our model. Five occupancy levels have been chosen to generate decision trees due to the maximum number of occupants met while collecting the dataset.

3.2 Resulting Occupancy Estimators

From the collected data in the office H358, a training dataset covering 11 days from 04-May-2015 to 14-May-2015 has been used. Moreover, a validation dataset is collected over 4 days from 17-May-2015 to 21-May-2015. Figure 7 shows the result obtained from the decision tree and random forest, considering the three features. It leads to occupancy estimation with an accuracy of 81.7% and an average error of 0.26 person, while random forest accuracy is 84%, and the average error is 0.26 person (Table 2).

The above results indicate that using the decision tree and random forest rules give quite a reasonable estimation of occupancy. Because of the limitation of the need to have labeled training data when deploying supervised learning, unsupervised learning based on collecting knowledge and questioning will be discussed in the next section. It will help to facilitate and generalize the occupancy estimation process.

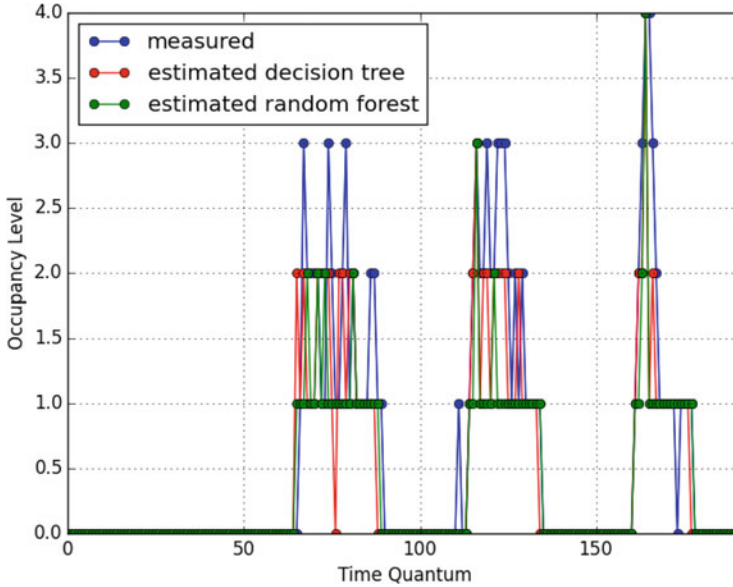


Fig. 7 Occupancy estimation from DT using three features

Table 2 Decision tree classification results after selecting main features

	Average error decision tree	Support
Class 1	0.02	132
Class 2	0.49	22
Class 3	0.88	23
Class 4	1.27	11
Class 5	1.75	4
Avg/total	0.26	192

3.3 Designing Estimators from Knowledge and Adjusting from Data

Similarly to the first approach, designing estimators from knowledge is based on sensor data and knowledge coming, respectively, from observations and questionnaires to build the estimation model. The proposed technique relies on a Bayesian Network (BN) algorithm to model human behavior with probabilistic cause-effect relations and states based on knowledge and questionnaire [94, 95].

The same case study of an office (H358) is considered as a simple and essential one-zone context with lots of sensors. Motion detection, power consumption, and acoustic pressure recorded by a microphone are used to feed this model. Collecting occupancy and activity feedbacks is very easy in the office context. Besides, there is a facility of questioning the occupants during design and validation periods of occupancy model. Unsupervised learning algorithms are used to solve problems

where the solution is not known. In this case, usually, the structure is derived by clustering the sensor data based on relationships among the variables. While in the case of collecting training period, it becomes similar to supervised learning methods with the difference in prediction techniques. For each feature, different levels have been considered. For example, the power consumption values discretize in three levels: low consumption, medium consumption, and high consumption, or L, M, and H, respectively. It gives a probability table with nine values. The probability table for power consumption has been defined by proposing different questions to the office occupants. For example: when occupants are arriving and leaving the office? What is the average time for using the laptop during the working hours? According to the user answers, the conditional probabilities are either calculated or filled directly in the tables. The same process can be repeated for the recorded signal from the microphone. At the same time, two different levels have been defined for the microphone low acoustic pressure and high acoustic pressure or L and H, respectively, see Fig. 8. Three occupancy levels have been considered to generate a Bayesian Network (BN): Low, Medium, and a High number of occupants. While the probabilities table for motion counter has been suggested according to the general knowledge for three different cases, low motion, medium motion, and high motion, or L, M, and H, respectively. Figure 8 shows the results obtained from the Bayesian network for three levels and three main features. Both actual and estimated occupancy profiles have been plotted in a graph with the number of occupants and time relations (quantum time was 30 min). The accuracy achieved from the Bayesian network was 91% (the number of correctly estimated points divided by the total number of points), and the average error was 0.08 persons. Table 3 represents the average error values for each class of estimation. While “support” indicates the number of events (sensor data each quantum time) in each class, and average support indicates the sum of all events in the three classes (Fig. 9).

Fig. 8 Bayesian network (BN) structure of an office H358

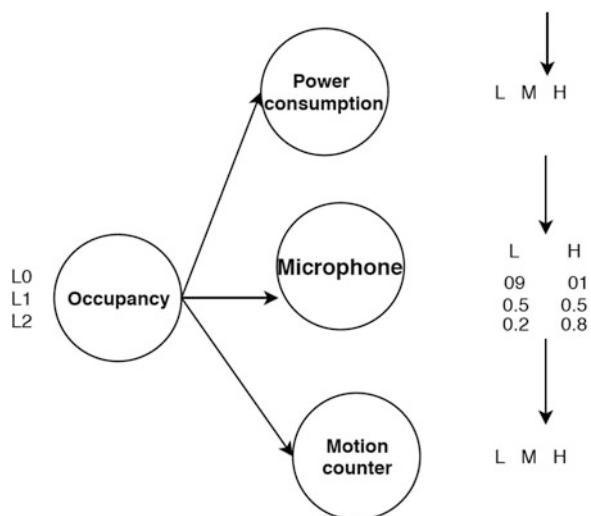


Table 3 Bayesian network estimation results

Classes	Accuracy-BN	Average error-BN	Support
Class 1	97%	0.001	400
Class 2	84%	0.2	170
Class 3	79%	0.5	54
Average	91%	0.09	624

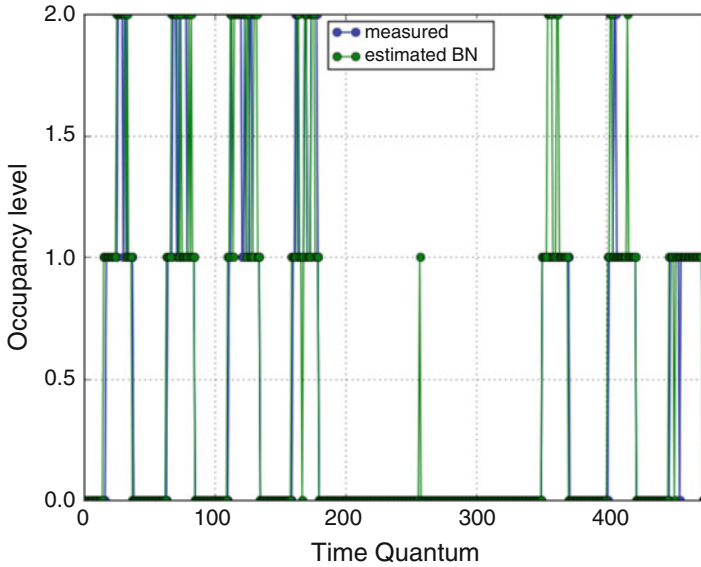


Fig. 9 Occupancy estimation from Bayesian network

Using the knowledge domain and questionnaire with data sensors in the unsupervised learning method is more flexible and open for different types of applications, with acceptable average errors for occupancy estimations. Besides, avoiding the use of video cameras has been achieved. This approach can be used widely in different contexts. Still, due to a few possibilities to validate the estimation model and poor performance in some testing period, a new innovative approach is proposed in the next section. It depends on estimating occupancy with a set of sensors, and self-labeling by occupants.

3.4 Designing Estimators from Interactive Learning

A novel way of supervised learning is analyzed to estimate the occupancy in a room where actual occupancy is interactively requested to occupants when it is the

most relevant to limit the number of interactions. Occupancy estimation algorithm relies on machine learning: it uses information gathered from occupants. In this section, an interactive technique has been investigated to solve the problem of getting the required labels used in the supervised method. In practical applications, the limitation arises due to the occupant's privacy issues. Accurately estimating occupancy with a set of sensors and self-labeling by interaction with occupants are the main goals of this section.

3.4.1 The Principle of Interactive Learning

Obtaining training data is a challenging task for smart home applications in general and activity recognition in particular. Some approaches have been proposed to involve the occupants to collect informative training data. An interesting approach called interactive learning has been proposed in [96]. Interactive learning is a process involving an exchange of information with the users to collect some essential data according to the problem context. In supervised learning methods, which are widely used in a lot of applications, the problem of the required target arises in the determination of the number of occupants, i.e., the labeling issue is usually tackled using video cameras. Utilizing a camera is generally not acceptable in many places to respect the privacy of occupants. Interactive learning is an extension of supervised learning that determines the occupancy by collecting the required labeling from the occupants themselves. The problem statement of occupancy estimation has been explained in [96].

Three rules are considered to determine whether an interaction space (*ask*) is potentially useful or not:

1. The density of the neighborhood: It is the number of existing records in the neighborhood of a potential *ask*. The neighborhood is defined by the infinite distance with a radius equal to one, because of the normalization. The record is a vector of features obtained in which values are obtained from the sensors. The neighborhood can be modified according to $\epsilon \in [0, 1]$.
2. The classifier estimation error in the neighborhood of the potential *ask* leads to the concept of neighborhood quality. If the classifier estimation error is very high for a record, this record is removed from the neighborhood because of the poor quality. $E_r \in [1, 2)$ typically is an error ratio that can be adjusted. However, a value smaller than 1 means a record is considered as good. Conversely, if E_r is big, equal to 2, for instance, it means you accept error twice as big as the average error. Theoretically, E_r belongs to $[0, \infty)$, but it is limited in our experiments to 2.
3. The minimum class weight: i.e., the minimum number of records for each class. The minimum class weight, $\text{weight}(\text{class } x) < C_w$, which can be adjusted according to the problem.

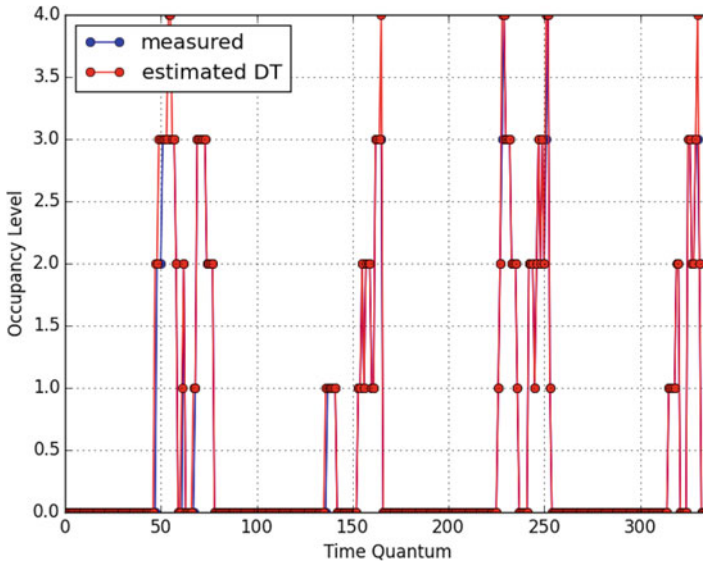


Fig. 10 Occupancy estimation from interactive learning

Table 4 Number of asks

Day	1	2	3	4	5	6	7	8	9	10
Number of <i>asks</i>	10	4	0	0	2	1	2	1	0	0

All the potential *asks* that satisfy the above three rules are asked to the occupants in order to become an additional record, possibly. The three previous rules have been checked with each new record. As a first validation, the occupant reaction has to be taken into account as a response probability whether the occupants answer or not. In a given context, the number of asks relies on the classifier used for estimation occupancy. To evaluate the interactive approach, we deploy the decision tree to compare it with the manual label approach. According to our study in [96, 97], Five occupancy levels to generate decision tree with an average error of 0.03 (see Fig. 10). Decision tree needs 21 *asks* for training data to build an acceptable estimator see the following Table 4.

Occupancy estimation using decision tree and interactive learning with an average error 0.03 person is more efficient than using decision tree and manually labeling from the video camera with an average error of 0.2 person. The precise answers to the questions can explain this improvement in occupancy estimation results. An occupant has replied to them during a training period of the decision tree. While in manually labeling from a video camera, average values of occupancy have been obtained, with some human mistakes during labeling. Probably the average error will decrease if the end-user does not feel concerned by the estimation process.

Table 5 Knowledge based vs manual labeling vs interactive learning occupancy estimation comparison

Case study office H358	Supervised learning	Interactive learning	Based knowledge learning
Average error	0.18	0.03	0.19
Requirements	Labels from video, keyboard	Frequent questions	Collecting knowledge from questioning and observation
Adaptation with high number of occupants	+	++	+/-

4 Conclusion

The Internet-of-Things (IoT) revolution has provided a variety of affordable sensors that new buildings are equipped with as well as data acquisition devices, and cloud storage. This has resulted in an unprecedented generation of raw data from sensors and smart meters. Many data mining approaches and machine learning techniques have been proposed to extract hidden knowledge from these data and then to build learning machines for a variety of applications and tasks. Activity recognition in smart buildings is one of the tasks that received a lot of attention due to its importance in energy management systems, for instance. The goal of this chapter was to review a variety of machine learning techniques that have been applied for activity recognition. Moreover, a case study and a methodology that concern occupancy estimation and that can be easily adopted for activity recognition have been presented and discussed. The results in this case study lead to the conclusion that the interactive learning approach is more efficient for occupancy estimation than the other methods taking into account the context. Two points can explain occupancy estimation improvement using interactive learning: firstly, the probability of making some human mistakes during manually labeling while using the video camera; secondly, the training period cannot be sufficient by missing some cases from the studied area. Using the ask technique considers all the events that occur when a new question is sent for each unique and different situation. This allows also to take into account the quality of the training data as deeply discussed in [98]. Interactive learning is the primary step to collect knowledge about the relations between user behavior and energy use. Moreover, its deployment allows involving occupants and increasing their awareness of energy systems. It depicts the future vision to develop energy systems, and it presents how much it is essential to put occupants in the energy process loop (Table 5).

Acknowledgments This work has been supported by the Fonds de Recherche du Québec—Nature et Technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC). This work benefits from the support of the INVOLVED ANR-14-CE22-0020-01 project of the French National Research Agency. It is also supported by the French National Research Agency in the framework of the EquipEx program AmiQual4Home ANR -11-EQPX-00 and the Investissements d’avenir program (ANR-15-IDEX-02) as the cross disciplinary program Eco-SESA.

References

1. T. Labeodan, W. Zeiler, G. Boxem, Y. Zhao, Occupancy measurement in commercial office buildings for demand-driven control applications - a survey and detection system evaluation. *Energy Build.* **93**, 303–314 (2015)
2. W. Khalid, I.A. Shah, Q. Mehfooz, N. Irshad, W. Mahmood, Reduction in building energy requirements by modern energy conservation techniques, in *2015 Power Generation System and Renewable Energy Technologies (PGSRET)*, June 2015, pp. 1–5
3. D.-M. Petroeanu, G. Carueaeu, N.L. Carueaeu, A. Parjan, A review of the recent developments in integrating machine learning models with sensor devices in the smart buildings sector with a view to attaining enhanced sensing, energy efficiency, and optimal building management. *Energies* **12**(24), 4745 (2019)
4. J.L. Crowley, J. Coutaz, An ecological view of smart home technologies, in *AMI 2015 - European Conference on Ambient Intelligence, Athens*, Nov 2015
5. A. De Paola, M. Ortolani, G. Lo Re, G. Anastasi, S.K. Das, Intelligent management systems for energy efficiency in buildings: a survey. *ACM Comput. Surv.* **47**(1), 1–38 (2014)
6. L. Jiang, D.-Y. Liu, B. Yang, Smart home research, in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, vol. 2 (2004), pp. 659–663
7. E. Soltanaghaei, K. Whitehouse, Walksense: classifying home occupancy states using walkway sensing, in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (2016), pp. 167–176
8. J. Scott, A. Bernheim Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, N. Villar, Preheat: controlling home heating using occupancy prediction, in *Proceedings of the 13th International Conference on Ubiquitous Computing* (Association for Computing Machinery, New York, 2011), pp. 281–290
9. C. Sarkar, S.N.A.U. Nambi, V. Prasad, iLTC: achieving individual comfort in shared spaces, in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks* (Junction Publishing, New York, 2016), pp. 65–76
10. Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, T. Weng, Occupancy-driven energy management for smart building automation, in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building* (Association for Computing Machinery, New York, 2010), pp. 1–6
11. K. Basu, L. Hawarah, N. Arghira, H. Joumaa, S. Ploix, A prediction system for home appliance usage. *Energy Build.* **67**, 668–679 (2013)
12. H.A. Aglan, Predictive model for CO₂ generation and decay in building envelopes. *J. Appl. Phys.* **93**(2), 1287–1290 (2003)
13. D. Yan, W. Oabrien, T. Hong, X. Feng, H.B. Gunay, F. Tahmasebi, A. Mahdavi, Occupant behavior modeling for building performance simulation: current state and future challenges. *Energy Build.* **107**, 264–278 (2015)
14. K. Schakib-Ekbatan, F.Z. Cakici, M. Schweiker, A. Wagner, Does the occupant behavior match the energy concept of the building? Analysis of a German naturally ventilated office building. *Build. Environ.* **84**, 142–150 (2015)
15. P. Hoes, J. Hensen, M. Loomans, B. de Vries, D. Bourgeois, User behavior in whole building simulation. *Energy Build.* **41**(3), 295–302 (2009)
16. A. Ebadat, G. Bottegal, D. Varagnolo, B. Wahlberg, K.H. Johansson, Estimation of building occupancy levels through environmental signals deconvolution, in *Proceedings of the 5th ACM Workshop on Embedded Systems for Energy-Efficient Buildings* (Association for Computing Machinery, New York, 2013), pp. 1–8
17. M. Milenkovic, O. Amft, An opportunistic activity-sensing approach to save energy in office buildings, in *Proceedings of the Fourth International Conference on Future Energy Systems* (Association for Computing Machinery, New York, 2013), pp. 247–258
18. H. Nguyen, M. Rahmanpour, N. Manouchehri, K. Maanicshah, M. Amayri, N. Bouguila, A statistical approach for unsupervised occupancy detection and estimation in smart buildings, in

- 2019 *IEEE International Smart Cities Conference, ISC2 2019, Casablanca, Morocco, 14–17 Oct 2019* (IEEE, Piscataway, 2019), pp. 414–419
19. D. Ankam, N. Bouguila, M. Amayri, Beta-Liouville regression and applications, in *6th International Conference on Control, Decision and Information Technologies, CoDIT 2019, Paris, France, 23–26 April 2019* (IEEE, Piscataway, 2019), pp. 1740–1745
 20. N. Zamzami, M. Amayri, N. Bouguila, S. Ploix, Online clustering for estimating occupancy in an office setting, in *28th IEEE International Symposium on Industrial Electronics, ISIE 2019, Vancouver, BC, Canada, 12–14 June 2019* (IEEE, Piscataway, 2019), pp. 2195–2200
 21. N. Manouchehri, J.S. Kalsi, M. Amayri, N. Bouguila, Finite two-dimensional beta mixture models: model selection and applications, in *28th IEEE International Symposium on Industrial Electronics, ISIE 2019, Vancouver, BC, Canada, 12–14 June 2019* (IEEE, Piscataway, 2019), pp. 1407–1412
 22. M. Amayri, S. Ploix, F. Najar, N. Bouguila, F. Wurtz, A statistical process control chart approach for occupancy estimation in smart buildings, in *IEEE Symposium Series on Computational Intelligence, SSCI 2019, Xiamen, China, 6–9 Dec 2019* (IEEE, Piscataway, 2019), pp. 1729–1734
 23. K. Padmanabh, A. Malikarjuna, S. Sen, S.P. Katru, A. Kumar, S.K. Vuppala, S. Paul, iSense: a wireless sensor network based conference room management system, in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings* (Association for Computing Machinery, New York, 2009), pp. 37–42
 24. T.A. Nguyen, M. Aiello, Beyond indoor presence monitoring with simple sensors, in *Proceedings PECCS* (2012), pp. 5–14
 25. C. Miller, Z. Nagy, A. Schlueter, A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings. *Renew. Sust. Energ. Rev.* **81**, 1365–1377 (2018)
 26. B. Dong, B. Andrews, K.P. Lam, M. Höyneck, R. Zhang, Y.-S. Chiou, D. Benitez, An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network. *Energy Build.* **42**(7), 1038–1046 (2010)
 27. R. Nasfi, M. Amayri, N. Bouguila, A novel approach for modeling positive vectors with inverted Dirichlet-based hidden Markov models. *Knowl. Based Syst.* **192**, 105335 (2020)
 28. Z. Li, Y. Jiang, Q. Wei et al., Survey on energy consumption of air conditioning in summer in a residential building in Beijing. *J. Heat. Vent. Air Cond.* **37**(4), 46–51 (2007)
 29. A.L. Pisello, F. Asdrubali, Human-based energy retrofits in residential buildings: a cost-effective alternative to traditional physical strategies. *Appl. Energy* **133**, 224–235 (2014)
 30. M. Amayri, A. Arora, S. Ploix, S. Bandyopadhyay, Q.-D. Ngo, V.R. Badarla, Estimating occupancy in heterogeneous sensor environment. *Energy Build.* **129**, 46–58 (2016)
 31. D. Beymer, K. Konolige, Real-time tracking of multiple people using continuous detection, in *IEEE Frame Rate Workshop* (1999), pp. 1–8
 32. O. Ardakanian, A. Bhattacharya, D. Culler, Non-intrusive techniques for establishing occupancy related energy savings in commercial buildings, in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (Association for Computing Machinery, New York, 2016), pp. 21–30. <https://doi.org/10.1145/2993422.2993574>
 33. R.H. Dodier, G.P. Henze, D.K. Tiller, X. Guo, Building occupancy detection through sensor belief networks. *Energy Build.* **38**(9), 1033–1043 (2006). <http://www.sciencedirect.com/science/article/pii/S0378778806000028>
 34. K. Basu, V. Debusschere, S. Bacha, U. Maulik, S. Bondyopadhyay, Nonintrusive load monitoring: a temporal multilabel classification approach. *IEEE Trans. Industr. Inform.* **11**(1), 262–270 (2015)
 35. S.T.M. Bourbou, Y. Yoo, User activity recognition in smart homes using pattern clustering applied to temporal ANN algorithm. *Sensors* (Basel, Switzerland) **15**(5), 11953–11971 (2015)
 36. S. Daoaca, T. Hong, Occupancy schedules learning process through a data mining framework. *Energy Build.* **88**, 395–408 (2015)

37. L. Chen, J. Hoey, C.D. Nugent, D.J. Cook, Z. Yu, Sensor-based activity recognition. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **42**(6), 790–808 (2012)
38. L.M. Candanedo, V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and CO₂ measurements using statistical learning models. *Energy Build.* **112**, 28–39 (2016)
39. F. Pernkopf, J. Bilmes, Efficient heuristics for discriminative structure learning of Bayesian network classifiers. *J. Mach. Learn. Res.* **11**, 2323–2360 (2010)
40. N. Bouguila, Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Trans. Knowl. Data Eng.* **24**(12), 2184–2202 (2012)
41. N. Bouguila, O. Amayri, A discrete mixture-based kernel for SVMs: application to spam and image categorization. *Inf. Process. Manag.* **45**(6), 631–642 (2009)
42. N. Bouguila, Deriving kernels from generalized Dirichlet mixture models and applications. *Inf. Process. Manag.* **49**(1), 123–137 (2013)
43. T. Bdiri, N. Bouguila, Bayesian learning of inverted Dirichlet mixtures for SVM kernels generation. *Neural Comput. Appl.* **23**(5), 1443–1458 (2013)
44. A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323 (1999). <https://doi.org/10.1145/331499.331504>
45. X.J. Zhu, Semi-supervised learning literature survey. University of Wisconsin-Madison, Department of Computer Sciences, Tech. Rep., 2005
46. L. Deng, D. Yu, Deep learning: Methods and applications. *Found. Trends Signal Process.* **7**(3–4), 197–387 (2014)
47. L. Chen, C.D. Nugent, H. Wang, A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. Knowl. Data Eng.* **24**, 961–974 (1970). <https://doi.org/10.1109/tkde.2011.51>
48. C. Hu, Y. Chen, X. Peng, H. Yu, C. Gao, L. Hu, A novel feature incremental learning method for sensor-based activity recognition. *IEEE Trans. Knowl. Data Eng.* **31**(6), 1038–1050 (2019)
49. U. Maurer, A. Smailagic, D.P. Siewiorek, M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)* (2006), pp. 4–116
50. N.C. Krishnan, D.J. Cook, Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **10**, 138–154 (2014)
51. W. Fan, N. Bouguila, Online learning of a Dirichlet process mixture of Beta-Liouville distributions via variational inference. *IEEE Trans. Neural Networks Learn. Syst.* **24**(11), 1850–1862 (2013)
52. C. Hu, Y. Chen, L. Hu, X. Peng, A novel random forests based class incremental learning method for activity recognition. *Pattern Recognit.* **78**, 277–290 (1970). <https://doi.org/10.1016/j.patcog.2018.01.025>
53. T. Gu, L. Wang, Z. Wu, X. Tao, J. Lu, A pattern mining approach to sensor-based human activity recognition. *IEEE Trans. Knowl. Data Eng.* **23**, 1359–1372 (1970). <https://doi.org/10.1109/tkde.2010.184>
54. M. Raeiszadeh, H. Tahayori, A. Visconti, Discovering varying patterns of normal and interleaved ADLs in smart homes. *Appl. Intell.* **49**(12), 4175–4188 (2019). <https://doi.org/10.1007/s10489-019-01493-6>
55. W. Li, B. Tan, R. Piechocki, Passive radar for opportunistic monitoring in e-health applications. *IEEE J. Transl. Eng. Health Med.* **6**, 1–10 (2018)
56. N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data, in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, ser. IAAI'05 (AAAI Press, 2005), pp. 1541–1546. <http://dl.acm.org/citation.cfm?id=1620092.1620107>
57. P.S. Colin Brennan, G.W. Taylor, Designing learned CO₂-based occupancy estimation in smart buildings. *IET Wirel. Sensor Syst.* **8**, 249–255(6) (2018)
58. Y. Tian, X. Wang, L. Chen, Z. Liu, Wearable sensor-based human activity recognition via two-layer diversity-enhanced multiclassifier recognition method. *Sensors* **19**(9), 2039 (2019). <http://dx.doi.org/10.3390/s19092039>

59. Y. Liu, L. Nie, L. Liu, D.S. Rosenblum, From action to activity: sensor-based activity recognition. *Neurocomputing* **181**, 108–115 (2016). Big Data Driven Intelligent Transportation Systems
60. E.M. Tapia, S.S. Intille, K. Larson, Activity recognition in the home using simple and ubiquitous sensors, in *Pervasive Computing*, ed. by A. Ferscha, F. Mattern (Springer, Berlin, 2004), pp. 158–175
61. F. Ordóñez, D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors* **16**(1), 115 (2016)
62. L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in *Pervasive Computing*, ed. by A. Ferscha, F. Mattern (Springer, Berlin, 2004), pp. 1–17
63. H. Yu, G. Pan, M. Pan, C. Li, W. Jia, L. Zhang, M. Sun, A hierarchical deep fusion framework for egocentric activity recognition using a wearable hybrid sensor system. *Sensors (Basel, Switzerland)* **19**(3), 546 (2019). <https://www.ncbi.nlm.nih.gov/pubmed/30696100>
64. X. Guo, R. Su, C. Hu, X. Ye, H. Wu, K. Nakamura, A single feature for human activity recognition using two-dimensional acoustic array. *Appl. Phys. Lett.* **114**(21), 214101 (2019)
65. F.A. Machot, A.H. Mosa, M. Ali, K. Kyamakyia, Activity recognition in sensor data streams for active and assisted living environments. *IEEE Trans. Circuits Syst. Video Technol.* **28**(10), 2933–2945 (2018)
66. H. Zhao, Q. Hua, H.-B. Chen, Y. Ye, H. Wang, S.X.-D. Tan, E. Tlelo-Cuautle, Thermal-sensor-based occupancy detection for smart buildings using machine-learning methods. *ACM Trans. Des. Autom. Electron. Syst.* **23**(4) (2018). <https://doi.org/10.1145/3200904>
67. D.J. Patterson, D. Fox, H. Kautz, M. Philipose, Fine-grained activity recognition by aggregating abstract object usage, in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, Oct 2005, pp. 44–51
68. S. Ali, N. Bouguila, Variational learning of Beta-Liouville hidden Markov models for infrared action recognition, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019
69. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**(2), 257–286 (1989)
70. C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, Berlin, 2006)
71. L. Rabiner, B. Juang, An introduction to hidden Markov models. *IEEE ASSP Mag.* **3**(1), 4–16 (1986)
72. M. Rodriguez, C. Orrite, C. Medrano, D. Makris, One-shot learning of human activity with an MAP adapted GMM and simplex-HMM. *IEEE Trans. Cybern.* **47**(7), 1769–1780 (2017)
73. M. Wang, S. Abdelfattah, N. Moustafa, J. Hu, Deep gaussian mixture-hidden Markov model for classification of EEG signals. *IEEE Trans. Emerg. Top. Comput. Intell.* **2**(4), 278–287 (2018)
74. J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *Int. Comput. Sci. Inst.* **4**, 126 (1998)
75. J.A. Hartigan, *Clustering Algorithms*, 99th edn. (Wiley, New York, 1975)
76. N. Oliver, A. Garg, E. Horvitz, Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.* **96**(2), 163–180 (2004). Special Issue on Event Detection in Video
77. T. van Kasteren, A. Noulas, G. Englebienne, B. Kröse, *Accurate Activity Recognition in a Home Setting* (Association for Computing Machinery, New York, 2008), pp. 1–9. <https://doi.org/10.1145/1409635.1409637>
78. G. Singla, D.J. Cook, M. Schmitter-Edgecombe, Recognizing independent and joint activities among multiple residents in smart environments. *J. Ambient Intell. Humaniz. Comput.* **1**(1), 57–63 (2010)
79. T.V. Duong, H.H. Bui, D.Q. Phung, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1 (2005), pp. 838–845

80. B. Minor, D.J. Cook, Regression tree classification for activity prediction in smart homes, in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Association for Computing Machinery, New York, 2014), pp. 441–450
81. A. Tsanas, A. Xifara, Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy Build.* **49**, 560–567 (1970)
82. L.G. Fahad, S.F. Tahir, M. Rajarajan, Activity recognition in smart homes using clustering based classification, in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 1348–1353
83. M. Stikic, D. Larlus, B. Schiele, Multi-graph based semi-supervised learning for activity recognition, in *2009 International Symposium on Wearable Computers*, Sept 2009, pp. 85–92
84. N. Bouguila, Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Trans. Knowl. Data Eng.* **24**(12), 2184–2202 (2012)
85. B.M. Shahshahani, D.A. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. Geosci. Remote Sens.* **32**(5), 1087–1095 (1994)
86. H.M.S. Hossain, M.A.A.H. Khan, N. Roy, Active learning enabled activity recognition. *Pervasive Mob. Comput.* **38**, 312–330 (2017). Special Issue IEEE International Conference on Pervasive Computing and Communications (PerCom) 2016. <http://www.sciencedirect.com/science/article/pii/S1574119216302073>
87. Y. Chiang, C. Lu, J.Y. Hsu, A feature-based knowledge transfer framework for cross-environment activity recognition toward smart home applications. *IEEE Trans. Hum. Mach. Syst.* **47**(3), 310–322 (2017)
88. T. Huynh, B. Schiele, Analyzing features for activity recognition, in *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies, ser. sOc-EUSAI '05* (ACM, New York, 2005), pp. 159–163
89. Q. Zhu, Z. Chen, Y.C. Soh, Smartphone-based human activity recognition in buildings using locality-constrained linear coding, in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)* (2015), pp. 214–219
90. E. Kim, S. Helal, D. Cook, Human activity recognition and pattern discovery. *IEEE Pervasive Comput.* **9**(1), 48–53 (2010)
91. D. Tran, A. Sorokin, Human activity recognition with metric learning, in *Computer Vision – ECCV 2008*, ed. by D. Forsyth, P. Torr, A. Zisserman (Springer, Berlin, 2008), pp. 548–561
92. N. Oliver, E. Horvitz, A. Garg, Layered representations for human activity recognition, in *Proceedings of Fourth IEEE International Conference on Multimodal Interfaces*, Oct 2002, pp. 3–8
93. O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutor.* **15**(3), 1192–1209 (2013)
94. M. Amayri, S. Ploix, H. Kazimi, Q. Ngo, A. Safadi, Estimating occupancy from measurements and knowledge using Bayesian network for energy management. *Sensor* **7**, 53932–53944 (2019)
95. M. Amayri, S. Ploix, Q.-D. Ngod, Estimating occupancy from measurement and knowledge with Bayesian networks, in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (2016)
96. M. Amayri, S. Ploix, N. Bouguila, F. Wurtz, Estimating occupancy using interactive learning with a sensor environment: real-time experiments. *IEEE Access* **7**, 53932–53944 (2019)
97. M. Amayri, S. Ploix, P. Reignier, S. Bandyopadhyay, Towards interactive learning for occupancy estimation, in *ICAI'16 - International Conference on Artificial Intelligence (As Part of WORLDCOMP'16 - World Congress in Computer Science, Computer Engineering and Applied Computing)*, Las Vegas, July 2016. <https://hal.archives-ouvertes.fr/hal-01407401>
98. M. Amayri, S. Ploix, N. Bouguila, F. Wurtz, Database quality assessment for interactive learning: application to occupancy estimation. *Energy Build.* **209**, 109578 (2020)